Hack the Future: A Gen AI Sprint Powered by Data

Data and AI Week

Team details

TEAM NAME : mansishah9865

Team Leader

Mansi Shah
NIT SURAT
mansishah9865@gmail.com
9328348836

Entry Submission Summary

Idea Title	Problem Statement 1: Optimizing Retail Inventory with Multi Agents
Team Name	mansishah9865
Problem Statement	Problem Statement 1: Optimizing Retail Inventory with Multi Agents. The goal is to create a multi-agent AI system that can predict demand, ensure product availability, reduce inventory holding costs, and improve supply chain efficiency. Your solution should enable seamless collaboration among different agents to manage inventory proactively, avoid stockouts, and minimize the excess holding of inventory, thereby maximizing sales and improving operational efficiency.
Proposed Solution	Multi Agent Framework used to develop our Research Agent capable of all sorts of queries handeling and enabling all features to optimise retialInventary with Multi Agents

Proposed Solution Overview

The solution is a comprehensive retail analytics system built using **LlamaIndex** and **Groq's** LLM models. It creates an AI-powered agent that can answer natural language queries about retail data by:

1. Processing multiple retail datasets (demand forecasting, inventory monitoring, pricing optimization)
2. Using specialized query engines for each dataset that convert natural language to executable Python code
3. Employing a multi-agent architecture with three ReAct agents working in parallel:
    - Agent 1: Primary query processor using Gemma 2-9B
    - Agent 2: Secondary processor using Qwen 2.5 Coder
    - Agent 3: Consensus builder that synthesizes results from the other agents

The system presents results through an interactive Streamlit dashboard that shows both final answers and the reasoning process. This approach combines the strengths of different LLMs and allows transparent, step-by-step reasoning for data analysis.

Technologies Used

LLMs:

- Gemma 2-9B-IT hosted on Groq's cloud model for general reasoning
- Qwen 2.5 Coder (32B) hosted on Groq's cloud for code generation and data analysis

Frameworks & Libraries:

- LlamaIndex: For creating query engines and ReAct agents
- Pandas: For data manipulation and analysis
- Streamlit: For interactive web dashboard

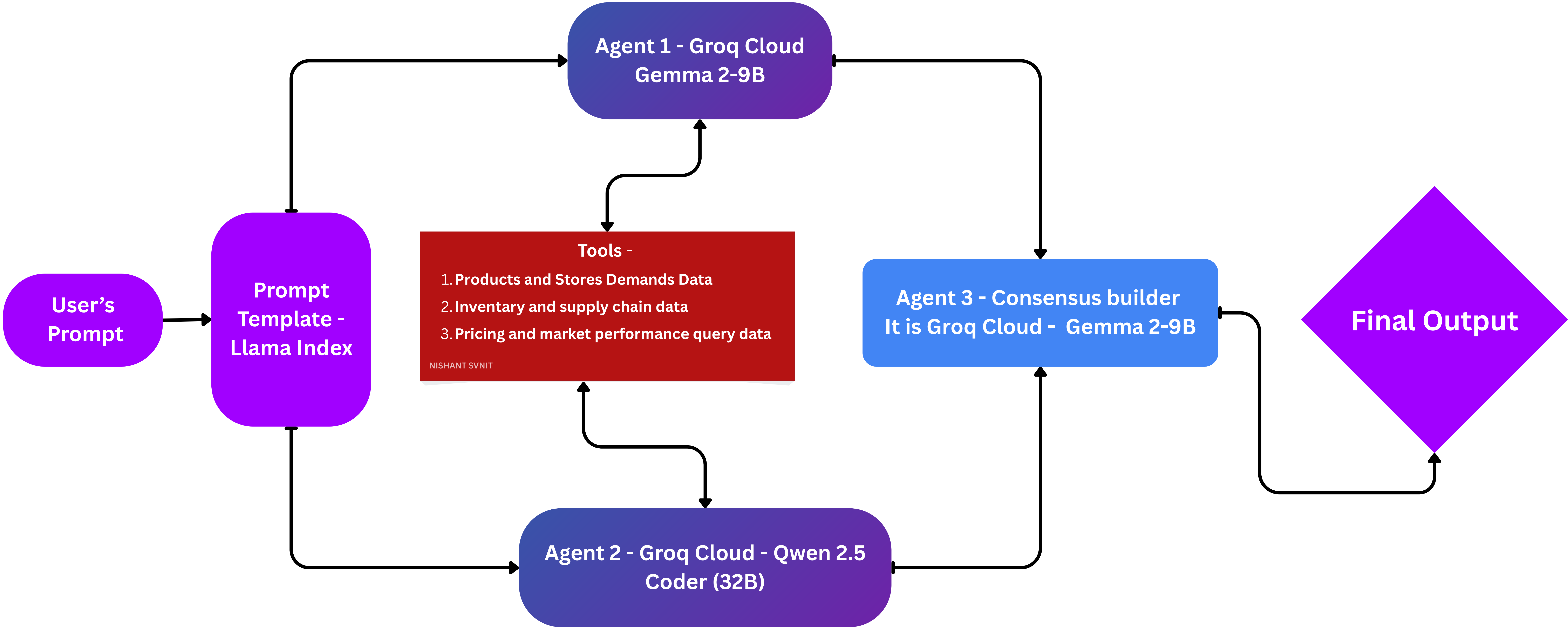
Development Tools:

- Python: Core programming language
- Git: Implied for version control
- Custom prompt templates for enhancing LLM performance

Approaches:

- ReAct framework: For reasoning and acting iteratively
- Chain of thought processing: Making agent reasoning transparent
- Parallelized agent architecture for better results

Agents' interaction design



Code structure

The codebase follows a modular structure:

Data Processing Layer:
CSV data loading via Pandas
Three separate datasets for different aspects of retail analytics

Query Engine Layer:
Three PandasQueryEngine instances (one per dataset)
Custom prompt templates with detailed schema information
Instruction strings to guide code generation

Agent Layer:
Three ReAct agents with different roles
Tool definitions connecting agents to query engines
Verbose output logging for transparency

UI Layer:
Streamlit-based dashboard
Split panel design with chat interface and reasoning display
Custom styling with CSS
Expandable dataset previews

Helper Functions:
format_agent_output(): For cleaning and formatting agent outputs
parse_agent_output(): For processing agent reasoning steps
Session state management for preserving conversation history

Conclusion

This solution effectively addresses retail analytics challenges by combining the power of large language models with structured data analysis. The multi-agent architecture provides more robust and accurate answers by leveraging the strengths of different models and allowing them to compensate for each other's weaknesses.

The **system's key strengths are:**

Accessibility: Complex data analysis is accessible through natural language queries
Transparency: Users can see each step of the reasoning process
Accuracy: Multiple agents working in parallel improve result quality
Versatility: Can analyze patterns across sales, inventory, and pricing data
Scalability: Architecture supports adding new datasets and capabilities

By implementing this solution, retailers can make more informed decisions through natural language interaction with their data, removing technical barriers to insights and empowering business users to perform sophisticated data analysis without specialized training.

References/Other details

- LlamaIndex Main Documentation: https://docs.llamaindex.ai/
- ReAct Agent Documentation: https://docs.llamaindex.ai/en/latest/examples/agent/react_agent/
- PandasQueryEngine Guide: https://docs.llamaindex.ai/en/latest/examples/query_engine/pandas_query_engine/
- Tools Framework Documentation: https://docs.llamaindex.ai/en/latest/module_guides/querying/tools/tools_overview/
- Multi-Modal Agents Guide: https://docs.llamaindex.ai/en/latest/examples/agent/multi_modal_rag/
- Customizing Prompts: https://docs.llamaindex.ai/en/latest/module_guides/prompts/prompt_hub/
- Memory Module Documentation: https://docs.llamaindex.ai/en/latest/module_guides/observability/memory/

Thank You