InsightForge: AI-Powered Business Intelligence Dashboard

Author: Chris Hauser
Date: October 17, 2025

Project Goals

Analyze and visualize multi-dimensional sales data. Generate automated insights using large language models (LLMs). Support interactive exploration of regional, demographic, and product trends. Evaluate the accuracy of AI-generated insights using LangChain tools.

Key Features

Data Management: Uploads and processes a structured CSV file containing sales, region, product, customer demographic, and satisfaction data. Adds derived columns like Month, AgeGroup, and Customer Segments.
Filtering: Users can filter the dataset by region and product. Filter history is tracked in session state.
Visualizations: Includes line and bar charts for monthly trends, region and product performance, gender-based sales, and demographic analysis.
LLM Integration: Uses ChatOpenAI (GPT-4o-mini) to summarize regional, product, satisfaction, and demographic trends.
RAG Integration: Constructs a FAISS vectorstore from structured documents and retrieves relevant context for LLM prompts.
Evaluation: Uses LangChain Evaluator to assess LLM output based on correctness. Ragas support is scaffolded for advanced evaluation.

LLM Prompt Engineering

InsightForge uses a structured prompt template with variable slot-filling and formatting constraints. It includes filters for Region and Product and inserts context and statistics into the prompt for consistent summaries.

Metrics Calculated

All metrics are computed in parallel using ThreadPoolExecutor. Metrics include sales by month, sales by region, satisfaction, gender-based sales, product comparison, demographic summaries, and customer segmentation.

Technologies Used

UI: Streamlit. Backend: pandas, matplotlib, seaborn. AI: LangChain, OpenAI, FAISS. Evaluation: LangChain Evaluator, optional Ragas.

Usage Instructions

From the project root: activate the virtual environment and run the app using 'streamlit run app.py'. Ensure a valid OpenAI API key is set in the .env file.

Error Handling

Each metric is wrapped in try-except blocks. Vectorstore creation is exception guarded. Summaries fallback to static outputs if generation fails. Filter and summary histories are tracked with limits.

Future Improvements

Enable full Ragas evaluation. Add file upload for custom CSVs. Introduce memory support for session continuity. Package the app as a deployable web service with authentication.