A Hands-On Mini Project

LLM-Powered Table Discovery and Evaluation

Objective

This mini-project invites you to explore how Large Language Models (LLMs) can support natural language interaction with tabular data. The goal is to build a simple assistant that performs schema summarization, table search, and result evaluation.

You are **not expected to complete all tasks fully**. Instead, try out as many parts as you can within the two-day timeframe. You are encouraged to experiment with different ideas and approaches—focus on learning, exploration, and what you find most interesting along the way.

1 Task 1: Table Schema Summarisation

Input: A small collection of CSV tables (15-20), each with several columns and potentially cryptic column names (e.g., `emp_id`, `trans_dt`, etc.).

Goal: Design prompts that enable an LLM (e.g., GPT-4, Claude, Gemini) to generate a user-friendly summary of each table's schema—explaining what the table is about and describing the purpose of each column in natural language.

Deliverables:

- Your designed prompt(s)
- Schema summaries for each table
- A short explanation of your prompt design choices

Hints:

- Ask the LLM to describe the *purpose* of the table, not just rephrase the column names.
- If available, include a few rows of sample data in the prompt to help infer semantics.
- Try different phrasing styles (e.g., "Explain this schema to a data analyst" or "Summarise what this table is about").

2 Task 2: Natural Language Table Search

Input: A user issues a natural language query (e.g., "Which table contains information about employee salaries?"). Use the schema summaries from Task 1 to match this query to the most relevant table(s).

Goal: Match the user query to the best-fit table(s) using LLM capabilities or other retrieval strategies.

Deliverables:

- Your implementation (Python or Jupyter notebook)
- Output examples: natural language query → matched table(s)
- A brief explanation of your approach

Hints:

- Try using LLM prompts like: "Given these table descriptions and a query, which is most relevant?"
- Alternatively, embed the query and summaries using sentence embeddings (e.g., `sentence-transformers`) and compute similarity scores.

3 Task 3: LLM-Based Evaluation of Retrieved Tables

Goal: Use an LLM to evaluate whether the retrieved table(s) from Task 2 are relevant to the user's query.

Instructions:

- Write a prompt that asks the LLM to assess and explain the relevance of the selected table(s).
- Test your prompt using at least 2–3 different queries.

Deliverables:

- Your evaluation prompt(s)
- LLM-provided relevance ratings and explanations
- A short reflection on whether the LLM's judgment agrees with your own

Hints:

- Try prompting: "User query: ...; Table summary: ...; Rate the relevance from 1 to 5 and explain why."
- Ask the LLM to identify missing or irrelevant information in the matched table.
- Use a mix of vague and specific queries to test robustness.

Final Submission Instructions

- Upload your project to a public or private GitHub repository.
- Include a `README.md` that:
  - Explains how to run your code.
  - Lists all required packages or dependencies.
  - Includes 1–2 example queries and their outputs.
  - Summarizes key insights and challenges encountered.