AI E-commerce Agent: Final Report

1. Introduction

This report outlines the development of an AI-powered agent that enables users to ask questions about their e-commerce data using natural language. The AI agent translates user queries into SQL, executes them on a structured database derived from CSV datasets, and returns insights in both textual and visual form.

2. Objective

The primary goal of this project is to build a conversational AI agent that can:
- Answer natural language questions related to e-commerce datasets.
- Convert the queries into SQL statements and fetch accurate answers.
- Support communication through API endpoints.
- (Bonus) Visualize relevant insights through plots or charts.

3. Datasets Used

The following datasets were used and converted into SQL tables:
1. Product-Level Ad Sales and Metrics
2. Product-Level Total Sales and Metrics
3. Product-Level Eligibility Table

4. System Architecture & Components

The solution integrates the following core components:
- CSV Data Preprocessing and SQLite Database Construction
- LLM Integration (Google Gemini 2.5 API)
- Query Engine: Converts natural language to SQL
- API Layer: Flask-based backend for handling questions
- Frontend: Streamlit application for visualization and interaction

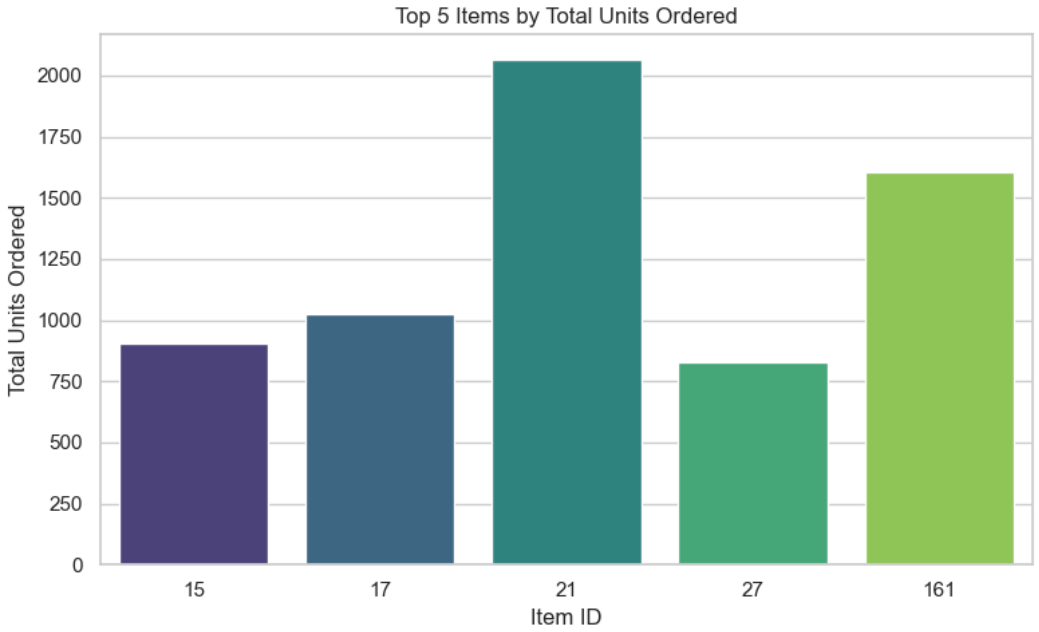
5. Workflow Overview

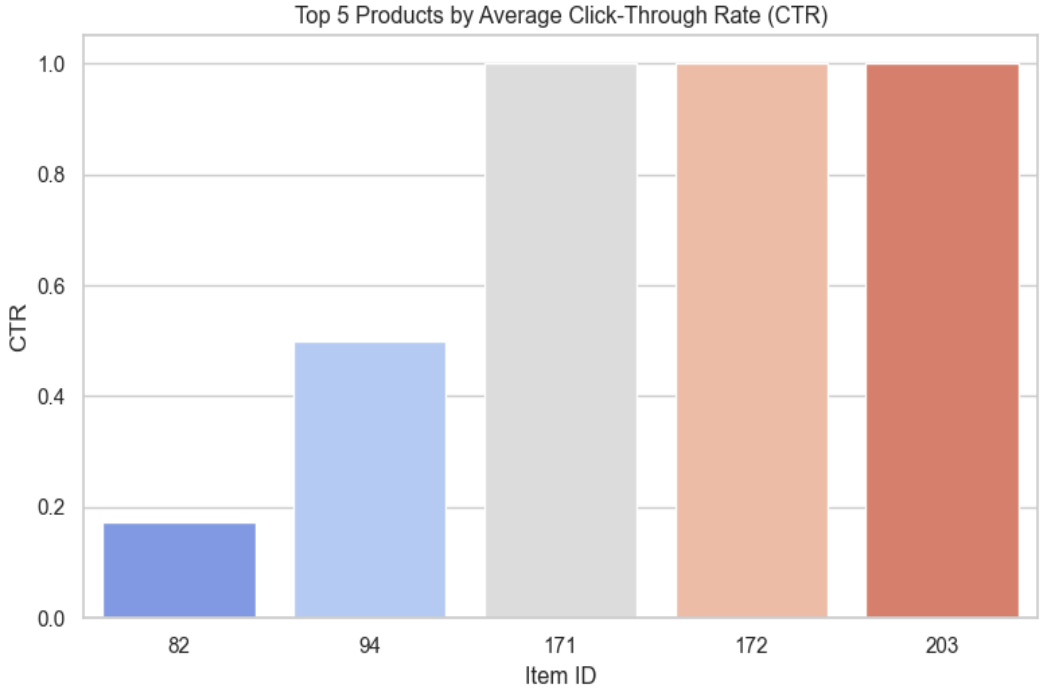
1. Data from CSV files is cleaned and inserted into SQLite tables.
2. User inputs a question on the Streamlit UI.
3. The backend sends the question to the Gemini LLM.
4. Gemini returns a generated SQL query.

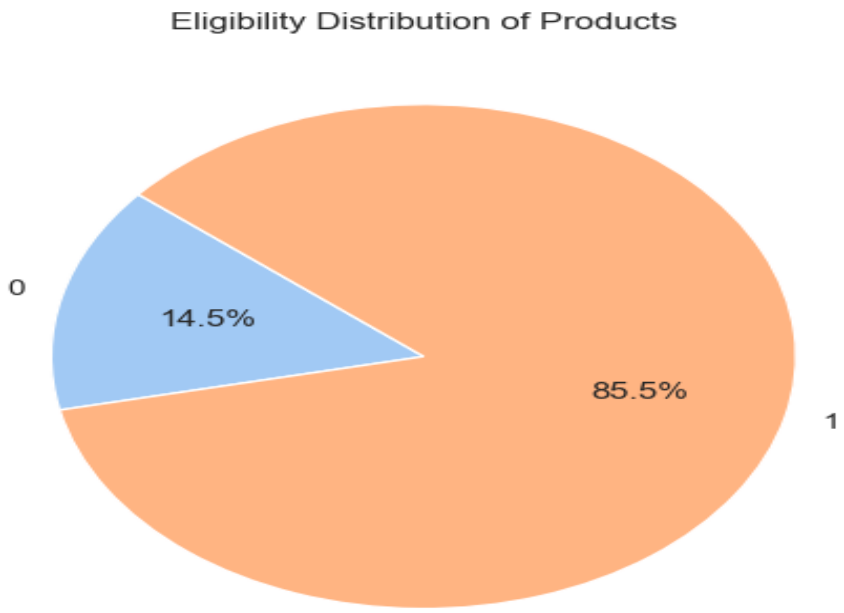
5. SQL query is executed on the database and results are returned.
6. Visuals are generated where applicable (e.g., sales trends, top products).

6. Business Intelligence Visualizations

visualizations generated by the AI agent:







7. Example Questions Answered by the Agent

The AI agent is capable of responding to several business queries, such as:
- Show products with low click-through rate.
- Compare ad spend vs ad sales by date.
- List top 5 items with highest impressions.
- Plot daily trends in unit sales.
- Identify products not eligible for promotion.

8. Technologies Used

- Python 3
- Pandas, SQLite, Matplotlib
- Flask (for backend API)
- Streamlit (for frontend UI)
- Gemini 2.5 (Google LLM for natural language understanding)

9. Challenges Faced

- Mapping ambiguous user questions to accurate SQL queries
- Cleaning inconsistently formatted CSV data
- Designing an intuitive and responsive Streamlit UI
- Handling API request failures and timeouts robustly

10. Conclusion

This AI E-commerce Agent demonstrates the effective integration of Large Language Models with real-world datasets for business intelligence purposes. The system is modular, user-friendly, and capable of offering insights through both textual explanations and visual analytics. It sets the foundation for future enhancements such as multilingual support, predictive analysis, and voice-enabled interfaces.