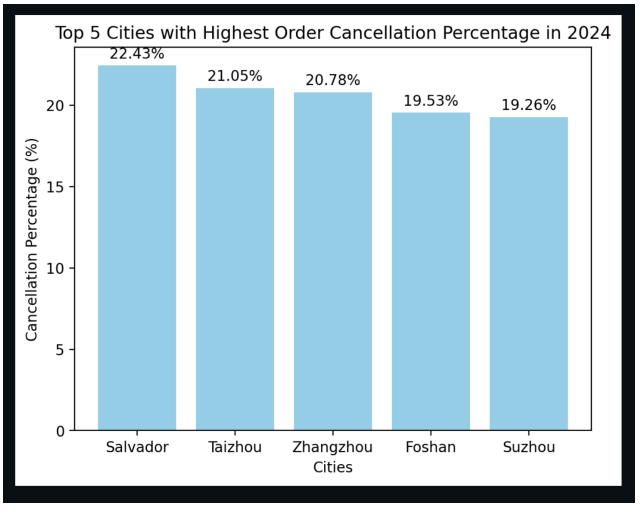
Automated Insights Generation

Motivation

In today's data-driven world, the ability to extract meaningful insights from vast datasets is crucial for informed decision-making across all levels of an organization. However, this process often requires a deep understanding of complex querying languages and coding skills, which can be a significant barrier for many users. This project aims to democratize data analysis by building an intuitive, user-friendly system that empowers users to generate insights from their data using natural language inputs and provide high quality visual and textual insights based on the data provided.

Objectives

- Build a system which takes natural language as input and generate insights from the data for users across the board without the requirement of writing complex queries and code
- Build the data pipeline which loads all csv files into SQL DB and utilises LangChain toolkit to connect with the DB
- Develop a chat based application that has an ability to converse with the user and generate insights on the go using frameworks such as LangChain, streamlit



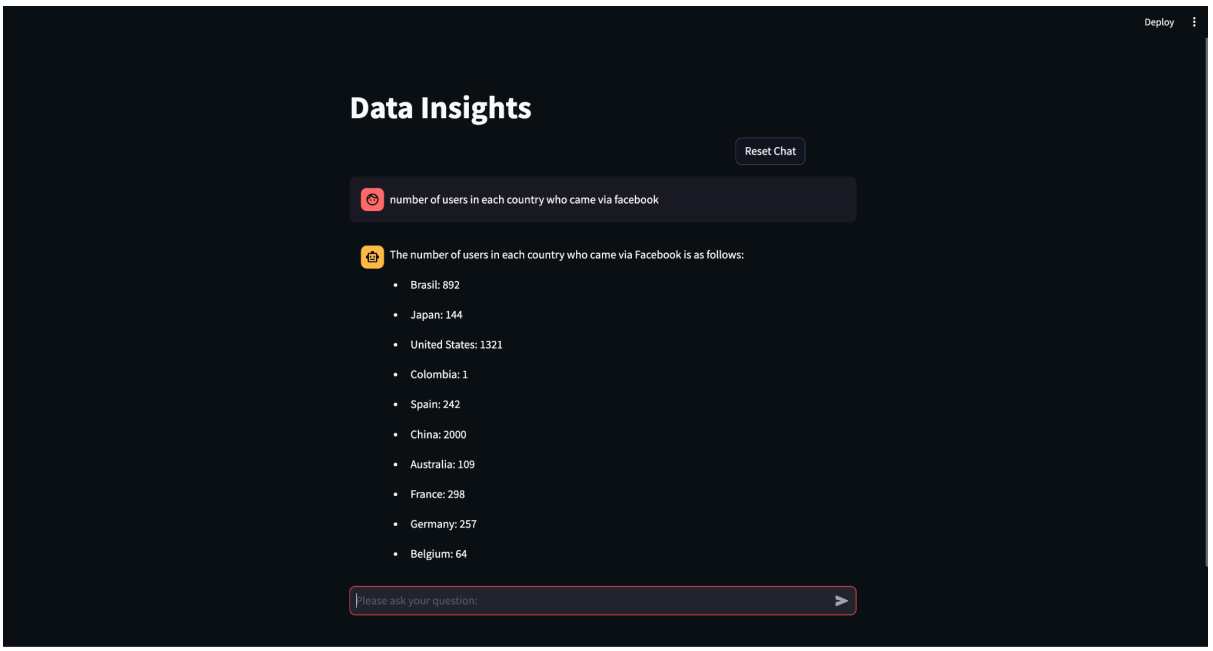
How is Today’s Content Relevant to my Role?

- **PMs**: Gain quick, data-driven insights to inform strategic decisions and enhance product development without writing any queries and code.
- **TPMs**: Facilitate efficient project management by identifying potential bottlenecks in GenAI projects and optimizing workflows.
- **SDEs**: Develop and integrate LLM-based components for natural language input processing and insights generation.
- **Engineering Managers**: Enhance team performance and GenAI project outcomes to guide new age engineering practices and resource allocation.
- **DevOps Engineers**: Build and maintain the infrastructure for scalable, reliable deployment and continuous integration of the LLM system.

Dataset

We will be using the following e-commerce dataset:
https://drive.google.com/drive/folders/1FYM_baitHLWu6LZ6MNExqxqBm5OXM6I8
Though you can use any other openly available datasets as well.

Example Output



Prerequisites

- SQL - https://www.w3schools.com/sql/
  - to store datasets and connect with LLM
- Python - https://www.w3schools.com/python/
  - Programming language
- Langchain - https://python.langchain.com/v0.2/docs/introduction/

    - for building prompts and orchestrating calls to LLMs and SQL DB
- Streamlit - https://docs.streamlit.io/develop/tutorials
    - or building interactive chat console

Get Ready for the Session

Everyone needs to run these steps before the session so that you are well prepared.

A. Setting up MySQL DB (MacOS, Windows)

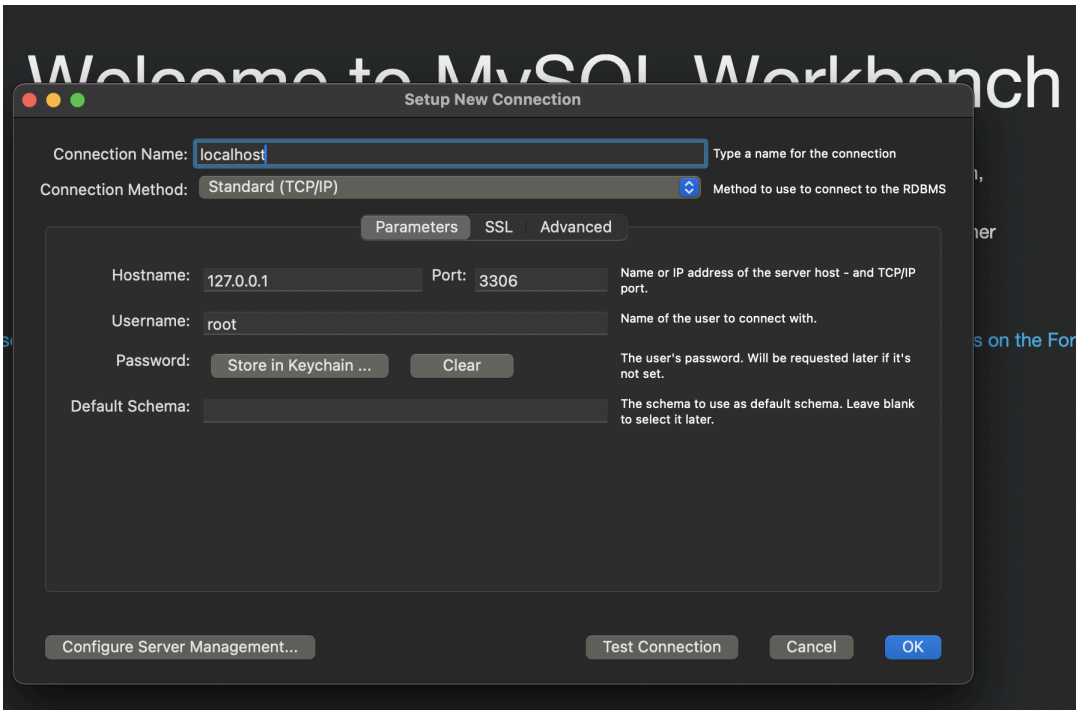
a. **Step by Step Video 1 for Mac** (feel free to post questions in the comments)

b. **Step by Step Video 2 for Mac** (feel free to post questions in the comments)

B. Setting up MySQL Workbench (MacOS, Windows)

a. **Step by Step Video 3** (feel free to post questions in the comments)

b. **Step by Step Video 4** (feel free to post questions in the comments). Make sure to create a localhost connection with user=**root** and port=**3306.**



C. **Installing Python (Windows PowerShell) - Recommended (Python 3.12.4)**

a. In Windows, open Powershell. Press **`Windows`** key or click on the **`Start`** menu icon → Type **`PoweShell`** and press **`Enter`**.

b. Run `python --version` to see if you already have python installed

c. **Download** the Python Installer. To do that, pick your machine architecture (64-bit, ARM64, 32-bit) etc and then run the following powershell command: `curl -o python-3.12.4-amd64.exe https://www.python.org/ftp/python/3.12.4/python-3.12.4-amd64.exe`

d. **Run the Downloaded Installer:** `.\python-3.12.4-amd64.exe /quiet InstallAllUsers=1 PrependPath=1`

e. **Verify** the installation with this powershell command: `python --version`

THAT'S ALL YOU NEED FOR NOW

MISCELLANEOUS:

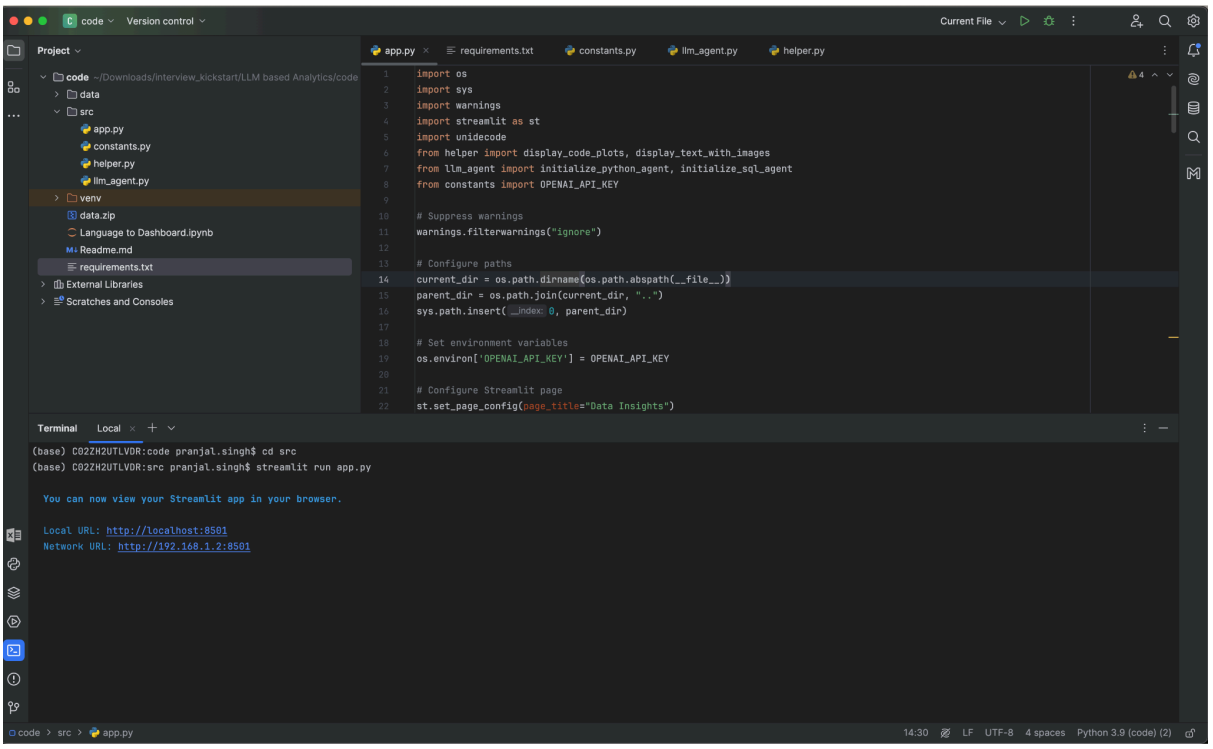
Understanding various modules present.

a. **app.py** - This code sets up a Streamlit web application that interacts with Python and SQL agents for data analysis and visualization. It configures paths and environment variables, initializes session states, defines functions for generating responses based on user inputs, and manages chat interactions with the user through a chat interface that processes and displays messages.

b. **llm_agent.py** - This code configures and initializes Python and SQL agents using the LangChain library for executing tasks related to data processing and querying. It sets up SQL database connections, defines custom instructions for handling queries, and provides functions to create language model agents tailored for specific tasks, such as running Python code or executing SQL queries.

c. **helper.py** - This code provides two functions for a Streamlit app: **display_code_plots** extracts Python code from a text block enclosed in triple

backticks, and **display_text_with_images** displays text interspersed with images, ensuring proper formatting by handling and splitting URLs within the text.



Milestones

- **Data Pipeline and Integration** - Build a robust data pipeline that automatically loads all CSV files into a SQL database. Create relevant databases and tables along with the right schema.
- **Explore LangChain** - Utilise the LangChain toolkit to establish connections with the database for efficient data retrieval. Ensure proper handling of data ingestion, transformation, and storage to facilitate easy access for insights generation.
- **Prompt Engineering** - Explore SQLDatabaseToolkit and various prompt templates to build sql queries and fetch relevant data from the LLM
- **Chat application** - Develop the chat application using streamlit in python and build the capability to retain conversation memory.
- **Dashboard integration** - Create graphs and charts in the streamlit application based on the data provided.

Model

- **gpt-4-0125-preview** - OpenAI
    - GPT-4 is a large multimodal model (accepting text or image inputs and outputting text) that can solve difficult problems with greater accuracy than any of our previous models, thanks to its broader general knowledge and advanced reasoning capabilities. GPT-4 is optimized for chat but works well for traditional completions tasks using the Chat Completions API.

Future Directions

- Connecting to other databases such as MongoDB, Spark
- Deployment of insights as a cron/scheduled process
- Export of insights/queries to PowerBI/Tableau/GDS

Common FAQs

Q1. What type of input and output does the tool expect and provide?
Ans: The tool expects textual input. It provides textual results as well as graphs as output.

Q2. Who can benefit from this system?

Ans: This system is designed for users across all levels of an organization, especially those who may not have technical expertise in data analysis or querying. It empowers non-technical users to access and understand their data, making data-driven decision-making more inclusive.

Q3: What types of insights can the system provide?

Ans: The system can provide a wide range of insights, including statistical summaries, trends, correlations, and data visualizations. The specific insights will depend on the data and the questions asked by the user.

Q4. How are CSV files loaded into the SQL database?

Ans: The data pipeline is designed to automate the process of loading

CSV files into a SQL database. It reads the CSV files, transforms the data as needed, and loads it into the database. This ensures that the data is organized and ready for querying.

Q5. What is the role of the LangChain toolkit in the data pipeline?

Ans: The LangChain toolkit is used to connect the natural language processing component with the SQL database. It facilitates the translation of natural language queries into SQL queries, enabling seamless interaction between the user's input and the database.

Q6. How does the chat-based application work?

Ans: The chat-based application provides an interactive interface where users can type their questions or requests. Using frameworks like LangChain and Streamlit, the application processes the user's input, queries the database, and returns the relevant insights.

Q7. What technologies are used in this project?

Ans: The project utilizes several technologies, including natural language processing (NLP), the LangChain toolkit for connecting with the SQL database, and Streamlit for building the chat-based application. The data pipeline is designed to handle CSV files and SQL databases.

Q8: What are some sample questions which can be asked?

Ans:

- Plot the sales trend over years
- Which channel is working best for us?