Smart Data Pre-processor – A Web-Based Tool for ML Dataset Cleaning & Visualization

Project Introduction

- We are **Team Sapphire**, and we are excited to present our project:
- **"Smart Data Pre-processor: A Web-Based Tool for Cleaning and Visualizing Machine Learning Datasets."**
- This project aims to simplify the often complex and time-consuming process of preparing datasets for Machine Learning and Deep Learning models.

Team Members

- **Atharv Gangwar** – Team Lead – 2318555
- **Dhruv Negi** – 23011451
- **Abhishek Negi** – 23011732

Problem Statement

Most machine learning models don’t work well with dirty data.
But cleaning data takes a lot of time and usually requires programming skills.

Many users — especially students and beginners — don’t know how to write code for data preprocessing.

Our project solves this by providing a simple tool where anyone can **upload a CSV file**, **clean the data**, and **visualize**

What’s Unique About Our Project?

Unlike other tools, our solution:

Is **100% no-code:** Users can clean and visualize datasets without writing a single line of code.

Combines **preprocessing and visualization** in one place.

Supports **CSV uploads**, **automated data cleaning**, and **interactive charts**.

Offers a **modular backend** that can be plugged into any ML pipeline.

Technology Stack & Justification

Layer	Tech Used	Why We Chose It
Frontend	React.js + Chart.js	Fast, interactive UI + chart rendering
Backend	Java (Spring Boot)	Scalable API development + Java ecosystem
Database	MongoDB	Flexible schema for raw and cleaned data
Processing	Weka, Apache Commons CSV, ND4J	Trusted Java libraries for data handling
Deployment	Netlify (Frontend), AWS/GCP (Backend)	Free + scalable cloud options

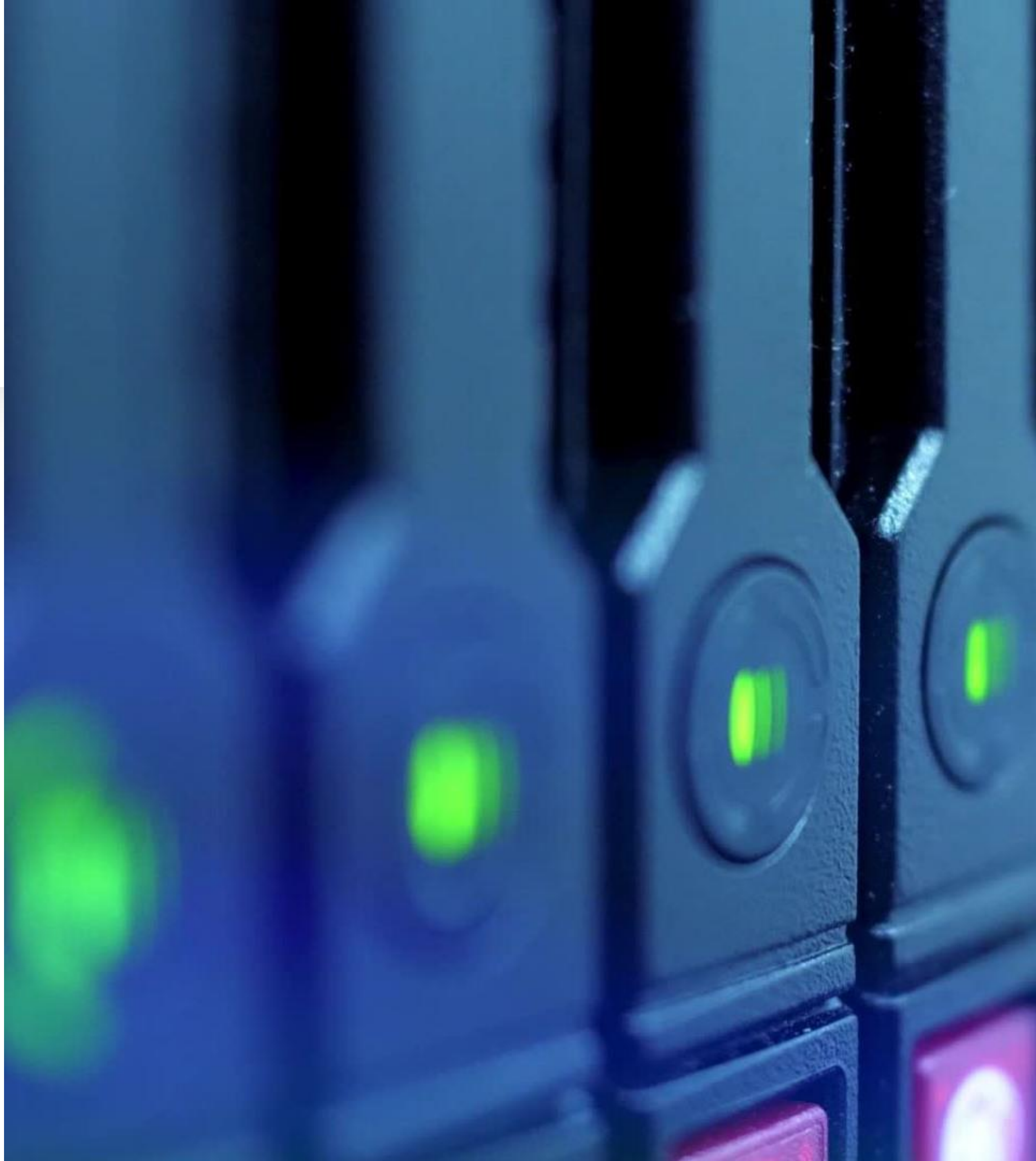
Step-by-Step Implementation Plan

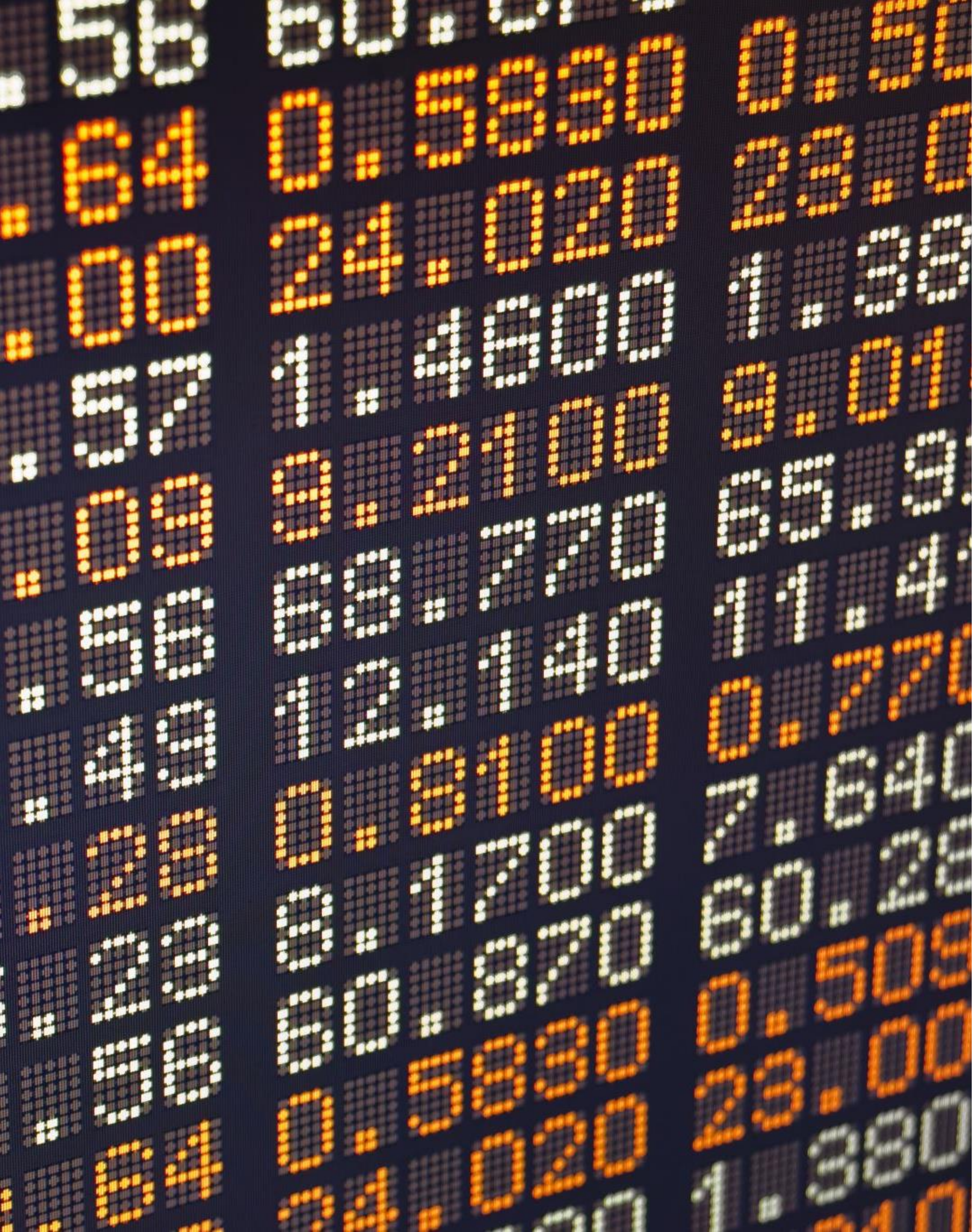
Step 1: File Upload (Frontend + Backend)

- User uploads a **CSV file** using the React frontend.
- File is sent to the **Java backend** via a REST API.

Step 2: Data Preprocessing (Backend - Java)

- Using **Apache Commons CSV** to read the file.
- Clean the data by:
  - Filling missing values
  - Removing outliers
  - Normalizing numeric columns
- Use **Weka** and **ND4J** for preprocessing operations.





Step-by-Step Implementation Plan

Step 3: Save Cleaned Data (Backend + Database)

- Store cleaned data and its summary into **MongoDB**.

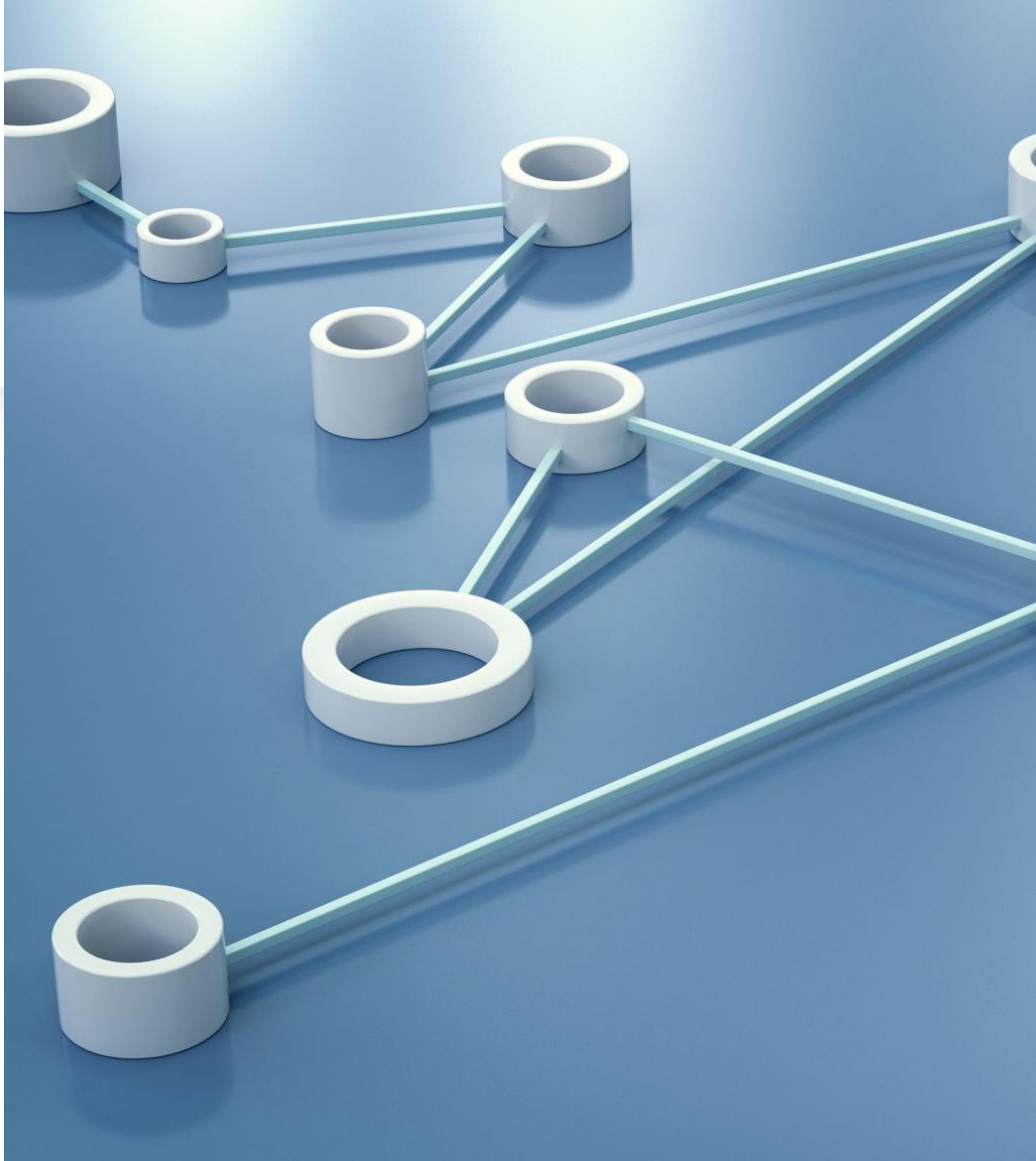
Step 4: Visualize Data (Frontend)

- Use **Chart.js** to show:
  - Histograms
  - Line graphs
  - Summary stats (mean, median, etc.)

Step-by-Step Implementation Plan

Step 5: Deploy the Web App

- **Frontend on Netlify** (free hosting)
- **Backend on Render**
- **MongoDB Atlas** as the cloud database



Project Deliverables

- Fully working web application
- Support for:
    - Uploading CSV
    - Data cleaning
    - Data visualization
- REST APIs for integration with ML models
- Project report documenting all phases

Future Scope

- Add auto-generated ML models using Scikit-Learn/TensorFlow
- Enable user-based dashboards and model deployment via API
- Monetization via freemium model (students, startups, data analysts)

Conclusion

- Our Smart Data Pre-processor makes ML easier for everyone. It bridges the gap between technical and non-technical users by offering a simple, interactive, and powerful data cleaning platform.

We are happy to take any questions.

Thank you!