Assignment: RAG for Summarization

Goal: To design a RAG based summarization agent

Tasks:

1. Create an LLM based summarization agent that summarizes an automotive issue using a given dataset. Input to the agent comes in the form of a json object with the following structure

   Input = { 'make': 'ford',
   'model': 'escape,
   'year': '2001',
   'issue': 'stuck throttle risk'}

   The agent should retrieve the relevant documents in the dataset and summarize the issue using those documents. The final response of the agent must include both the retrieved documents as well as the final summarization.

   Dataset: Use the NHTSA Recalls dataset. The dataset can be downloaded from the following link https://www.nhtsa.gov/nhtsa-datasets-and-apis. Download the FLAT_RCL.zip file. Relevant columns for this assignment are {make, model, year, defect summary, consequence summary, corrective summary}. All the three summaries could be merged and considered as a single document for computing vector embeddings. For this assignment, you may take a subset of this dataset by selecting data for only Ford and Toyota makes.

   Note: You may choose the embedding model of your choice. For summarization, use the LLaMA 3.1 8b model. Do not use chat-gpt. Use google collab or Kaggle for GPU resources.

2. The RAG approach mentioned in the task-1, works well when the issue you are searching for exists in a few documents. However, if a user is interested in issues like 'most frequent recall', 'top 5 recalls' that occur for a make, model, year combination i.e. a user is interested in a holistic view of the dataset instead of a local view, under such situation the basic RAG fails.

   Based on the claim above, answer the following questions
   1. Why do you think the basic RAG approach fails in such situation?
   2. What are some methods that can be employed to improve RAG for holistic questions?
   3. Can you think of some preprocessing that can be done on the dataset to aid in the existing RAG pipeline?

   Hint: Investigate clustering, Graph-RAG

Submission Guideline: Upload your code for task-1 to a github repo and share the link as part of your submission. Prepare to discuss your answers for task-2 in the interview.

For any questions, feel free to contact abhishek.kumar@predii.com