**PROJECT DOCUMENTATION**

**HEALTHCARE AND PERSONAL CARE PRODUCT REVIEWS**

**Overview:**This project processes and integrates datasets containing both structured (e.g., product reviews) and unstructured data (e.g., textual user inputs). The results are stored in a database and made retrievable using a retrieval system based on embeddings. By employing Retrieval-Augmented Generation (RAG), the system can summarize retrieved results in response to user prompts.

**Core Use Case:**

* Users submit reviews of products they have purchased along with the store from which the product was bought (randomly chosen from the database).
* The system retrieves similar comments from the database.
* The retrieved data enables the agent company to:
* Identify patterns in reviews.
* Filter stores by the nature of customer feedback.
* Enhance their business model based on insights.

**THOUGHT PROCESS:  
  
Dataset:**Customer/Product Review based datasets are best examples of structured and unstructured data. This dataset is a collection of two documents, one contains the user id, review, product id, and date of the review etc. The other document contains the meta data about the product that the user has reviewed. With this data we provide a business solution for the Agents who deliver the products to customers.  
  
**Database**:  
For the storage of records and vectors, MongoDB is used. Simple to use and supports vector storage. MongoDB provides a great leverage for efficient storage and retrieval of data.   
  
**LLM and Retrieval strategy**:

Sentence Transformers from hugging face is a light weight and open-source library for a wide range of embedding models. Here “all-MiniLM-L6-v2” model is simple and widely used for text embedding. After the data is vectorized and stored in dataset, we follow these steps for retrieval:

* Fetch all the records of the store name matching with user review.
* cosine similarity is used to retrieve the similar reviews of the records.
* Validate similarity with a default threshold value.

With this we will get records with similar reviews for further analysis. If no matching records are found, then new records should be added for the user, product and the store.

**SCHEMA DESIGN:**

The schema design for data storage is the combination of all the dataset documents with distinct fields and one primary field from all documents. With the necessary fields, create a JSON schema and include validation for each field, which can be used for data storage in MongoDB based on the validation.

**FUTURE ENHANCEMENTS:**

1. **User Feedback Loop**:
   * Allow users to rate the relevance of retrieved reviews to refine retrieval algorithms.
2. **Multi-language Support**:
   * Extend embeddings to support reviews in different languages.
3. **Advanced Summarization**:
   * Integrate summarization techniques like fine-tuned GPT models for better insights.
4. **Scalability**:
   * Transition to distributed databases for handling larger datasets.

**SOURCE CODE:**Find my git repository for complete source code of the application and, refer README for running the application.