End-to-End RAG-based Product Recommendation System using Sentence Transformers and GPT-4o
Dataset: Myntra Fashion Product Recommendation
Student Name: Priyanuj Misra
Course: Generative AI

1. Objective

The objective of this project is to develop a Retrieval-Augmented Generation (RAG) system tailored for recommending fashion products from the Myntra catalog. By leveraging semantic embeddings and large language models (LLMs), the system is designed to return contextually relevant and user-friendly product recommendations in response to natural language queries.

2. Dataset Description

The dataset used for this project is a structured product catalog from Myntra, consisting of several thousand rows. Each row corresponds to a unique product and includes the following columns:

Product Name: The name/title of the product.

Products/ Category: The fine-grained product category (e.g., Sneakers, Kurtas, Heels).

Rating Count: The number of user ratings received.

Average Rating: The average user rating, on a 5-point scale.

Description: A marketing description highlighting product features, material, design, or use cases.

Metadata: Additional identifiers such as product ID or source page reference for citation.

3. Approach Overview

This system follows the standard structure of a RAG pipeline:

Embedding Generation: Transforming structured tabular data into semantically rich vector representations using sentence transformers.

Semantic Retrieval: Identifying the top N relevant products by comparing a user's query embedding with product embeddings using similarity measures.

Response Generation: Using a large language model to generate natural language responses grounded in the retrieved product information.

4. Preprocessing and Embedding

To prepare the data for semantic search, the columns category, rating_count, average_rating, and description were combined into a single natural language text string per product. This allowed the model to interpret both qualitative and quantitative information contextually.

A pre-trained SentenceTransformer model, specifically all-MiniLM-L6-v2, was used to generate fixed-length embeddings for these combined text entries. These embeddings capture the semantic content of the product details and enable effective similarity-based retrieval.

5. Embedding Storage and Handling
Since CSV files do not support storing Python lists natively, each product's embedding (which is a list of floats) was converted into a comma-separated string for storage. The final dataset with embeddings was saved as a CSV file and optionally stored in Google Drive for accessibility.

Upon reloading the CSV, the embedding strings were parsed back into lists of floats to be used in downstream similarity calculations.

6. Retrieval Mechanism
When a user submits a query (e.g., "Show me stylish white sneakers under ₹3000"), the query is embedded using the same sentence transformer model. Cosine similarity is then used to compare this query embedding against all product embeddings to retrieve the top 3 semantically similar products.

7. Prompt Design and Response Generation
To convert the retrieved product information into a helpful and engaging user response, a carefully crafted prompt was developed and submitted to OpenAI's GPT-4o model.

The prompt provides the model with:

The user query.

A system message describing its role as a fashion assistant.

The top 3 retrieved product descriptions.

Instructions on how to cite the product name and metadata.

Guidelines to generate user-friendly, concise, and accurate recommendations.

The assistant is instructed to avoid internal details and return only relevant, customer-facing information, optionally formatted in text or tabular form. It must conclude the response with citations of the recommended products.

8. Final Output Format
The system returns a coherent recommendation narrative, addressing the user's fashion needs and highlighting key product features such as category, ratings, and descriptions. The output ends with a citation section listing the recommended product names along with their IDs or metadata for traceability.

9. Key Benefits and Applications

This RAG-based recommendation approach offers multiple advantages:

Personalization: It interprets nuanced user intent via LLMs.

Context-Awareness: Embeddings provide contextual understanding beyond keywords.

Explainability: The generated responses clearly cite why and what was recommended.

Scalability: The method scales well with new product data and changing customer needs.

Such a system can be directly applied to real-world e-commerce platforms like Myntra to enhance customer experience and improve conversion rates.

10. Future Scope

Potential extensions of this project include:

Incorporating price and size availability in the recommendation criteria.

Allowing for multilingual queries and responses.

Building a live web app interface with search and recommendation capabilities.

Experimenting with larger models like E5-large or embedding fine-tuning for domain specificity.

Indexing embeddings in vector databases like FAISS or Qdrant for faster retrieval.

11. Conclusion

This project demonstrates how combining sentence-level embeddings and generative AI models can result in a powerful, scalable, and user-friendly product recommendation engine. By grounding LLMs in structured domain data, the system brings together the best of both worlds — retrieval accuracy and natural language fluency.