Work flow of the project

This project is a Retrieval-Augmented Generation (RAG) system designed to answer data science-related queries using a structured pipeline involving document processing, vector-based retrieval, and an interactive chatbot interface.

The system begins with document ingestion, where data is extracted from a PDF file using PyPDFLoader. The extracted text undergoes preprocessing, where it is split into manageable chunks using RecursiveCharacterTextSplitter. This ensures that the system can efficiently retrieve relevant information without processing large, unstructured text blocks. These chunks are then converted into numerical representations using HuggingFaceEmbeddings, enabling similarity searches. The embeddings are stored in a FAISS vector database, which allows for fast retrieval of information based on user queries.

For query handling, the system first searches the FAISS index to find the most relevant text chunks. This similarity search ensures that the most appropriate context is retrieved based on the query. The retrieved text chunks are formatted into a structured prompt using ChatPromptTemplate, which is then passed to the ChatGroq model. This model, specifically the LLaMA-3.1-8B-Instant, processes the input and generates a response based on the provided context. To maintain response quality, a threshold is applied to similarity scores, filtering out irrelevant or low-confidence results. The response is then structured in a clear and concise manner while ensuring that the source of the information is retained.

The user interface is developed using Streamlit, providing an interactive and user-friendly chatbot experience. The application initializes session states to store chat history and employs caching mechanisms to optimize response retrieval. When a user inputs a query, the system first checks if a cached response exists; if not, the query is processed through the retrieval module. The conversation flow is maintained by displaying previous queries and responses, ensuring a seamless interaction. The interface also includes visual elements such as an image and formatted text to enhance usability.