ISR Project Report: AI-Driven Superstore Sales Analysis:

Project Overview:

The project presents an AI-driven dashboard for analyzing a Superstore sales dataset using natural language queries. It combines LangChain, OpenAI's GPT-4, and a Pandas-based data agent to create a conversational data exploration interface. The system enables users to ask questions and get real-time insights from the dataset using natural language.

Objective:

The project aims to develop a natural language interface using GPT-4 and LangChain that enables business users to interact with and extract insights from Superstore sales data without writing any code. It allows questions to be asked in plain English and produces real-time answers, such as statistics, trends, and data-driven decisions.

Technical Stack
Language: Python

Libraries:

- pandas, numpy, openai
- langchain, langchain-openai, langchain-experimental

Platform: Google Colab (for development), Jupyter Notebook.

Model: OpenAI GPT-4 via LangChain.

Dataset: Superstore sales data (9994 records, 21 columns).

Key Functionalities

- Uses create_pandas_dataframe_agent() to link GPT with the DataFrame.
- Handles various query types (Simple, medium, complex)
- Accept user input via natural language and display analysis dynamically.

AI Agent Capabilities
The AI agent supports

- Column inspection
- Statistical summaries
- Group-y operations
- Correlation analysis
- Time based trend evaluations
- Custom visualizations (e.g, line charts for sales trends)

User Interface

- As shown in the dashboard screenshot (Image-2.png), the interface includes:
- A query input field (left panel)
- A dynamically updated QA section (right panel)
- A clean and responsive layout

Project Workflow:

- Installation of required packages (langchain, openai, pandas, etc.).
- API key setup (hardcoded temporarily for testing).
- Data read from a CSV into a Pandas DataFrame.

AI Agent Initialization:

A LangChain agent is created.

```
agent = create_pandas_dataframe_agent(
   ChatOpenAI(model="gpt-4", temperature=0),
   data,
   verbose=True,
   allow_dangerous_code=True,
   agent_type=AgentType.OPENAI_FUNCTIONS,
)
```

This agent acts as the interface between the user's question and the DataFrame.

Ground Truth Development:

To validate the model, a series of hardcoded Python queries (ground truth) were created for comparison:

```
# Example: Ground truth for unique customers
unique_customers = data['Customer ID'].nunique()

# Ground truth for top 5 cities by sales
top_cities = data.groupby('City')['Sales'].sum().nlargest(5)
```

Validation Process Using Ground Truth:

For each natural language query passed to the AI agent, a corresponding **ground truth computation was run manually** in the notebook. The results were then compared for consistency.

Examples of Validation

Question	Ground Truth	AI Agent Response	Match
Total number of unique customers	793	793	yes
Most common sub-category	‘Binders’	‘Binders’	yes
Average discount	0.16	0.156	almost
Top 5 cities by sales	NY, LA, Seattle, SF,	Same	yes

	Philly		
Month with highest sales	11 (November)	11 (November)	yes
Correlation between discount & profit	-0.22	-0.22	yes
Segment with most sales	Consumer	Consumer	yes
State with highest profit	California	California	yes

All tested questions showed full alignment with the ground truth values, confirming the model’s reliable and accurate performance.

Natural Language Query Interface

A function **ask_agent(question)** was defined, allowing users to input English queries such as:

ask_agent("Which state has the highest total profit?")
ask_agent("What is the average profit?")
ask_agent("Find the month with the highest sales volume.")

It calls the LangChain agent and returns a textual response. This simulates a **chatbot-like experience for data analytics**.

Null & Missing Data Analysis

Agent accurately identifies that there are **no null values** in the dataset.

Testing

Validation Through Ground Truth: Every insight obtained from the agent was tested against Pandas-generated results.

Future Improvement

Improve the frontend of the chatbot-like experience for data analytics.

Conclusion:

This project presents a promising approach for integrating large language models like GPT-4 with tabular data analysis using tools such as LangChain. By allowing users to ask natural language questions and receive data-driven insights, the system demonstrates the potential to simplify data exploration for non-technical users. Throughout the process, the results provided by the AI agent were cross-verified against manually computed values, offering confidence in the systems performance for this specific dataset. While further testing and improvements would be necessary for deployment in real-world applications, the current implementation serves as a strong foundation for future work in AI-powered business intelligence.