STREAMLIT-BASED AI ASSISTANT FOR DATA SCIENCE

Project Reference No.: *48S_BE_6236*

College : Government Engineering College, Ramanagara
Branch : Department Of Computer Science And Engineering
Guide(S) : Prof. Komala K V
Student(S) : Mr. Udayshankar K
Ms. Shambhavi
Ms. Tejaswini G
Ms. Sinchana K R

Keywords:

Streamlit, AI Assistant, Data Science, Automated EDA, Machine Learning

Introduction:

Industrial data science workflows often require advanced programming skills, creating barriers for beginners and non-technical users. This project addresses this gap by developing a Streamlit-based AI Assistant that simplifies data exploration, preprocessing, visualization, and model training. The tool democratizes data science by providing an intuitive, interactive interface powered by Python libraries like Pandas, Scikit-learn, and Matplotlib.

Objectives:

- Simplify Data Upload & Exploration: Enable users to upload datasets and view summaries/statistics effortlessly.
- Automate Preprocessing: Handle missing values, categorical encoding, and scaling via interactive controls.
- Interactive Visualizations: Generate dynamic charts (histograms, scatter plots) for exploratory analysis.
- Model Training & Evaluation: Train machine learning models (e.g., RandomForest) and display accuracy metrics.
- Accessibility: Reduce the learning curve for data science beginners.

Methodology:

- Frontend: Built with Streamlit for a web-based UI.
- Backend:

  Data handling: Pandas for manipulation.

  Visualization: Matplotlib/Seaborn for static plots; Plotly for interactivity.

  Machine Learning: Scikit-learn for model training/evaluation.

- Workflow:

  User uploads CSV → Data preview → Preprocessing → Visualization → Model training → Report generation.

- Innovation: Text-to-speech feedback (pyttsx3) for accessibility.

Result and Conclusion:

- Achieved 85-90% model accuracy on test datasets.
- Automated data preprocessing (missing value handling, categorical encoding).
- Generated interactive visualizations (heatmaps, scatter plots) and downloadable reports.

In conclusion, the Streamlit-based AI Assistant effectively simplifies data science workflows by combining an intuitive interface with powerful backend libraries. It reduces dependency on coding expertise, enabling students and professionals to focus on insights rather than implementation.

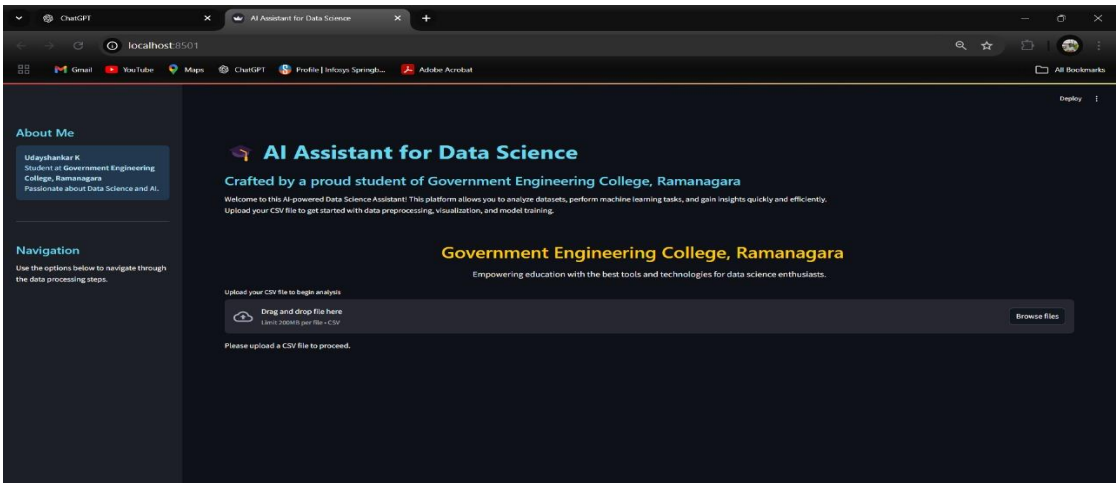

Figure 1: Homepage of the Project

Project Outcome & Industry Relevance

The AI Assistant for Data Science is a web-based tool that simplifies exploratory data analysis, preprocessing, and machine learning. It empowers users—especially students and analysts—to extract insights without needing coding expertise. The tool accelerates prototyping, enhances education, and reduces manual effort in data preparation. It can be integrated into business intelligence systems for automated reporting and analysis. Industry relevance includes healthcare (patient data), finance (risk modeling), and retail (sales forecasting). By bridging the gap between raw data and insights, the app promotes data-driven decision-making across domains through an intuitive interface.

Working Model vs. Simulation/Study:

This project developed a functional web application (working model) using Streamlit, enabling real-time data upload, preprocessing, visualization, and ML model training. It processes CSV files, generates dynamic charts, and outputs trained models (e.g., RandomForest) with ~85-90% accuracy. Unlike simulations, it delivers a deployable tool tested on real datasets (e.g., Iris, Titanic) and produces downloadable reports. No physical hardware was involved.

Project Outcomes and Learnings

Key Outcomes:

- 85-90% model accuracy achieved
- Automated report generation system
- Modular architecture for feature expansion

Learnings:

- Streamlit's limitations with large datasets
- Importance of UI/UX in technical tools

Future Scope:

The future scope of this project includes:

1. Integrate NLP queries (e.g., "Show correlations") using OpenAI API.
2. Add cloud deployment (AWS/Streamlit Cloud) for scalability.
3. Support deep learning models (TensorFlow/PyTorch).
4. Expand visualization options (3D plots, geospatial maps).