**ADF Gen AI Hackathon Submission Format**

1. **Team Details**
   * Team - 1
   * Gurpreet Singh, Suraj Kumar Padhy, Abhishek Kumar Sharma
2. **Problem Statement**

* **Frequent Ad-Hoc Data Requests:** Multiple teams often submit ad-hoc data requests that require immediate attention, creating a bottleneck in data retrieval processes and diverting focus from core responsibilities.
* **Repetitive Nature of Requests:** Many requests are similar, differing only in specific segments or filters, leading to redundant query execution and inefficient use of resources.
* **Time-Consuming and Resource-Intensive:** Crafting, running, and formatting queries for each request takes significant time, delaying responses and impacting the team's ability to focus on strategic tasks.

1. **Solution**

* **Step 0: Store General Queries**

Identify and compile 20-30 commonly used queries relevant to the domain or use case.

Represent these queries as vector embeddings using a pre-trained model (e.g., Sentence Transformers or OpenAI embeddings).

Store these embeddings in a Vector Database for efficient similarity search.

* **Step 1: Retrieve Relevant Queries**

When a user inputs a query, embed the input using the same embedding model used in Step 0.

Perform a similarity search in the Vector-DB to retrieve the top N relevant queries.

* **Step 2: Re-Rank Queries**

Utilize a Language Model (LLM) to re-rank the retrieved queries.

Score the queries based on their semantic similarity to the user's input and contextual relevance.

* **Step 3: Select Top Queries**

Extract the top K queries with the highest similarity scores from the re-ranked list.

* **Step 4: Generating the Query**

We send both user requirements, schema of the DB and top similar query for given requirement and LLM generates the target query.

* **Steps 5 & 6: Execute Query**

Execute the generated query on the RDS (Relational Database Service) server.

Retrieve the relevant data from the database.

* **Step 7: Deliver Results**

Summarize the retrieved data using the LLM.

Return the data and the summarized report to the user in an easy-to-understand format.

1. **Source Code**

https://github.com/adf570/Gen-AI-Hackathon-Text-to-Data-Project

1. **Value to ADF**

* **Faster Analysis:** This tool enables rapid generation of requested analyses, saving time and effort compared to traditional methods.
* **Ease of Use:** Users can perform complex data analyses without requiring advanced SQL knowledge.
* **Improved Decision-Making:** As many business decisions revolve around fundamental data analyses, this tool accelerates the process, enabling quicker and more informed decision-making.

1. **Supporting Materials** (if applicable)

**Solution Architecture:**

A diagram of a diagram

Description automatically generated