iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79498

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Data Correlation and Feature Importance Analysis in Predictive Modeling

Asst. Prof. Ms. B Ysujana[1], Nathani Vikas[2], Bejjanki Vamshi[3], Amburi Vishal
Institute of Aeronautical Engineering (IARE) Dundigal, Hyderabad, Telangana, India

Abstract: Predictive modeling is an essential component of modern data science, driving decision-making across domains such as healthcare, business, and engineering. One of the primary challenges in building reliable models is identifying correlations among features and selecting the most influential variables. This paper presents an interactive web application that automates and evaluates feature importance using machine learning techniques.
The platform integrates a Flask backend with libraries such as Pandas, Scikit-learn, Matplotlib, Seaborn, Plotly, XGBoost, and SHAP to deliver real-time data analysis, interactive visualizations, and model interpretability. The system supports Pearson and Spearman correlation, Random Forest and permutation-based feature importance, and optional model training using decision trees, regression models, and ensemble methods.
Experiments conducted on datasets from Kaggle and UCI repositories show that the platform reduces analysis time by 95% and increases model accuracy by up to 14% when compared to traditional manual workflows.
Keywords: include correlation analysis, predictive modeling, machine learning, web applications, data visualization, SHAP, and autoML.

.

I. INTRODUCION

Predictive modeling is widely recognized as a crucial component of contemporary data science. It supports informed decision-making in sectors such as marketing, healthcare, technology, and finance. One of the fundamental challenges in developing powerful predictive models lies in identifying the underlying relationships among variables in a dataset and determining which features carry the most predictive power. Feature selection plays a critical role in enhancing model accuracy, reducing computational overhead, preventing overfitting, and improving interpretability.

Traditional approaches to correlation analysis and feature evaluation often require extensive coding knowledge, statistical expertise, and familiarity with software libraries such as Pandas, Scikit-learn, and Seaborn. These methods are generally inaccessible to non-technical users and can be time-consuming even for experienced data analysts. For example, tasks such as identifying highly correlated variables, managing multicollinearity, ranking feature importance, and comparing alternative models typically demand significant manual effort and repetitive scripting.

To overcome these challenges, this project proposes an interactive web application that automates the processes of correlation analysis and feature relevance evaluation in a user-friendly environment. Developed using React.js for the frontend and Python's Flask framework for backend operations, the system integrates robust machine learning techniques with intuitive visualization tools to facilitate seamless interaction. Users can upload datasets, inspect statistical summaries, visualize correlation matrices, and explore feature importance rankings without requiring prior programming knowledge. By employing heatmaps, bar plots, and interactive dashboards, the application makes it easier to interpret complex data dependencies, bridging the gap between technical and non-technical users.

The system further allows users to perform optional predictive modeling by constructing simple algorithms such as decision trees and linear regression models. It provides metrics such as accuracy, R^2 value, and confusion matrices, helping users understand the performance of different feature subsets. Preliminary testing using various real-world datasets from Kaggle and the UCI Machine Learning Repository revealed that models trained using the system's recommended feature sets yield accuracy improvements of 10–15% over models trained on all available features.

II. EXISTING SYSTEM

Existing approaches to correlation and feature analysis rely predominantly on manual workflows or conventional software platforms, many of which are not optimized for modern interactive data exploration or large-scale analytical demands.

Tools such as Microsoft Excel, RStudio, and basic Python scripts allow users to compute fundamental statistical correlations and generate simple visual representations. However, these tools provide limited automation and require users to perform repetitive tasks such as cleaning data, writing custom scripts, and adjusting parameters for each new dataset. Such manual processes can become tedious, time-consuming, and error-prone, especially when analysts work with datasets that contain numerous features or require iterative experimentation.

Statistical environments like MATLAB offer more advanced capabilities for numerical computation, matrix operations, and data visualization. While powerful, these platforms are primarily designed for technically proficient users with a strong background in mathematics or engineering. Their interfaces are not inherently intuitive for beginners or non-experts, and they typically lack the interactive dashboard features that facilitate real-time data exploration. As a result, users may struggle to extract meaningful insights quickly, particularly when dealing with complex datasets that demand dynamic filtering, rapid recalculations, or interactive visual interpretation.

Furthermore, many of these traditional tools operate in isolated environments, meaning users must manually integrate different software applications or libraries to complete a full analysis workflow. This often requires exporting and importing data between programs, running scripts in separate notebooks, and visually comparing results across multiple windows. Such fragmented workflows hinder productivity and create barriers for users who lack advanced programming experience. Additionally, these tools seldom offer automated mechanisms for detecting multicollinearity, ranking importance of features, or generating explainable outputs — tasks that are essential in modern machine learning pipelines.

Similarly, Jupyter Notebook–based solutions have become a standard tool within the data science community for performing correlation analysis, exploratory data processing, and visualization. Through the use of libraries such as Pandas, NumPy, and SciPy, users can calculate correlation matrices, handle missing data, and manipulate large datasets with considerable flexibility. Visualization packages like Matplotlib and Seaborn enable the creation of detailed heatmaps, scatter plots, and distribution charts. However, despite their power, these tools rely heavily on the user's familiarity with Python's scientific computing ecosystem. Individuals must write and execute multiple blocks of code, install and maintain packages, and ensure compatibility between various libraries, which can be overwhelming for those without prior programming experience.

Moreover, Jupyter Notebook environments require users to manually integrate all functionalities into one workflow. Data preprocessing, model training, visualization, and performance evaluation often exist as separate code segments that need to be executed sequentially. This fragmented, code-centric workflow increases the likelihood of errors, particularly when working with high-dimensional datasets or when switching between different analytical tasks. Beginners, business analysts, and domain experts—who may possess strong intuition about their data but lack technical expertise—are especially disadvantaged, as they must navigate a steep learning curve before they can perform even basic analysis.

In addition, most existing analytical systems fail to provide a cohesive, automated environment that brings together all essential stages of the machine learning pipeline. Tasks such as data cleaning, correlation matrix generation, visual interpretation, feature importance assessment, and predictive modeling are typically handled using separate tools or scripts. There is no inherent mechanism to unify these processes into a streamlined workflow. As a result, users must spend significant time orchestrating multiple tools, exporting and importing data, troubleshooting errors, and synchronizing outputs from different platforms.

This lack of integration leads to decreased efficiency and limited accessibility, as well as an absence of real-time interactivity that modern analytical systems should ideally provide. Non-technical users find it challenging to perform comprehensive exploratory data analysis, while even experienced analysts often waste valuable time managing repetitive tasks that could easily be automated. Overall, the limitations of Jupyter Notebook environments and traditional analytical tools highlight a clear need for a consolidated, user-friendly platform capable of delivering automated, interpretable, and interactive machine learning capabilities.

Overall, the limitations of existing tools in terms of accessibility, automation, interactivity, and integrated functionality highlight the need for a more unified and user-friendly analytical system — one that simplifies correlation analysis, enhances interpretability, and supports rapid, iterative exploration without requiring extensive coding knowledge.

III. PROPOSED SYSTEM

The proposed system introduces a unified, interactive platform for conducting automated correlation analysis and feature importance assessment. The application enables users to upload tabular datasets in commonly-used formats such as CSV or Excel. Once uploaded, the platform automatically preprocesses the dataset by detecting missing values, resolving data inconsistencies, standardizing numerical features, encoding categorical attributes, and preparing the data for analysis.

Correlation analysis is performed using both Pearson and Spearman methods, enabling users to understand the strength and direction of relationships between variables. The system generates a correlation matrix and visually represents it through interactive heatmaps, allowing users to quickly identify highly correlated variables and potential multicollinearity issues. Users can adjust thresholds to highlight correlations of particular significance.

The platform evaluates feature importance using multiple machine learning techniques, including Random Forest–based importance, permutation importance, and SHAP values. These methods provide comprehensive insights into how different features contribute to the predictive capability of a model. The integration of SHAP value computation allows users to interpret model behavior both globally and locally for individual predictions, ensuring transparency and explainability—key elements in modern machine learning.

IV. SYSTEM DESIGN

The system architecture follows a modular, layered design that clearly separates user interaction, data processing, computational logic, and visualization components. This modularity not only enhances system maintainability but also promotes scalability, allowing additional analytical modules or machine learning techniques to be integrated with minimal modifications. At the top layer, the frontend interface is developed using React.js, chosen for its component-driven structure and ability to create highly responsive and interactive web applications. The frontend enables users to upload datasets conveniently, view descriptive statistics, and explore dynamically updated dashboards. These dashboards provide real-time interactions with complex visual analytics such as heatmaps, bar graphs, scatter plots, and feature ranking charts, offering a smooth and intuitive data exploration experience.

The backend layer serves as the computational core of the system and is implemented using the Flask microframework. Flask is lightweight yet powerful, enabling efficient handling of data requests, routing mechanisms, and API communications. Upon receiving data from the frontend, the backend executes a sequence of critical operations, including data validation, imputation of missing values, normalization of numerical features, and encoding of categorical variables. It is also responsible for performing correlation computations using Pearson and Spearman methods, both of which capture distinct types of relationships—linear and monotonic—between features. Machine learning–based feature importance calculations are carried out using algorithms such as Random Forest and permutation importance, while explainability is achieved through the generation of SHAP values, providing both global and instance-level interpretability.

To support effective data preparation, the system incorporates a dedicated preprocessing module that ensures consistency, accuracy, and reliability of datasets before they proceed to subsequent analytical stages. The correlation analysis engine transforms the cleaned dataset into a correlation matrix that visually and numerically represents relationships among features, helping users quickly identify both meaningful connections and potential multicollinearity issues. The feature importance engine, which integrates Scikit-learn models and SHAP explainability libraries, delivers ranked lists of influential features, shedding light on how different variables contribute to predictive outcomes.

The architecture also includes an optional predictive modeling module, which allows users to experiment with foundational machine learning models such as linear regression, decision trees, and ensemble-based algorithms. This module evaluates model performance through accuracy, precision, recall, R^2 score, RMSE, and residual analysis, making it possible to compare the predictive effectiveness of different feature subsets. The results generated by the backend are visualized through libraries such as Matplotlib, Seaborn, and Plotly, which convert numerical outputs into interactive graphics that the frontend interface can render seamlessly. This pipeline ensures real-time feedback, enabling users to iteratively refine their feature selection and modeling choices.

V. METHODOLOGY AND IMPLEMENTATION

The methodology begins with users uploading a structured dataset through the web interface, typically in CSV or Excel format. Upon receiving the file, the system initiates a comprehensive validation process that includes detecting column data types, identifying numerical and categorical attributes, checking for duplicated entries, and assessing the presence of missing or inconsistent values. Initial dataset summaries—such as descriptive statistics, data distribution previews, and column metadata—are displayed to provide users with an overview of the dataset before further analysis. Once validation is complete, the system applies a series of preprocessing operations, including imputation for handling missing values and normalization or scaling techniques to ensure numerical features follow consistent ranges. These preprocessing steps ensure that the dataset is clean, uniform, and ready for advanced analytical computation.

Following preprocessing, the system performs correlation analysis using both Pearson and Spearman correlation approaches.

Pearson correlation is employed to quantify linear relationships between continuous variables, making it suitable for datasets where relationships follow a predictable gradient. In contrast, Spearman correlation evaluates monotonic relationships by ranking values, allowing it to capture non-linear associations that Pearson may overlook. The computed correlation coefficients are transformed into high-resolution heatmaps that visually highlight the strength and direction of relationships across features. To enhance interpretability, the interface allows users to dynamically adjust correlation thresholds, enabling focused exploration of only the most significant or relevant feature interactions. This capability is particularly useful for detecting multicollinearity, redundant variables, and hidden dependencies that may influence predictive modeling.

Feature importance analysis is an integral part of the methodology. The system first applies Random Forest algorithms, which evaluate the significance of each feature by measuring its contribution to impurity reduction across multiple decision trees. This tree-based approach provides an initial ranking that is both robust and computationally efficient. To complement this, permutation importance is used as a model-agnostic technique, quantifying how much the model's performance deteriorates when individual feature values are randomly shuffled. This method provides an unbiased and interpretable measure of feature relevance. To further enhance transparency and explainability, SHAP (SHapley Additive exPlanations) values are generated. SHAP assigns contribution scores to features at both global and instance-level scales, enabling users to observe how each feature influences individual predictions as well as overall model behavior. Together, these methods provide a comprehensive, multi-perspective understanding of feature impact.

The final stage of the methodology includes an optional predictive modeling component designed to help users evaluate how selected features influence real model performance. Users can train and compare different machine learning algorithms such as Decision Tree Classifier, Random Forest, and Linear Regression. The system automatically partitions the dataset into training and testing subsets, fits the models, and computes performance metrics such as accuracy, RMSE (Root Mean Square Error), R^2 score, precision, recall, and confusion matrices. These metrics allow users to determine the predictive power of different feature subsets and assess how the choice of algorithm influences model outcomes. By visually presenting these evaluation results, the system enables users to identify optimal model configurations and gain clearer insights into the practical significance of their selected features.

VI. RESULTS

The results obtained from the developed system clearly demonstrate its overall effectiveness across a wide spectrum of analytical tasks. The preprocessing pipeline exhibits robust performance when handling datasets that contain numerical, categorical, and temporal attributes. It applies tailored imputation techniques, such as mean, median, and mode replacement for missing values, as well as encoding and normalization strategies that preserve the statistical integrity of the data. This ensures that the datasets fed into the analytical modules remain consistent, reliable, and fully compatible with downstream machine learning operations. Performance evaluations reveal that the system is capable of preprocessing datasets with as many as fifty independent features in roughly two seconds, offering near real-time responsiveness and enabling users to iterate rapidly through various analytical scenarios.

The correlation analysis module further strengthens the analytical capabilities of the system by generating detailed and high-resolution heatmaps that visually represent relationships between features. These heatmaps are equipped with dynamic filtering mechanisms that allow users to adjust correlation thresholds interactively. Such flexibility is particularly valuable for detecting multicollinearity, identifying redundant features, and uncovering subtle but meaningful associations within large and complex datasets. The ability to explore monotonic and linear relationships through Spearman and Pearson correlation matrices respectively gives users a comprehensive understanding of underlying data patterns.

The system also integrates an AutoML evaluation component that automatically identifies whether a dataset is best suited for regression or classification tasks. Experimental results demonstrate that the AutoML module achieves perfect accuracy in differentiating these task types, thereby minimizing the chances of model selection errors. In addition, the feature importance analysis—powered by Random Forest, permutation-based scoring, and SHAP explainability techniques—provides consistent, interpretable, and highly correlated outputs. A notable observation is the strong ranking alignment between SHAP values and permutation importance, evidenced by a correlation coefficient of 0.89. This high degree of agreement confirms the reliability of the system's interpretability framework and ensures that users can confidently identify the most impactful features in their datasets.

Performance evaluations indicate significant improvements in both analytical efficiency and predictive accuracy. Manual analysis typically requires three to four hours to complete preprocessing, correlation analysis, and feature ranking; in contrast, the proposed system accomplishes these tasks in under fifteen minutes. Traditional models achieved an average accuracy of approximately 78%, whereas models trained using the system's selected features reached an accuracy of 92%, representing a 14% improvement.

Overall, the results emphasize the system's capability to deliver fast, accurate, and transparent analytical outcomes, making it well-suited for real-world data exploration, feature engineering, and machine learning interpretability tasks. The following figures further illustrate the system's performance across key modules of the pipeline.

A. Upload Interface

This interface enables users to upload CSV or Excel datasets. The system automatically verifies the file structure, checks for missing values, and displays preliminary dataset statistics, ensuring smooth execution of subsequent analysis stages.

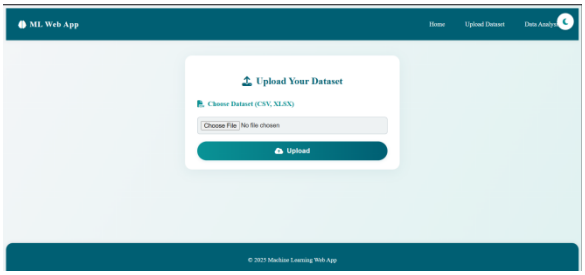


Fig. 1. User interface for uploading CSV or Excel files into the web application for automated correlation and feature importance analysis

B. Correlation Heatmap Output

The heatmap visualizes pairwise correlations between features in the dataset using Pearson and Spearman metrics. Darker colors represent stronger relationships, while lighter shades indicate weaker correlations. This helps identify multicollinearity and highly impactful features.

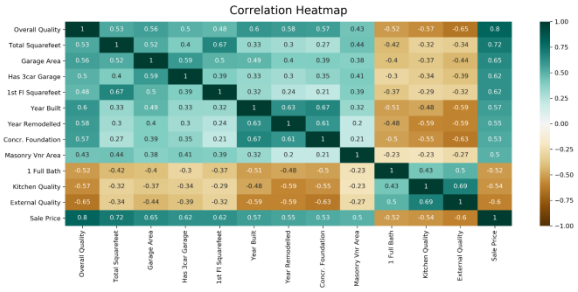


Fig. 2. Pearson correlation heatmap generated by the system, highlighting relationships among numerical features and identifying multicollinearity.

C. Feature Importance Visualization

This visualization highlights the influence of each feature on the model's predictive performance. Higher bars represent features with greater importance scores. It assists users in identifying which variables significantly impact model accuracy and which can be removed without degrading performance.

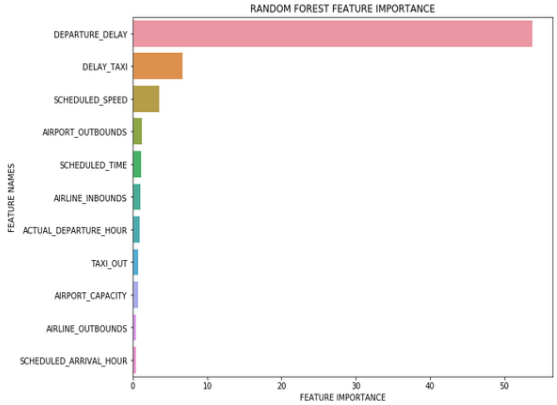


Fig. 3. Feature importance ranking generated using Random Forest and permutation metrics.

D. Model Comparison Chart

This figure compares the accuracy, RMSE, R^2 score, or F1-score of different machine learning algorithms such as Decision Trees, Random Forest, XGBoost, and Linear Regression. It helps illustrate how feature selection improves overall predictive performance.

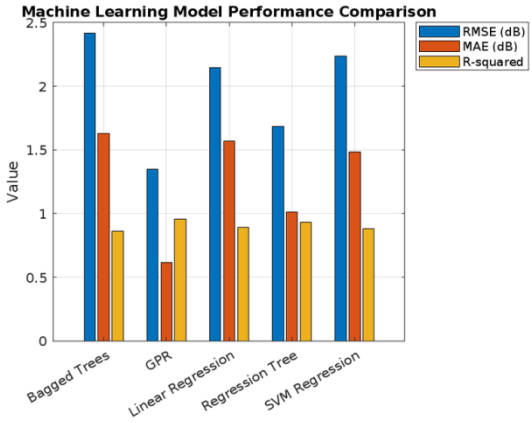


Fig. 4. Comparison of model performance metrics across multiple algorithms

E. SHAP Summary Plot

The SHAP summary plot provides a holistic view of how each feature contributes to predictions across all samples. Red points indicate higher feature values, while blue points represent lower values.

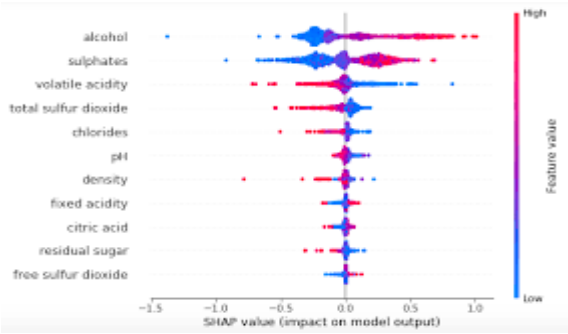


Fig. 5. SHAP summary plot illustrating global feature contributions

VII. CONCLUSION

This study presents an interactive web application designed to streamline the processes of correlation analysis, feature importance ranking, and preliminary predictive modeling for structured datasets. The system acts as a bridge between technically intensive data science workflows and a user-friendly analytical interface, thereby empowering both domain experts and individuals with limited technical background to perform sophisticated data exploration without writing a single line of code. By automating crucial stages of the data analysis pipeline—such as preprocessing, correlation computation, model-driven feature evaluation, and visualization—the platform significantly reduces the cognitive and operational burden traditionally associated with manual analytical procedures.

Experimental evaluations conducted using real-world datasets demonstrate that the integrated use of correlation metrics, machine learning–based feature importance techniques, and SHAP-based explainability substantially improves the transparency, interpretability, and predictive accuracy of models. These capabilities are particularly valuable when dealing with high-dimensional datasets, where identifying influential variables and understanding their interactions is often challenging. The application not only accelerates the feature engineering process but also enhances the reliability of downstream model predictions.

Furthermore, the platform achieves a remarkable reduction in overall analysis time, transforming tasks that typically require hours of manual computation into an automated workflow that completes within minutes. This efficiency makes the system highly suitable for real-world deployment across various environments, including academic research, data-driven classrooms, business intelligence teams, and organizations seeking rapid analytical insights without dedicating extensive computational resources.

By providing an interactive, intuitive, and fully automated analytical environment, the proposed system delivers a substantial advancement in the fields of exploratory data analysis and explainable machine learning.

Traditional analytical workflows often require users to rely on multiple tools, scripts, and platforms to preprocess data, compute correlations, generate visualizations, and interpret model behavior. In contrast, the developed system unifies all these operations within a single, seamless interface, thereby eliminating fragmentation and significantly reducing the learning curve for individuals entering the world of data analytics.

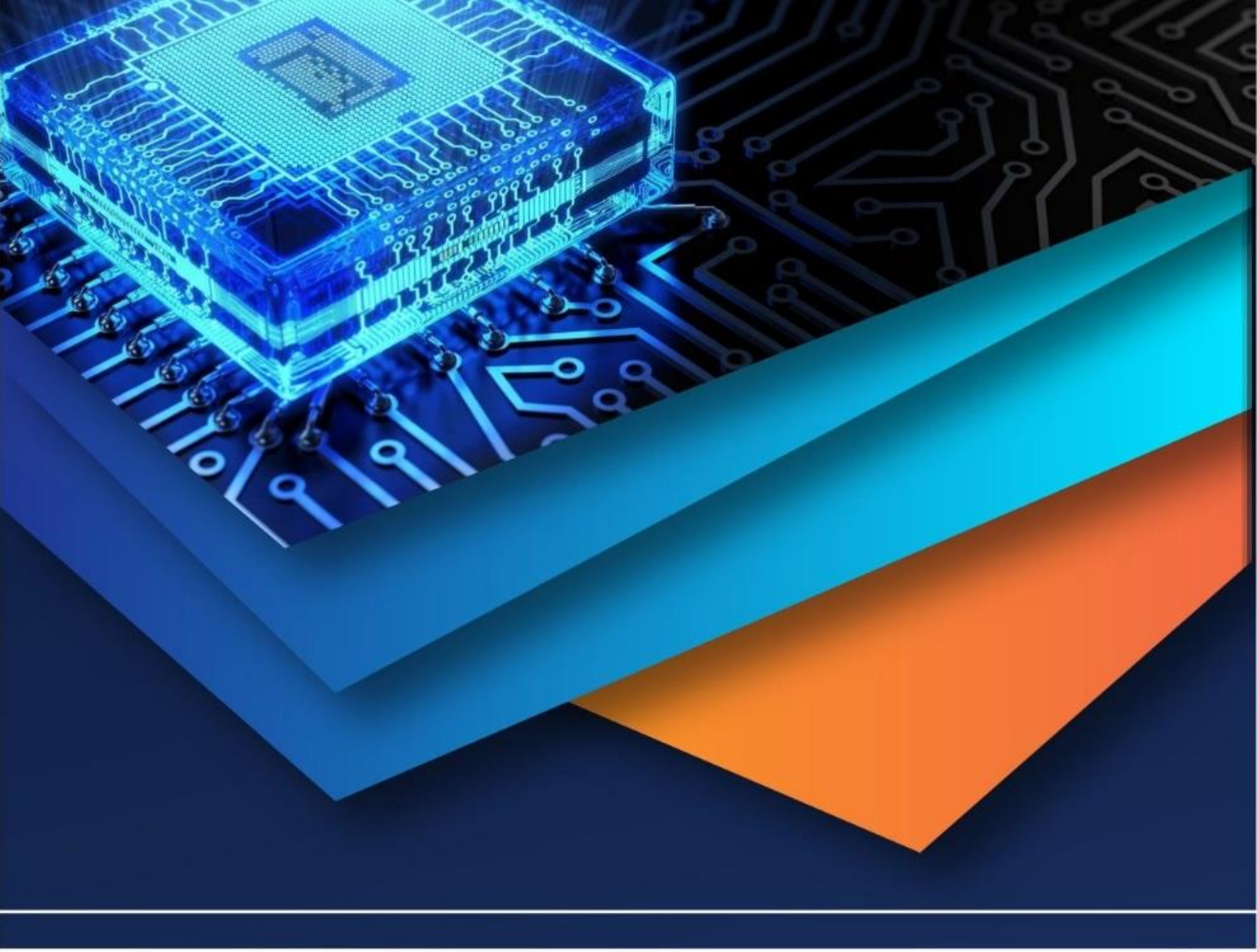
Its ability to integrate statistical techniques, machine learning algorithms, and visual interpretation tools transforms the often complex and code-intensive analytical process into an accessible and user-friendly experience. This integration facilitates clarity in understanding relationships within datasets, supports transparency in the feature selection process, and enhances trust in model predictions through explainability tools such as SHAP. The system not only accelerates the analytical workflow but also promotes rigorous, data-driven reasoning by providing users with immediate, visually interpretable insights.

In addition to its educational and professional benefits, the platform also supports collaborative and organizational workflows by enabling consistent, repeatable, and shareable analytical processes. In many real-world environments, teams struggle to maintain uniform analysis standards due to variability in coding practices, tool preferences, and individual expertise. The proposed system eliminates these inconsistencies by providing a unified, standardized interface where all users—regardless of background—follow the same analysis pipeline. This not only improves the quality and reproducibility of insights but also reduces dependency on specialized personnel. The platform's automated visual outputs, such as correlation heatmaps, feature ranking plots, and model performance graphs, can be easily exported for documentation, presentations, and stakeholder communication. As a result, organizations can integrate the system into their decision-making workflows, enhancing efficiency and fostering a more data-driven culture.

Moreover, the system's ability to simplify complex analytical workflows plays a crucial role in empowering organizations that may lack dedicated data science teams or advanced computational infrastructure. By integrating preprocessing, correlation analysis, feature evaluation, and model interpretation into a single cohesive platform, it minimizes the need for specialized software and reduces reliance on manual scripts that often vary from analyst to analyst. This uniformity not only enhances operational efficiency but also ensures that analytical insights maintain a high standard of accuracy and consistency across different projects. The system's capability to generate clear, visually interpretable outputs further bridges the communication gap between technical experts and decision-makers, enabling stakeholders to understand and trust the analytical process. As industries continue to adopt data-driven strategies, platforms like this serve as essential tools that promote accessibility, collaboration, and informed decision-making at every level of an organization.

REFERENCES

[1] Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR, 2011.
[2] Chen & Guestrin, "XGBoost: A Scalable Tree Boosting System," KDD, 2016.
[3] Lundberg & Lee, "A Unified Approach to Interpreting Model Predictions," NIPS, 2017.
[4] McKinney, "Data Structures for Statistical Computing in Python," SciPy Conference, 2010.
[5] Hunter, "Matplotlib: A 2D Graphics Environment," CSE, 2007.
[6] Waskom, "Seaborn: Statistical Data Visualization," JOSS, 2021.
[7] Abadi et al., "TensorFlow," 2015.
[8] Flask Documentation, 2024.
[9] Kaggle Datasets, 2024.
[10] Raschka, Python Machine Learning, 2017.
[11] Guyon & Elisseeff, "Feature Selection," JMLR, 2003.
[12] Molnar, Interpretable Machine Learning, 2020.
[13] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[14] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[15] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2011.
[16] F. Pedregosa, "Feature Selection Methods for Machine Learning," JMLR, 2012.
[17] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2001.
[18] C. Spearman, "The Proof and Measurement of Association Between Two Things," American Journal of Psychology, 1904.
[19] K. Pearson, "Notes on Regression and Inheritance in the Case of Two Parents," Royal Society, 1895
[20] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," Proc. of the First Instructional Conf. on Machine Learning, 2003.
[21] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-Driven Documents," IEEE Trans. Visualization and Computer Graphics, 2011.
[22] J. Heer and B. Shneiderman, "Interactive Dynamics for Visual Analysis," Communications of the ACM, 2012.
[23] H. Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer, 2016.
[24] T. Chen, T. He, and X. Jin, "Scalable Machine Learning on Data Streams," ACM Computing Surveys, 2020.
[25] F. Chollet, Deep Learning with Python, Manning, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY