

---

# Potoo Solutions: AI for BI

12/04/2024

## Group 5

Likhitha Guggilla, Nishita Rao, Srijanani Achuthan,  
Vinay Kiran Reddy Chinnakondur

---

# Agenda

1

## Project overview

- a. Scope of project
- b. Project boundaries
- c. Project Approach

2

## Testing Approaches

- a. Approach 1: Agentic system with API data
- b. Approach 2: Workflow with file upload to generate visuals
- c. Demonstration of how it works

3

## Evaluating the LLM

- a. Demonstrate how the workflow passes evaluation criteria
- b. Project's Challenges

4

## Scope for future works

# Project Overview

## Scope

1. Build a working agentic workflow for Potoo Solutions that allows users to ask questions in natural language and get answers about Potoo Solution's demo brand *Wusthof*.
2. Augment LLM to cater to diverse user tastes and answer queries accurately in both text and visual formats.

## Boundaries

- Scope of project may not cover Potoo's real-time customer queries leaving room for project expansion.

**Data Format:** Structured Data (Excel, Database)

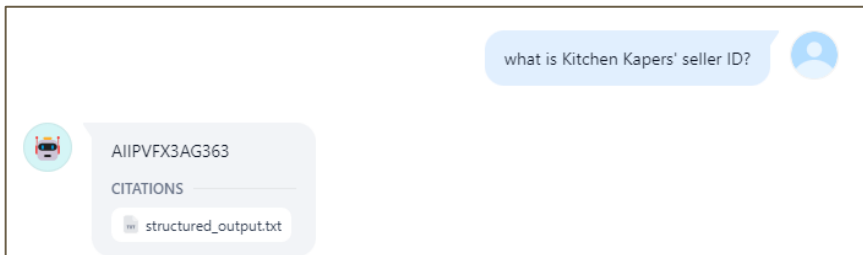
# RAG Suitability

- ❖ RAG (Retrieval-Augmented Generation) is useful for querying and answering questions/prompts supported by **unstructured data**

DATE	UPC	RETAILER_NAME	SELLER_ID
2024-08-24	4002293103464	Kitchen Kapers	AIIPVFX3AG363
2024-09-04	4002293104720	Kitchen Kapers	AIIPVFX3AG363
2024-08-26	4002293103464	Kitchen Kapers	Kitchen Kapers

There are **3 records** of Retailers named “Kitchen Kapers” and **2 distinct** Seller IDs:

- AIIPVFX3AG363
- Kitchen Kapers



**Result:** RAG only produces one result when prompted indicating **failure of interpreting Potoo's structured data knowledge base**

# Approach

- ❖ Created multiple flows: Aim to parse structured data accurately
  1. Agentic system using API
  2. Workflow using File upload

# Approach 1: API Access Agentic System



**Orchestrate**

**Agentic System**

Agent Settings | gpt-4o CHAT | Publish

**Instructions**

# Job description: Data Analysis Copilot

## Character

You are an experienced data analyst. Using extensive and detailed data from a database, you can do a detailed analysis of inventory information (based on data accessed from API). You will do all fundamental and technical data analysis for the inventory data as an expert.

## Tools

You will have one tool available for you to access inventory data and complete the analysis based on the question asked.

### Tool 1: Access and process data using 'AlforBI'

## Workflow

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.

All relevant rows from the original data must be used to respond to the user's question.

75%

**Tools**

1/1 Enabled + Add

AI for BI - API Access [WORKING] AlforBI

**Debug & Preview**

I am your data analysis copilot. Please tell me your question with your brand of interest and the date range of your query.

Talk to Bot

Features Enabled Manage

## Step 1: Create a Workflow

- ❖ HTTP Request node accesses real-time data from Potoo Solution's API in *json*
- ❖ JSON Parse tool parses the data to only feed specified columns/variables to LLM for further processing
- ❖ GPT 4o is used to build LLM to process the real-time data

## Step 2: Build an Agentic System

- ❖ Prompt engineering is used to give the system instructions to perform as a data analysis copilot
- ❖ Workflow is fed as a tool into the agentic system

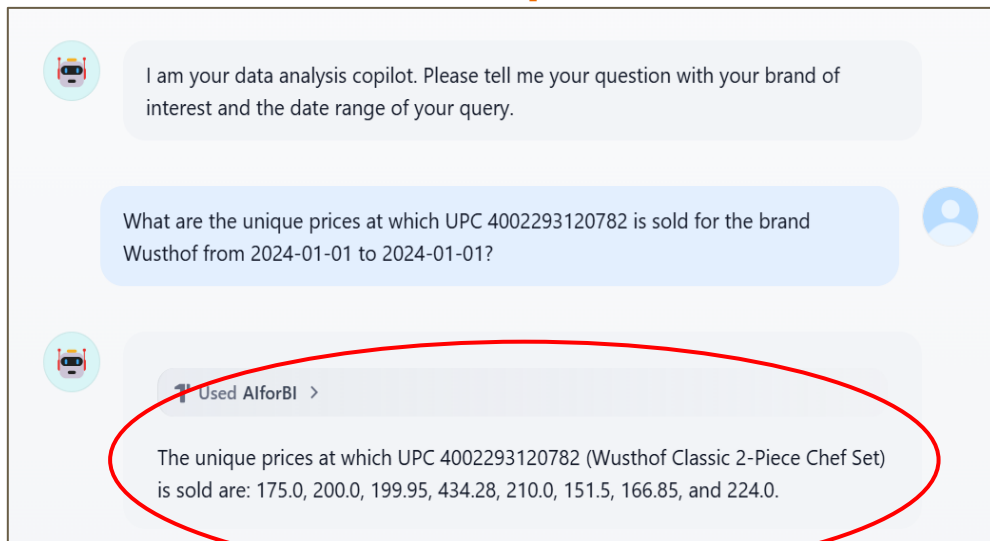
# Performance Example 1

## Key inputs to be entered

- ❖ Date range of query
- ❖ Brand of interest
- ❖ User question

*E.g. "What are the unique prices at which UPC 4002293120782 is sold for the brand Wusthof from 2024-01-01 to 2024-01-01?"*

## Demo question



I am your data analysis copilot. Please tell me your question with your brand of interest and the date range of your query.

What are the unique prices at which UPC 4002293120782 is sold for the brand Wusthof from 2024-01-01 to 2024-01-01?

Used AlforBI >

The unique prices at which UPC 4002293120782 (Wusthof Classic 2-Piece Chef Set) is sold are: 175.0, 200.0, 199.95, 434.28, 210.0, 151.5, 166.85, and 224.0.

## Real answer

```
[47] # Unique price range for UPC 4002293120782
      filtered_df['RETAILER_ITEM_PRICE'].unique()

array([175. , 200. , 199.95, 434.28, 210. , 151.5 , 166.85, 224. ])
```

## Verifying results with Python

- ❖ There are **8 distinct prices** for UPC 4002293120782

# Performance Example 2

## Inputs to be entered

*E.g. "Which UPC has the highest product price for the brand Wusthof from 2024-01-01 to 2024-01-01? What is its product name and what is the price?"*

## Real answer

```
# Find the row with the highest RETAILER_ITEM_PRICE
max_price_row = df.loc[df['RETAILER_ITEM_PRICE'].idxmax()]

# Extract SKU, PRODUCT_NAME, and RETAILER_ITEM_PRICE from that row
sku = max_price_row['SKU']
product_name = max_price_row['PRODUCT_NAME']
retailer_item_price = max_price_row['RETAILER_ITEM_PRICE']

# Print the results
print(f"SKU: {sku}")
print(f"Product Name: {product_name}")
print(f"Retailer Item Price: {retailer_item_price}")
```

```
SKU: 17-105499
Product Name: Wusthof Classic Chef's Knife
Retailer Item Price: 437.1
```

## Demo question

Which UPC has the highest product price? What is its product name and what is the price?

To provide an accurate answer, please specify a date range and a brand of interest. If you want to search across all dates and brands, you can mention that as well.

I am looking at the brand Wusthof for the date range from 2024-01-01 to 2024-01-01

1 Used AlforBI >

The UPC with the highest product price for the brand Wusthof on 2024-01-01 is "17-105499". The product name is Wusthof Classic Chef's Knife, and the price is \$437.10.

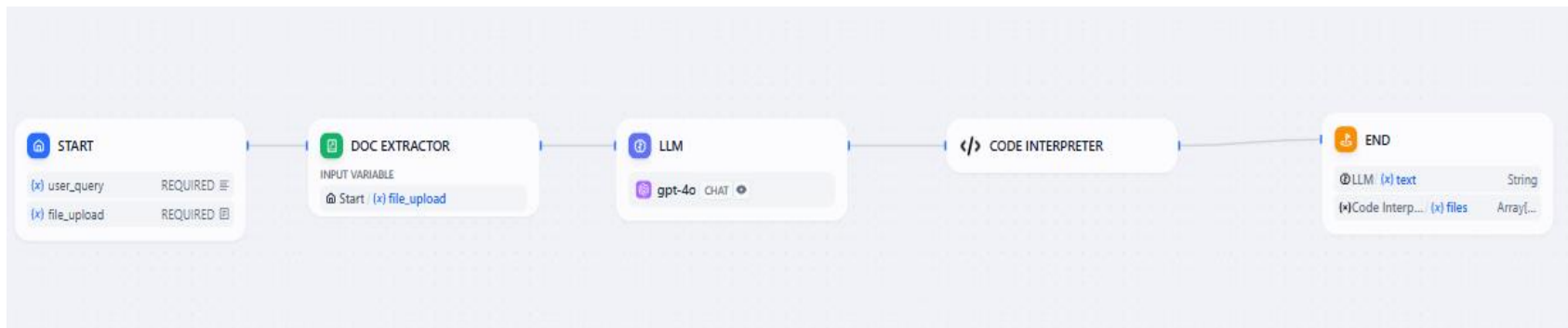
## Verifying results with Python

- ❖ There are **8 distinct prices** for UPC 4002293120782



# Approach 2: File Upload Workflow

## Overview of workflow



- ❖ Workflow approaches breaking down model building by uploading Potoo Solution's sample database (.xls) into the 'Start' node - *Upload file* and *User query*
- ❖ 'Doc extractor' node parses the sample data and feeds it into LLM
- ❖ LLM feeds python code into code interpreter for plot generation
- ❖ Code interpreter executes the code and 'End' node outputs the result

# Performance and Demo

**User Query :** How is the price trend for the brand 'Wusthof' in September 2024?

## Input

INPUT

RESULT

DETAIL

TRACING

user\_query

How is the price trend for the brand 'Wusthof' in September 2024?

file\_upload

Local upload

Paste file link

X

Potoo\_Sample\_Cleaned\_Data.xlsx

XLSX • 28.88KB

Start Run

## Workflow Output

INPUT

RESULT

DETAIL

TRACING

Python

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 data = {
5     'DATE': ['2024-09-02', '2024-09-02', '2024-09-02',
6             '2024-09-03', '2024-09-03', '2024-09-04', '2024-09-04',
7             '2024-09-06'],
8     'RETAILER_ITEM_PRICE': [210, 69, 417.39, 130.1, 170, 116.32, 135, 135]
9 }
10 df = pd.DataFrame(data)
11 df['DATE'] = pd.to_datetime(df['DATE'])
12 average_price_per_day = df.groupby('DATE')['RETAILER_ITEM_PRICE'].mean()
13
14 plt.figure(figsize=(10, 5))
15 average_price_per_day.plot(marker='o')
16 plt.title('Average Price Trend for Wusthof in September 2024')
17 plt.xlabel('Date')
18 plt.ylabel('Average Price')
19 plt.xticks(rotation=45)
20 plt.grid()
21 plt.show()
```

## Excel Data

DATE	RETAILER_ITEM_PRICE	BRAND
2024-09-04	135	Wusthof
2024-09-02	210.00	Wusthof
2024-09-02	69.00	Wusthof
2024-09-02	417.39	Wusthof
2024-09-04	164.10	Wusthof
2024-09-03	130.10	Wusthof
2024-09-03	170.00	Wusthof
2024-09-04	116.32	Wusthof
2024-09-06	135.00	Wusthof

- ❖ **Observation:** The data used in the Python code accurately reflects the Excel data, and the generated code is fully aligned with the user query.
- ❖ **Demo Query:** What is the average price of each unique product sold by brand Wusthof in september 2024?

# Limitations of Dify

Dify **lacks pre-installed modules** like pandas and matplotlib, causing a **ModuleNotFoundError** as shown

```
INPUT  RESULT  DETAIL  TRACING

Failed to invoke tool: Traceback (most recent call last):
File
"/var/sandbox/sandbox-
python/tmp/373be421_4261_400f_a84a_0e870221b874.py", line
48, in <module> File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'pandas' error: exit
status 255

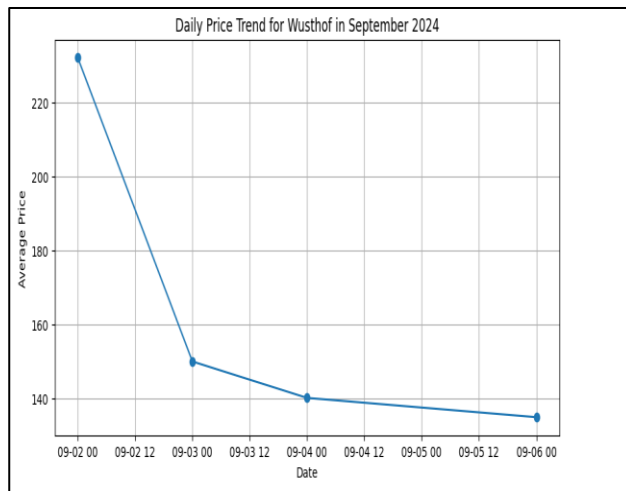
import pandas as pd
import matplotlib.pyplot as plt

data = {
"DATE": ["2024-09-02", "2024-09-02", "2024-09-03", "2024-
09-03", "2024-09-04", "2024-09-04", "2024-09-04", "2024-
09-06"],
"RETAILER_ITEM_PRICE": [210, 69, 130.1, 170, 116.32, 135,
164.1, 135]
}

df = pd.DataFrame(data)
df['DATE'] = pd.to_datetime(df['DATE'])
average_price_per_day = df.groupby('DATE')
['RETAILER_ITEM_PRICE'].mean()

plt.figure(figsize=(10, 5))
plt.plot(average_price_per_day.index,
average_price_per_day.values, marker='o')
plt.title('Daily Price Trend for Wusthof in September 2024')
plt.xlabel('Date')
plt.ylabel('Average Price')
plt.grid(True)
```

The **generated Python code** has been thoroughly tested and verified to **work** seamlessly in the Colab environment.



# Comparative Analysis

Type of system

Accuracy Score

Faithfulness

Answer Relevancy

Retrieval  
Augmented  
Generation (RAG)

Workflow (with *File  
Upload*) + PE

Agentic system (*API  
data*) + PE

# Project's Challenges

## Challenges faced while developing both approaches

- ❖ Difficulty in processing large volume of data at once due to LLM's input token limitations
  - Workflow and agentic system was built to test data from 1 January 2024 comprising 560 rows and 5 columns
- ❖ Consistency of responses by the LLM
  - Refined prompting was not able to circumvent inconsistency in LLM's responses
- ❖ Working in Dify
  - While it is a smooth experience with limited coding, there are many features that are not available (e.g. pre-installed coding modules for Python) for faster processing

# Scope for Future Works

Future direction could primarily focus on:

1. Incorporating and experimenting with larger volume of API accessed data
2. Extend LLM's capabilities to generate data visualizations directly in the workflow/chatflow
3. Incorporate Potoo's real-time query documentation on common customer queries

**Thank you!**