

# DeepStat:

# A Data Analyser

A Capstone Project

Aakanksha Pote (D014)

Preet Jain (D023)

Sakshi Prabhu (D026)



# Introduction

## Research Gap

Many people, especially those from non-technical backgrounds, struggle to analyze data due to lack of coding skills.

## Purpose of the Project

Our goal is to bridge this gap by providing a user-friendly platform that simplifies data analysis.

## Target

Aim to make data analytics and data science accessible to everyone



From Raw Data to Clarity

DeepStat is a user-friendly and comprehensive web-based platform designed to simplify data analysis for individuals and businesses, regardless of their technical expertise. It eliminates the need for multiple tools and complex programming, making data science more approachable for all.

With DeepStat, users can effortlessly upload datasets, clean and preprocess data, identify correlations, and generate insightful visualizations—all without writing a single line of code.



# DeepStat

From Raw Data to Clarity

[Go to Website](#)

# What Makes DeepStat Better?



## No Code

Use advanced ML and statistical technique without coding

## Beginner-friendly Interface

Simple, guided workflow using a clean sidebar and step-by-step interaction

## All in one platform

From data upload → cleaning → visualization → modeling— everything in one flow.

## Completely Free

No login. No installation headaches. Just upload and go!

# Methodology

## System Architecture

It is designed to ensure smooth data processing, model execution, and user interaction—all through a simple web interface.

## Technology Used

Python language was used throughout whereas libraries like pandas, streamlit, matplotlib, etc were used.

## Application Workflow

The workflow makes it possible for even non-technical users to conduct end-to-end data analysis with minimal effort.

## Usability Testing

Testing was conducted to identify pain points, improve the user experience, and validate the platform's effectiveness.

# System Architecture



01

## Frontend

Implemented using Streamlit (interactive UI)

02

## Backend

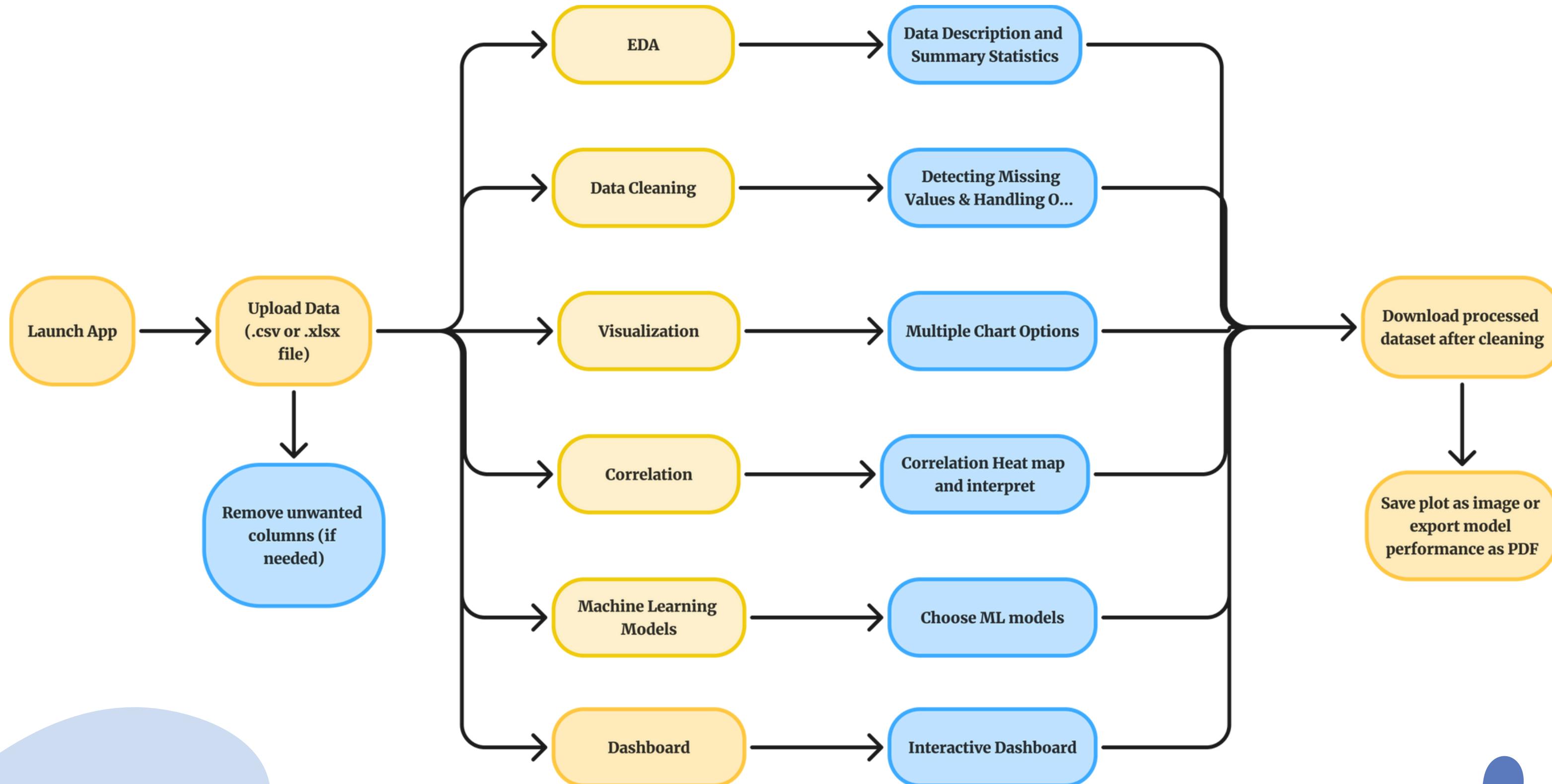
Data processing and machine learning using Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn and Plotly.

03

## Data Flow

How user-uploaded data is processed (cleaning, transformation, visualization, modeling).

# Application Workflow (User Interface Experience)



# Usability Testing

**01**

## Participant Selection & Testing Approach

- The application was tested by a small group of students and peers from non-technical backgrounds and a task-based approach was adopted.

**02**

## Key Findings

- Users were able to upload datasets and visualize data without external help.
- Most users appreciated the dropdown-based selection of models and automatic generation of evaluation metrics.
- Minimal errors occurred when users uploaded incorrectly formatted data, and appropriate error messages were displayed.

**03**

## Improvements based on Feedback

- More descriptive labels were added.
- Button placements were reorganized for better flow.

# Results

## User Interface (UI)



### Clean and Intuitive Design

Simple navigation for users with no technical background.

### Dashboard Overview

Displays key insights, data previews, and analysis options.

### Interactive Visualizations

Dynamic charts, graphs, and correlation heatmaps.

### Step-by-Step Workflow

Guided process for data cleaning, EDA, and modeling

# Conclusion

## Bridging the Gap

DeepStat simplifies data analysis for users without a technical background.

## All-in-One Platform

Offers data cleaning, visualization, correlation analysis, and machine learning in a single interface.

## User-Friendly Interface

Streamlit makes it interactive and easy to use.

## Efficient & Accurate Analysis

Uses powerful libraries like Pandas, NumPy, Seaborn, and Scikit-learn for seamless data processing and model evaluation.

# Future Work

## Add Hyperparameter Tuning Options

Handle unstructured data formats like text (NLP analysis) and images. Implement clustering (e.g., K-Means, DBSCAN) and dimensionality reduction (e.g., PCA) to expand analytical capabilities.

## Real-Time Data Processing

Enable real-time analytics by integrating with streaming data sources and live databases.

## Introducing Deep Learning Capabilities

Integrate deep learning frameworks like TensorFlow and PyTorch for advanced predictive modeling.

## Mobile & Tablet Compatibility

Develop a mobile-friendly interface to allow users to perform quick analyses on the go.

# Applications

**01**

## Healthcare Analytics

Customize modules for analyzing patient data and aiding clinical research.

**02**

## Educational Tools

Develop a classroom version to introduce students to data science through guided examples.

**03**

## Marketing Dashboards

Build domain-specific templates for customer segmentation, campaign tracking, and KPI monitoring.

# Team behind DeepStat



**Aakanksha Pote**



**Preet Jain**



**Sakshi Prabhu**

# Acknowledgement

We would also like to thank our mentor, **Dr. Suresh Pathare**, for his guidance and support throughout this project. His expertise and insights were essential for our project's satisfactory results.







# Website Interface

**Data Input & Analysis Options**

Upload a CSV file

Drag and drop file here  
Limit 200MB per file • CSV

Browse files

## DeepStat: From Raw Data to Clarity

Description   About the Team

### Project Description



DeepStat is a comprehensive data analysis tool designed to simplify the process of extracting meaningful insights from raw data. It provides a user-friendly interface to upload, explore, clean, visualize, and model data, empowering users to make data-driven decisions.

Key Features Include:

- Data Upload and Exploration: Easily upload CSV files and get a quick overview of the data.
- Data Cleaning: Tools to handle missing values and outliers.
- Data Visualization: Generate various plots to understand data patterns.
- Model Training and Prediction: Train machine learning models and make predictions.

# Application Workflow



- 01 **Data Upload**  
Users upload CSV files through the Streamlit interface
- 02 **Data Preprocessing**  
Handling missing values, Outliers.
- 03 **Exploratory Data Analysis (EDA)**  
Visualization using bar charts, scatter plots, histograms
- 04 **Data Summary**  
Provides with the statistical properties of data
- 05 **Machine Learning Model Selection**  
Classification, regression, clustering and many more
- 06 **Result Interpretation and Prediction**  
Metrics like accuracy, precision.

# Limitations



01

## Missing Advanced Features

The lack of real-time collaboration, user login, and cloud database integration limits scalability for enterprise and research environments.

02

## Limited Customization of Machine Learning Models

Users can apply predefined models but have restricted control over hyperparameter tuning, advanced feature selection, and deep learning methods.

03

## Performance Issues with Large Datasets

Handling large datasets may slow down performance due to memory constraints and the limitations of Streamlit's real-time execution.

04

## Dependency on Structured Data

DeepStat is designed for structured tabular data and lacks support for unstructured data types like text, images, or audio processing.

# Working



- 01 **Upload Data**  
Users can upload CSV or Excel files
- 02 **Exploratory Data Analysis (EDA)**  
Gain insights into data distributions and relationships.
- 03 **Data Cleaning**  
Handle missing values, remove outliers and prepare datasets for analysis.
- 04 **Visualization**  
Create clear and impactful charts and graphs to interpret trends.
- 05 **Correlation Analysis**  
Identify relationships between variables and understand dependencies.
- 06 **Machine Learning Models**  
Implement ML algorithms with built-in preprocessing for better accuracy.

# Machine Learning & Preprocessing

- scikit-learn (sklearn)- A comprehensive machine learning library used for data preprocessing, model training, and evaluation. Key components include:

1. **Preprocessing:** OneHotEncoder, LabelEncoder, MinMaxScaler for transforming and normalizing data.
2. **Model Training:** LogisticRegression, RandomForestClassifier, SVC, KNeighborsClassifier for predictive analytics.
3. **Evaluation:** accuracy\_score, precision\_score, recall\_score, and f1\_score for assessing model performance.

## System Utilities

- time (import time) – Used for measuring execution time and optimizing performance.
- os (import os) – Provides file and directory management, enabling data import/export functionalities.
- PIL (Python Imaging Library) (from PIL import Image) – Used for handling and processing image data, useful for visualizations and uploading image-based datasets.