




PROJECT AND TEAM INFORMATION

Project Title

Smart Data Pre-processor: A Web-Based Tool for Cleaning and Visualizing Machine Learning Datasets

Student / Team Information

<i>Team Name:</i>	Sapphire
Team member 1 (Team Lead)	<p>Gangwar, Atharv – 230112386</p> <p>atharvgangwar8@gmail.com</p> 
Team member 2	<p>Negi, Dhruv – 23011451</p> <p>negi67291@gmail.com</p> 
Team member 3	<p>Negi, Abhishek – 23011732</p> <p>abhisheknegi75054@gmail.com</p> 

PROPOSAL DESCRIPTION

Motivation

Machine Learning (ML) and Deep Learning (DL) models rely heavily on high-quality datasets. However, raw datasets often contain missing values, outliers, and inconsistent formats, leading to poor model performance. Currently, data scientists spend significant time manually cleaning data, which is inefficient and error-prone. Our project aims to simplify this process by providing an intuitive web-based application that automates dataset pre-processing, ensuring that users can quickly prepare their data for ML/DL models. We also have data visualization feature for users to understand their data in a better way.

State of the Art / Current solution

Currently, data pre-processing is performed using programming libraries like Pandas (Python), Weka (Java), and Apache Spark. However, these solutions require coding knowledge and are not accessible to users without programming expertise. Some existing web-based tools focus on visualization but lack comprehensive pre-processing capabilities. Our project bridges this gap by providing an easy-to-use web application that combines **data cleaning, pre-processing, and visualization** in one platform.

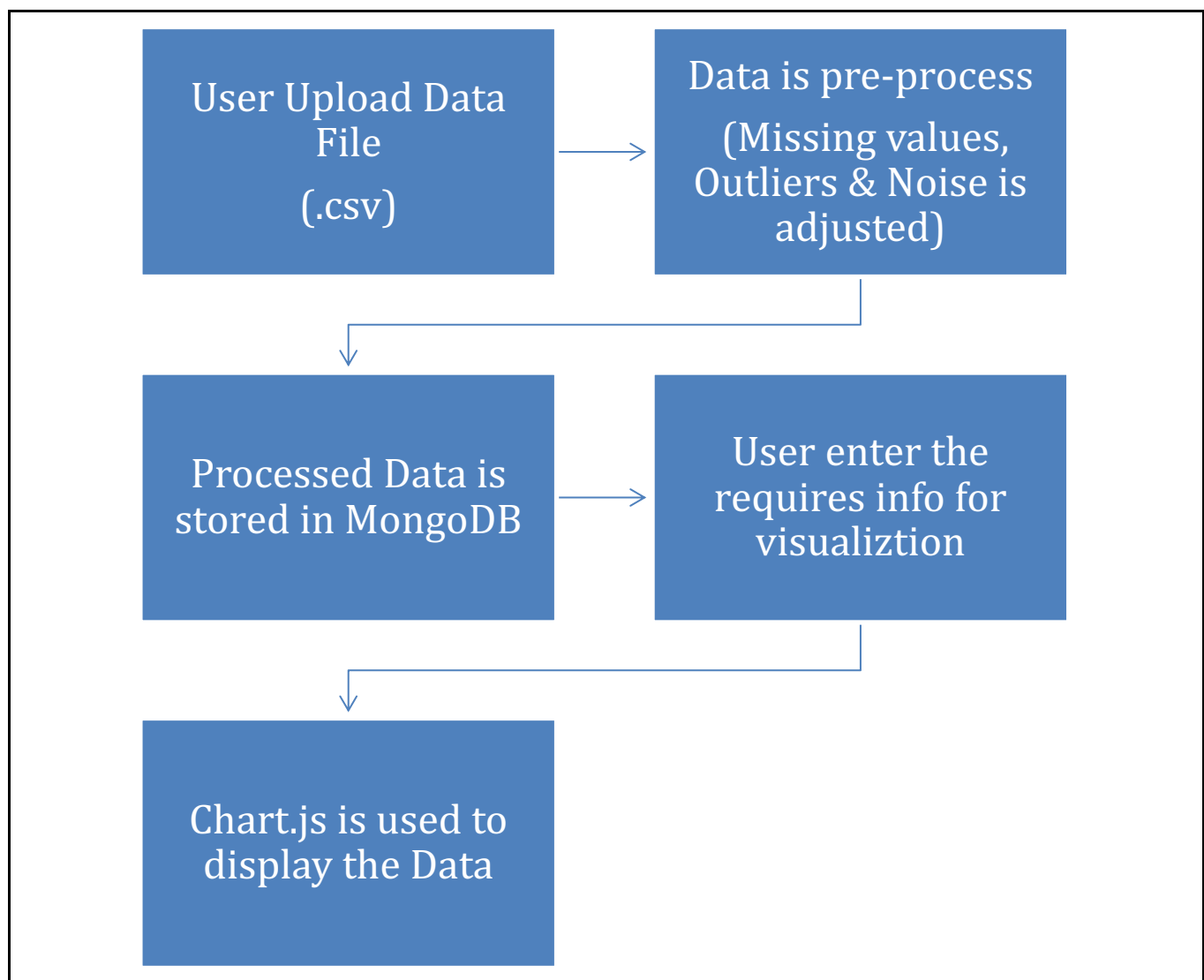
Project Goals and Milestones

- Develop a **file upload system** to handle CSV datasets.
- Implement **data cleaning features** (handling missing values, removing outliers, and normalizing data).
- Store cleaned data in a **MongoDB database** for further processing.
- Create a **React frontend** for intuitive user interaction.
- Integrate **Chart.js** for interactive data visualization.
- Deploy the system on **Netlify** for accessibility.

Project Approach

- **Frontend:** Built using **React** with **Chart.js** for visualizing processed datasets.
- **Backend:** Developed in **Java (Spring Boot)** for handling data processing and API management.
- **Database:** **MongoDB** will be used for storing raw and processed datasets.
- **Processing Libraries:** **Apache Commons CSV** (file handling), **Weka** (data cleaning & ML preprocessing), **ND4J** (numerical processing for DL).
- **Deployment:** Hosted on **Netlify**.

System Architecture (High Level Diagram)



Project Outcome / Deliverables

- A **fully functional web application** for dataset pre-processing.
- Support for **CSV uploads, data cleaning, and visualization**.
- API endpoints to allow integration with ML pipelines.
- A final project report documenting the implementation and outcomes.

Assumptions

- Users will upload **structured datasets (CSV format)**.
- Users will require basic pre-processing, including **handling missing values, outlier detection, and data normalization**.
- The application will initially support small to medium-sized datasets.

References

- Weka: <https://www.cs.waikato.ac.nz/ml/weka/>
- Apache Commons CSV: <https://commons.apache.org/proper/commons-csv/>
- ND4J: <https://deeplearning4j.konduit.ai/nd4j>
- Chart.js: <https://www.chartjs.org/>