**Task1**

1. Describe the default chunking strategy that you will use.

I would use a simple method for a quick assessment, applying **recursive chunking** to get an overview of the system. In the context of machine learning, we always start with the simplest model: linear regression. This recursive approach lets me experiment with different partitions quickly and with low computational costs.

1. Articulate a chunking strategy that you would also like to test out.

After reviewing the length and division of the text within the documents, I believe it would be beneficial to test a **Semantic Chunking** strategy. Given that the topics and definitions are shorter texts, semantic chunking allows us to split the content into smaller, meaning-preserving units. This approach maintains the integrity of ideas in a more concise form, resulting in lightweight chunks that are easier for the embeddings model (based on **mixedbread-ai/mxbai-embed-large-v1**) to process efficiently.

1. Describe how and why you made these decisions

Before building a basic **RAG** system, I decided to retrieve chunked documents using both **Recursive Chunking** and **Semantic Chunking** strategies. By comparing these two options, I aim to determine which method makes it easier for the model to grasp the necessary context to answer questions based on the retrieved text. This comparison allows me to verify which chunking strategy better preserves meaning and enhances the model's ability to generate accurate and contextually relevant responses.

Task 2

1. Build a prototype and deploy to a Hugging Face Space, and include the public URL link to your space create a short (< 2 min) loom video demonstrating some initial testing inputs and outputs.

URL

1. How did you choose your stack, and why did you select each tool the way you did?

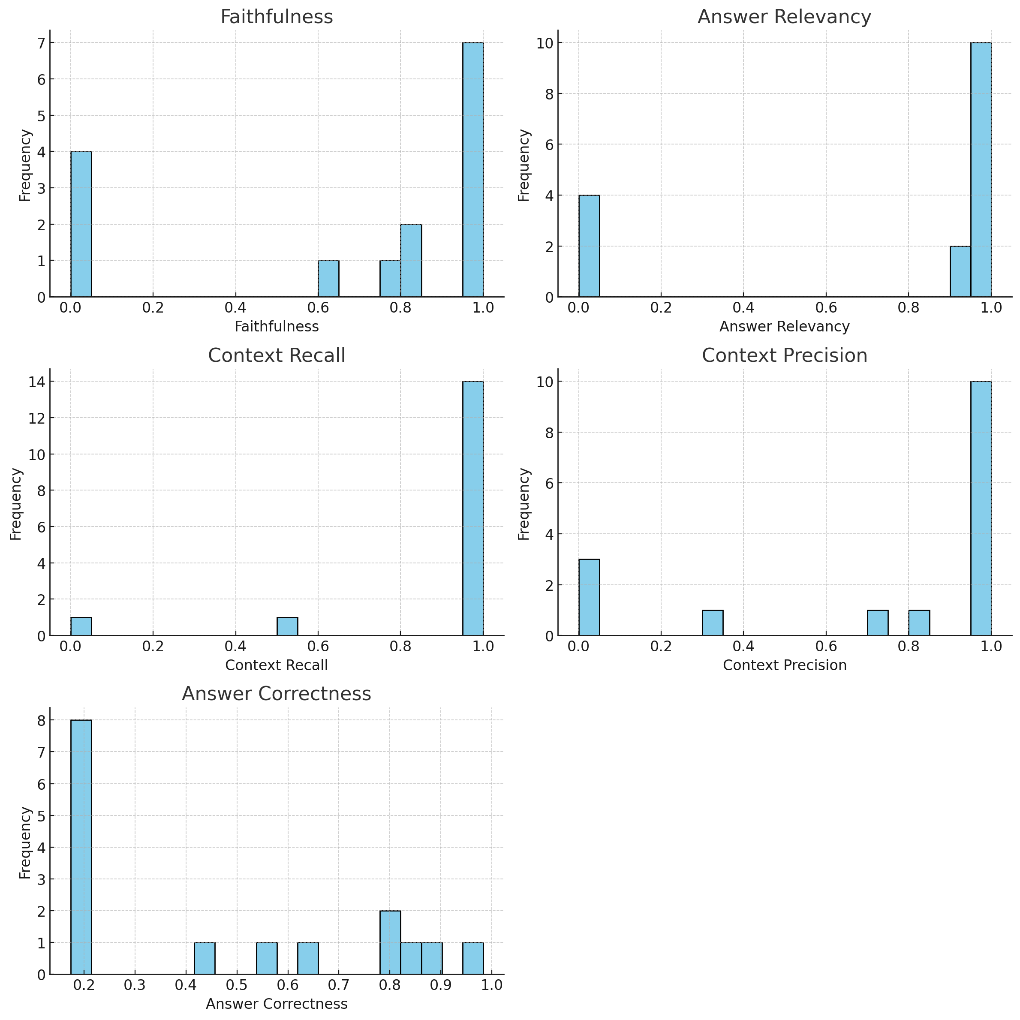
Since there were no security requirements, we chose a stack based on the free public cloud from **Hugging Face**, with a simple frontend developed in **Chainlit**, eliminating the need to manage authentication or sessions. For the logic and orchestration of the RAG, we use **Langchain** because of its versatility.

As for the vector database, we went with **Quadrant**, which strikes a good balance between performance and cost by using the free version in the cloud.

For **SDG** and **RAG evaluation** tools I used **Ragas** provide a straightforward set of classes that allow for the evaluation of various metrics associated with RAG to understand the performance and the reasons behind the responses.

Task 3

1. Assess your pipeline using the RAGAS framework including key metrics faithfulness, answer relevancy, context precision, and context recall. Provide a table of your output results



The RAG system effectively retrieves relevant context and maintains high recall. However, improvements are needed in **faithfulness** and **answer accuracy**, as some responses are factually incorrect or stray from the retrieved context. A promising enhancement could be in the retriever area, and for this purpose, the initial test employed the **Semantic Chunking** strategy.

Task 4

1. Swap out your existing embedding model for the new fine-tuned version. Provide a link to your fine-tuned embedding model on the Hugging Face Hub.

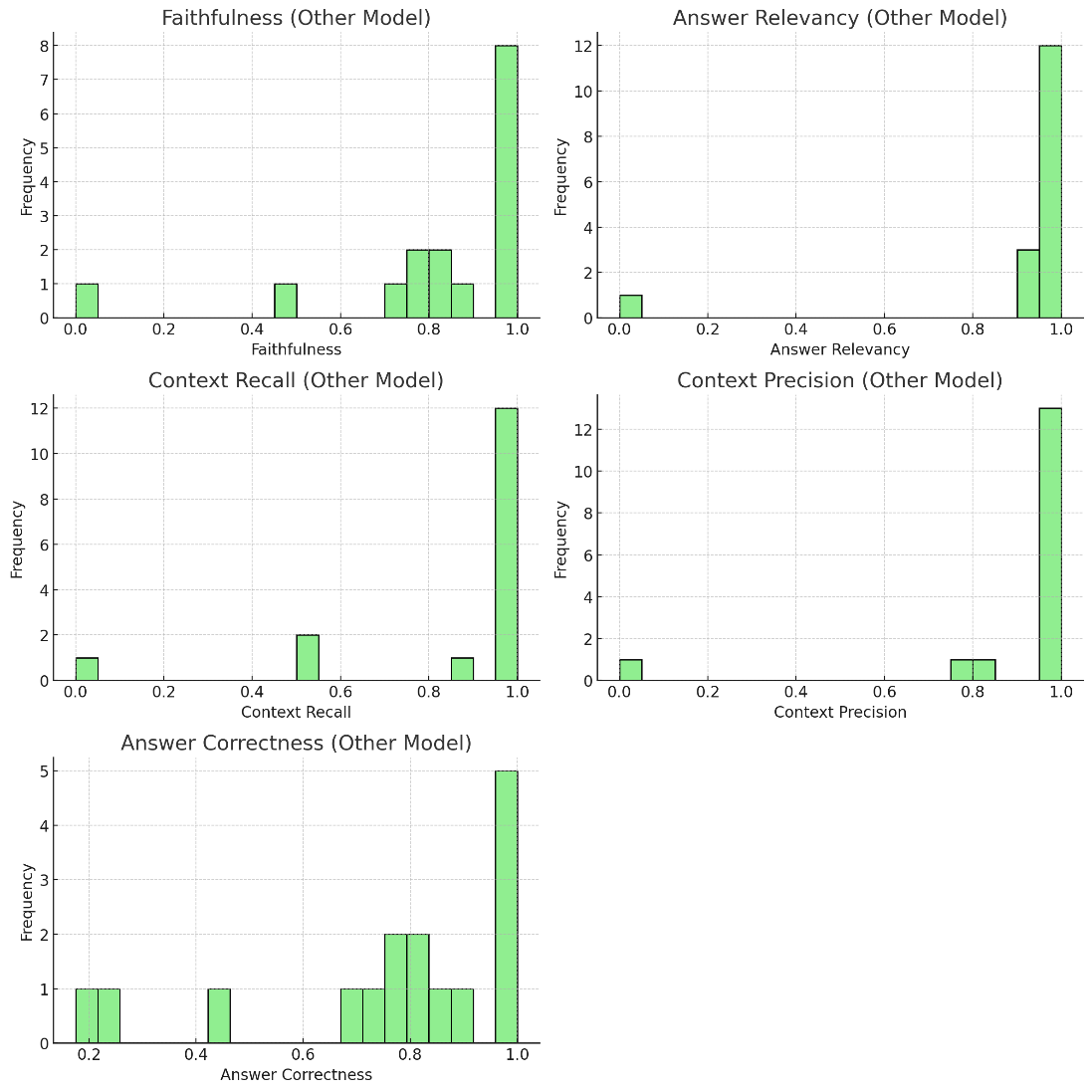
https://huggingface.co/CamiloGC93/bge-large-en-v1.5-etical

1. How did you choose the embedding model for this application?

One of the services I use for local testing is Ollama. According to the documentation, many recommendations about embeddings suggest that the bge-large-en model outperforms OpenAI's ada-02. You can check it out at: https://huggingface.co/BAAI/bge-large-en.

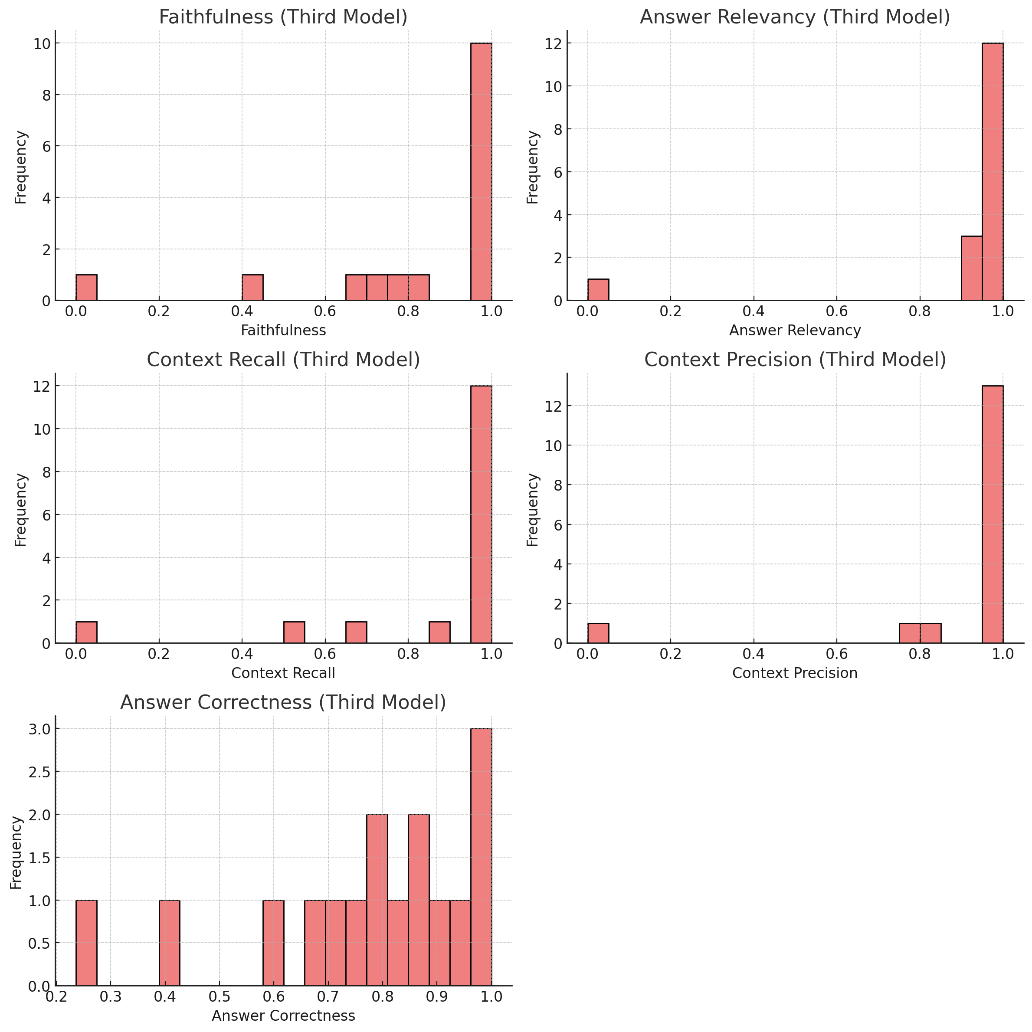
Task 5

1. Test the fine-tuned embedding model using the RAGAS frameworks to quantify any improvements. Provide results in a table.



Rag evaluation with the embedding model finetuned and Semantic Chunk Stagey

1. Test the two chunking strategies using the RAGAS frameworks to quantify any improvements. Provide results in a table.



Rag evaluation with the embedding model finetuned and Recursive chunking Stagey

1. The AI Solutions Engineer asks you “Which one is the best to test with internal stakeholders next week, and why?”

Most of the time, stakeholders care more about the results than the implementations. The best strategy to visualize results, thanks to Ragas, is semantic chunking with pre-trained embeddings. This is because, even though it's a bit more complex to grasp, the visual outcome stands out and the numbers provide greater reliability.

1. What is the story that you will give to the CEO to tell the whole company at the launch next month?

The best way to understand something is to ask someone who knows more than we do, someone who can simplify it with examples. That's why a bot with up-to-date knowledge is perfect for clearing up uncertainty about a new topic and building confidence by providing the information needed to implement AI ethically.