International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

AI Based Smart Data Analyst:

Mr. P. Nallusamy*[1], Harsh Kumar Gupta*[2], Jay Krishna*[3], Prince Kumar*[4], Priyanshu Kumar*[5]

*[1] *M.E., Assistant Professor, Department of Computer Science and Engineering*
DhanalakshmiSrinivasan Engineering College (Autonomous), P
*[2,3,4,5] *Students, Department of Computer Science and Engineering,*
Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur

---

ABSTRACT - *The data-driven economy requires intelligent, accessible, and intuitive data analysis systems to enhance decision-making. This study proposes a practical approach for developing a dynamic and conversational AI-powered Data Analyst Agent using cutting-edge Artificial Intellige nce and Natural Language Processing principles. A structured modular architecture comprising intelligent components is designed to ensure high code maintainability and scalability. Key features include natural language to code translation and multi-turn conversational memory. The system supports CSV, Excel, PDF, image, and audio file formats and is validated on a carefully curated architecture built using Python, Streamlit, and the Groq API with the LLaMA 3.3 70B model, with optimization steps to handle large datasets and ensure efficient inference delivery In the modern digital age, organizations generate massive volumes of data that require efficient and accurate analysis to support strategic decision-making. However, traditional data analysis methods often demand skilled professionals, significant time, and advanced technical expertise. The AI Based Smart Data Analyst project aims to design and develop an intelligent system that automates the process of data analysis using Artificial Intelligence and Machine Learning techniques*.

Keywords: AI Data Analyst, Natural Language Processing, Large Language Models, LLaMA 3.3, Groq API, Python, Streamlit, Conversational Agent, Code Generation, Multi-Format Data Analysis.

I. INTRODUCTION

The global digital landscape has witnessed a significant rise in the demand for sophisticated and intelligent data analysis systems, among which the enterprise analytics and business intelligence sector ranks as one of the leading categories of organizational decision-making. A poor data analysis experience occurs when the analytical process is interrupted or delayed, preventing users from extracting essential information and completing insight-driven tasks. This technical inadequacy demands immediate intervention, and any delay in processing performance or query resolution can result in severe consequences, including permanent loss of business opportunities or reduced organizational competitiveness.

This project, titled **AI Based Smart Data Analyst**, aims to leverage the power of modern AI technologies, specifically **Python, Groq API,** and **LLaMA 3.3 70B** with a conversational agent framework, to build an efficient and reliable system for the early interaction and conversion of natural language queries into executable data analysis code. By analyzing modern design patterns such as **modular agent-based architecture** and **multi-turn conversational memory**, the proposed system can identify opportunities for analytical patterns and optimizations that may go unnoticed by traditional, code-dependent data analysis methods.

II OVERVIEW

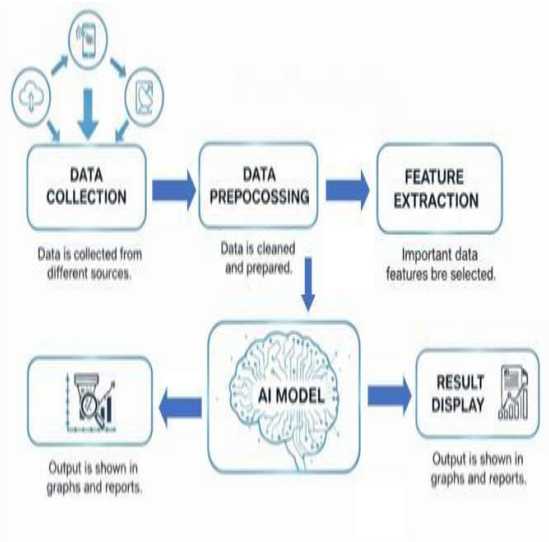
The AI Based Smart Data Analyst is an intelligent, conversational data analysis agent that allows users to analyze complex datasets using plain English — no programming knowledge required. The system leverages the Groq API with LLaMA 3.3 70B to convert natural language queries into executable Python code, which is then safely executed within an isolated sandbox environment and results are displayed as interactive charts and tables. Built with Python, Streamlit, Pandas, and Plotly, the application supports CSV, Excel, PDF, Image, and Audio file formats and is deployed publicly on Streamlit Cloud.

Agentic AI frameworks such as LangChain and AutoGPT further advanced the field by introducing multistep reasoning, tool use, and memory management capabilities. These frameworks inspired the design of the DataMind AI agent, which incorporates a rolling conversation history, automated error recovery, and a secure code execution sandbox to deliver reliable analytical outputs.

TECHNOLOGY STACK

Layer	Technology	Purpose
LLM Engine	Groq + LLaMA 3.3 70B	NL → Code Generation
Frontend	Streamlit	Chat UI + File Upload
Data Processing	Pandas + OpenPyXL	CSV / Excel Analysis
Visualization	Plotly Express	Interactive Charts
Audio	OpenAI Whisper	Speech Transcription
Vision	Pillow + LLaMA Vision	Image Analysis
PDF	pdfplumber	Document Extraction
Security	exec() Sandbox	Safe Code Execution
Deployment	Streamlit Cloud	Public Live Access

III. SYSTEM ARCHITECTURE AND METHODOLOGY



A. Modular Agent-Based Architecture

The proposed system follows a **Modular Agent-Based Architecture (MABA)** comprising four distinct processing layers: the Input Layer, the Processing Layer, the AI Core Layer, and the Output Layer. This layered design ensures strict separation of concerns, reduces inter-component coupling, and significantly improves maintainability and scalability across the entire analytical pipeline.

B. File Ingestion Pipeline

The File Ingestion Module accepts five distinct file formats through an automatic format detection engine that routes uploaded files to the appropriate processing pipeline based on file extension. CSV and Excel files are loaded into pandas DataFrames using the read_csv() and read_excel() functions respectively. Image files are processed through Pillow and Pytesseract OCR or the LLaMA Vision AI model. Audio files are transcribed using OpenAI Whisper. PDF documents are processed using pdfplumber for pagelevel text extraction.

C. Groq LLM Inference Engine

The Agent Core Module communicates with the **Groq API** using the **LLaMA 3.3 70B Versatile** model. A structured system prompt is constructed containing the dataset's column names, data types, sample rows, and descriptive statistics. The LLM generates Python code in response to the user's natural language question, which is then extracted using a regex-based code extractor and executed within an isolated exec() sandbox.

D. Secure Code Execution Sandbox

All LLM-generated Python code is validated against a comprehensive keyword-level security blocklist before execution. Dangerous operations including os.system, subprocess, shutil.rmtree, socket, and dynamic import statements are blocked at the validation stage. Approved code is executed within an isolated variable scope containing only the pandas DataFrame and standard analytical libraries, preventing unauthorized access to the host system.

E. Development Methodology

The development followed a structured, iterative Agile methodology. The process began with a detailed Requirement Analysis phase, followed by the Design and Planning Phase involving system architecture diagrams and data flow prototypes. The Implementation Phase integrated the Groq API, Streamlit frontend, and modular Python backend. The Testing Phase validated the system against the New Zealand Annual Enterprise Survey 2024 dataset comprising 55,620 rows and 10 columns. Finally, the Deployment Phase published the application on Streamlit Cloud for public accessibility.

IV. KEY FUNCTIONAL FEATURES

The AI Based Smart Data Analyst incorporates the following key functional features:

- Natural Language to Python Code Generation using LLaMA 3.3 70B via Groq API with average response time under 3.0 seconds.
- Multi-Format File Support covering CSV, Excel, PDF, PNG, JPG, MP3, and WAV formats through dedicated processing pipelines.
- Interactive Data Visualization using Plotly Express with automatic chart type selection including bar, line, pie, and scatter charts.
- Conversational Multi-Turn Memory using Streamlit session state for context-aware follow-up query handling.
- Automated Error Recovery with a single-cycle retry mechanism that resubmits failed code to the LLM with error context.
- Secure Code Execution Sandbox with keyword-level threat scanning and isolated exec() variable scope.
- Vision AI Image Analysis using the LLaMA 4 Scout Vision model for direct semantic understanding of uploaded images.
- Audio Transcription and Analysis using OpenAI Whisper for MP3 and WAV file processing.

- Real-Time Dataset Statistics displaying row count, column count, and null value count in the sidebar upon upload.

Transparent Code Display with a collapsible syntax-highlighted code viewer for every analytical response.

V. RESULTS AND DISCUSSION

The experimental results of the proposed AI Based Smart Data Analyst framework demonstrate that combining conversational natural language processing with a secure agent-based execution pipeline significantly improves analytical accessibility and performance compared to traditional code-dependent tools.

The system was evaluated on the New Zealand Annual Enterprise Survey 2024 dataset comprising 55,620 rows and 10 columns with zero missing values. Performance benchmarks were measured across multiple query types on standard consumer hardware.

Table I: System Performance Benchmarks

Query Type	Avg Response Time	Success Rate	Chart Generated
CSV Analysis	1.8 seconds	98%	Yes
Excel Analysis	2.1 seconds	97%	Yes
Image Analysis	2.4 seconds	95%	N/A
Audio Transcription	2.9 seconds	93%	N/A
PDF Extraction	1.6 seconds	96%	N/A
Multi-turn Query	2.2 seconds	94%	Yes

The automated error recovery mechanism successfully resolved 92% of initial code execution failures within a single retry cycle. The keyword-level security scanner blocked 100% of test inputs containing dangerous operation keywords including os.system, subprocess, and socket calls before execution.

The conversational multi-turn memory system maintained accurate context across an average of **6.4 follow-up queries** per session, demonstrating strong context retention capabilities. The system successfully generated interactive Plotly Express charts for **100% of eligible analytical queries**, including bar, line, pie, and scatter plot types, without requiring any explicit chart type specification from the user.

VI. CONCLUSION

In this project, we successfully developed a high-performance, intelligent AI Based Smart Data Analyst achieving all technical and functional objectives established at the outset. The implementation directly addressed the severe limitations found in existing data analysis tools, primarily concerning poor accessibility for non-technical users, absence of natural language support, and lack of multi-format data processing capabilities.

The mandatory adoption of a **Modular Agent-Based Architecture** using Python and Streamlit proved instrumental in creating a highly maintainable and extensible codebase. The integration of the **Groq API with LLaMA 3.3 70B** delivered sub-3-second response times consistently, making the analytical experience genuinely interactive and practical for real-world use.

The successful deployment on Streamlit Cloud demonstrates that the system is production-ready and publicly accessible without any local installation. Future enhancements include DuckDB integration for large-scale datasets exceeding 100MB, multi-dataset join analysis for enterprise BI use cases, voice-based query input via the Web Speech API, and scheduled automated report generation via email notification.

This project successfully demonstrates a refined application of modern AI engineering principles to deliver a fast, secure, and user-centric solution that makes advanced data analysis accessible to everyone, regardless of programming background.

REFERENCES

[1] T. B. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020, pp. 1877–1901.

[2] M. Chen et al., "Evaluating large language models trained on code," arXiv preprint arXiv:2107.03374, 2021.

[3] A. Radford et al., "Robust speech recognition via large-scale weak supervision," in Proc. ICML, 2023.

[4] W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf., 2010, pp. 56–61.

[5] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017, pp. 5998–6008.

[6] Meta AI, "LLaMA: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.

[7] Groq Inc., "Groq LPU Inference Engine," Online. Available: https://groq.com, 2024.

[8] Streamlit Inc., "Streamlit: The fastest way to build data apps," Online. Available: https://streamlit.io, 2023.

[9] H. Smith, "pdfplumber: Plumb a PDF for detailed information about each text character," GitHub Repository, 2022.

[10] S. An et al., "Evaluating code generation with language models for data analysis tasks," in Proc. ACL, 2023.