AI Integration For Dish Data Project

**By: Dhanush H**

**Data Documentation Generator**

Implemented AI integration capabilities within the data engineering pipeline by leveraging an LLM (OpenAI GPT-4) to automatically generate a data dictionary for any BigQuery table in the project.

This showcases how generative AI can enhance metadata management, documentation automation, and data governance.

**Script Name:** src/generate\_data\_dictionary.py

**Core Features:**

1. AI-powered documentation: Uses OpenAI GPT-4 to generate human-readable column descriptions.
2. Automatic dataset detection: Determines if a table belongs to the raw, stage, or mart dataset.
3. Dynamic column selection:

\* By default, documents all columns in the table.

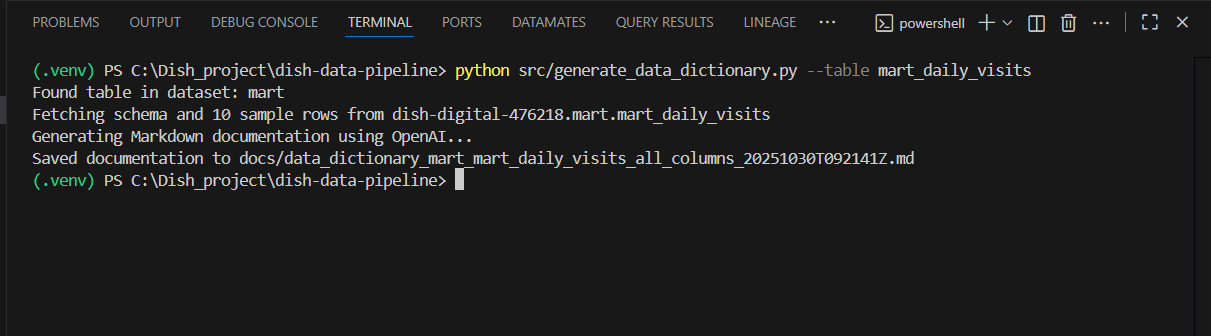
\* Optionally, We can specify specific column names using --columns argument.

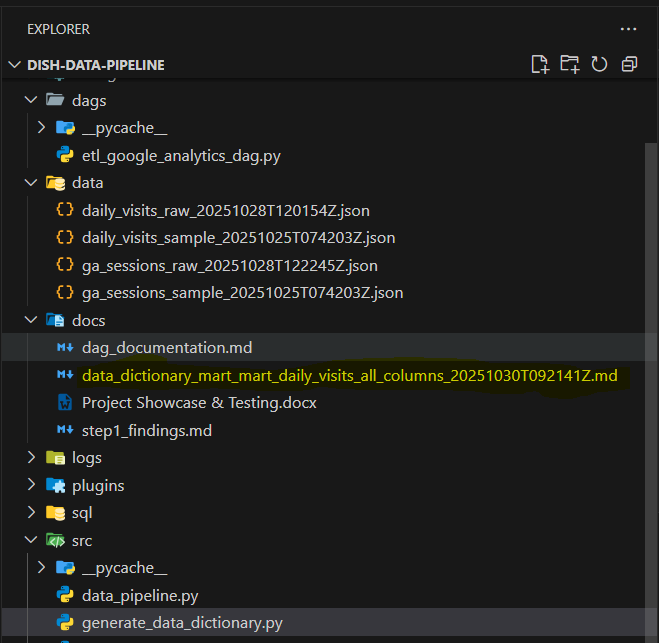
1. Schema & sample-based insights: The script retrieves both the schema and a few sample rows from BigQuery to give the model context.
2. Documentation storage: Generated Markdown files are saved locally.
3. Command-line friendly: Works via CLI with minimal arguments.
4. Environment-driven configuration: Securely reads environment variables from .env (no hardcoded keys).

**Testing:**

1. Testing Scenario 1 --> Documents all columns in the table.

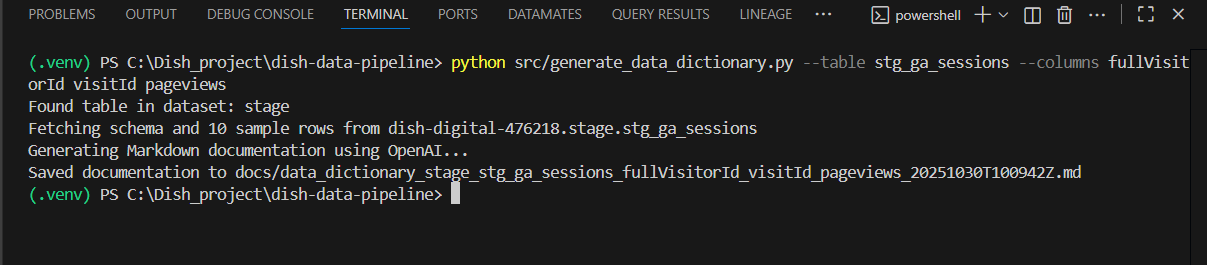
Command: python src/generate\_data\_dictionary.py --table mart\_daily\_visits

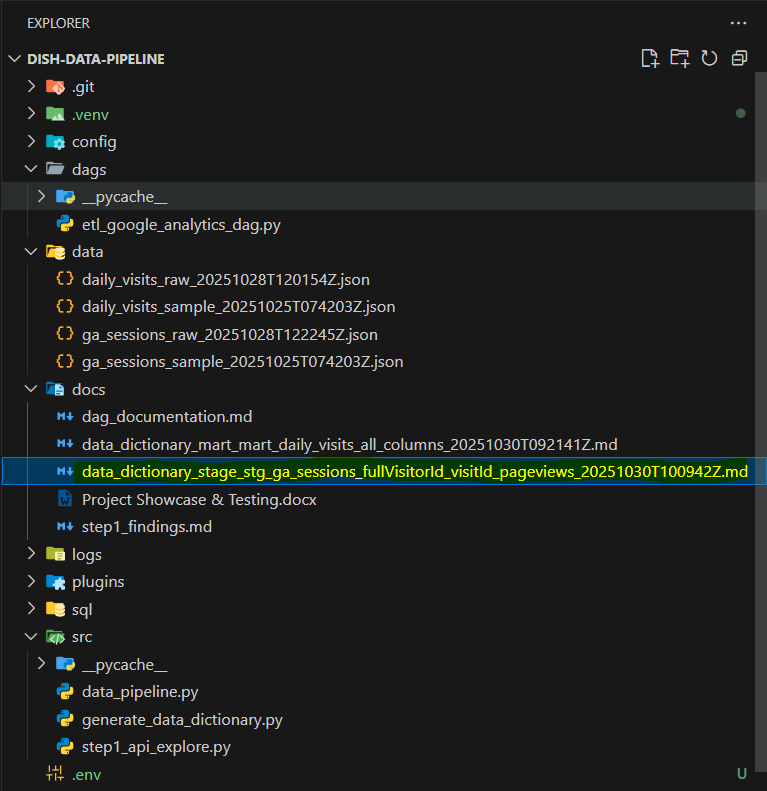




1. Testing Scenario 2 --> Specify specific column names using --columns argument

Command: python src/generate\_data\_dictionary.py --table stg\_ga\_sessions --columns fullVisitorId visitId pageviews





**Reflection: My Learning Journey with AI and LLM Integration**

**My First Step into AI & LLMs:**

Before this project, I had never worked with Large Language Models (LLMs) or AI integration. I had read about them and seen how tools like ChatGPT or Gemini could generate text, but I wasn’t sure how these models could be applied in a real data engineering workflow. This project gave me the perfect opportunity to explore and understand how AI can support data engineering tasks in a practical, value-adding way.

**Understanding How LLMs Work:**

I started by learning the basics of how LLMs work, how they take prompts as input and generate intelligent responses based on context.

Once I understood that, I explored how to connect an LLM through APIs, using OpenAI’s API key to interact with GPT models directly from my Python script. It was both challenging and exciting to see the model respond dynamically to data coming from BigQuery.

At first, I spent a lot of time experimenting with prompts tuning the instructions and formats until the model produced structured, meaningful Markdown output instead of free-form text.

**Building My First AI Integration – The Data Documentation Generator:**

For the integration task, I decided to build a Data Documentation Generator. The goal was to automatically generate human-readable data dictionaries from my BigQuery tables.

This feature uses an LLM:

1. Analyze the table schema and sample data,

2) Generate natural language descriptions for each column,

3) Include examples and validation hints, and

4) Suggest relevant data quality checks at the end.

I found it fascinating how the model could interpret column names and sample values to produce clear, business-level explanations something that usually takes a lot of manual documentation effort.

**Lessons Learned about applying AI in Data Engineering:**

This step taught me more than just how to call an API — it showed me how AI can complement data engineering by automating repetitive documentation tasks and improving knowledge sharing within teams.

I also learned about responsible AI practices:

1. I ensured no sensitive or personally identifiable information (PII) was passed to the model.
2. Only schema and limited sample data were used as inputs.

This experience made me more aware of the importance of data privacy, governance, and ethical AI usage.

**Looking Ahead for My Continued AI Journey**

Overall, this was a huge learning experience for me. I went from having no hands-on exposure to LLMs to successfully integrating one into a real data pipeline. I now have a better understanding of how prompt engineering works, how to connect AI models using APIs, and how they can enhance traditional data workflows. This experience sparked my interest in exploring more AI-driven automation from generating SQL queries to building intelligent data quality reports — and I’m genuinely excited to continue learning in this space.