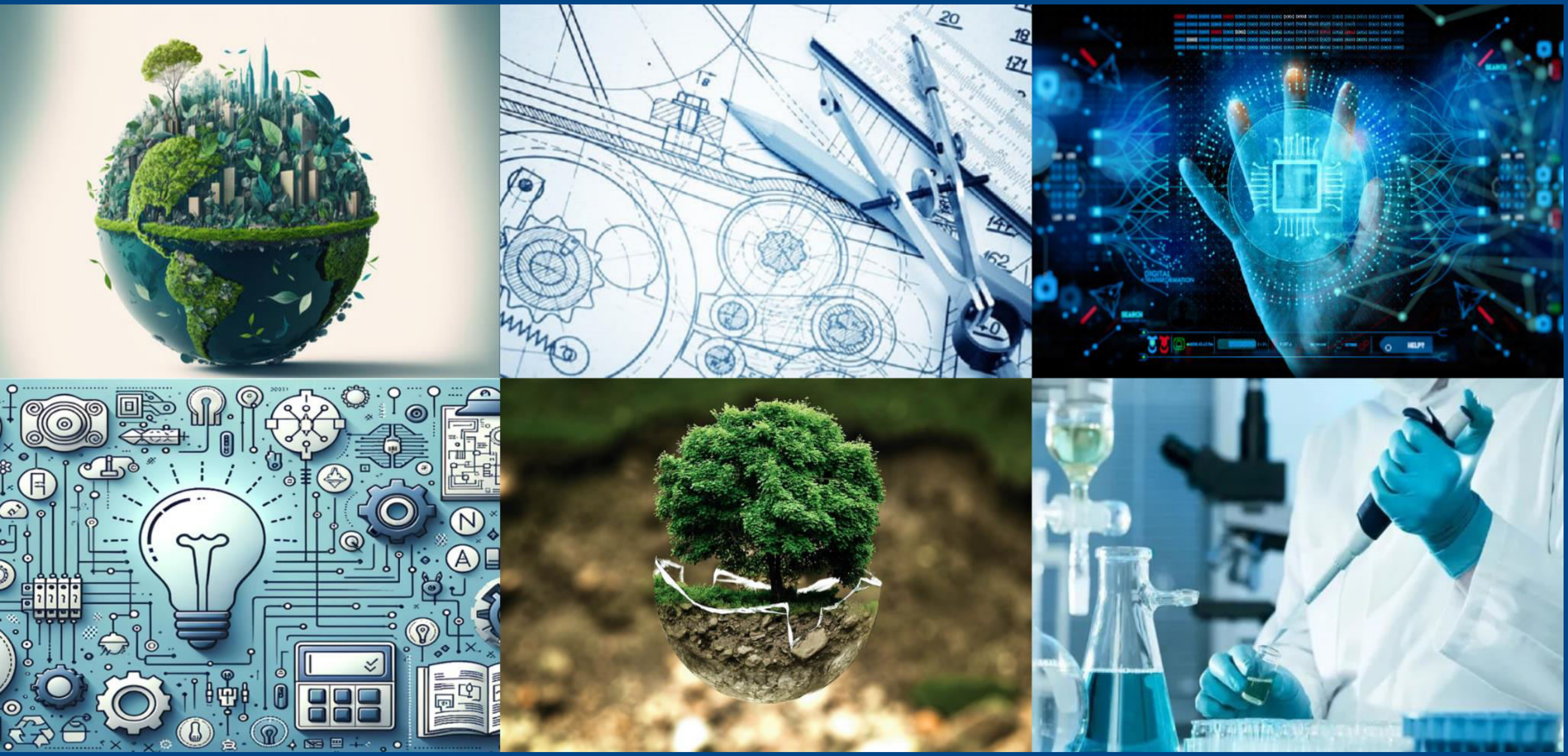
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 4, April 2025

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

DataBuddy: No-Code Data Science Tool

Kunal Yadav

B. Tech, Department of (CSE), Amity University, Uttar Pradesh, Lucknow, India

ABSTRACT: Data analysis and machine learning are increasingly essential in various industries; however, the complexity of existing tools creates barriers for non-technical users. This paper presents DataBuddy, a no-code tool designed to automate data analysis and machine learning processes through an intuitive graphical interface. DataBuddy integrates Python libraries like Pandas, Matplotlib, Seaborn, and Scikit-Learn, providing features such as automated exploratory data analysis (EDA), dynamic visualizations, and machine learning model training — all without requiring programming knowledge. The tool allows users to upload datasets, select columns, visualize data, and train predictive models through simple checkboxes and dropdown menus. Performance metrics and analysis reports are generated automatically and saved in organized folders, ensuring accessibility and efficiency. DataBuddy empowers users from diverse backgrounds to derive insights, make data-driven decisions, and build machine learning models with minimal effort. The results demonstrate the tool's effectiveness in simplifying complex workflows, reducing analysis time, and bridging the gap between technical and non-technical users.

Despite the growing importance of data analysis and machine learning across industries, many existing tools remain inaccessible to non-technical users due to their reliance on coding expertise. DataBuddy is a no-code tool that overcomes these challenges by providing a user-friendly graphical interface that allows users to perform data preprocessing, exploratory data analysis (EDA), visualization, and machine learning tasks without writing any code. The platform integrates Python libraries such as Pandas, Matplotlib, Seaborn, and Scikit-Learn to automate complex processes while offering simplicity through dropdowns, checkboxes, and guided workflows. Users can upload datasets, manipulate and analyze data, and train predictive models with minimal effort. DataBuddy aims to empower professionals from non-technical backgrounds to make informed, data-driven decisions with ease, significantly reducing the time and resources needed for comprehensive analysis.

KEYWORDS: No-Code Data Analysis, Machine Learning, Data Visualization, Exploratory Data Analysis (EDA), Predictive Modeling, Python, User-Friendly Interface, Automation, Data Science, Non-Technical Users.

I. INTRODUCTION

A. Problem Statement: In the current data-centric landscape, organizations heavily depend on data analysis and machine learning to derive insights and optimize decision-making. However, traditional data science tools often require a strong command of programming languages, creating a steep learning curve for users without technical expertise. This limitation significantly restricts the ability of professionals in various domains to independently utilize data-driven techniques, thus necessitating the development of tools that can bridge this gap.

B. Motivation and Vision: The motivation behind this project stems from the desire to democratize access to data science by eliminating the barriers posed by complex coding requirements. The vision is to create an intuitive, interactive platform that allows users to seamlessly perform data analysis and build machine learning models using a no-code interface, thereby expanding the reach of advanced analytics to users from diverse professional backgrounds.

C. Objectives: The core objectives of DataBuddy are to develop a fully functional, no-code platform that simplifies and automates the entire data science pipeline. This includes enabling data input and preprocessing without scripting, offering automated exploratory data analysis with visual insights, providing tools for customizable data visualizations, integrating machine learning models for prediction and classification, and generating reports that summarize results for sharing and documentation. The focus remains on accessibility, user guidance, and performance efficiency.

II. RELATED WORK

Criteria	Excel	Power BI	Google Data Studio	AutoML	DataBuddy
Ease of Use	Moderate	High	High	Moderate	High
Data Input Options	Local Files	Local, Cloud	Cloud-based	Local, Cloud	Local, URL
Visualization Support	Basic Charts	Advanced	Advanced	Limited	Customizable
Machine Learning Integration	None	Limited	None	Advanced	User-Driven
Cost	One-time	Subscription	Free	High	Free

Existing no-code platforms each offer a mix of features, but most fail to provide end-to-end functionality combining preprocessing, EDA, visualization, and machine learning. DataBuddy fills this gap by providing a unified workflow for non-technical users, combining the strengths of visualization tools like Power BI and Google Data Studio with the machine learning capabilities of platforms like AutoML, all while maintaining a user-friendly and cost-effective environment.

III. LITERATURE REVIEW

A. Data Handling and Preprocessing: DataBuddy relies heavily on the Pandas library, which has been recognized for its robustness in data wrangling, cleaning, and transformation. As emphasized by McKinney [1], Pandas enables efficient handling of missing values, column transformations, and data merging. Brownlee [7] supports the importance of exploratory data analysis techniques, including statistical summaries and correlation matrices, both of which are integrated into DataBuddy.

B. Machine Learning Implementation: The machine learning backbone of DataBuddy is powered by Scikit-Learn, a widely-used Python library for implementing a range of ML algorithms. According to Pedregosa et al. [2], Scikit-Learn supports linear regression, decision trees, and support vector machines (SVM), all of which are incorporated in the tool. Géron [6] further promotes the concept of reusable ML pipelines, an idea reflected in DataBuddy's GUI-driven model-building process.

C. Data Visualization: Visualization is crucial for interpreting data insights. Matplotlib, introduced by Hunter [3], offers flexibility in plotting, while Seaborn, as described by Waskom [4], simplifies statistical visualizations. Both libraries have been integrated into DataBuddy, allowing users to create visually appealing and informative charts without coding.

D. GUI and Workflow Design: DataBuddy's graphical user interface is built using Tkinter, which is known for its simplicity and effectiveness in developing desktop applications. Python's official documentation [5] provides the framework for Tkinter integration. Additionally, Turtle graphics are used for an engaging welcome screen.

E. Development Environment: Anaconda, as referenced by Anaconda Inc. [9], plays a crucial role in managing dependencies and ensuring compatibility among various libraries, which is essential for a tool like DataBuddy that relies on multiple packages.

IV. PROPOSED METHODOLOGY

A. System Architecture: The architecture of DataBuddy comprises three key layers: the user interface (UI) layer built with Tkinter for managing all user interactions; the processing layer, which utilizes Pandas, NumPy, and Scikit-Learn to perform data operations and machine learning; and the storage layer, which organizes and maintains input datasets, visualizations, trained models, and reports for future reference.

B. Data Input and Preprocessing: Users can upload datasets in CSV or Excel formats or provide URLs for data access. Upon upload, the data is automatically converted into a Pandas DataFrame. The system detects and handles missing values, supports column renaming and type conversion through the interface, and allows users to preview their data before proceeding.

C. Data Analysis and Visualization: DataBuddy automates EDA using Pandas' describe() and correlation analysis tools. Users can select visualization types from dropdowns and assign columns using checkboxes. Charts are generated instantly and saved in an organized folder structure. A summary chart combining multiple visualizations offers a complete overview of the dataset.

D. Machine Learning Model Integration: DataBuddy enables users to choose from several machine learning models through a dropdown interface. Parameters like train-test split, learning rate, and iteration count can be modified using GUI components. After training, performance metrics such as accuracy, precision, and recall are displayed, along with a visual history of training progression. All models and reports are stored automatically.

E. No-Code Implementation: The platform's no-code philosophy is implemented through intuitive visual elements and automated processes. Users are guided at every step with prompts and labels, and all backend operations like missing value imputation, data visualization, and model training are executed without requiring programming knowledge.

Stage	Traditional (Jupyter Notebook)	DataBuddy
Data Loading	Requires coding	One-click file upload or URL input
Preprocessing	Manual scripting	Automated handling with GUI options
Visualization	Requires custom code	Select from dropdowns and checkboxes
Model Training	Manual algorithm setup	Click-and-train interface
Result Interpretation	Manual graphing	Auto-generated metrics and visualizations

V. SYSTEM IMPLEMENTATION

A. Tools and Technologies Used: DataBuddy is developed using Python, with libraries including Tkinter for GUI, Pandas for data manipulation, Matplotlib and Seaborn for visualizations, and Scikit-Learn for machine learning algorithms. The OS module ensures structured storage of files, while Turtle enhances the user interface.

B. User Interface Design: The application features a welcoming screen, dataset input options (upload or URL), a preview window for verifying data, and control panels for data operations and model training. Users can select visualizations, configure ML parameters, and view results—all within the application.

C. Folder Structure and Data Management: DataBuddy maintains a consistent directory structure with separate folders for Input Data, Analysis Results, Extracted Models, and Reports. These folders are created automatically during the first execution, ensuring all outputs are organized and accessible.

D. Data Processing Workflow: After dataset acquisition, preprocessing includes handling missing values and performing basic transformations. EDA is conducted using descriptive statistics and correlation analysis, followed by visualization generation. ML models are trained, evaluated, and stored with complete documentation. Reports summarizing each step are generated automatically.

E. Error Handling and User Guidance: The tool detects invalid files, prompts users with helpful messages, and includes fallback mechanisms for data issues. Descriptive labels and real-time prompts ensure a smooth user experience at each stage of the workflow.

VI. RESULTS AND DISCUSSION

A. Exploratory Data Analysis (EDA) Results: DataBuddy provides comprehensive statistics such as mean, median, standard deviation, and range using the describe() method. Missing values are detected using isnull(), and correlation matrices help identify feature relationships. All results are stored as text reports.

B. Visualization Insights: Users can generate a variety of plots including histograms, scatter plots, box plots, and heatmaps. Visualizations are customizable through dropdowns and checkboxes. Summary charts provide an overarching view of dataset trends and are saved for future reference.

C. Machine Learning Model Performance: Users can select models like linear regression, decision trees, or SVM, and the tool automatically handles data splitting and training. Key metrics including accuracy, precision, recall, and F1-score are displayed. Training history is visualized, and all models and associated reports are stored in the Extracted Models folder.

D. Comparative Analysis with Existing Tools: DataBuddy eliminates the need for coding, streamlines the entire data workflow into a few clicks, and consolidates features found in multiple tools into a single platform. Compared to tools like Excel, Power BI, and AutoML, DataBuddy offers superior ease of use, customization, and cost-efficiency.

E. User Feedback and Applications: Test users praised the intuitive interface and clear guidance provided at every step. Practical applications include small business analytics, academic research projects, and classroom use by students unfamiliar with coding.

Criteria	Excel	Power BI	Google Data Studio	AutoML	DataBuddy
Ease of Use	Moderate	High	High	Moderate	High
Data Input Options	Local	Local/Cloud	Cloud-based	Local/Cloud	Local/URL
Visualization Support	Basic Charts	Advanced	Advanced	Limited	Customizable
Machine Learning Integration	None	Limited	None	Advanced	User-Driven
Cost One-time Subscription	Free High	Free			

Model	Training Time	Accuracy	Interpretability	Use Case
Linear Regression	Fast	Moderate	High	Continuous Prediction
Decision Trees	Moderate	High	Moderate	Classification
SVM	Slow	High	Low	Complex Classification
DataBuddy Models	Customizable	High	Moderate	User-Defined Tasks

VII. CONCLUSION

DataBuddy successfully addresses a critical need for accessible data science tools by enabling users without technical backgrounds to conduct in-depth data analysis and machine learning. The tool integrates preprocessing, EDA, visualization, and model training into a cohesive platform that requires no programming knowledge. Users benefit from automation, efficiency, and guided interactions that make complex tasks manageable.

Limitations: The current version supports a limited set of machine learning models and does not accommodate real-time data processing. Performance can be optimized further for larger datasets.

Future Scope: Future development will include the addition of ensemble methods (e.g., Random Forest, XGBoost), real-time data stream support, cloud deployment for collaboration, advanced dashboards using interactive libraries like Plotly, and automatic report generation with AI-powered text summarization.

REFERENCES

[1] W. McKinney, *Python for Data Analysis*, O'Reilly, 2017.
[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *JMLR*, 2011.
[3] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Sci & Engg*, 2007.
[4] M. Waskom, "Seaborn: Statistical Data Visualization," *JOSS*, 2021.
[5] K. Oku, R. K. Vaddy, A. Yada, and R. K. Batchu, "Data Engineering Excellence: A Catalyst for Advanced Data

Analytics in Modern Organizations," International Journal of Creative Research in Computer Technology and Design, vol. 6, no. 6, pp. 1–10, 2024

[6] Python Software Foundation, Tkinter Documentation.

[7] A. Géron, *Hands-On ML with Scikit-Learn*, O'Reilly, 2019.

[8] J. Brownlee, "Exploratory Data Analysis," *Machine Learning Mastery*, 2019.

[9] Turtle Graphics, Python Docs.

INTERNATIONAL JOURNAL OF
MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com