RAG pipeline flow

Approach

1. Designed a simple RAG pipeline as promptflow is a new framework, needed to see how langchain and promptflow interact.
2. For evaluation, trulens RAG triad is used to keep things simple as it needs API calls to open AI models.
3. For regression testing, langchain was used to generate question, context, answer pairs using GPT-3.5 to use as groundtruth for evaluation using RAGAS. The script regression-tests.py is built for this purpose. It takes lot of resources and API calls for running this script. I have run it in advance and showed the results in comments for a sample.
4. I have included open AI key in the code to allow smoother run, it may happen that the budget gets finished. New key can be added to inputs if flow in openai field.

Areas of Improvement

Below are some improvements which can be done on the project but due to constraints like time, cost, latency etc, could not be done.

1. Web scraping to find more relevant documents
2. Hybrid search to make retrieval more robust for industry specific terms
3. Hide Open AI key exposed in input
4. Reranker after retrieval to retain only most relevant information
5. More experiments with embedding models like open ai ada, matryoshka embeddings
6. More comprehensive evaluation of RAG using hit rate calculation to evaluate retrieval, and RAGAS framework for more metric calculations
7. The output from LLM is getting truncated, this needs to be explored
8. More powerful and bigger LLMs can be used. Currently, Mistral 7b model is being used
9. Latency can be improved by using more powerful machines or self hosting the model.
10. More regression tests can be designed. Need to do more research in this area.