International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Data Forge Shape Your Data into Clarity

Lohitha Lakshmi K, Hema Sri S, Shaik Reshma, Hima Sai Nandhan P, Manoj Kumar Reddy S D V
Department of Artificial Intelligence & Data Science, Vasireddy Venkatadri Institute of Technology, Andhra Pradesh, India

Abstract- Data plays a key role in analysis and machine learning, but working with real-world datasets is often challenging because they usually contain missing values, duplicate entries, inconsistencies, and noise that can affect the accuracy of results. Data cleaning is therefore an essential step, yet it can be time-consuming and often requires programming knowledge, making it less convenient for many users. In this work, we present DataForge, a data preprocessing system designed to make the cleaning process simpler and more accessible. The platform allows users to upload datasets and perform cleaning operations without writing code, using a mix of statistical methods and simple intelligent techniques to handle issues such as missing data, outliers, and duplicate records. Overall, DataForge focuses on reducing the effort required for data preparation while still helping users work with more reliable datasets. This approach also helps users get a clearer idea of their data without going into too much technical detail.

Keywords- Airbnb Analytics, Data Visualization, Sharing Economy, Hospitality Industry, Exploratory Data Analysis (EDA), Customer Reviews, Pricing Analysis, Geographic Visualization, Business Intelligence, Data-Driven Decision Making, Tourism Analytics, Interactive Dashboards.

I. INTRODUCTION

The Data is being used more and more in different areas today, especially for analysis and machine learning, but working with real-world data is not always simple. In many cases, datasets come with problems like missing values, repeated entries, or inconsistent formats, and these can affect the results if they are not handled properly. Because of this, cleaning the data becomes an important step before doing any kind of analysis.
Data Even though it is important, data cleaning can be a bit time-consuming and not always easy to manage. Most of the time, people depend on tools like Python or SQL, which means some level of programming knowledge is required. This can make things difficult for users who are not from a technical background. Also, many existing tools only solve parts of the problem, and users still need to do a lot of things manually. Sometimes, tasks like understanding the data and cleaning it are handled separately, which makes the overall process feel a bit disconnected.

To make this easier, this paper introduces DataForge, a data preprocessing system developed to simplify the cleaning process. The idea is to provide a platform where users can upload their datasets and apply different cleaning steps without writing code. The system uses a combination of basic statistical methods and a few simple intelligent techniques to handle common issues such as missing values, duplicates, and outliers. By bringing everything into a single flow, the system reduces manual effort and helps users work with cleaner and more reliable data in a more straightforward way.

Another thing is that most users today prefer tools that are simple to use and don't take much time to understand. In many cases, people still end up using multiple tools or doing extra steps just to clean and check their data, which can feel a bit unnecessary and tiring. From our side, the idea was to keep things more simply direct, so users don't have to worry too much about the process itself. Instead, they can focus on their data and what they actually want to do with it. This also makes the overall workflow feel easier to follow, especially for someone who is just getting started. At the same time, it also reduces the chances of mistakes that usually happen with too many manual steps. We also felt that keeping things simple would make the system more practical in real use. Overall, the focus is more on making the experience smooth rather than making it complicated.

II. LITERATURE REVIEW

Literature survey is a crucial step in any software development process. Before actually building the system, we felt it was important to first look at what is already available and how

similar problems are being handled. This gives a rough idea of the existing approaches and also helps in noticing where things are not working that well. In data preprocessing, there are already quite a few methods and tools, but in practice, they are not always as simple or convenient as they seem.

In earlier methods, people mostly depended on spreadsheets or programming libraries like pandas to clean data. These are useful for handling things like missing values, duplicates, and basic formatting. But at the same time, they need manual work and some programming knowledge, which not everyone is comfortable with. Because of this, the whole process can feel a bit slow and sometimes even confusing, especially for someone who is just starting out. Also, one small issue is that these methods don't really show clearly how much the data has improved after cleaning.

To make things easier, some tools were later introduced with more interactive features. Tools like OpenRefine allow users to perform operations like filtering, grouping, and formatting in a more visual way. This does make things slightly easier, but still, most of the decisions depend on the user. There is not much support for automatic suggestions, and understanding the overall quality of the data is still not very clear in many cases.

More recently, machine learning-based approaches have also been used in data cleaning. For example, methods like MICE try to fill missing values by learning patterns from the data, and techniques like the Interquartile Range (IQR) are used to identify outliers. These methods can give better results, but they are not always easy to apply directly and usually need a bit more understanding. Because of that, they are not very friendly for beginners or non-technical users.

Even with all these options, there still seems to be a gap when it comes to having something that is both simple and complete at the same time.

Many tools only handle certain parts of the process, so users often end up switching between different tools or doing extra steps, which can feel unnecessary. This also makes the workflow less smooth. In our approach, we tried to keep things simple and combined these steps into one system, where users can clean, understand, and work with their data in a single place without too much complexity.

Another thing we noticed while going through existing approaches is that many tools focus more on the technical side rather than the user experience. While they do provide useful features, they are not always easy to understand for someone who just wants to quickly clean their data and move on. In some cases, even simple tasks take multiple steps, which can feel a bit unnecessary. Because of this, users may spend more time figuring out the tool itself rather than actually working with their data, which is not ideal.

III. PROPOSED SYSTEM

The proposed system, DataForge, was mainly created to make data cleaning feel less confusing and easier to handle. In many cases, people end up using different tools or writing code just to clean their data, which can take extra time and effort. Here, the idea was to keep everything simple by bringing all the steps into one place, where users can just upload their dataset and work on it directly. The system takes care of common issues like missing values, duplicates, and messy data in the background, so users don't have to worry too much about how it works. Overall, it is designed to make the process more straightforward, so users can spend less time on cleaning and more time actually using their data.

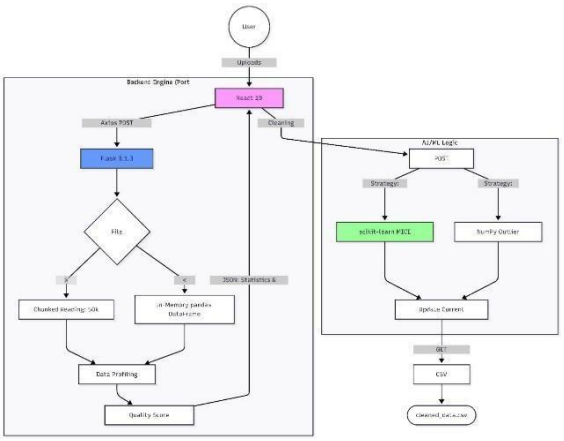

Fig: System Architecture

The system architecture of DataForge is kept simple so that users don't feel overwhelmed while using it. Once a dataset is uploaded through the interface, the system takes care of the rest by checking the file and processing it step by step in the background. It handles common issues like missing values, duplicates, and inconsistent data without requiring much input from the user.

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Instead of splitting everything into multiple complicated steps, the flow is kept smooth and connected, making it easier to follow. After processing, the cleaned data is shown to the user and can be downloaded if needed, keeping the overall process quick and straightforward.

Block Diagram

The block diagram of DataForge just gives a simple idea of how the system is set up and how the data moves through it. It's not meant to be too detailed or technical, but more like a quick way to understand what happens behind the scenes. Starting from the user side, it shows how a dataset is uploaded and then passed through different parts of the system. Each block represents a small part of the overall process, and when you look at it together, it gives a clear picture of how everything is connected and working in sequence.

Once the file is uploaded, it goes to the backend where the main processing happens. The system takes care of things like missing values, duplicates, and messy data on its own, so the user doesn't have to worry much about it. After that, the cleaned data is sent back to the interface, where it can be viewed and downloaded easily. This way, the whole flow feels simple and easy to follow without making things look too technical.

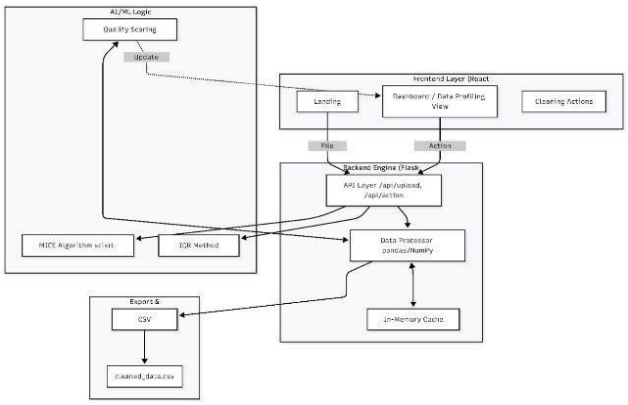

Fig: Block Diagram

Class Diagram

The class diagram of DataForge is just there to give a rough idea of how the system is set up from the inside. It's not really about showing what happens step by step, but more about how different parts are connected and what each one is responsible for. We didn't want to make it look too heavy or detailed, so it's kept quite simple, just enough to understand how the system is organized.

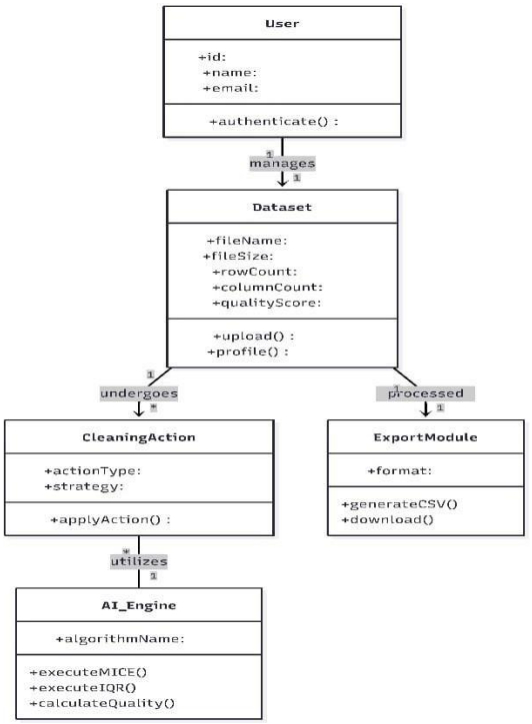

Fig: Class Diagram

In the diagram, you can see that the user part is where everything starts, like uploading a file or triggering the cleaning process. After that, other parts of the system take care of storing the file, processing the data, and sending the results back. All these parts are connected in a simple way, so the data just moves from one place to another without any unnecessary steps.

Overall, it gives a basic picture of how everything works together inside the system without making it look too complicated.

Sequence Diagram

The sequence diagram of DataForge is just there to show what really happens when someone uses the system, step by step. It doesn't go into too much detail, it just gives a simple idea of how things move from one part to another once the user starts interacting with it. From uploading the file to getting the result back, it shows the order in which things happen so it's easier to understand the flow without overthinking it

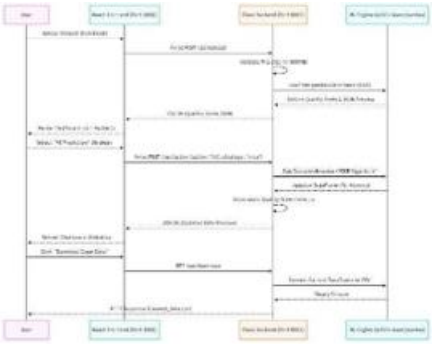

Fig: Sequence Diagram

So basically, it starts when the user uploads a dataset, and the system receives it and does a quick check to see if everything looks fine. Then it passes the data to the backend where all the actual work happens, like cleaning the data and fixing common issues. Once that part is done, the result is sent back, and the user can just view it or download it. In the end, it's just a simple back-and-forth between the user and the system, nothing too complicated, just enough to show how things work. It mainly helps in getting a clear picture of how the system responds when a user gives an input.

One more thing is that the sequence diagram also helps in seeing how smoothly everything works when a request is given. It gives a simple idea of how the system responds at each step, from taking the input to sending the result back. Even though there are multiple parts involved, the whole process still feels quite simple and connected, which is what we were trying to achieve.

IV. METHODOLOGY

The methodology of DataForge is designed to make data cleaning simple and easy to handle. The process begins when a user uploads a dataset in CSV or Excel format, after which the system quickly checks if the file is valid. Once accepted, the data is loaded and examined to understand its condition. Instead of directly cleaning it, the system first identifies issues like missing values, duplicates, and data types. It also gives a basic quality score so users can clearly see how much improvement the data needs.

After understanding the data, the system applies suitable cleaning methods. Missing values are filled using mean, median, or mode, and in more complex cases, the MICE method is used for better accuracy. Duplicate entries are removed, and outliers are filtered using the IQR technique to keep the data consistent. Built with Flask and React, the platform shows changes in real time, making the process easy to follow. In the end, the system ensures the data becomes cleaner, more reliable, and ready for further use with less manual effort.

In addition to this, the system is designed in a way that users don't feel lost at any stage of the process. The overall idea is not just to clean the data, but to make the entire experience smooth, clear, and less time-consuming for the user

V. RESULT

a. User Interface

The Home Page of DataForge gives a very simple and comfortable starting point for users. From the moment the page opens, it clearly shows what the platform is meant for turning raw and messy data into something cleaner and more meaningful. The design doesn't feel heavy or confusing; instead, it keeps things neat so users can quickly understand what they are looking at without overthinking.

Fig: Home Page

Looking at the page, it feels like it gently guides the user rather than forcing too much information at once. The main idea of improving data quality is presented in a straightforward way, making it easy to connect with. Overall, the Home Page does a good job of setting the tone for the project by keeping everything clear, simple, and easy to follow, which makes users feel comfortable to move ahead and start using the system.

Fig: About Page

The About Page of DataForge simply explains what the platform is about in a very easy and clear way. It shows that the main idea behind the system is to make data cleaning less confusing and less time-consuming. Instead of making users deal with complicated steps, the platform tries to handle things in a smoother and more understandable way using AI along with a simple interface.

When looking at the page, it feels like the focus is more on helping users rather than showing too many technical details. It talks about things like better accuracy, faster results, and the ability to handle different sizes of data, which makes it clear that the system is built to be practical. Overall, the page gives the impression that DataForge is meant to make working with data easier, so users can spend less time fixing their data and more time actually using it.

Fig: How It Works Page

The How It Works section explains the process in a very simple and easy way by breaking it down into three basic steps. It starts with the user uploading a dataset in formats like CSV or Excel, which is made quick and straightforward so users don't face any confusion. After the upload, the system takes care of the main part by looking into the data, identifying common issues like missing values, duplicates, or unusual patterns, and fixing them using suitable methods. Since most of this is handled automatically, users don't need to worry much about the technical side. Once the cleaning is done, the updated data is shown to the user, and they can easily download it. Overall, the entire process feels smooth and guided, making data cleaning much easier and less time-consuming.

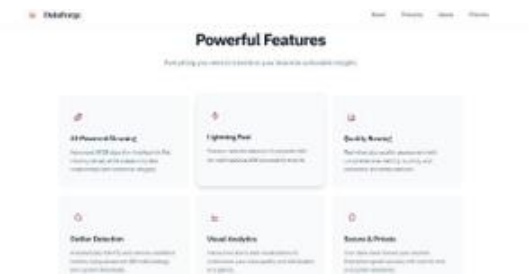

Fig: Features Page

The Features section of DataForge simply shows what the platform is capable of in a very easy and understandable way. It mainly focuses on how the system makes data cleaning less of a burden by using AI to handle things like missing values in a smarter way, without affecting the overall data.

The platform also feels quick in handling data, even when the dataset is large, which saves a lot of time. It gives a data quality score as well, so users can easily get an idea of how their data looks before and after cleaning.

Along with this, it can automatically spot unusual values and remove them, helping keep the data more consistent. The visual charts make it easier to understand what is happening with the data without needing deep technical knowledge. At the same time, the system keeps user data safe, which adds a sense of trust. Overall, all these features come together to make the whole process feel simple, smooth, and less time-consuming for anyone using it.

Fig: Upload Page

The Upload Page of DataForge is where everything starts, and it feels very simple and easy to use right from the beginning. When looking at the page, it doesn't feel confusing or overloaded users can quickly understand what to do and upload their dataset without thinking too much.

Whether it is a CSV or Excel file, the process feels smooth, and the system quietly handles the checks in the background to make sure everything is fine. What makes this step feel comfortable is that it doesn't require any technical effort from the user. The design is clean, and the whole experience feels quick and straightforward, which helps users move ahead without hesitation. Overall, the Upload Page creates a relaxed and clear starting point, making it easy for anyone to begin the data cleaning process without any difficulty.

Fig: Messy Dataset Uploaded

This page shows how the data looks right after it is uploaded, and it clearly feels unorganized at first glance. You can easily see problems like missing values, repeated rows, and some incorrect entries, which affect the quality of the data. The system points out these issues and also gives a basic quality score along with simple details, so users can quickly understand what needs to be fixed before cleaning.

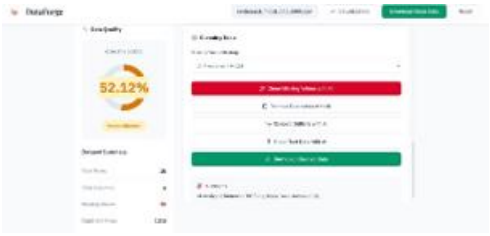

Fig: Removing Duplicate values with AI

After the dataset is uploaded, the system evaluates its quality by identifying missing values, duplicates, and inconsistencies, resulting in an initial low-quality score of about 19.91%.
When the AI-based duplicates removal applied, the system automatically improves the data by predicting missing values and enhancing consistency based on existing patterns.

This process significantly increases the data quality score to 52.12%, producing a duplicate removal and more reliable dataset that is ready for further analysis with minimal user effort.

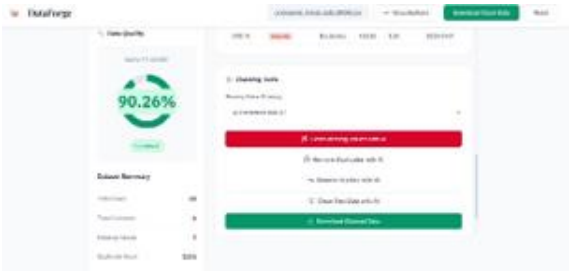

Fig: Removing Outliers with AI

After the missing values are cleaned, the system continues by removing outliers to further improve the dataset. In this step, the AI carefully examines the data to find values that do not naturally fit with the rest of the dataset and may have occurred due to mistakes, noise, or unusual conditions. Instead of relying on fixed rules, the system learns from the data itself to decide which values are truly abnormal. By removing these irregular entries, the dataset becomes more balanced and consistent, leading to an improved data quality score of 90.26% and making the data more dependable for analysis and future modeling tasks.

Fig: Cleaning Text data with AI

After completing the numerical cleaning steps, the system cleans the text data using AI to remove unwanted characters, symbols, and formatting issues that commonly appear in real-world datasets.

The AI carefully processes each text entry, keeping meaningful content while eliminating noise that can affect analysis.

This step cleans thousands of text records in a single operation and helps finalize the dataset, resulting in a 100% data quality score, which indicates that the data is fully clean, consistent, and ready for further use.

Fig: Visualization pages

After the data is cleaned, the system provides a data visualization page that makes it easier for users to understand what the dataset looks like at a glance. Instead of going through rows of raw data, users can see simple charts that show where missing values were present, how different data types are distributed, and how values like price and quantity change across records.

These visualizations help confirm that the cleaning steps worked as expected and give users more confidence in the quality of the data and also the user can download the needed visualizations. Overall, this module helps users better connect with the dataset and ensures it is ready for meaningful analysis or further use.

VI. CONCLUSIONS

DataForge was developed to address a very practical and common problem faced today working with messy, unprepared data. In many real-world scenarios, data is incomplete, inconsistent, and difficult to analyze without spending significant time on cleaning. This project brings together multiple preprocessing steps into a single, easy-to-use web platform, allowing users to upload a dataset, clean it using AI-based options, and understand the results through clear visualizations. By automating tasks such as handling missing values, removing outliers, and cleaning text data, DataForge reduces the manual effort traditionally required in data preparation.

One of the key strengths of this project is its simplicity combined with intelligent automation. Users do not need advanced technical knowledge to prepare high-quality datasets, making the tool useful for students, researchers, and professionals alike. The visible improvement in data quality scores and the immediate visual feedback help users trust the cleaning process and better understand their data. In today's data-driven world, where accurate insights depend heavily on data quality, DataForge offers a practical and efficient solution. Its uniqueness lies in integrating AI-assisted cleaning, quality scoring, and visualization within a single workflow, making data preparation faster, more intuitive, and more reliable for real-world analytical and machine learning applications.

REFERENCES

1. Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann Publishers, Elsevier, USA.
2. García, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Intelligent Systems Reference Library, Springer, Cham.
3. Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23(4), 3–13.
4. Kandel, S., Heer, J., Plaisant, C., Kennedy, J., Van Ham, F., Riche, N. H., Weaver, C., Lee, B., & Stasko, J. (2011). Research directions in data wrangling: Visualizations and transformations for usable and credible data. Information Visualization, 10(4), 271–288.
5. Aggarwal, C. C. (2017). Outlier Analysis (2nd ed.). Springer International Publishing, Switzerland.
6. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly Media, USA.
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
8. McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 56–61.
9. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press, USA.
10. IBM Corporation. (2020). Data Quality Dimensions. IBM Knowledge Center.
11. Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data cleaning: Overview and emerging challenges. Proceedings of the 2016 International Conference on Management of Data (SIGMOD), 2201–2206. ACM, USA.
12. Abedjan, Z., Golab, L., & Naumann, F. (2016). Profiling relational data: A survey. The VLDB Journal, 24(4), 557–581.
13. Zhang, Z. (2018). Missing data imputation: Focusing on single imputation. Annals of Translational Medicine, 6(4), 1–9.
14. Wilkinson, L. (2005). The Grammar of Graphics (2nd ed.). Springer Science + Business Media, New York, USA.

15. Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM Computing Surveys, 41(3), Article 16.