- The dataset should have at least 30 rows.  
- Make sure your query-response pairs are diverse and are not concentrated on a specific query type, document section or page.  
- Explain why you feel this is a comprehensive dataset to gauge the performance of your chatbot.

* How you constructed your dataset
* How and why you chose these evaluation metrics
* What did you try to improve the accuracy

Initially, when I received the take-home project, my first idea was to design the pipeline so it could work with any PDF document. The document would be embedded, stored in a vector database, and the database would be used as a retriever. It would query relevant documents and pass them to the LLM for generation. I chose LangChain as my framework because it is user-friendly and offers many excellent integrations. Additionally, I appreciate LangChain for its comprehensive documentation.

How did I construct the dataset: - For creating the dataset I used Ragas test generator initially I used different distributions for {simple, reasoning, }

Ragas Faithfulness and answer relevancy

Regular RAG

A red and green striped pattern

Description automatically generated

{'faithfulness': 0.8067, 'answer\_relevancy': 0.9225}

Multi-Query:-

A red and green striped fabric

Description automatically generated

{'faithfulness': 0.8115, 'answer\_relevancy': 0.9487}

Hyde

A red green and black text

Description automatically generated

{'faithfulness': 0.4553, 'answer\_relevancy': 0.4708}

Re-Ranking

A red and green striped pattern

Description automatically generated

{'faithfulness': 0.8033, 'answer\_relevancy': 0.9552}

RAG with colbert

A screen shot of a screen

Description automatically generated

{'faithfulness': 0.8968, 'answer\_relevancy': 0.9121}

Ragas Faithfulness and answer relevancy

A screenshot of a computer

Description automatically generated

{'faithfulness': 0.9246, 'answer\_relevancy': 0.9458}