Chat-Based Semantic Analysis of Tabular Data using LLMs

Abstract

This project explores the use of Large Language Models (LLMs), specifically GPT-3.5-turbo, to analyze structured CSV data using natural language. By applying a semantic layer on top of a heart disease dataset, we enable intuitive and flexible interactions between users and data through conversational queries. The project aims to demonstrate how LLMs can improve data accessibility, generate insights, and support non-technical users in data analysis workflows.

Introduction

Traditional data analysis requires knowledge of query languages or programming (e.g., SQL, Python). This project eliminates that barrier by enabling users to ask questions in plain English using a semantic layer that adds context to raw tabular data. We leverage LLMs via PandasAI to process natural language queries and return textual and visual insights.

Motivation

- Simplify structured data analysis through natural language
- Enable non-programmers to explore data using LLMs
- Demonstrate semantic enrichment of datasets for AI interpretation
- Bridge the gap between raw data and intuitive decision-making tools

Previous Work

- SmartDataframe interfaces with GPT for natural language queries (PandasAI v1)
- SQL-to-NL and NL-to-SQL tools
- LLM-powered BI tools (e.g., Tableau GPT, ChatGPT Code Interpreter)
- This project differs by integrating semantic context directly into the dataset layer for enhanced accuracy and reuse

Problem Formulation

- **Input**: A heart disease dataset in CSV format with semantic metadata describing each column
- **Process**:
    - Attach a semantic layer to the DataFrame
    - Use an LLM (GPT-3.5-turbo) to interpret natural language queries
- **Output**: Text-based responses and visualizations based on user queries
- **ML Task**: LLM-driven semantic analysis and visualization (zero-shot prompting)

Datasets

- **Source**: Heart Disease Dataset (public CSV)
- **Size**: 300+ samples, 13 features
- **Preprocessing**:
    - Rename and define each column via metadata
    - Normalize value formats where needed

Tools & Libraries

- OpenAI API (GPT-3.5-turbo)
- PandasAI (with LiteLLM backend)
- pandas, matplotlib, seaborn
- Jupyter Notebook

Expected Performance

- **Accuracy**: Reliable interpretation of most well-formed queries
- **Visualization**: Clear charts rendered by LLM instruction (e.g., scatter, bar, histogram)
- **Interpretability**: Enhanced by the semantic layer guiding LLM responses
- **Usability**: Reusable chat interface with rich user prompts and flexible exploration