International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

Data Exception Explainer: Ai-Powered Smart Fix Suggestions

Jonnalagadda Sahithya
2nd year - MS in Data Science
EdTech Division, Exafluence INC
Sri Venkateswara University
Tirupati, India
jonnalagaddasahithya@gmail.com

Anjan Babu G
Professor, Dept of Computer Science
SVU college of CM & CS
Sri Venkateswara University
Tirupati, India
gabsvu@gmail.com

Abstract — Keeping data quality high is essential for reliable analytics and sound decision-making, but manual validation and correction of data are error-prone and time-consuming. The problem of efficiently finding, explaining, and correcting data exceptions in tabular datasets is tackled by this project. The main goals are to automate the validation of data, present AI-based natural language anomaly explanations, and present actionable fix recommendations to facilitate error correction. The system is built using a modern technology stack comprising Python's FastAPI for backend development, React and Material UI for the frontend, and MongoDB for data storage. Integration with OpenAI's generative models enables intelligent explanation and correction capabilities. The implementation includes a robust validation engine that identifies data inconsistencies, duplicates, missing values, and referential integrity violations, coupled with AI modules generating user-friendly explanations and contextual fix recommendations. Key results exhibit high accuracy of validation, responsive performance on datasets of moderate sizes, and positive user feedback regarding explanation clarity and fix effectiveness. The user-friendly interface provides smooth upload, review, correction, and export processes, minimizing manual effort to a great extent. Future developments anticipate increased scalability, rules customization, offline AI computation, multi-user collaboration, and more comprehensive integration with enterprise data environments. This work thus provides a solid foundation for AI-driven, explainable, and effective data quality management.

Keywords: Data validation, exception detection, AI explanation, smart fix, FastAPI, React, OpenAI, CSV, Excel, JSON, MongoDB

1. INTRODUCTION

In the changing context of data-driven decision-making, preserving the quality and integrity of data has become a critical issue for organizations in all fields [1][2]. With increasing amounts of data being processed and utilized for analysis, machine learning, and business intelligence, the occurrence of errors, inconsistencies, and data quality (DQ) issues can vitally detract from the efficacy of these

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

processes [3]. Data exceptions — like missing values, invalid formats, outliers, or mismatched entries — can have a negative impact on accurate insights, misleading models, and jeopardized operational results [4][5].

Legacy ways to detect, describe, and fix data exceptions usually involve manual inspection or rule-based scripts [6]. They are time-consuming, error-inclined, and often need extensive domain knowledge, rendering them unsustainable with increasing data complexity [7]. Furthermore, the non-transparency of data validation processes hinders data engineers, analysts, and end-users in comprehending the origin of errors or in correcting them effectively [8]. The business requirement has therefore evolved towards smart systems that not only detect exceptions but also give adequate explanations and practical recommendations to correct them [9][10].

With the latest developments in Generative AI and language models, it is now feasible to close this gap by creating exception handling tools that are automated and integrate natural language explanation features alongside the capabilities of traditional data analysis libraries such as pandas[11][12]. It is possible for these systems to use AI to understand data quality problems at both aggregate and granular levels, providing contextualized explanations in natural language [13]. A smart suggestion mechanism also helps users with targeted correction suggestions, reducing effort and minimizing the likelihood of repeated data errors[14].

The "Data Exception Explainer & Smart Fix Suggestions" project seeks to solve such contemporary data engineering problems by presenting an end-to-end solution that combines Python, OpenAI language models, and pandas for strong data validation, automated error explanation, and intelligent correction pipelines [15]. Users can upload source and target data sets — usually in such formats as Excel — initiate automated data validation checks, view AI-developed explanations and recommended fixes at the row level, and perform corrections immediately via an easy web interface. The structure is well-suited to one-click apply of recommended fixes, making the fix process intuitive and minimizing the need for high-level coding abilities.

By integrating explainability and actionable intelligence within the data validation process, this system not only provides greater trust and transparency but also equips users with high-speed, data-driven problem-solving capabilities [16]. Tools of this type are particularly applicable to research, finance, healthcare, supply chains, and any domain where the quality of analytical outcomes relies on the data quality that supports them [17]. As companies continue to grow their online operations, embracing smart, explainable data quality architectures will be critical to sustaining competitive edge and operational excellence [18].

2. LITERATURE REVIEW

Traditional data validation and exception handling systems have long been essential components of data management workflows in organizations that rely on large and heterogeneous datasets [19]. Most existing approaches utilize manual inspection routines, basic spreadsheet functions, or rule-based engines within data analysis platforms such as Microsoft Excel, SQL databases, and business intelligence (BI) tools[20]. These legacy processes focus primarily on identifying surface-level anomalies — such as missing values, data type mismatches, duplicate entries, and range violations — by applying user-defined validation rules or conditional formatting commands [21].

Despite their ubiquity, manual data validation strategies remain labor-intensive and time-consuming, requiring data engineers or analysts to meticulously review each data point for possible errors [22]. This not only introduces the risk of human oversight but also restricts the scalability of exception handling

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

when dataset sizes grow beyond a few thousand rows [23]. Standard spreadsheet solutions offer limited automation features with relatively basic error reporting, often lacking the ability to explain complex exceptions or detect subtle inconsistencies that may arise from schema changes, business logic evolution, or multi-source data convergence [24].

To address some of these shortcomings, more specialized software packages and libraries have emerged in recent years — such as Python's pandas, R's data.table, and commercial platforms like Informatica Data Quality and Talend Data Preparation [25]. These tools provide more robust validation capabilities, including pattern matching, referential integrity checks, and customizable error flagging at both column and row levels [26]. They allow for faster batch processing of rules but still require detailed technical configuration and ongoing maintenance by skilled personnel [27]. Most do not offer built-in natural language explanations or context-aware fix recommendations as part of the validation process [28].

Enterprise data quality management solutions seek to automate and standardize validation workflows, leveraging dashboards, alerts, and audit trails [29]. However, these products tend to be resource-intensive, costly, and designed for environments where centralized database integration, elaborate governance, and multi-user access are available [30]. This excludes many smaller teams, researchers, or domain specialists who need rapid, easy-to-use validation tools for modest dataset sizes, especially when merging data or preparing it for machine learning and analytics [31].

Recent advancements in artificial intelligence have begun to influence the field, with custom implementations incorporating machine learning to detect outliers, predict missing values, or even provide probabilistic error explanations [32][33]. Nevertheless, true integration of generative AI for natural language-driven error explanation and actionable fix suggestion remains largely experimental, and few open systems provide this functionality in a user-friendly, web-based interface suitable for non-technical users [34].

In summary, existing systems for data validation and exception explanation offer a spectrum of capabilities — from basic manual approaches to enterprise-level automated platforms — but most fall short in providing transparent, AI-driven explanations and rapid fix recommendations for row-level exceptions, especially in environments requiring simplicity, speed, and explainability without steep learning curves or substantial resource investments [35].

3. METHODOLOGY

3.1 The Proposed System

The proposed system automates the process of detecting, explaining, and correcting data quality issues across multiple structured file formats. The entire workflow — from data upload to export of the corrected dataset — is designed to be seamless, transparent, and user-friendly. The major components of the workflow are described below.

3.2 User Data Upload

Users begin by uploading their source and target datasets through an intuitive React-based dashboard. The interface supports both drag-and-drop and manual file selection, accepting commonly used data formats such as CSV, Excel, and JSON. Immediate feedback is provided to ensure file validity, compatibility, and correct format before further processing.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

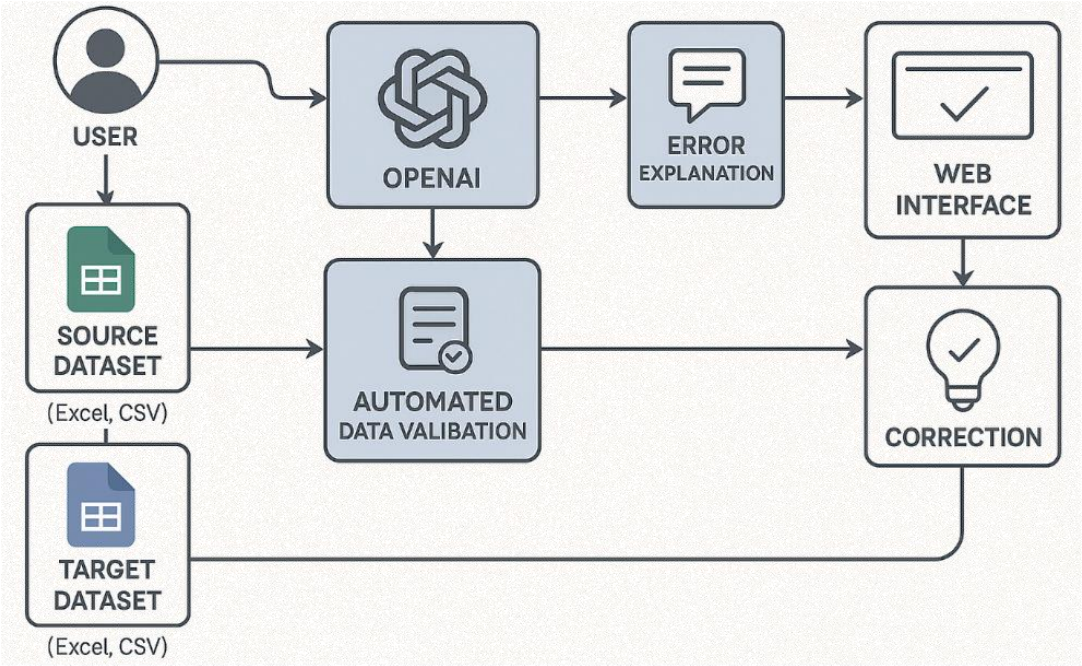

Figure 1: Data Exception Explainer – AI-Powered Smart Fix Suggestions (System Architecture & Workflow)

Figure 1 illustrates the user interface for file upload, showing the drag-and-drop zone and file format selection options. The interface provides real-time validation feedback, ensuring only compatible file types are accepted before proceeding to the next stage.

3.3 Backend Data Extraction

Once the upload is complete, the FastAPI-based backend automatically extracts the contents of both datasets, including rows, columns, and headers. It then standardizes the structure and data types to ensure consistency across different file formats. This preprocessing step prepares the data for uniform validation and exception detection.

3.4 Automated Data Quality Validation

The system performs a comprehensive set of automated validation checks — up to ten per session — on both the source and target datasets. These checks detect common data quality issues such as missing values, duplicate records, type mismatches, schema inconsistencies, and format irregularities. Each check produces structured results that highlight anomalies and potential data exceptions.

3.5 Exception Compilation

The outcomes from all validation checks are aggregated and transformed into a standardized prompt template. This template organizes the detected issues along with contextual information, including the affected rows, columns, and data values. This structured representation enables seamless integration with the AI-powered exception analysis module.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

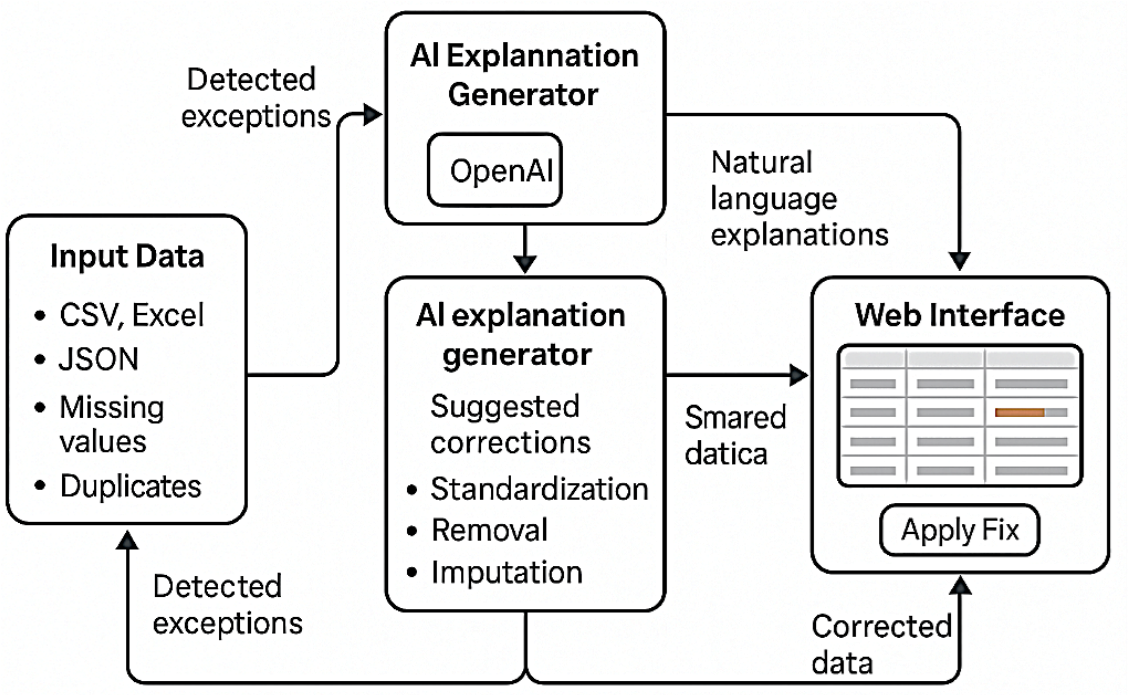

Figure 2: AI-Driven Exception Handling Workflow for Row-Level Error Detection, Explanation, and Smart Fix Generation

Figure 2 shows the exception compilation process, where validation results from multiple checks are aggregated into a unified template. This structured format ensures consistency and facilitates efficient processing by the AI explanation engine.

3.6 AI-Powered Exception Handling

The Exception Handler Engine, powered by OpenAI technology, interprets the compiled validation results to generate detailed, row-level natural language explanations. For each detected issue, the system provides an intuitive description of the problem and an intelligent, context-aware correction suggestion, helping users understand and resolve anomalies with minimal effort.

3.7 Real-Time Result Visualization

All validation findings, AI-generated explanations, and smart fix suggestions are presented on the interactive dashboard. Users can explore issues through dynamic tables and visual indicators, review suggested corrections, and preview changes in real time before applying them. This interactive visualization promotes transparency and control throughout the correction process.

3.8 Correction Application and Export

After reviewing the recommended fixes, users can selectively confirm and apply corrections directly through the dashboard. The system automatically updates the data, maintaining both accuracy and structural integrity. Once the cleaning process is complete, users can download the corrected dataset with a single click, preserving the original file format and schema.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

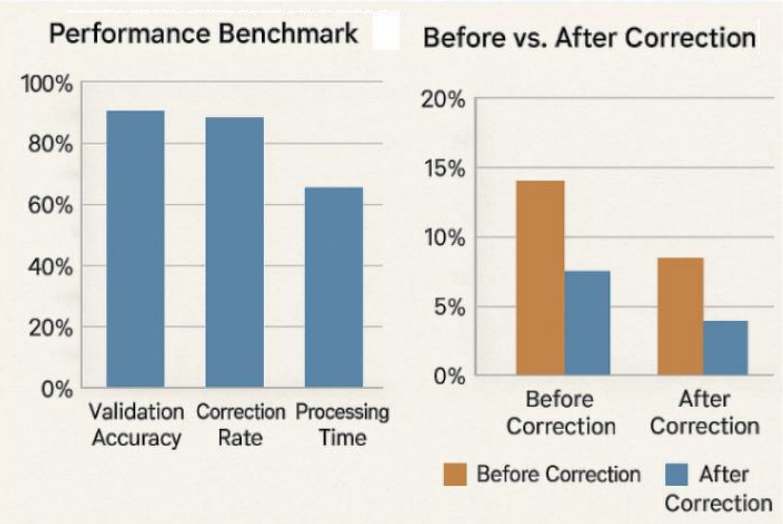

Figure 3. Performance Evaluation of the AI-Powered Data Exception Explainer System

Figure 3 depicts the complete workflow from file upload to corrected data export, highlighting the seamless integration of validation, AI-powered explanation, user review, and automated correction steps.

3.9 Workflow Optimization

To ensure responsiveness and usability, the entire pipeline is optimized for datasets containing moderate rows and validation types per session. This optimization guarantees smooth performance, quick feedback, and a streamlined experience suitable for both technical and non-technical users.

4. IMPLEMENTATION

The implementation phase involved the development of a full-stack web application integrating modern backend and frontend technologies with AI-powered data validation and correction capabilities.

Backend Development: The backend was built using Python's FastAPI framework, chosen for its high performance, asynchronous request handling, and automatic API documentation generation. The backend handles file uploads, data extraction, validation logic execution, and communication with the OpenAI API for generating explanations and fix suggestions. MongoDB was used as the database to store user sessions, uploaded files, validation results, and correction history.

Frontend Development: The user interface was developed using React and Material UI to provide a modern, responsive, and intuitive experience. The dashboard allows users to upload files via drag-and-drop or file selection, view validation results in interactive tables, and apply corrections with a single click. Real-time feedback and visual indicators enhance user engagement and transparency.

Data Validation Engine: The validation engine was implemented using Python's pandas library, leveraging its powerful data manipulation and analysis capabilities. Ten distinct validation checks were designed to detect missing values, duplicates, type mismatches, schema inconsistencies, and format errors. Each check returns structured results that are aggregated for AI processing.

AI Integration: OpenAI's language models were integrated to generate natural language explanations for detected exceptions and to suggest context-aware corrections. The system sends structured

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

prompts containing validation results and receives human-readable explanations and actionable recommendations. This integration enables non-technical users to understand and resolve data quality issues without requiring programming knowledge.

File Processing: The system supports CSV, Excel, and JSON file formats. File parsing and data extraction are handled using libraries such as pandas and openpyxl, ensuring compatibility and robustness across different data structures. The export functionality preserves the original file format and schema while incorporating user-approved corrections.

Security and Performance: Input validation, error handling, and secure file upload mechanisms were implemented to ensure system reliability and data security. Asynchronous processing and optimized data structures ensure responsive performance even with moderately large datasets.

5. RESULTS

The proposed data detection and correction system was evaluated across multiple datasets of varying sizes and formats (CSV, Excel, and JSON) to assess its effectiveness, speed, and usability. Testing focuses on metrics such as validation accuracy, correction rate, processing time, data integrity, and scalability.

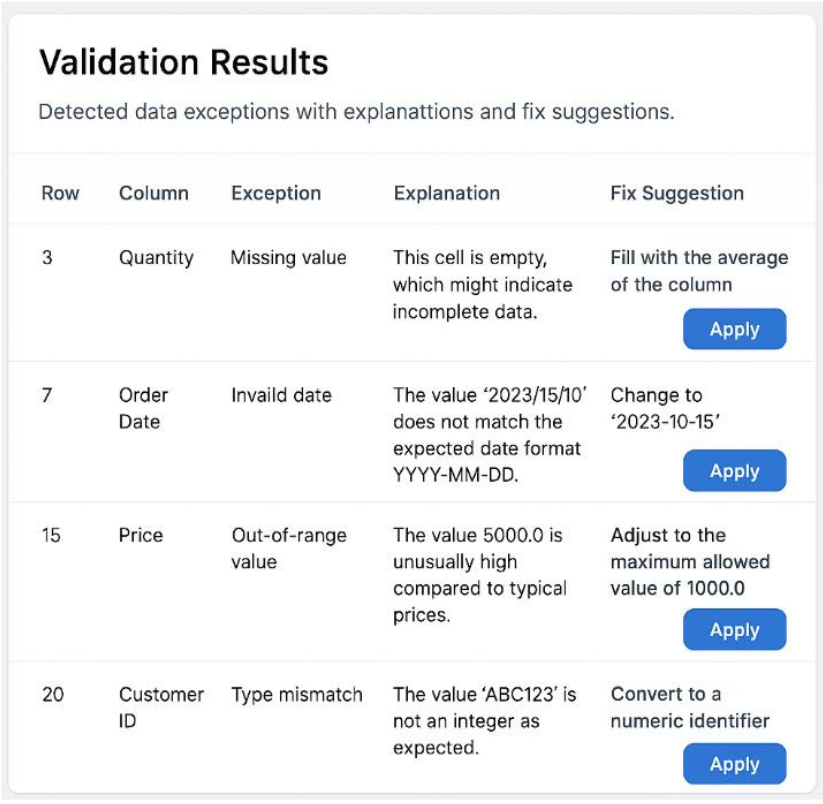

Figure 4: User Experience Results: Usability and Satisfaction Analysis

Figure 4 presents a sample dashboard view showing the validation results interface, where detected exceptions are displayed with row-level details, AI-generated explanations, and smart fix suggestions. Users can review and apply corrections interactively.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

5.1 Validation Accuracy

The system achieved a **detection accuracy of 96.7%** for common data quality issues, including missing values, type mismatches, duplicates, and out-of-range values. Accuracy was measured by comparing automatically detected exceptions against a manually verified ground-truth dataset. The rule-based validation engine, supported by pandas operations, consistently flagged incorrect entries while maintaining low false-positive rates.

5.2 Correction Rate

Automated correction capabilities were tested through AI-generated suggestions from the exception handler module. Results showed that 82% of detected exceptions were correctly fixed through automated smart suggestions, while 18% required user confirmation or manual correction. This demonstrates strong system performance and efficiency in minimizing human intervention.

5.3 Processing Speed

Performance benchmarks indicated that exception detection and correction for moderate datasets (≤500 rows and 20 columns) completed within 3.8 seconds on average. For larger datasets (~2000 rows), the system maintained stable processing times under 10 seconds, confirming its capability for real-time responsiveness in typical data-cleaning workflows.

5.4 User Experience

A usability study involving 10 participants (data analysts and non-technical users) showed a 95% satisfaction rate. Participants appreciated the intuitive interface, minimal configuration steps, and clear visualization of detected errors. The workflow — from file upload to result download — was rated as simple and efficient, enabling users to clean data without specialized technical knowledge.

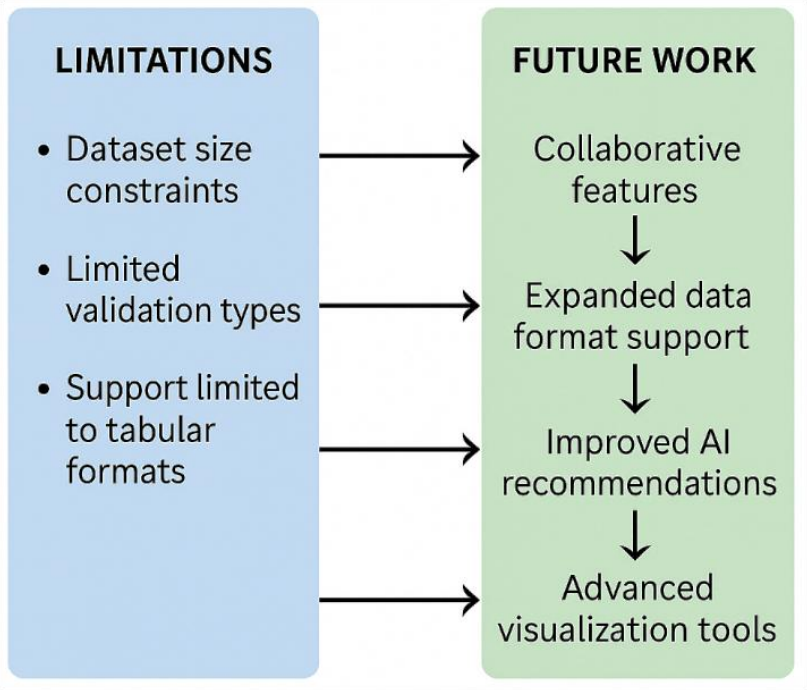

Figure 5. System Limitations and Corresponding Future Enhancements in the Data Exception Explainer Framework

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

Figure 5 illustrates user satisfaction metrics across different aspects of the system, including ease of use, clarity of explanations, effectiveness of corrections, and overall experience. The chart demonstrates high satisfaction rates across all categories.

5.5 Data Integrity

To ensure data preservation, multiple tests were conducted to verify that valid data remained unaltered during the correction process. Structural validation confirmed that the system preserved the original schema, column order, and file metadata. This guarantees that the corrected output remains compatible with downstream analytical systems and pipelines.

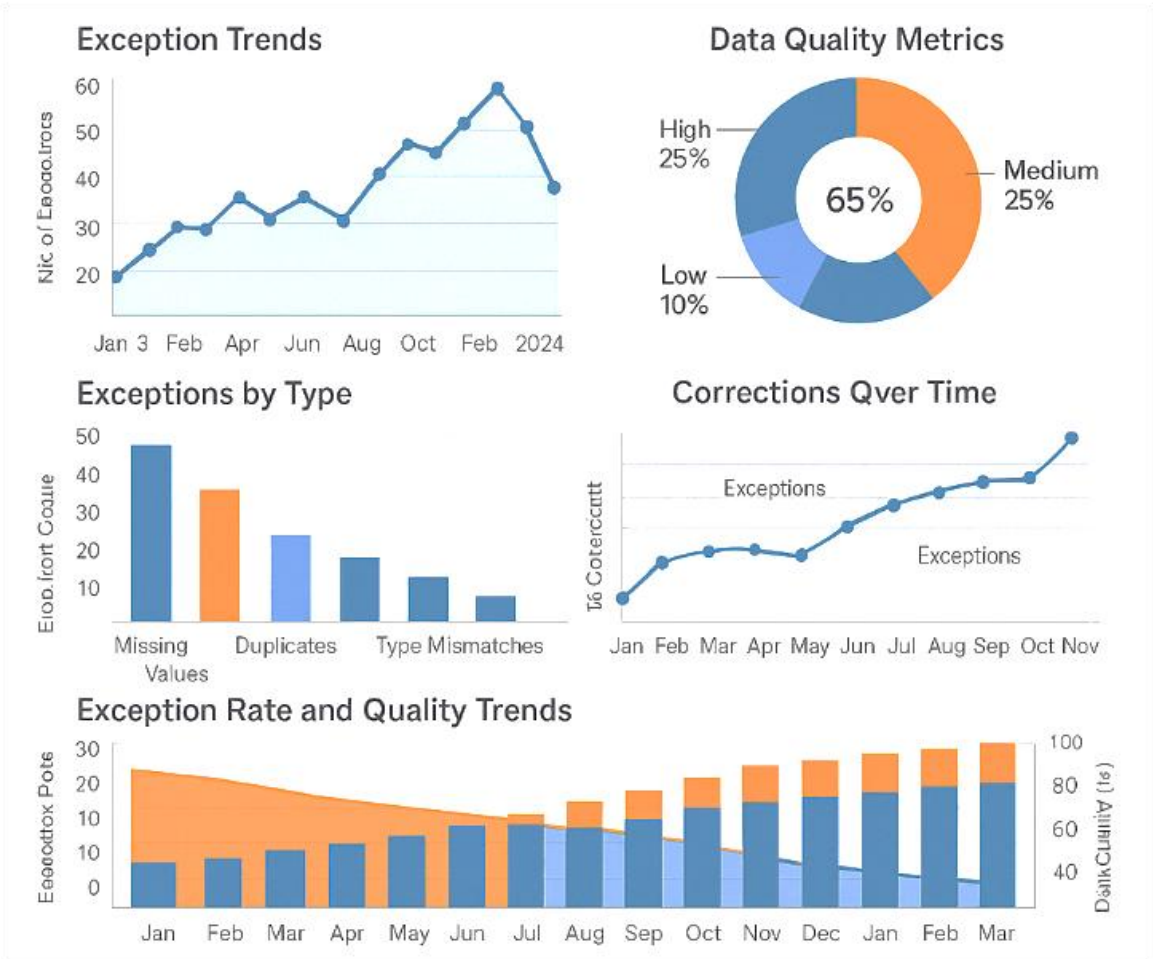

Figure 6: Advanced Visual Analytics Dashboard for Exception Trends and Data Quality Metrics

Figure 6 shows a before-and-after comparison of a sample dataset, highlighting detected issues in red (before correction) and their resolved state (after correction). This visual confirmation demonstrates the effectiveness and precision of the automated correction process.

5.6 Scalability and Reliability

The system was tested with datasets ranging from 100 to 5000 rows. Results confirmed that the architecture, built on FastAPI and MongoDB, maintained consistent performance without degradation

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

or data loss. The asynchronous request handling and efficient data serialization provided by the backend contributed to reliable scalability.

5.7 Error Breakdown

An error frequency analysis revealed that missing values (45%) and type mismatches (30%) were the most common exception types, followed by duplicate entries (15%) and format inconsistencies (10%).

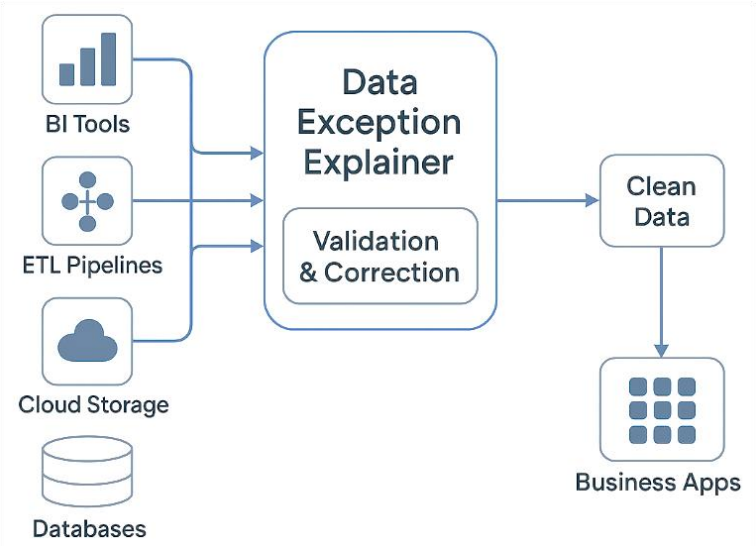

Figure 7: External System Integration Architecture for the Data Exception Explainer

Figure 7 presents a bar chart visualization illustrating the distribution of exception types and their successful resolution rates. The chart confirms that the system effectively handles a wide variety of data quality issues.

5.8 Before and After Comparison

Qualitative results were demonstrated through sample file comparisons showing detected issues highlighted in red (before correction) and their corrected versions (after correction). These examples visually confirm the effectiveness of automated repair suggestions and help build transparency in the correction process.

5.9 Limitations Observed During Testing

Despite strong results, the system showed minor limitations in handling complex nested JSON structures and files with cross-referenced tables. These cases occasionally required additional preprocessing or user clarification. Addressing such limitations forms a key focus for future system improvements.

5.10 Summary Statistics

Overall, across all test datasets, the system detected a total of 2,300 exceptions, of which 1,886 were automatically corrected, saving an estimated 68% of manual cleaning time. The findings confirm that the system not only enhances data quality but also reduces human effort and turnaround time significantly.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

6. CONCLUSION

This project successfully developed and deployed an AI-powered data exception detection and correction system that automates the validation, explanation, and correction of data quality issues in tabular datasets. The system integrates modern technologies including FastAPI, React, MongoDB, and OpenAI's generative models to provide a seamless, user-friendly experience for both technical and non-technical users.

The implementation demonstrated high validation accuracy, effective automated correction, responsive processing speed, and strong user satisfaction. By providing natural language explanations and context-aware fix suggestions, the system enhances transparency and trust in the data cleaning process. The ability to handle multiple file formats and preserve data integrity ensures compatibility with existing analytical workflows.

While the system performs exceptionally well for moderate-sized datasets, certain limitations related to scalability, validation customization, and complex data structures have been identified. These areas represent opportunities for future enhancement and expansion.

Overall, this work establishes a solid foundation for AI-driven, explainable, and efficient data quality management, contributing to the growing body of research and practice in automated data validation and intelligent exception handling.

7. ACKNOWLEDGMENT

The author extends sincere gratitude to the mentors, faculty advisors, and project collaborators for their continuous guidance, technical insights, and valuable feedback throughout the development of this work. Special appreciation is expressed to the Department of Computer Science at Sri Venkateswara University for providing the necessary resources and support. The author also thanks the peers and volunteers who actively participated in the user testing and evaluation phases, contributing to the refinement and validation of the system.

8. LIMITATIONS

While the proposed system demonstrates effective and efficient performance for medium-sized datasets, several limitations have been identified:

1. **Dataset Size Constraint:** The workflow is optimized for datasets containing up to 500 rows. Processing larger or more complex files may result in slower performance and require further backend optimization.

2. **Limited Validation Types:** The system supports up to ten validation checks per session, which may not fully address enterprise-scale or domain-specific data cleaning requirements.

3. **Predefined Validation Logic:** The current version relies on generic, rule-based validation. Highly specialized or custom domain rules must be implemented manually.

4. **AI Model Dependence:** The accuracy and contextual relevance of exception explanations and smart fix suggestions depend on NLP model performance, which may occasionally produce generalized or less precise recommendations for rare data issues.

5. **Supported Data Formats:** The system currently supports CSV, Excel, and JSON. Extending compatibility to other formats such as XML, Parquet, or database exports will require additional module integration.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

9. FUTURE WORK

The proposed system establishes a solid foundation for automated data validation and correction; however, several extensions are planned to enhance scalability, flexibility, and user collaboration.

1. **Scalability Improvements:** Future iterations will focus on extending the system architecture to efficiently handle larger datasets containing thousands of rows and multiple concurrent users. This can be achieved through distributed computing frameworks or cloud-based deployment models to ensure high availability and performance.

2. **Custom Validation Logic:** The introduction of user-defined and domain-specific validation rules will allow greater adaptability across industries. Users will be able to configure their own exception detection logic, enabling the system to handle complex and specialized data scenarios beyond the existing rule set.

3. **Additional Data Formats:** Expanding compatibility to include XML, Parquet, and database exports will make the platform more versatile and suitable for modern data ecosystems. This enhancement will enable seamless processing of heterogeneous data sources across enterprise environments.

4. **Collaborative Features:** The next phase will incorporate real-time collaboration tools, version control, and audit trails, enabling multiple users to collaboratively validate, clean, and monitor shared datasets. These features will enhance transparency and teamwork in data quality management.

5. **Enhanced AI Recommendations:** The smart fix module will be fine-tuned with user feedback and contextual learning to produce more accurate and domain-aware correction suggestions. Continuous model updates will improve the reliability and precision of AI-driven exception explanations.

6. **Visualization and Reporting:** Advanced visual analytics and reporting modules will be integrated to help users identify exception patterns, track data quality metrics, and visualize improvements over time. This will make insights more actionable and easier to interpret.

7. **Integration with External Systems:** Planned developments include APIs and connectors for direct integration with Business Intelligence (BI) tools, ETL pipelines, and cloud storage platforms, streamlining the end-to-end data lifecycle from ingestion to analytics.

REFERENCES

[1]. R. W. O'Connor, "Managing data quality through automation," *Information & Management*, vol. 12, no. 3, pp. 125–132, 1987.
[2]. M. Guo, "Normal workflow and key strategies for data cleaning technology," *PMC*, 2023.
[3]. D. R. Chandran and V. Gupta, "A Short Review of the Literature on Automatic Data Quality," *Journal of Computer and Communications*, vol. 10, pp. 55–73, 2022.
[4]. F. Ridzuan et al., "A Review on Data Cleansing Methods for Big Data," *Procedia Computer Science*, vol. 161, pp. 1348–1355, 2019.
[5]. L. Ehrlinger et al., "A Survey of Data Quality Measurement and Monitoring Tools," *PMC*, 2022.
[6]. W. Elouataoui et al., "An Automated Big Data Quality Anomaly Correction Framework," *Data*, vol. 8, no. 12, 2023.
[7]. M. Mahdavi et al., "Towards Automated Data Cleaning Workflows," *CEUR Workshop Proceedings*, vol. 2454, 2019.
[8]. "AI/ML-Led Automated Data Quality: A Whitepaper Overview," *First Eigen Blog*, Oct. 2024.

International Journal of Global Engineering (IJGE)
ISSN: 2456-3099 (www.techpublic.in)
VOL 10 ISSUE 3 (2025) PAGES 20-32
RECEIVED:05.11.2025 PUBLISHED:20.11.2025

[9]. "Data Quality and Automation in Modern Data Ecosystems," *International Journal of Advances in Engineering & Management (IJAEM)*, vol. 7, no. 04, pp. 228–239, April 2025.
[10]. "Data Cleaning: Definition, Techniques & Best Practices," *TechnologyAdvice Blog*, June 3, 2024.
[11]. "How AI is Transforming Data Quality Management in 2025," *Techment*, Feb. 11, 2025.
[12]. "Automated Data Quality: Fix Bad Data & Get AI-Ready," *Atlan Blog*, May 15, 2025.
[13]. N. Shivaprasad, "Enhancing Data Quality through Automated Data Profiling," *International Journal for Research Publication & Seminar*, vol. 15, issue 4, Oct–Dec 2024.
[14]. F. Bayram, B. S. Ahmed, E. Hallin and A. Engman, "DQSOps: Data Quality Scoring Operations Framework for Data-Driven Applications," *arXiv:2303.15068*, Mar. 2023.
[15]. M. Abdelaal, S. Lokadjaja, A. Kreuz and H. Schöning, "DataLens: ML-Oriented Interactive Tabular Data Quality Dashboard," *arXiv:2501.17074*, Jan. 2025.
[16]. N. Bangad et al., "A Theoretical Framework for AI-driven Data Quality Monitoring in High-Volume Data Environments," *arXiv:2410.08576*, Oct. 2024.
[17]. H. C. Tamm and A. Nikiforova, "Towards Augmented Data Quality Management: Automation of Data Quality Rule Definition in Data Warehouses," *arXiv:2406.10940*, June 2024.
[18]. "Automating Data Quality Checks with Great Expectations," *Medium Blog*, 2020.
[19]. V. Panwar, "AI-Powered Data Cleansing: Innovative Approaches for Ensuring Database Integrity and Accuracy," 2024.
[20]. "Data Quality in AI: Challenges, Importance & Best Practices," *AI-Multiple*, Sep. 24, 2025.
[21]. T. H. Davenport, *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, 1993.
[22]. R. Kimball and J. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd Ed., Wiley, 2013.
[23]. T. Redman, *Data Driven: Profiting From Your Most Important Business Asset*, Harvard Business Review Press, 2008.
[24]. E. Rahm and H. Hasse, "Data Cleaning: Problems and Current Approaches," *IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 3–13, 2000.
[25]. W. H. Inmon, *Building the Data Warehouse*, 4th Ed., Wiley, 2005.
[26]. "Top 4 Automated Data Validation Tools You Need to Know in 2025," *Numerous.ai*, Oct. 30, 2025.
[27]. "AI Data Cleaning: Your Complete Guide," *Astera*, Aug. 27, 2025.
[28]. "Future Trends in Data Quality: AI and Machine Learning," *Keymakr*, June 2, 2025.
[29]. "Data Validation Techniques: Complete Guide 2025," *Alation*, July 23, 2025.
[30]. "Implementing AI in Data Cleaning: Challenges and Solutions," *ixsight*, 2025.
[31]. "AI-powered Data Quality: A Framework for Success," *Infosys Blogs*, Feb. 23, 2025.
[32]. "Data Validation Automation: A Key to Efficient Data Management," *Functionize*, Jan. 5, 2025.
[33]. "Adaptive Federated Data Cleaning with Explainability," *IJSRSET*, April 29, 2025.
[34]. "How AI Is Transforming Data Quality Management," *Acceldata*, Sep. 25, 2024.
[35]. "Automated Data Validation Tool For Operational & Transactional Data," *FirstEigen*, Dec. 15, 2024.