International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Cleansight, An Interactive AI-Based Tool for Refined Data Cleaning and Analysis

Prishita Matreja [1], Akshat Diwan [2], Shreya Garg [3] , Dr. Rakesh Kumar Arora [4]
Bhagwan Parshuram Institute of Technology, Delhi
prishita.matreja@gmail.com, akshatdiwan31@gmail.com, shreyagarg12347@gmail.com,
dr.rakeshkarora@gmail.com
[1],[2],[3] – Students in BTech. CSE-DS at Bhagwan Parshuram Institute of Technology, Delhi
[4] – Professor at Bhagwan Parshuram Institute of Technology, Delhi

Abstract: Data quality is a fundamental prerequisite for reliable analysis of data and machine learning. Raw datasets in real-world scenarios routinely suffer from missing values, outliers, duplicate records, and inconsistent formatting, all of which severely degrade downstream model performance and analytical conclusions. This paper presents CleanSight, an interactive AI based data profiling and cleaning tool developed using Python, CSS and Streamlit. The system integrates automated CSV/Excel ingestion, column wise profiling, intelligent IQR based outlier detection, user chosen missing value imputation, duplicate removal, inconsistency detection, adaptive visualization and report generation all within a unified workflow. A hybrid AI architecture combines deterministic local query engine for dataset specific computations with Groq hosted Large Language Model fallback used for unresolved natural language questions. By resolving supported queries locally, the architecture reduces unnecessary API calls and reduces response latency. The AI layer uses structured dataset context rather than raw records and uses a rule based fallback for summary generation, when the external model is unavailable. After evaluation on three real world datasets including a dataset with 541,909 rows, showed successful handling of missing values while retaining 95–99% of rows. The results demonstrate that CleanSight provides an effective and accessible workflow for interactive data quality improvement, adaptive visualization, AI-assisted analysis, interactive cleaning and report generation.

Index Terms: *Data Profiling, Data Cleaning, Streamlit, Outlier Detection, Missing Value Imputation, Interquartile Range, Exploratory Data Analysis, Data Quality, Data Inconsistency Detection, Conversational AI, Large Language Models, Python, Plotly*

I. INTRODUCTION

The exponential growth of data across industries has made data quality management one of the most critical challenges in modern data science and analytics. Organizations increasingly rely on structured datasets for making decisions, forecasting and machine learning model development but there is still a lot of unprocessed and raw data. The real world data is often incomplete, inconsistent and noisy. The studies indicate that data scientists spend approximately 60% to 80% of their time on data preprocessing tasks such as handling missing values, detecting outliers and removing duplicates. Despite this widespread need, accessible, interactive and domain agnostic tools for automated and interactive data cleaning are limited.

For traditional data preprocessing there should be proficiency and knowledge in programming languages such as Python thus it acts as a significant barrier for non technical stakeholders, business analysts and the students as well as requiring a lot of time. Although several commercial tools such as OpenRefine, Trifacta and Talend provide automation, they are often expensive, complex to configure, or restricted in flexibility. This shows the need for a lightweight, open source and a user friendly solution that helps in data quality management.

The existing tools may provide profiling, cleaning, visualization or reporting individually, but there is need for an integrated workflow that has interactive data quality correction, adaptive visualization, interactive cleaning, structured context AI assistance, fault tolerant AI summarization and report generation.

To address these challenges, this paper presents the **CleanSight** tool, an interactive web based application developed using Streamlit framework in Python. The proposed system integrates multiple stages of the data preprocessing pipeline into a single cohesive interface including- file ingestion, statistical profiling, outlier detection and treatment, interactive missing value imputation, duplicate removal, data quality and inconsistency detection, AI generated insights, conversational assistant, visualization and report generation with cleaned dataset. The step by step workflow ensures that users can perform complete data cleaning operations without requiring programming expertise.

The key contributions for users and society of this web based application are as follows: (1) A unified data quality workflow that integrates profiling, issue detection, user controlled cleaning, visualization, AI-assisted analysis, and reporting within a single interactive environment;

(2) A context aware cleaning mechanism that checks columns before applying statistical operations and provides multiple user selectable treatments for the detected outliers, including IQR based handling, row removal, mean or median replacement and retention of the original values of them;

(3) A rule based categorical inconsistency detection and standardization mechanism that identifies case or whitespace variations, word to number representations and heterogeneous date formats and helps in handling that by making it standardized across the whole columns just by selecting the detected ones;

(4) An adaptive visualization mechanism that classifies columns according to data type, cardinality, identifier characteristics and variability to automatically select suitable visualization candidates while supporting user controlled generation of the charts;

(5) A structured context AI assistance layer that gives dataset specific responses using the computed dataset summaries rather than directly exposing complete raw records to the language model;

(6) An automated generation of report in the PDF format with visual analytics and being able to download cleaned dataset in CSV or Excel format as required by the user.

The rest of the paper is organized as follows. Section II has reviews related work. Section III describes the system architecture and methodology. Section IV shows implementation details. Section V discusses experimental results. Section VI provides a comparison with existing tools and outlines limitations and future work. Finally, Section VII helps conclude the paper.

II. RELATED WORK

Data cleaning and data profiling is widely studied in database systems and for data mining [1], [11], [12]. More recent studies focused on scalable techniques for detecting and correcting data inconsistencies in the uploaded large datasets [1], [8].

Handling missing values is main step in data preprocessing. Simple techniques like mean and median imputation are commonly used because of their efficiency and ease of implementation [13], [14]. More advanced techniques are the Multiple Imputation by Chained Equations (MICE) and k-Nearest Neighbour (KNN) imputation [13], [14] that provide with better statistical accuracy but cause higher computational complexity. The proposed CleanSight tool focuses on simple and efficient imputation methods to support real time usability.

Outlier detection techniques can be broadly categorized into statistical and machine learning based approaches. The Interquartile Range method is widely used due to its robustness and its simplicity [2], [15], [18]. Advanced methods such as Isolation Forest [20] and Local Outlier Factor provide better detection for complex datasets but are computationally expensive and are less suitable for interactive systems [2].

Several tools exist for data cleaning and profiling like OpenRefine that provides a graphical interface for data transformation but lacks in integrated statistical profiling. Pandas Profiling [8] generates automated exploratory reports but it does not support interactive data cleaning. Commercial tools such as Trifacta offer advanced features but are often costly and complex [11]. The CleanSight tool addresses these limitations by integrating profiling, interactive cleaning, report generation and cleaned dataset exporting capabilities into a single lightweight and open source solution [4], [7].

Streamlit [19] is a popular framework for building interactive data applications. It allows rapid development of web based interfaces having data visualization and real time interaction [5], [6]. However, comprehensive end to end data cleaning systems built using Streamlit are relatively limited which motivates the development of the proposed system.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The CleanSight tool has a stage based pipeline structure as illustrated in Figure 1. The workflow pipeline consists of eight major stages: (1) File Ingestion (2) Statistical Profiling (3) Outlier Detection and Treatment (4) Missing Value Handling (5) Duplicate Removal (6) Data Quality and Inconsistency Detection (7) AI Generated Insights and Conversational Assistant (8) Visualizations and PDF Export Each stage reads from and writes to shared Pandas DataFrame that is maintained in the Streamlit's session state, ensuring that the changes applied in one stage are then immediately reflected in the subsequent stages.

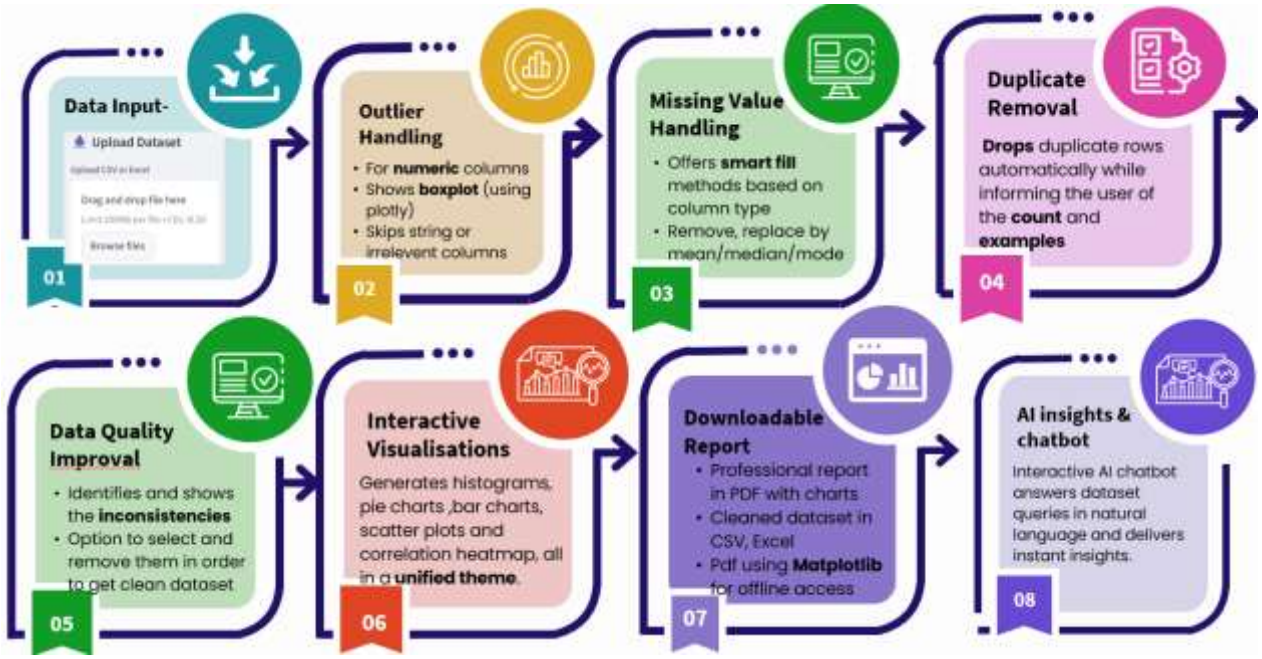


Fig. 1. CleanSight System Architecture Pipeline

Table 1: CleanSight Pipeline Stages and Components

Stage	Component	Technology
1. File Ingestion	CSV or Excel loading, delimiter detection	Pandas, NumPy
2. Statistical Profiling	Column summary, data types, null counts, sample values	Pandas, Streamlit
3. Outlier Detection	IQR based detection, box plots, treatment options	NumPy, Plotly
4. Null Imputation	Mean/median/mode/custom/drop strategies	Pandas
5. Duplicate Removal	Exact duplicate detection and elimination	Pandas
6. Data Quality & Inconsistency Detection	Case or whitespace, word-number and date format inconsistency detection and correction	Pandas
7. AI Insights & Conversational Assistant	AI generated summary; hybrid rule based LLM chatbot	Groq LLM API
8. Visualization & Export	Histograms, pie charts, scatter, heatmap; CSV or Excel or PDF	Plotly, Matplotlib, FPDF

B. File Ingestion and Delimiter Detection

The tool accepts CSV and Excel files through Streamlit's file uploader widget. CSV and Excel files are processed using Pandas. After successful ingestion the dataset is stored in the Streamlit session state and an independent copy of the original DataFrame is stored to provide a reference state for cleaning and duplicate detection. The interface also displays basic dataset statistics consisting of the number of rows, columns, numeric attributes, missing values and provides direct navigation buttons to do profiling, AI assistant and cleaning modules.

C. Statistical Profiling

Upon loading of the file, CleanSight generates an overview of the dataset containing the total number of rows and columns, missing values, duplicate records and overall data. The profiler provides with an interactive column selection interface in which individual variables can be selected for detailed inspection. A scrollable column panel allows users to navigate through dataset columns that have a large number of variables without displaying all column details simultaneously.

For each selected column the system displays the data category, data type, number of missing values and their percentage, IQR based outliers, number of unique values and a representative statistic such as the mean for numeric attributes or the most frequent value for non numeric attributes. The selected variable also shows a distribution view. Numeric attributes are represented using a histogram with descriptive statistics while non numeric attributes are summarized using the frequencies of their most common values. An overview mode additionally provides a small sample of the dataset.

D. Outlier Detection and Treatment

Outlier detection is applied only to numeric columns that pass a series of filters to reduce false positives. The system automatically skips: (1) non numeric columns (2) columns where all values are unique and the dtype is integer that indicates an identifier columns (3) binary columns with exactly two unique values (4) low cardinality columns with fewer than 10 unique values that are likely categorical encodings. This smart filtering prevents imprecise treatment of variables such as primary keys, flags, or ordinal codes.

For qualifying columns, outliers are identified using the IQR method. Given the first quartile Q1 and third quartile Q3, the IQR is defined as $IQR = Q3 - Q1$. A value x is classified as an outlier if $x < Q1 - 1.5 \times IQR$ or $x > Q3 + 1.5 \times IQR$. The user is presented with four treatment options for each affected column: keep all values, remove outlier rows, replace with the column median, or replace with the column mean. The applied treatment is logged to a cleaning audit trail. As shown in Fig. 2, outliers are identified using the IQR-based boxplot method.

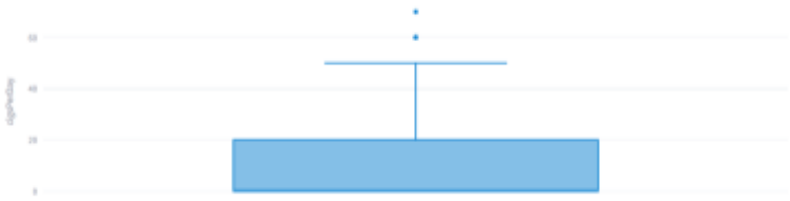


Fig. 2. Outlier detection using the IQR method

E. Missing Value Imputation

The tool provides an interface for columns containing detected null values. For numeric columns the handling strategies include filling with the column mean, filling with the column median, entering a user defined custom value or dropping rows. For categorical columns the available strategies include filling with the column mode, replacing missing values with the string “Unknown”, entering a custom value or dropping the affected rows. The custom value option uses a type aware casting function that converts the user provided input to the column's original data type and displays an error message when the value cannot be converted so helps in preventing invalid values from being inserted into the dataset. Each applied imputation is recorded in the cleaning log.

F. Duplicate Detection and Removal

Duplicate rows detection is done on the original DataFrame (pre cleaning) using Pandas duplicated() method with keep is equal to 'first' rule that sees all the rows that are the exact replicas of an earlier row. The number of duplicates is shown and upto five example duplicate rows are displayed in the clean part of data preprocessing. Their removal is applied automatically to the active DataFrame and the count of removed rows is logged.

G. Data Quality and Inconsistency Detection

Beyond structural cleaning, the tool also performs a dedicated pass over the categorical columns to look for semantic data quality issues that include numeric focused checks to make them consistent. Three inconsistency categories are detected: (i) case and whitespace variants where values are grouped using a lowercase and trimmed representation (ii) mixed word and number representations where number words such as 'one' are compared with their digit representations within the same column (iii) inconsistent date formats where values are checked against predefined date patterns that include YYYY-MM-DD, DD/MM/YYYY, DD-MM-YYYY, DD/MM/YY and YYYY/MM/DD.

Detected issues are presented through an interactive correction interface that displays the affected column, issue type, example values, affected row count and the correction status. Pending inconsistencies can be individually selected using checkboxes and corrected through the "Correct Inconsistencies" button.

H. AI Generated Insights Summary

After all cleaning stages are complete the system assembles a structured JSON dataset context containing the column level statistics that includes before and after cleaning, missing-value counts, duplicate counts, outliers and the full cleaning action log. When the user selects the Generate AI summary button, the system constructs a dataset specific prompt containing the current dataset dimensions, column names, remaining missing value information, numerical summary statistics and the stored cleaning actions.

The constructed prompt is first given to a Groq Large Language Model using the openai/gpt-oss-20b model. The model is instructed to generate a concise 150 to 220 words plain text summary having the overall data quality, cleaning operations and their purpose, data quality risks and practical recommendations. The generated response is stored in the Streamlit session state and displayed in the Reports interface. If the API call fails the application displays a fallback summary using rules.

I. Conversational AI Assistant (Chatbot)

CleanSight provides a conversational AI assistant through a dedicated Assistant page that allows users to ask natural language questions about currently loaded dataset. The assistant maintains the conversation history in Streamlit session state and provides predefined quick questions for common analytical tasks like dataset summarization, correlation analysis, cleaning recommendations and model selection guidance.

Before each request the system makes a textual dataset context consisting of the dataset dimensions, column names, data types, missing value information, numerical summaries and selected categorical information. The numerical context includes the mean and standard deviation for up to ten numeric columns while the categorical context contains the number of unique values and the three most frequent values. This context is combined with the conversation history and supplied to the Groq language model. The assistant is instructed to provide specific and practical responses based on given dataset context. Responses are displayed interactively in the chat interface while the conversation remains available during the active Streamlit session. A sidebar control is also given in order to clear the stored conversation history.

The conversational assistant therefore uses a hybrid AI architecture with two processing layers. Layer 1 performs deterministic local computation using Pandas and rule based logic before any language model call is done. It can directly calculate threshold counts, mean, median, minimum or maximum values, Pearson correlation and categorical distributions from active DataFrame. If the local query engine cannot resolve the question then Layer 2 acts as an LLM fallback having the user question, recent conversation context and the structured dataset context which are forwarded to the Groq model. Thus the resulting decision path is: User question → local query engine → answer found if → yes: deterministic answer; no: LLM fallback. This architecture reduces unnecessary external API calls and keeps numerically answerable queries grounded in exact local computations while retaining natural language flexibility for the questions outside the deterministic query patterns.

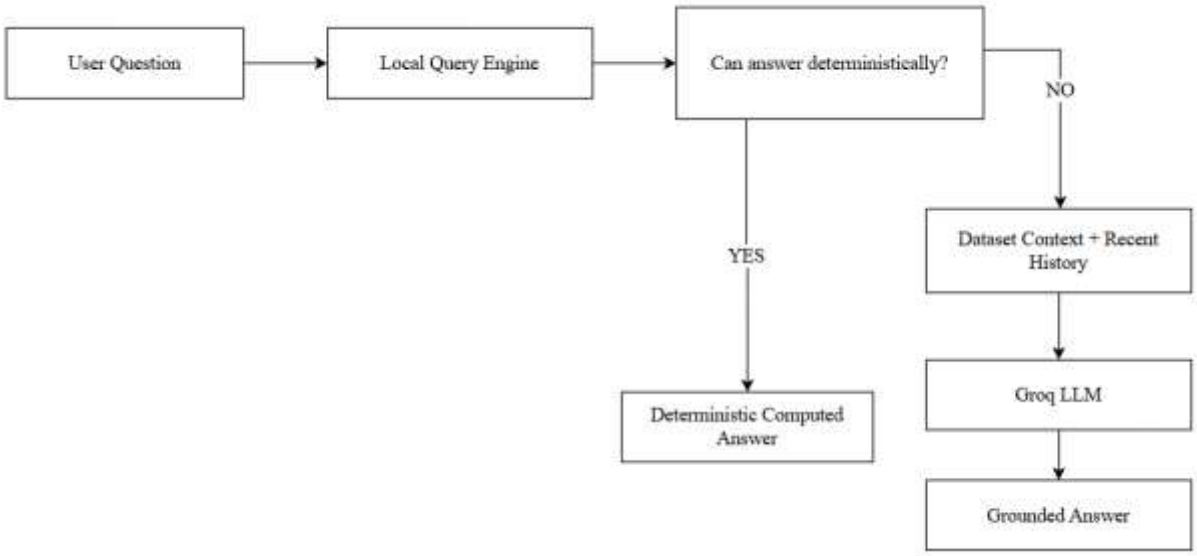


Fig. 8. CleanSight hybrid AI query architecture.

J. Visualisations

The visualization page provides two complementary modes that are an interactive chart builder for user directed exploration and an automatic insights mode for fast dataset level visual analysis. The interactive chart builder supports seven chart types: histogram, bar chart, scatter plot, box plot, pie chart, line chart and a correlation heatmap. Users select the required columns according to the selected chart type and the interface restricts column choices to only the compatible data types.

Each generated chart is accompanied by an automatically generated statistical explanation for it. For histograms the system reports the distribution shape together with the mean and median. For bar and pie charts it identifies the most frequent category and its proportion. For box plots it tells the median and number of values outside the IQR based whiskers. For scatter plots it tells the Pearson correlation coefficient together with the strength and direction of the relationship. For the line charts system estimates the trend direction using a first degree linear fit. For correlation heatmaps it identifies the strongest pairwise relationship among the selected variables.

The automatic insights mode applies rule based column classification before generating the visualizations. Columns are classified on the basis of their data type, cardinality and identifier characteristics. Likely identifier columns are excluded from adding in automatic visualization. Numeric columns with fewer than ten unique values are treated as low cardinality variables whereas the numeric columns with ten or more unique values are treated as continuous candidates. Categorical columns are classified according to the number of distinct values. Continuous numeric variables are ranked using relative variability that is calculated from standard deviation relative to the absolute mean and up to three high variability columns are selected for generating the histogram. Up to two categorical variables are selected for top category pie charts and up to three strongest numeric pairs are selected for scatter plots based on the absolute Pearson correlation. Additional bar charts are generated for selected categorical or low cardinality variables.

All interactive visualizations are made using Plotly. The automatic selection mechanism controls the number of generated charts and excludes unsuitable variables hence providing a compact visual overview while allowing users to manually explore additional relationships through the interactive chart builder.

K. Export Module

The cleaned dataset can be exported in CSV and Excel formats, giving users with convenient options for further analysis and uses in external applications. CSV export is done using Pandas' to_csv() method, while Excel export uses Pandas' to_excel() method with the openpyxl engine. In addition to the cleaned-data exports, CleanSight also generates a complete PDF report using the FPDF library. The multi-page report includes the dataset overview, cleaning actions and logs, duplicate statistics, outlier detection and results, AI-generated summary, and all the generated visualisations. The visualisations are rendered as Matplotlib PNG images, temporarily stored during report generation, and added into the PDF using FPDF's image insertion functionality. This provides both the cleaned dataset itself and a combined report including the major data-cleaning and analysis results.

IV. IMPLEMENTATION

A. Technology Stack

The application is implemented entirely in Python 3.13 and the core libraries used are: Streamlit 1.63 for the web interface; Pandas and NumPy for data manipulation; Plotly Express for interactive visualizations; Matplotlib and Seaborn for static PDF chart generation; FPDF for PDF assembly; openpyxl for Excel I/O; and Groq Python client for accessing a hosted LLM (openai/gpt-oss-20b) that is used for the conversational assistant and AI-generated insights and summary. The application is structured as a multi-page Streamlit app - a central entry point (Home.py) with dedicated pages for profiling, cleaning, visualization, the AI assistant, and report generation - enabling straightforward deployment on the Streamlit Community Cloud without requiring complex configuration [16].

B. Session State Management

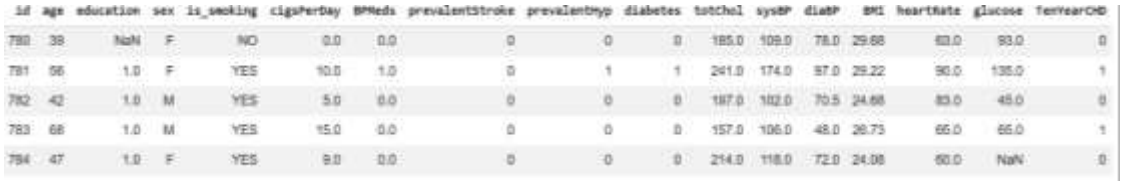
A key implementation challenge in Streamlit is its top-down execution model where the entire script reruns on each user interaction. To manage this- chart metadata such as type, title and data references are stored in session state. This ensures that visualizations generated during intermediate steps remain available for final report generation without any data loss taking place. Session state is similarly used to persist the chat conversation history, the currently open/closed state of the assistant panel, the list of already-accepted data-quality corrections, and a SHA-256 fingerprint of the uploaded file, so that these are correctly reset only when a genuinely new dataset is uploaded rather than on every script rerun.

C. User Interface Design

The user interface follows a simple and structured design with a consistent dark theme (#0F1117 background, #1A1D27 panels), teal (#14B8A6) accent color, and rounded card components. The application uses a sidebar-based navigation system that organizes the workflow into dedicated pages for the Home, Profiler, Assistant, Clean, Visualize, and Reports sections. The Home page provides the initial dataset upload interface along with an overview of the uploaded dataset. The Profiler page presents detailed column-level information and statistical summaries, while the Clean page contains missing-value handling, duplicate removal, outlier detection and treatment, and data inconsistency detection. The Visualize page provides interactive visualizations , and the Reports page provides access to download the generated reports and cleaned-data in PDF and CSV format respectively. Each page uses structured headings, cards, tables, interactive controls, and color-coded feedback to provide clear visual feedback on the operations performed.

The CleanSight Assistant is provided as a dedicated Assistant page within the sidebar's navigation structure. It provides a chat-based interface through which users can ask natural-language questions about the current loaded dataset and receive dataset-specific responses. The interface also provides suggested questions and maintains the conversation history during the active session.

To improve navigation and usability when working with larger datasets, the application uses page-based navigation that separates major data-processing and analysis tasks into dedicated sections. This allows profiling, cleaning, visualization, AI-assisted analysis, and reporting functions to be accessed independently rather than presenting the complete workflow on a single page, thereby reducing visual clutter and improving accessibility and usability.



id	age	education	sex	is_smoking	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
780	38	NaN	F	NO	0.0	0.0	0	0	0	185.0	109.0	78.0	29.68	63.0	93.0	0
781	56	1.0	F	YES	10.0	1.0	0	1	1	241.0	174.0	97.0	29.22	90.0	135.0	1
782	42	1.0	M	YES	5.0	0.0	0	0	0	197.0	102.0	70.5	24.66	83.0	48.0	0
783	68	1.0	M	YES	15.0	0.0	0	0	0	157.0	106.0	48.0	26.75	65.0	65.0	1
784	47	1.0	F	YES	9.0	0.0	0	0	0	214.0	118.0	72.0	24.08	60.0	NaN	0

Fig. 3. Raw dataset before preprocessing



id	age	education	sex	is_smoking	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
780	38	1.970836	F	NO	0.0	0.0	0	0	0	185.0	109.0	78.0	29.68	63.0	93.000000	0
781	56	1.000000	F	YES	10.0	1.0	0	1	1	241.0	174.0	97.0	29.22	90.0	78.000000	1
782	42	1.000000	M	YES	5.0	0.0	0	0	0	197.0	102.0	70.5	24.66	83.0	78.000000	0
783	68	1.000000	M	YES	15.0	0.0	0	0	0	157.0	106.0	82.0	26.75	65.0	65.000000	1
784	47	1.000000	F	YES	9.0	0.0	0	0	0	214.0	118.0	72.0	24.08	60.0	78.436839	0

Fig. 4. Cleaned dataset after preprocessing

Table 2: Key Functions and Their Roles

Function	Description
infer_type()	Infers a display type (numeric, category, boolean, datetime, string) for each column from its dtype and cardinality
is_id_column()	Flags likely identifier columns (fully unique, integer-typed or 'id'-named) to exclude from outlier detection
outlier_bounds()	Computes the IQR-based lower and upper outlier bounds for a numeric column
detect_delimiter()	Detects the delimiter used in CSV files to support reliable dataset ingestion.
run_local_query()	Processes dataset questions locally when the requested result can be computed directly from the uploaded data.
detect_inconsistencies()	Scans categorical columns for case/whitespace, word-number and date-format inconsistencies
run_auto_fix()	Applies selected automatic data-cleaning operations to the dataset.
drop_duplicates()	Removes exact duplicate rows from the DataFrame and logs the number removed
build_context()	Builds a compact statistical profile of the dataset as grounding context for the AI assistant
build_summary_prompt()	Assembles dataset statistics and the cleaning log into a prompt for the AI-generated report summary
build_pdf()	Assembles the cleaning summary and AI-generated narrative into a downloadable PDF report
classify_visual_columns()	Buckets columns into continuous, low-cardinality, pie, bar and identifier roles to drive automatic chart selection

V. EXPERIMENTAL RESULTS

The proposed CleanSight tool was evaluated on three real-world datasets to assess its effectiveness in improving data quality:

(1) Movies dataset taken from IMDB (4,803 rows, 24 columns),
(2) Cardiovascular Risk dataset from UCI (3,390 rows, 17 columns) and
(3) the Online Retail dataset (541,909 rows, 8 columns).

Table 3: Data Quality Metrics Before and After CleanSight Processing

Dataset	Nulls (Before)	Nulls (After)	Outliers Treated	Duplicates Removed	Rows Retained (%)
Movies Dataset	4,454	0	Multiple columns	Yes	97%
Cardiovascular Risk	510	0	Few columns	No	99%
Online Retail	136,534	0	Multiple columns	Yes	95%

The results demonstrate that the CleanSight tool effectively improves the data quality across the datasets of varying sizes and domains. All missing values in the datasets were successfully handled using the appropriate user based imputation techniques with default as remove.

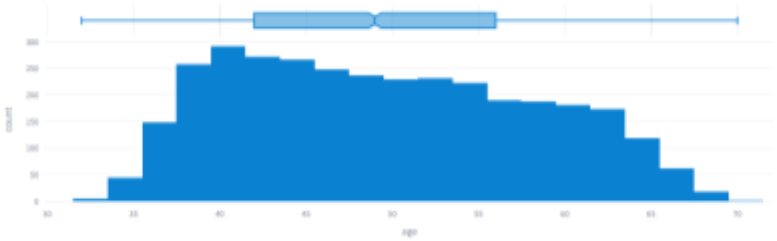


Fig. 5. Distribution of numerical data after preprocessing

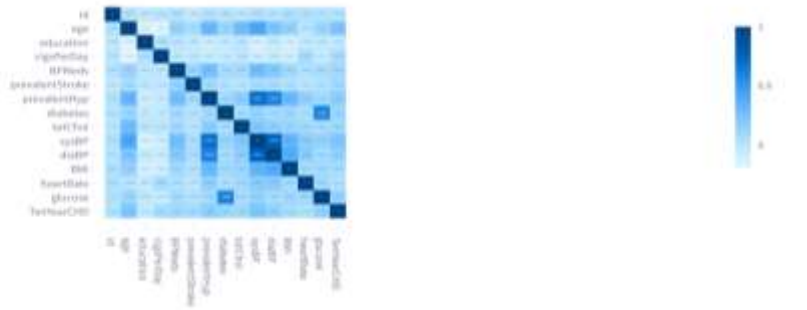


Fig. 6. Correlation heatmap of numerical features

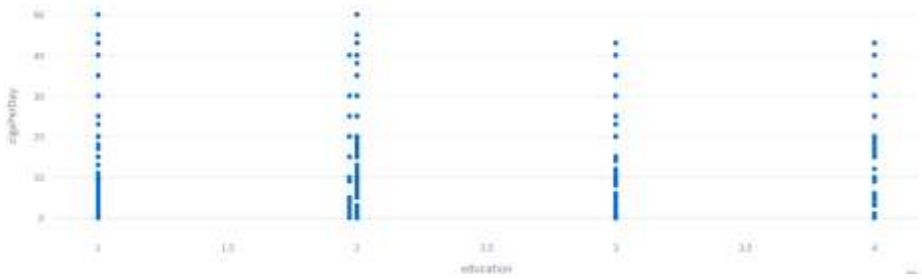


Fig. 7. Scatter plot showing relationship between two numerical features

The IQR based outlier detection method effectively identified the extreme values in numerical columns shown using boxplots while avoiding incorrect detection in categorical features. Duplicate records were also identified and removed in applicable datasets.

The tool processed both small and large datasets efficiently, including the Online Retail dataset with more than 500,000 rows in seconds, demonstrating its scalability and suitability for real-time use and data preprocessing.

Additionally, the generated reports included visualizations such as histograms, pie charts, scatter plots and correlation heatmaps along with the cleaning logs that makes the system easy to use and interpret.

VI. LIMITATIONS AND FUTURE WORK

Although CleanSight provides an effective and user-friendly interface for data preprocessing but some limitations still remain. The current system processes the datasets in-memory that may affect performance while handling extremely large datasets.

In addition, the tool currently supports basic missing value imputation techniques such as mean, median and mode. While these methods are efficient, some more advanced techniques could further improve data accuracy. Similarly, outlier detection is based on the IQR method which may not capture complex patterns in high dimensional data. Also, the current inconsistency detection module covers case/whitespace, word-number and date-format issues in categorical columns only, and the conversational assistant and AI-generated insights summary depend on an externally hosted Groq LLM, so the AI layer falls back to a rule-based summary whenever no API key is configured or the service is unavailable. The local query engine within the chatbot is similarly limited to a fixed set of question patterns (column lookups, threshold comparisons, correlations, and basic descriptive statistics); questions falling outside these patterns are only answered when the Groq API is connected, making the assistant's availability and responsiveness dependent on network availability and third-party API uptime.

Future work will focus on enhancing the scalability and intelligence of the system. This includes adding support for large scale data processing frameworks while including advanced imputation methods and improving outlier detection techniques. Additional enhancements may also include improved user interface features, support for cloud storage integration and the development of APIs in order to enable integration with automated data processing pipelines. Planned improvements to the AI layer include expanding the inconsistency detection module to cover more issue types such as unit and encoding conflicts, allowing configurable or locally hosted language models to reduce dependence on a single external LLM, and extending the AI assistant with the ability to suggest and apply cleaning actions directly from the chat.

VII. CONCLUSION

This paper presented the CleanSight tool which is a comprehensive and interactive web application that automates the end to end data preprocessing workflow that is built on the Streamlit framework with Python. The system addresses a practical gap in the data science ecosystem by providing a code-free and user friendly interface that combines statistical profiling, intelligent outlier detection, flexible null imputation, duplicate removal, interactive visualizations and multi-format export options of the cleaned dataset and the PDF report in a single cohesive pipeline.

The system's key innovations include automated and smart column pre-screening for outlier detection, type-safe custom imputation, session state driven PDF chart assembly, automated delimiter detection, a dedicated data quality and inconsistency detection module for case/whitespace, word-number and date-format issues, and an AI layer combining a hybrid rule-based/LLM based conversational assistant with an automatically generated, context-based insights and summary, which collectively enhances usability and reduces the risk of error filled data transformation. Experimental evaluation on real-world datasets also demonstrated the effectiveness of the tool in improving data quality while maintaining data integrity.

The column-wise profiling interface and paginated navigation further improved the usability of the system for large datasets by allowing users to inspect individual features without overwhelming the main dashboard.

The CleanSight tool contributes to making data preprocessing more accessible to analysts, researchers, and students by eliminating the need for advanced programming skills. A fully functional and deployed prototype of the system is publicly available and can be easily accessed at: https://cleansight.streamlit.app/

REFERENCES

[1] J. Zhu, X. Zhao, Y. Sun, S. Song, and X. Yuan, “Relational Data Cleaning Meets Artificial Intelligence: A Survey,” Data Science and Engineering, vol. 10, pp. 147–174, 2025.

[2] B. Mohammadivojdan, “Robust Algorithm for Automatic Outlier Detection in Data Cleaning,” Journal of Applied Statistics, vol. 52, no. 3, pp. 678–692, 2025.

[3] A. M. Sharifnia, “A Primer of Data Cleaning in Quantitative Research,” Journal of Advanced Nursing, vol. 81, no. 2, pp. 345–356, 2025.

[4] R. Gupta and S. Sharma, “Intelligent Data Preprocessing Framework for Analytical Applications,” Expert Systems with Applications, vol. 235, p. 121234, 2024.

[5] A. Satyanarayan et al., “Interactive Visualization Systems for Data Analysis,” IEEE Computer Graphics and Applications, vol. 44, no. 2, pp. 50–63, 2024.

[6] K. Lee and D. Kim, "Interactive Data Visualization Systems for Analytical Applications," Information Visualization, vol. 23, no. 2, pp. 180–195, 2024.

[7] L. Zhang et al., "Automated Report Generation from Data Analytics Systems," Expert Systems with Applications, vol. 230, p. 120456, 2024.

[8] S. Zhang, Z. Huang, and E. Wu, "Data Cleaning Using Large Language Models," arXiv preprint arXiv:2410.15547, 2024.

[9] V. Khatri, P. Panchal, and L. K. Meena, "Personalized Healthcare Systems Using AI," NanoTechnology Perceptions, vol. 20, no. S13, pp. 437–452, Oct. 2024.

[10] R. K. Arora and S. Singh, "Smart Irrigation and Monitoring Framework," in Proc. International Conference on Innovative Computing and Communications, Feb. 16–17, 2024, Springer.

[11] I. F. Ilyas and X. Chu, "Data Cleaning: Overview and Emerging Challenges," Proceedings of the VLDB Endowment, vol. 16, no. 12, pp. 4100–4103, 2023.

[12] Y. Li et al., "Automated Data Cleaning Techniques in Data Science Pipelines," IEEE Access, vol. 11, pp. 45000–45015, 2023.

[13] S. García, J. Luengo, and F. Herrera, "Data Preprocessing in Data Mining: Handling Missing Values," Knowledge-Based Systems, vol. 257, p. 109756, 2023.

[14] H. Wickham et al., "Data Tidying and Missing Data Handling Techniques," Journal of Statistical Software, vol. 105, no. 1, pp. 1–35, 2023.

[15] A. K. Singh and R. Kumar, "A Review on Outlier Detection Techniques in Data Mining," Journal of Big Data, vol. 10, no. 1, pp. 1–21, 2023.

[16] R. K. Arora and S. Singh, "Data Quality Management Framework for Data Warehouse," in Proc. International Conference on Artificial Intelligence, Advanced Materials and Mechatronics Systems, JECRC University, Jaipur, India, Nov. 3–4, 2023.

[17] R. K. Arora and S. Singh, "Analysis of Various Smart Healthcare Systems for Heart Disease Prediction," in Proc. International Conference on Machine Learning, Advances in Computing, Renewable Energy and Communication, Glocal University, Saharanpur, India, Nov. 28–29, 2023, Springer.

[18] C. C. Aggarwal, "Outlier Analysis: Concepts and Techniques," ACM Computing Surveys, vol. 56, no. 4, pp. 1–36, 2023.

[19] A. Treuille, T. Teixeira, and A. Zwiener, "Streamlit: The Fastest Way to Build and Share Data Apps," Streamlit, 2023.

[20] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proc. IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.