OPIM 5894: Special Topics
Generative AI in Business

Final Report:

Potoo Solutions'
AI for BI

Team 5: Likhitha Guggilla, Nishita Rao, Srijanani Achuthan,
Vinay Kiran Reddy Chinnakondu

Executive Summary

This project, in collaboration with Potoo Solutions, aimed to enhance business intelligence by transitioning from static Tableau dashboards to dynamic, conversational interfaces powered by Large Language Models (LLMs). These interfaces enabled users to query structured data in natural language and receive accurate, real-time responses, including visual insights, simplifying data exploration and decision-making processes.

To achieve these goals, two distinct workflows were developed and evaluated. The **Agentic Workflow** used real-time API connections to access and process Potoo Solutions' experimental data for the Wüsthof brand, dynamically retrieving data and leveraging an LLM to generate insights. The **Simple Workflow** focused on structured Excel data, enabling visual data generation and analysis through LLM processing and Python-executed visualizations.

The workflows were evaluated based on **Accuracy Score**, **Faithfulness**, and **Answer Relevancy**, with accuracy further assessed on data extraction, code functionality, and visualization accuracy. Results showed the Agentic Workflow excelled at handling real-time data, while the Simple Workflow proved effective for visual outputs.

The project faced some challenges, such as LLMs struggling with large datasets due to token limits and occasional inconsistencies in responses. Additionally, the Dify platform, though easy to use, lacked certain Python tools, which slowed down some processes. These challenges were managed through careful planning and adjustments.

Future enhancements could focus on handling larger datasets, integrating advanced visualization capabilities, and incorporating real-time customer query documentation. This project demonstrates the potential of AI-driven solutions to improve data access, provide actionable insights, and support smarter decision-making for Potoo Solutions' customers.

Project Goals

The primary objective of this project is to collaborate with Potoo Solutions to implement AI for BI processes. The project aims to enhance data accessibility by transforming static Tableau dashboards into dynamic, conversational interfaces. By addressing analytics-related queries directly from a structured database, this solution seeks to meet the diverse requirements of Potoo Solutions' customers, enabling them to derive greater value from their data and make informed decisions effectively.

To achieve this, advanced Large Language Models (LLMs) will be employed to facilitate seamless natural language interactions with the database. This integration will simplify data retrieval and provide real-time insights tailored to user needs. By combining conversational AI with intuitive data exploration, the project will enhance user engagement and streamline analytics processes, delivering a scalable and efficient solution for dynamic business intelligence.

Scope

To develop the initial prototype of the project, the scope has been refined to focus on creating a functional agentic system for Potoo Solutions. This prototype will enable users to ask questions in natural language and receive insights specifically related to Potoo Solutions’ demo brand, Wüsthof. The system will also integrate a Large Language Model (LLM) to address diverse user needs, delivering accurate responses in both text and visual formats.

Boundaries

The prototype will be designed to answer a predefined set of preliminary questions created by the team about the Wüsthof brand. Real-time customer queries and broader business use cases are intentionally excluded at this stage to allow for a focused and efficient development process. These limitations provide a controlled environment for building and validating the prototype while leaving room for future expansion and enhancement.

Data

The primary data is accessible in two formats namely MS Excel and through API access. The data covers inventory details for Potoo Solutions' experimental brand *Wusthof*.

GenAI application design

To tackle the multiple sources of data, this project explored creating multiple workflows aimed at parsing structured data accurately. The first approach created an agentic system using API to access the entirety of Potoo's experimental data through a live connection node and answer customer queries. The second approach created a workflow using a file upload option to gain access to their static sample database (*.xls* format) and answer customer queries.

Approach 1: Agentic Workflow

The workflow begins with a **Start** node, where four inputs are specified as required: start_date, end_date, brand (*e.g. Wusthof*), and user's question. These inputs form the foundational parameters for the process. Following this, the flow connects to an **HTTP Request** node, which sends a GET request to a specified API endpoint to access Potoo Solutions' experimental data for Wusthof. This request fetches data based on the input parameters from the **Start** node in JSON format.

Thereafter, the output from the **HTTP Request** node is routed to a **JSON Parse** node, which processes the response and ensures the data is in a format suitable for further processing by a suitable LLM. Given the large volume of data accessible from API, this tool helps to filter the database into smaller portions for easier processing by the LLM. In this project, the data was limited to:

Date range: 1 January 2024
Brand: Wusthof
Variables of interest: Date, SKU, Product Name, UPC and Retailer Item Price

Thereafter, the parsed data was passed to a **LLM** (GPT-4o) node, which uses the information to generate text or answers based on the inputs and the API response. In order to fine-tune the responses given by the LLM, the system was prompted to play the role of Data Analyst Assistant. Finally, the processed output from the LLM is sent to an **End** node, where the generated text is outputted as a string.

Image 1: Overview of HTTP Workflow

To further create a customer-friendly interface where the LLM is able to retain memory of conversations with the user, an agentic system was created using the HTTP workflow as a tool to support the LLM resolve customer queries. For better context, the LLM was instructed to play the role of an experienced data analyst with the ability to complete all fundamental and technical data analysis for the inventory data as an expert. To begin any interaction with the user, the chatbot asks the user for their question, a date range for their question including a start date and end date as well as their brand of interest. This allows the user query to match the Start node inputs from the HTTP workflow, thereby successfully activating the workflow.

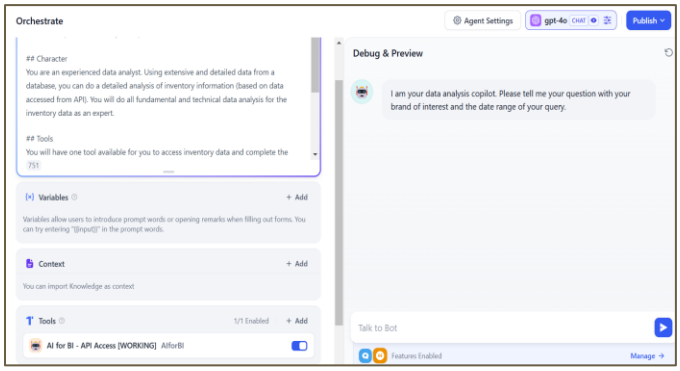
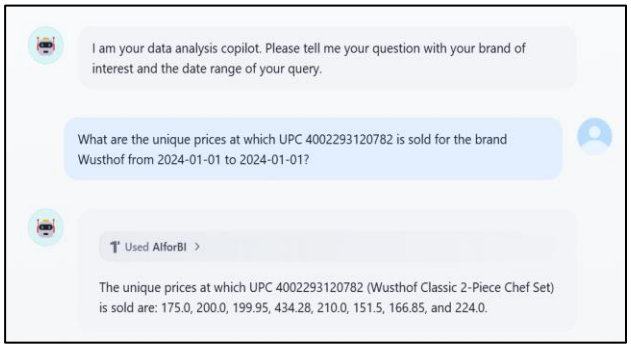


Image 2: Overview of Agentic System

A successful demonstration of the working agentic flow can be seen below. The chatbot is able to successfully identify unique prices at which UPC 4002293120782 is sold for the brand Wusthof from 2024-01-01 to 2024-01-01.



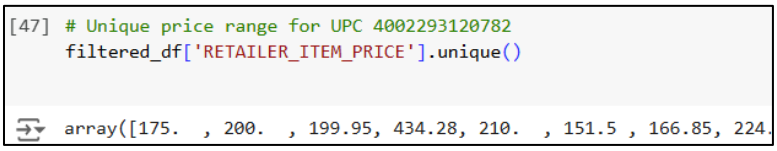


Image 3: Demonstration of chatbot in action (left). Verification of chatbot's response with actual data using Python (right)

Each time a question is asked to the chatbot, the HTTP workflow gets activated as a tool in the agentic system's environment. This allows Potoo's real-time data to be accessed through the HTTP node enabling the LLM to work with the most up-to-date data to process and respond to the user's question.

Approach 2: Simple Workflow

This workflow is designed to enable visual generation functionality using sample data provided by Potoo Solutions in an Excel (.xlsx) format.

The workflow initiates with a **Start node** that requires two inputs: a structured sample data file from Potoo Solutions for Wusthof and a user-provided question. Next, the process advances to a **Doc Extractor** node, which transforms the structured data into text format and passes it to a **LLM (GPT-4o)**.

The LLM leverages the extracted data to generate text or answers tailored to the user's query, which is provided as context. To enhance the relevance and precision of the responses, the LLM is guided to assume the role of a Data Analyst Assistant. Finally, the output generated by the LLM is directed to an **End node**, where the resulting code is delivered as a string.

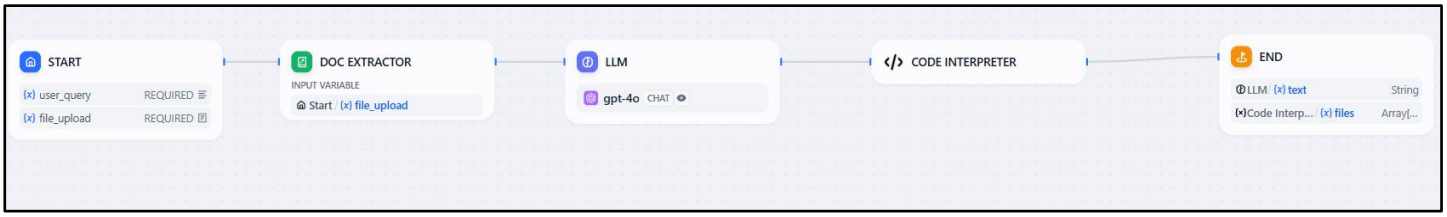


Image 4: Overview of Simple Workflow

In this approach, the variables of interest were:

Brand: Wusthof

Data Format: Excel File (.xlsx)

Key Parameters: User query, structured data fields, visualizations

The **Simple Workflow** was tested using a sample query to demonstrate its capabilities in generating accurate insights and visualizations based on structured data. Below is a detailed example of the workflow's performance.

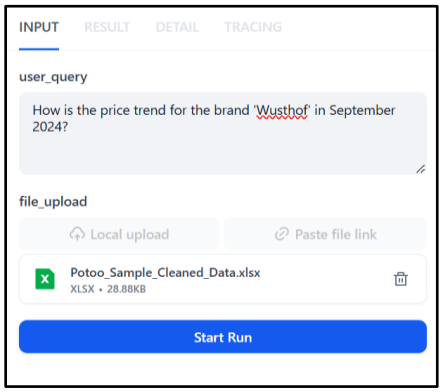


Image 5: Input query for Workflow

The workflow requires two inputs: (1) an uploaded structured data file (Potoo_Sample_Cleaned_Data.xlsx) and (2) the user's query specifying the data of interest.

The Python code generated by the workflow utilized the extracted data to generate a price trend plot for the brand 'Wusthof' during the specified time period (September 2024). The generated Python code was fully aligned with the query and accurately reflected the Excel data.

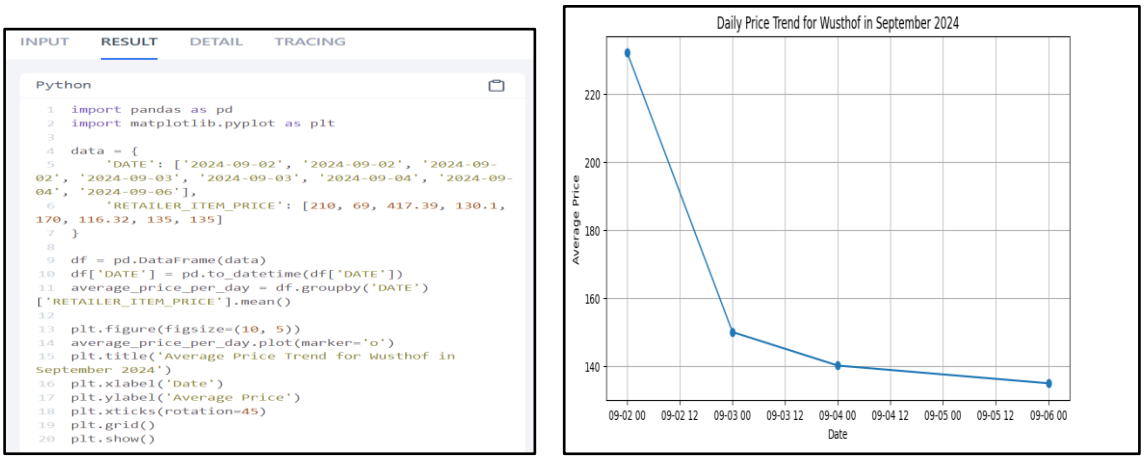


Image 6: Workflow output example and verification

Data Evaluation

Given the nature of the data and the multiple workflows explored, three evaluation metrics were identified to be critical in assessing which type of model works the best for this project.

First, accuracy score was considered which measures the proportion of correct predictions or outputs compared to the total number of predictions. This gives an indication of the reliability of a model in the long term. Second, faithfulness measures whether the answer given by the LLM for each query is factually correct or not. As many systems struggle to address hallucination, this measure helps to identify if hallucinations exist in the environment. Finally, answer relevancy checks whether the LLM retains context when giving responses to any user query. This will help judge whether the LLM understands the question that it is being asked.

For Accuracy Score, three important aspects were considered. First, whether the data extracted by the LLM was correct and matched the input dataset. Second, whether the workflow generated a code that ran successfully without any errors, ensuring it was working as intended. Third, whether the code produced the expected visualization accurately, including correct axes, labels, and graphs that aligned with the user's query.

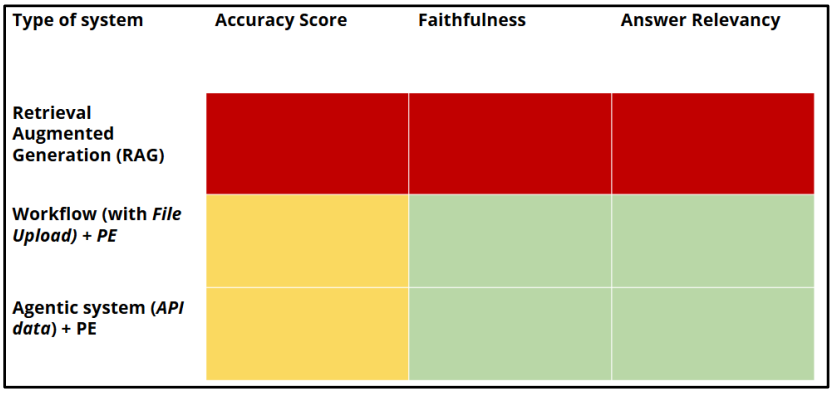


Image 7: Overview of Model Evaluation Results

The evaluation matrix reveals that the **Retrieval Augmented Generation (RAG)** approach struggled across all three metrics. Its limitations in processing structured data in low accuracy and inconsistent faithfulness, making it unsuitable for addressing complex queries. In contrast, the **Workflow with File Upload** approach did much better, especially in providing accurate and relevant answers. It was able to use the structured data effectively to produce meaningful responses. The **Agentic System**, which uses real-time API data, showed slightly better accuracy because it could retrieve and process data efficiently.

This evaluation highlights that both the **Workflow with File Upload** and **Agentic System** approaches worked well for the project. It also shows how important it is to have clear workflows and well-designed prompts to get accurate and reliable results from the system.

Challenges and Limitations

There were a few challenges and limitations faced when completing the project in the Dify environment. First, large language models (LLMs) were unable to process large volumes of data due to their input token constraints and this posed a significant challenge. The LLM struggled to handle the full dataset efficiently, highlighting the need for strategies like chunking data or integrating complementary tools to manage large volumes of information

effectively. To reduce the token usage, the HTTP workflow and agentic system were limited to test data for 1 January 2024 only, which included 560 rows and 5 columns.

Second, ensuring consistency in the LLM's responses proved challenging, even with refined prompting techniques. Variability in the outputs persisted, undermining reliability and suggesting potential limitations in the model's adaptability to specific requirements. This inconsistency underscores the need for further fine-tuning or leveraging external logic layers for critical tasks demanding high accuracy.

Finally, Dify provided a user-friendly platform for deploying workflows with limited coding expertise. However, the experience revealed certain gaps in functionality, such as the lack of pre-installed Python coding modules, which could significantly accelerate processing. While the platform's ease of use is commendable, expanding its feature set would enhance its utility for more complex or data-intensive operations.

Scope for Future Works

To enhance data processing and analysis, efforts can be directed toward incorporating and experimenting with larger volumes of API-accessed data. This approach should aim to leverage real-time and extensive datasets, pushing the boundaries of workflow efficiency and expanding the scope of data-driven insights. Integrating larger datasets effectively will require optimized strategies to handle increased complexity and ensure seamless processing within the existing system architecture.

Another key area of development could involve extending the LLM’s capabilities to generate data visualizations directly within workflows or chatflows. By enabling visual representation of data, users can quickly interpret trends, patterns, and anomalies, facilitating more informed decision-making. This feature would significantly enhance the functionality of workflows by offering a more interactive and insightful user experience.

Finally, incorporating Potoo Solutions’ real-time query documentation for common customer queries can streamline support processes and improve response accuracy.

References

[1] https://potoosolutions.com/

[2] https://docs.dify.ai/learn-more/extended-reading/how-to-use-json-schema-in-dify

[3] https://docs.dify.ai/guides/workflow/node/http-request

[4] https://docs.dify.ai/guides/workflow/node/code