Evaluation Strategy – RAG Chatbot (Amazon Annual Report)

Objective

To validate the accuracy, relevance, and faithfulness of the responses generated by the chatbot using a structured evaluation methodology. The goal is to ensure the system reliably:

- Retrieves relevant content from the document.
- Answers questions based on factual content.
- Aligns with the user's intent.

Steps-to-execute evaluation file

For generating the evaluation metric I have generated ground truth responses using LLM which are used as reference for analyzing the RAG chatbot LLM response.

To generate the evaluation score data from 6 sample question execute the ragas_evaluation.py file which will generate score of LLM response by compar it with ground truth.

Tools & Libraries Used

The following tools and libraries were used to build and evaluate the chatbot system:

Component	Library/Tool	Purpose
Vector DB	Chroma	Stores and indexes document chunks for retrieval.
Embeddings	OpenAIEmbeddings	Transforms chunks into semantic vectors.
LLM	OpenAI GPT-4o	Generates responses using retrieved context.
Evaluation	RAGAS	Framework to compute retrieval and generation metrics.
Wrapper	LangchainLLMWrapper	Integrates Langchain-based LLMs with RAGAS.

Evaluation Metrics

The system was evaluated using three core metrics provided by RAGAS:

1. LLMContextRecall
- Measures how well the retrieved context covers the necessary information needed to answer the question.

2. Faithfulness
- Checks if the LLM's answer is faithful to the retrieved context.

3. AnswerRelevancy
- Measures whether the generated answer truly addresses the user's original query.

Step-by-Step Evaluation Flow

1. Load Vector DB & Retriever
- A prebuilt Chroma DB is loaded using saved embeddings.
- A LangChain-based retriever is initialized (MMR-based).

2. Sample Dataset
- A small dataset of real user queries and expected (reference) answers is prepared.

3. Response Generation
- For each question:
  - Retrieve chunks from ChromaDB
  - Format with system prompt
  - Use OpenAI GPT-4o to generate a response

4. Dataset Assembly
- Each evaluation instance includes:
  - user_input, retrieved_contexts, response, reference

5. Run Evaluation
- Dataset evaluated using ragas.evaluate()
- Results saved as a Pandas DataFrame and exported

Output

CSV File: evaluation_chatbot.csv (included in the code base folder)

Benefits of This Approach

- Quantitative validation of both retrieval and generation.
- Easy to extend with more queries or metrics.
- Provides clarity on system performance.