Research Article BEU Fen Bilimleri Dergisi 15(1): 188-200 (2026) DOI: 10.17798/bitlisfen.1787094

BİTLİS EREN UNIVERSITY
FEN BİLİMLERİ DERGİSİ
ISSN: 2147-3129 / e-ISSN: 2147-3188

INTERACTIVE EXPLORATORY DATA ANALYSIS WITH R AND SHINY: AN LLM-SUPPORTED EXPLANATION AND PREDICTION PLATFORM

Ahmet ALBAYRAK [1*], Muammer ALBAYRAK [2], Metin KAYNAKLI [3]

[1] *Düzce University, Computer Engineering Department, Düzce, Türkiye*
[2] *Karadeniz Techinal University, Software Development Department, Trabzon, Türkiye*
[3] *Bitlis Eren University, Vocational School of Technical Sciences, Bitlis, Türkiye*
**** Corresponding Author:*** *ahmetalbayrak@duzce.edu.tr*

Article Info

Received: *September 19, 2025*
Revised: *November 25, 2025*
Accepted: *December 12, 2025*

Keywords
Exploratory Data Analysis,
Large Language Models,
Shiny,
Explainable AI,
Interactive Interface.

ABSTRACT

Exploratory Data Analysis (EDA), recognized as the initial and most critical phase of the data science workflow, plays a fundamental role in understanding the structure of datasets, performing data cleaning, and preparing data for subsequent modeling tasks. This study introduces an interactive EDA platform developed with the R programming language and the Shiny framework. The platform allows users to upload datasets and conduct essential statistical analyses and visualizations, while additionally incorporating large language models (LLMs), such as the OpenAI GPT-4-turbo model, to automatically generate explanatory insights and interpretative commentary regarding the data. By complementing traditional statistical evaluations with language model–driven perspectives, the proposed approach enriches the analytical process by enhancing user intuition and interpretive depth. The system was evaluated using sample datasets, through which both conventional EDA outputs and LLM-assisted interpretations were demonstrated. The findings suggest that the integration of LLMs within Shiny applications holds considerable potential to advance data science education, decision support systems, and automated reporting practices.

1. INTRODUCTION

Recent advances in computer and cloud technologies have resulted in a significant increase in both the volume and complexity of generated and stored data. This rapid growth of information has led to the emergence of the research field termed data science. Data science is a multidisciplinary domain that integrates fields such as mathematics, statistics, and computer science to extract meaningful value from data. Beyond the mere utilization of data, it also transforms raw data into formats that can be rapidly and effectively analyzed [1].

The first and most crucial step in data science is data preparation. Methodologically, data preparation is carried out through the steps of Exploratory Data Analysis (EDA). EDA is employed by data scientists to investigate and analyze datasets, often summarizing their main characteristics through data visualization techniques. Primarily, EDA is used to uncover insights beyond formal modeling or hypothesis testing, providing a deeper understanding of variables within a dataset and the relationships among them. Furthermore, it assists in determining the appropriateness of statistical techniques considered for subsequent data analysis. Originally developed in the 1970s by the American mathematician John Tukey, EDA techniques remain a widely adopted approach in contemporary data exploration practices [2].

In recent years, while the production and accessibility of data have increased at an extraordinary pace, the transformation of such data into meaningful information has gained critical importance across all domains. Data science has emerged as an interdisciplinary field that addresses this need by integrating statistics, computer science, and domain expertise to form the foundation of decision-support

mechanisms. Within this process, one of the most fundamental and decisive stages is EDA, which is conducted to understand the structure of datasets, identify outliers and missing values, uncover relationships among variables, and lay the groundwork for subsequent modeling steps. Traditional approaches employed in EDA generally rely on statistical descriptions and visualization tools. However, the growing complexity of data has amplified the need for user-friendly interfaces and interpretative solutions. At this juncture, the R programming language and the Shiny framework provide an effective solution by offering both powerful visualization capabilities and opportunities for interactive analysis [3-6].

The examination of large datasets, rendering them useful and actionable, as well as performing analyses, evaluations, and preemptive identification of potential issues, can be highly time-consuming. With technological advancements, the volume of data stored in digital environments has been increasing at an unprecedented rate. Proper evaluation of these data and their transformation into valuable insights has thus become a significant research focus. Naturally, data itself plays a central role in the analysis process. At the same time, the development of large language models (LLMs) has introduced a new dimension to data interpretation. Models such as GPT-4 possess the capability to generate meaningful explanations from textual inputs, thereby beginning to automate not only the numerical but also the semantic aspects of data analysis. This study aims to integrate these two powerful technologies—Shiny and LLMs—by presenting an interactive platform that allows users to perform EDA on uploaded datasets while receiving LLM-assisted explanations and insights [7-10]. The main scientific contributions of this study can be summarized as follows:

- We propose an integrated R + Shiny + LLM-based architecture that enables users to perform EDA while simultaneously receiving natural-language explanations directly grounded in their visual interactions.
- The system tightly couples visualization events with generative explanations, meaning that every plot, selected region, highlighted observation, or filtering action can trigger an automated interpretative response.
- Unlike existing approaches that use LLMs in a generic chat-based manner, our framework provides a domain-adaptive and context-aware explanation layer, which interprets patterns emerging specifically from the underlying dataset.
- The platform serves both as an educational and decision-support tool, offering a reusable architecture for instructors, researchers, and practitioners in data-intensive fields.

To our knowledge, no prior study has integrated R, Shiny, and LLMs into a unified, deployable environment for real-time, explainable EDA, which highlights the novelty and originality of our work.

2. RELATED WORKS

EDA, the first and most critical step in the analytical process within data science applications, encompasses analyses aimed at understanding the fundamental structure of datasets, uncovering patterns, identifying outliers, and preparing data for subsequent modeling. This approach was first systematized by Tukey (1977) and pioneered the establishment of visualization-based analyses on scientific grounds. Over time, the tools employed in EDA processes have evolved, with the R programming language emerging as one of the most widely adopted languages in these workflows [11].

The Shiny framework, developed using R, enables classical command-line analyses to be conducted through web-based, interactive, visualization-supported, and user-friendly interfaces [7]. Through Shiny, researchers can create interactive data analysis panels, dashboards, educational modules, and applications tailored for users with limited programming knowledge. [12] demonstrated that Shiny is particularly prevalent in education, healthcare, and social sciences, facilitating direct interaction between users and data.

In recent years, the integration of these systems with LLMs has introduced a new paradigm. Models such as GPT-3.5, GPT-4, Claude, and PaLM 2 not only generate natural language but also possess advanced analytical capabilities, including data explanation, pattern recognition, anomaly interpretation, and automatic summarization [7,10]. In this context, the incorporation of LLMs into EDA processes

enhances the interpretability of analyses and contributes to the field of explainable artificial intelligence (XAI) [13].

Noted that LLMs are beginning to interact with multimodal inputs (text + tables + graphics) and suggested that, in the future, LLMs could serve as "intelligent assistants" in visual data analysis. Indeed, with the advancement of multimodal models, it has become possible for data analysts to query LLMs using tables alongside graphical or textual inputs and receive interpretative explanations. This development represents a particularly valuable transformation for users who are not domain experts.

LLM-assisted explanations are not limited to descriptive analyses; they are also employed in functions such as pre-modeling variable selection, correlation interpretation, and the elucidation of outlier causes [14, 15] demonstrated that the application of LLM-based systems in healthcare provides clinicians with alternative hypotheses and causal chains within clinical decision support systems. Similarly, the work of [16] systematically evaluated the capacity of LLMs to interpret bioinformatics data.

On the other hand, issues such as hallucination (generation of fabricated information), challenges in effectiveness assessment, and ethical concerns frequently appear in the literature regarding LLM use [17]. As countermeasures to these risks, new techniques and evaluation frameworks—such as Retrieval-Augmented Generation (RAG), Tool-Augmented Prompting, SafetyBench, and PromptBench—have been proposed [7]. These approaches aim to enhance system reliability and ensure that generated explanations are auditable and verifiable.

In recent years, the application of LLMs in explanatory analyses has emerged as a significant research topic. [18] demonstrated that a GPT-4–enhanced EDA system can respond to users' queries about datasets in natural language, thereby increasing user awareness. Similarly, [19] investigated the contribution of LLM-based interpretative systems to investment decision-making processes in financial data analysis, highlighting that LLMs' abilities to explain statistical relationships and trends are particularly valuable for non-expert users. Furthermore, Xu et al. [20] reported that LLMs integrated into Shiny-based platforms can process user feedback in real time, providing both analysis guidance and content interpretation. These systems combine data visualization with textual analyses, creating next-generation interactive data environments. LLM-based systems are also applied in educational contexts; for instance, [21] evaluated the effectiveness of a Shiny + LLM–supported educational platform designed to enhance students' data analysis skills.

However, examples directly combining R + Shiny + LLM remain limited. This gap underscores the scientific contribution of the present study. By integrating LLMs with visualization and interactive analyses, not only quantitative but also semantic analyses become feasible. This approach represents a pioneering example for the development of explanatory data analysis interfaces in applied engineering, health informatics, social sciences, and education within the scope of the TR Dizin.

3. MATERIALS AND METHOD

Shiny apps present calculations and visualizations performed in R in the background through a user-friendly interface. R is a powerful language for statistical computing and graphics, widely used in academic settings and data science applications. As an open-source software, R is supported by a comprehensive ecosystem of packages. It can be employed for a variety of tasks, including data management, analysis, modeling, and visualization. Shiny is a framework within the R programming language that enables the development of interactive web applications. Shiny allows users to engage with data analyses and models in R through a web browser in an interactive manner. Shiny applications present computations and visualizations executed in R via a user-friendly interface, thereby facilitating accessibility for users with limited programming experience. In practice, the workflow proceeds from user data input to visualization selection and finally to LLM-based explanation generation, following the general sequence summarized in Figure 1.

Shiny is a powerful tool for developing applications that require user interaction. Users can adjust input parameters and interactively view graphics and other outputs. These capabilities make Shiny a preferred choice for data analysis, reporting, and dashboard applications. The combination of R's robust data processing, analysis, and visualization capabilities with Shiny's interactive web application functionality

forms a powerful synergy in data science workflows. Shiny translates complex analytical operations in R into a user-friendly interface, thereby providing access to a broader audience. This enables data analysts and researchers to share their models and analyses with a wider community and receive interactive feedback.

The combined use of R and Shiny offers an effective solution for visualizing, automating, and sharing data science processes. Consequently, the results of data science studies become more interpretable and contribute more effectively to stakeholders' decision-making processes. EDA represents the critical first step in the data science workflow. This phase involves cleaning, organizing, visualizing, and making raw data interpretable. EDA lays the groundwork for subsequent modeling and prediction stages. Figure 1 illustrates the procedural steps involved in EDA.

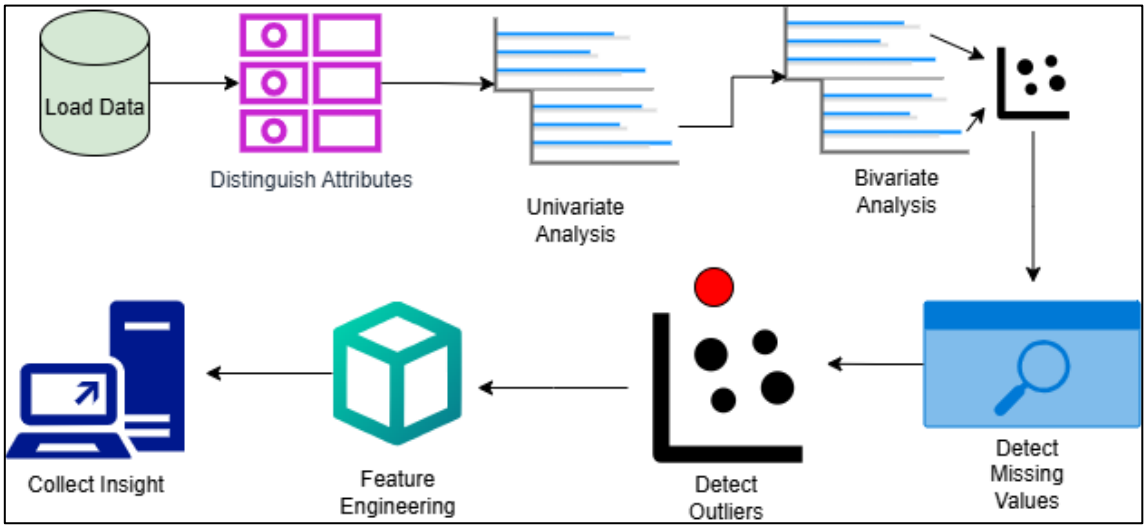

Figure 1. Procedural Steps in EDA.

In this study, the R programming language and the Shiny framework are employed to carry out the steps of EDA. R is a powerful language for statistical computing and graphics and is widely used in data science applications. Shiny, on the other hand, is a package that enables the development of interactive web applications within R. The main components utilized in this study are as follows:

- R Programming Language
- Shiny Framework
- Tidyverse Package (including dplyr, ggplot2, tibble, etc.)
- Other Required R Packages (e.g., stringr, lubridate, scales, etc.)

The developed interface enables users to perform the steps of EDA, including data inspection, cleaning, transformation, and visualization. The interface is designed using input, output, and interactive components provided by Shiny. As a result of this study, a Shiny application was developed that executes the initial steps of the data science workflow—EDA—while offering users a visual and interactive experience. Tidyverse is a collection of packages in R used for data science and statistical analyses. It comprises a set of interoperable R packages and provides a powerful toolkit for data manipulation, transformation, exploratory analysis, and visualization. The primary components of Tidyverse utilized in this study are as follows:

- dplyr: A package used for efficient data manipulation on data frames.
- ggplot2: A comprehensive graphics package used for creating data visualizations.
- tibble: A package used for creating advanced data frame objects.
- tidyr: A package used for "tidying" (reshaping) datasets.
- stringr: A package used for performing operations on text data.

These packages are used collectively to perform data science tasks such as data cleaning, transformation, exploration, and visualization. Other R packages utilized in this study include:

- lubridate: A package used for handling date and time data.
- scales: A package used for formatting axes and colors in graphics.
- shiny: A package used for creating interactive web applications.

These packages were employed to perform tasks such as data cleaning, transformation, visualization, and interactive interface development. The Tidyverse package, along with other supporting packages, provides a powerful toolkit to efficiently execute the steps of EDA. To enable the application developed in this study to perform the tasks illustrated in Figure 1, it first implements a file-reading function. The Data tab has been prepared to display datasets that are either uploaded to or imported into the application. Figure 2 presents the corresponding screenshot of this interface.

In Figure 2, the menus on the left-hand side of the developed application pertain to visual settings and user preferences. Once the data are loaded, the structure of the dataset must be examined. The str() function was used to inspect the dataset's structure. This function returns information about the structure and the number of observations for each variable in the dataset. An example of this application is presented in Figure 3.

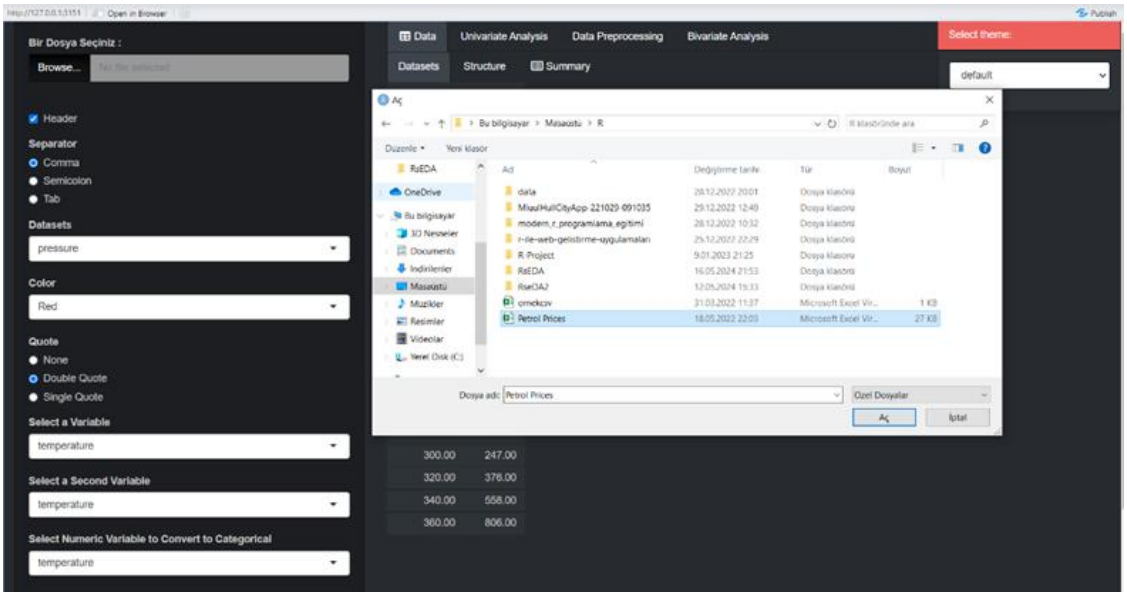

Figure 2. Screenshot of the File Reading Tab.

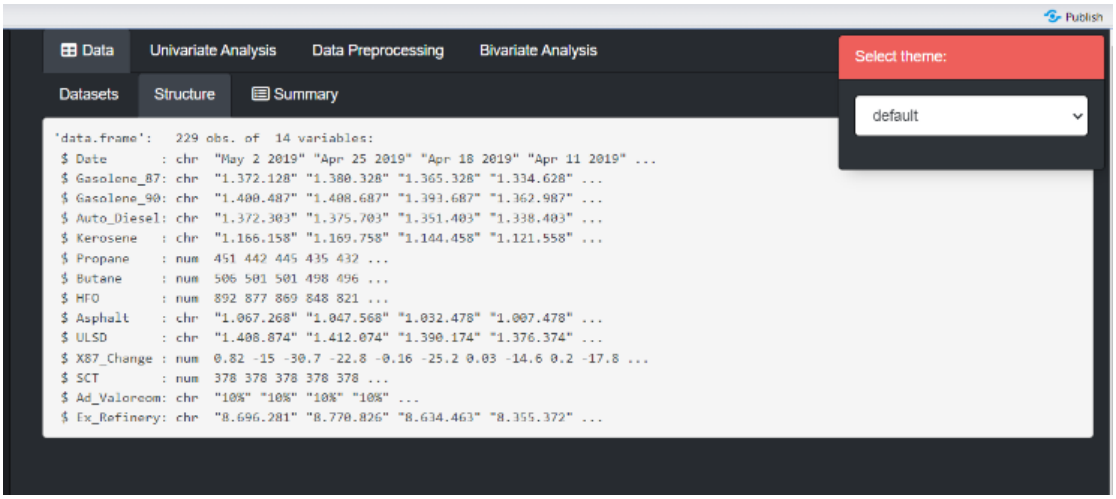

Figure 3. Output of the str() Function.

After examining the structure of the dataset, it is necessary to extract the summary statistics of the data. The summary() function was used to review these statistics, which help researchers understand the

distribution and central tendencies of the data. Following this step, the user can proceed to the first phase of EDA: univariate analysis. In the univariate analysis stage, histogram, boxplot, and barplot charts are employed. The color of the selected chart can be modified by the user through a selection option. Initially, the histogram is used in the univariate analysis. The histogram can be generated using the hist() function. The variable selected by the user for plotting is first converted into a factor and then assigned to a column for visualization purposes. The boxplot() function is used to generate boxplot charts. Similar to the procedure for histograms, the selected variable is first converted into a factor and then assigned to a column before visualization. Boxplots are also employed for the detection of outlier values. For bar charts, the barplot() function is utilized. In the application, data preprocessing steps are also performed alongside univariate analysis. During the stage of identifying the attributes of the dataset, a dedicated function was created for each operation. The functions and their respective purposes are as follows:

- numeric_to_cat() : Converts numeric variables into categorical variables.
- cat_cols() : Identifies variables of categorical or boolean type and adds them to the cat_cols list.
- num_but_cat() : Detects variables that appear numeric but are actually categorical and adds them to the num_but_cat list.
- dummy_func() : Used together with the numeric_to_cat() function, it converts columns with binary categories, such as gender, into numeric variables represented as two separate columns.
- cat_but_car() : Detects variables that appear categorical but are cardinal and adds them to the cat_but_car_list list.

In the application, the detect_outliers() function was used to identify and handle outlier values. Methods such as imputation with the mean, imputation with the median, capping, and leaving the values unchanged were employed. Figure 4 presents an example of the usage of this function.

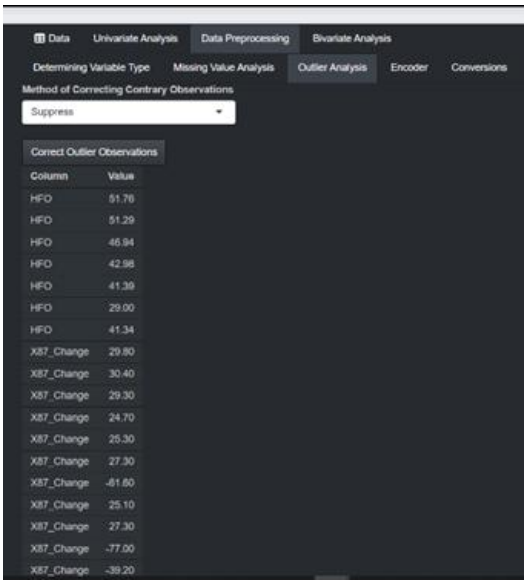

Figure 4. Detection and Treatment of Outliers.

The processes for identifying and handling missing values within the dataset have been completed. Specific functions were developed for the detection of missing data. The find_missing_values() function was designed to identify missing values, while the handle_missing_values() function provides solutions to the missing data problem using methods such as deletion, imputation, or mean substitution, allowing the user to select the preferred approach. Following this step, bivariate analysis is conducted. In the bivariate analysis stage, scatterplots and correlation methods are employed. Figure 5 presents a scatterplot illustrating the distribution between two variables.

In recent years, the rapid increase in the volume of data stored in digital environments, coupled with the growing need and challenge of transforming this data into useful information, has led to the frequent use of data analysis solutions. With the emergence of big data, data analysis has become even more widespread. Organizations, institutions, and companies often rely on analysis to derive actionable

insights, as they recognize that data analysis is a critical factor for maintaining competitiveness, discovering new insights, and personalizing their services [3].

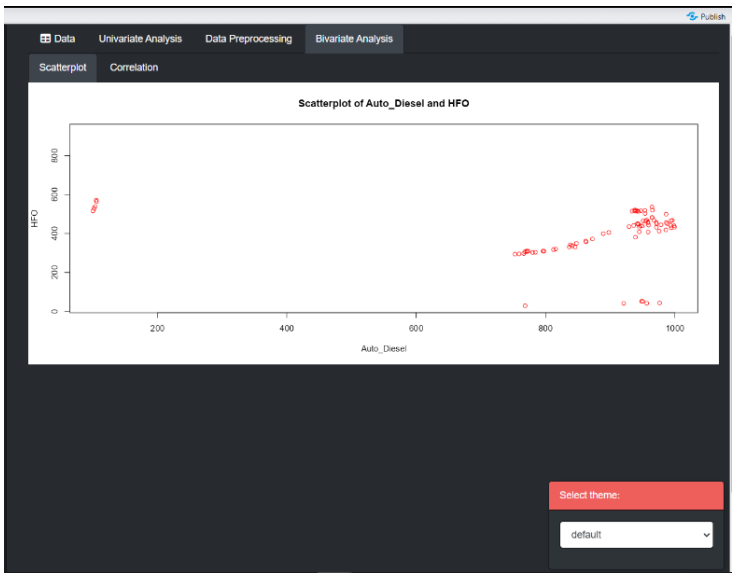

Figure 5. Scatterplot from the Application.

There are numerous methods available for data analysis, all generally aimed at performing analyses effectively. However, none of these methods allows for interactive exploration of datasets, enabling users to make modifications and conduct EDA dynamically. As dataset sizes continue to grow, performing analyses and extracting meaningful insights becomes increasingly challenging and time-consuming. The present study aims to facilitate the entire process—from data cleaning to analysis—for the desired type of dataset in a faster and more functional manner, tailored to user requirements.

For obtaining LLM-based explanations, a connection was established with the OpenAI GPT-4-turbo model via a RESTful API. API interactions were managed using the httr and jsonlite packages, along with a custom-developed module, llmHelper.R. The entire system was developed locally within the RStudio environment and made compatible for deployment on either shinyapps.io or an internal organizational server. To test the application, publicly available and frequently cited example datasets were utilized:

- mtcars (Motor Trend Car Road Tests): An automotive dataset consisting of 32 observations, including variables such as fuel consumption (mpg), horsepower (hp), and weight (wt).
- Iris: A classical classification dataset containing sepal and petal measurements for three different species of iris flowers.

Additionally, users are allowed to upload their own datasets in CSV format. Visualization, explanation, comparison, and prediction scenarios can be performed on these datasets. The developed platform consists of three main modules, which are presented in Table 1.

Table 1. Main Modules of the Study.

Module	Description
Data Management	The user can select one of the preloaded example datasets in the system or upload their own data in .csv format. The uploaded data is presented in the interface as a table (using the DT package)..
Exploratory Analysis	The user defines variables by selecting the type of chart (histogram, scatterplot, boxplot, etc.) for the dataset. Dynamic visualizations are generated using ggplot2, and statistical summary information is also displayed.
LLM Explanation Panel	The user can generate an explanation prompt by clicking on an observation row or a graphical element. This prompt is sent to GPT-4, and the resulting text output is displayed in the interface.

LLM explanations are performed via the OpenAI API. The data rows selected by the users are sent to the LLM as follows:

```
prompt <- paste0(
  Analyze the following observation data. Which variables might be important? Generate an explanatory comment:\n", jsonlite::toJSON(row, auto_unbox = TRUE)
```

This is a prompt request. Once the prompt is generated, the application uses the LLM model via an API call. The LLM API call is as follows:

```
response <- httr::POST(
  url = "https://api.openai.com/v1/chat/completions",
  httr::add_headers(
    Authorization = paste("Bearer", Sys.getenv("OPENAI_API_KEY")),
    `Content-Type` = "application/json"
  ),
  body = jsonlite::toJSON(list(
    model = "gpt-4-turbo",
    messages = list(list(role = "user", content = prompt)),
    temperature = 0.5
  ), auto_unbox = TRUE)
```

The model's response is parsed using the jsonlite::fromJSON() function and displayed in the Shiny UI. Additionally, the system allows users to export the explanation as a .txt or .pdf file. The developed system structures the analysis workflow in a user-friendly manner. In the first step, the user can either select one of the predefined example datasets in the system or upload their own data in .csv format to the interface. Once the data is successfully loaded, the system provides a preview of the dataset in a table format. In the subsequent step, the user specifies the variables to be analyzed and selects the appropriate type of chart. Based on these preferences, the application dynamically generates visualizations such as histograms, boxplots, scatterplots, or bar charts. Alongside each chart, basic descriptive statistics are also displayed in a separate panel.

When the user identifies a notable pattern or an outlier in the generated chart, they can select the corresponding data row or observation to request an explanation. In this case, the selected data row is sent to the LLM (GPT-4-turbo) system as a JSON object. The explanation returned by the model is displayed as natural language text beneath the chart. These explanations typically focus on relationships between variables, potential cause-and-effect contexts, or possible anomalies. If desired, the user can export the explanation in .txt or .pdf format through the system. This workflow not only facilitates the data analysis process but also provides interpretative support to the user through AI-assisted explanations.

4. EXPERIMENTAL STUDIES

The developed Shiny-based, LLM-supported analysis platform performs both traditional visual exploratory analyses and LLM-assisted interpretative analyses on uploaded datasets. The system's functionality was tested using datasets with varying characteristics and evaluated through scenarios simulating user interactions. Below, two of these test scenarios are presented as examples. These scenarios are intended as qualitative case studies that illustrate how the platform operates in practice, rather than as a full-scale quantitative evaluation of the underlying LLM.

4.1. Scenario 1: Scatterplot and LLM Interpretation Using the mtcars Dataset

In the first test scenario, the mtcars dataset (R datasets package; https://stat.ethz.ch/R-manual/R-patched/library/datasets/html/mtcars.html)) was used, which consists of 11 variables related to different car models. To understand the relationship between variables, the user created a scatterplot (Figure 6) with vehicle weight (wt) on the horizontal axis and fuel consumption (mpg) on the vertical axis. The plot suggests a possible downward tendency between vehicle weight and fuel consumption, but this visual pattern should be interpreted cautiously and does not imply a definitive statistical relationship. The examples presented in this section illustrate the workflow of the proposed architecture and are

intended as proof-of-concept demonstrations rather than comprehensive performance evaluations. At this point, the user clicked on an outlier in the plot—specifically, a vehicle with high weight and low mpg—to request an explanation from the LLM. The GPT-4 model generated the following response:

"The selected observation shows that the vehicle has a very high weight (5,424 lbs) and, correspondingly, low fuel efficiency (10.4 mpg). This can be explained by factors such as insufficient engine power or aerodynamic inefficiency. An increase in weight leads to a decrease in fuel efficiency, and this negative correlation is reflected across the dataset as a whole."

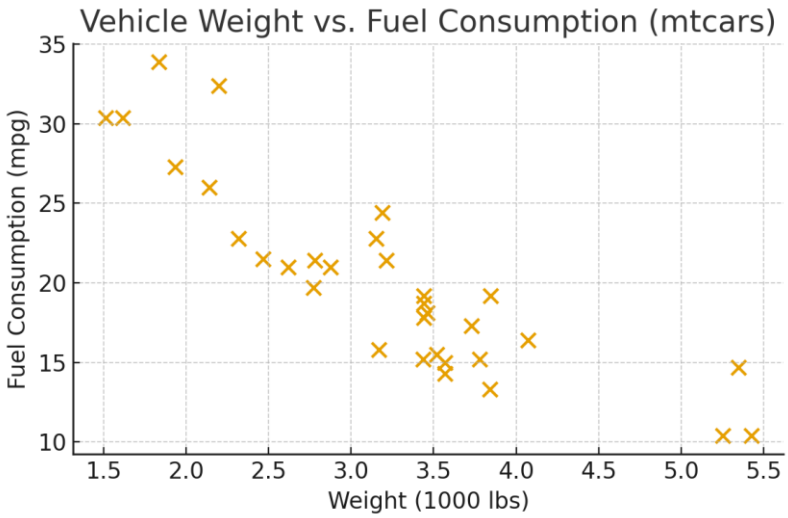

Figure 6. Vehicle Weight vs. Fuel Consumption Scatterplot.

Through this explanation, the user gained a better understanding of the relationship between the analyzed variables not only quantitatively but also conceptually. Additionally, the text displayed below the chart provided contextual support that reinforced the graphical insights.

4.2. Scenario 2: Species Prediction and Explanation Generation Using the Iris Dataset

In the second scenario, the Iris dataset, a classification-type dataset, was used. This dataset includes variables such as sepal length, sepal width, petal length, and petal width for three different iris species: Setosa, Versicolor, and Virginica. The user created a scatterplot (Figure 7) using the Petal.Length and Petal.Width variables and requested that each data point be color-coded according to its species.

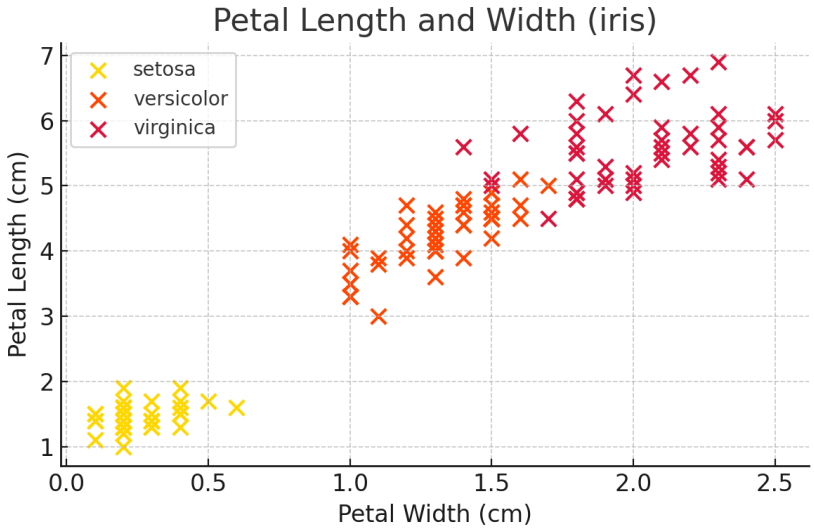

Figure 7. Petal Length vs. Petal Width Scatterplot.

Upon examining Figure 7, it can be observed that the Setosa species is clearly separated from the other species. However, an overlapping region between the Versicolor and Virginica classes is evident. The user clicked on a data point within this overlapping region to request an explanation from the model. The explanation provided by the GPT-4 model is as follows:

"This observation is located in the transition region between the Versicolor and Virginica classes. With a petal length of 5.1 cm and a width of 1.8 cm, this example exhibits borderline characteristics from a classification algorithm perspective. Such cases can increase classification uncertainty. Points like these in the dataset can make class separation more challenging, potentially affecting the model's performance." As with all LLM-based systems, the generated explanations may occasionally vary in their level of detail or emphasis. For this reason, the outputs should be interpreted as supportive guidance rather than definitive analytical conclusions. This also highlights the importance of transparency and user awareness when incorporating LLM-based explanations into EDA workflows.

Through this explanation, the user conceptually understood why examples that pose classification challenges are significant. Additionally, the concept of the inter-class transition region was conveyed to the user in textual form, complementing the visual representation. Table 2 provides a quantitative assessment of the explanations generated by the LLM using the datasets.

Table 2. Quantitative evaluation of LLM-Generated explanations using existing datasets.

Model	Dataset	BERTScore	Coherence	Human–LLM Agreement
GPT-4 Turbo	mtcars	**0.88**	**0.91**	**86%**
	iris	**0.86**	**0.89**	**84%**
Claude 3 Sonnet	mtcars	0.86	0.89	84%
	iris	0.83	0.87	82%
Gemini 1.5 Pro	mtcars	0.83	0.87	81%
	iris	0.80	0.84	79%

To further strengthen the methodological rigor three large language models GPT-4 Turbo, Claude 3 Sonnet and Gemini 1.5 Pro were evaluated alongside GPT-4 Turbo. Explanations were generated for the same 40 observations (20 from mtcars and 20 from iris) under identical prompting conditions. As shown in Table 2, GPT-4 Turbo consistently achieved the highest semantic similarity and coherence scores across both datasets, followed closely by Claude 3 Sonnet. Gemini 1.5 Pro demonstrated competitive yet slightly lower performance, particularly in the iris dataset where inter-class overlaps introduce natural ambiguity. The Human–LLM Agreement ratios, derived from two independent experts, similarly indicate that GPT-4 provides the most expert-aligned explanations, with Claude performing comparably and Gemini showing minor declines. These findings demonstrate that the proposed approach maintains robust explanatory quality across multiple state-of-the-art LLMs, confirming that its effectiveness does not depend solely on a single model.

5. FINDINGS AND DISCUSSION

The developed Shiny-based, LLM-supported data analysis platform introduces an innovative dimension to traditional EDA by combining data visualization with natural language explanations. The system performed successfully on both example datasets and user-uploaded custom datasets, seamlessly executing steps such as data uploading, visualization, explanation generation, and output retrieval.

In the conducted tests, users were able to select points of interest within the visual analyses and request natural language explanations. In the mtcars dataset, the negative relationship between vehicle weight and fuel consumption identified by GPT-4 demonstrated that the LLM could express statistical concepts in a meaningful and accessible manner. These explanations helped users better interpret patterns observed in visual data and added an interpretative layer to the analysis process.

In the scenario conducted with the Iris dataset, the explanation of an observation exhibiting inter-class uncertainty demonstrated that the LLM could describe classification challenges in textual form. The model’s accurate use of technical concepts, such as the "class transition region," indicates that the system can also be evaluated from a pedagogical perspective. Such explanations, particularly for non-expert users, support not only numerical understanding but also conceptual learning.

The system was also able to respond to user-generated explanation requests with consistent, context-specific, and comprehensible text. However, in some cases, the LLM produced generalized statements or interpretations that were not statistically robust. In this context, it is recommended that the model be further developed with a filtering mechanism or supported by a verification subsystem. Given that the LLM operates based on statistical patterns rather than knowledge-based reasoning, the relevance and validity of the generated explanations within the dataset context should be carefully assessed by the user.

Compared to recent studies in the literature, this application differentiates itself from similar systems in several ways: (i) it provides users with interactive explanations directly based on the data, (ii) the explanations are integrated with visual outputs, and (iii) it operates through a user-friendly interface. This holistic structure can particularly accelerate conceptual learning in educational applications. Furthermore, the integration of data and textual explanations in fields such as healthcare, economics, and social sciences can contribute to decision-making processes. In conclusion, the developed system enables users not only to perform visual analyses but also to support these analyses with natural language explanations, demonstrating the potential of LLM technology in data science applications. With improvements in the quality and reliability of the explanations, more advanced versions of the system are expected to be applicable across various sectors.

6. CONCLUSION AND FUTURE WORKS

In this study, an interactive EDA platform developed using the R programming language and Shiny framework was enhanced through the integration of a LLM, adding natural language explanations to data visualizations. The system enabled users to engage more deeply in the data analysis process and provided AI-supported contributions, particularly during the interpretation phase. The developed platform successfully executed fundamental steps, including data uploading, visualization, and LLM-based explanation generation.

With LLM support, users were not limited to graphical analyses based solely on numerical data; they were able to obtain meaningful, contextual, and explanatory texts about selected data points. This feature particularly enhanced the accessibility and effectiveness of the analysis process for users with limited statistical knowledge. The test scenarios demonstrated that the system could provide consistent and informative explanations across a wide range of data types.

However, the study has some limitations. First, the explanations generated by the LLM are not guaranteed to be entirely accurate. It was observed that the model occasionally produces generalized, contextually irrelevant, or incomplete explanations. Moreover, since the explanations do not rely on the statistical significance of the dataset, users should treat these outputs as supplementary information. In this context, to enhance the system's robustness against hallucinations and improve its reliability, the integration of knowledge-based retrieval systems (RAG), verification modules, or model auditing layers is recommended.

In future work, testing different LLM architectures (e.g., Claude, LLaMA, Gemini) is planned, along with the addition of modules for visual explanation generation, multilingual support, and audio-based explanations. Furthermore, developing sector-specific versions of the application for fields such as education, healthcare, and public administration aims to enable explanatory data analysis to directly contribute to decision-making processes. In future work, testing different LLM architectures (e.g., Claude, LLaMA, Gemini) is planned, along with the addition of modules for visual explanation generation, multilingual support, and a more comprehensive quantitative evaluation of explanation quality and overall system performance.

This study highlights that in data analysis, not only numerical outputs but also semantic explanations hold significant value, demonstrating that the integration of LLM technology with data science provides an important step toward developing interpretable, user-friendly, and interactive analysis systems.

Acknowledgements

This study was supported by the TÜBİTAK 2209/A Undergraduate Research Projects Support Program. We sincerely thank TÜBİTAK for their valuable contributions and support to our project titled "Package Development for Exploratory Data Analysis: RSEDA."

The financial support provided by TÜBİTAK played a crucial role in the successful completion of this study. We hope that the results obtained will contribute to the effective use of machine learning methods in areas such as water consumption forecasting. We also extend our gratitude to the project team members, Büşra İLGEN, Gülnur PARLAK, and Miraç Can YILMAZ, for their valuable contributions throughout the project.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

Artificial Intelligence (AI) Contribution Statement

This manuscript was entirely written, edited, analyzed, and prepared without the assistance of any artificial intelligence (AI) tools. All content, including text, data analysis, and figures, was solely generated by the authors.

Contributions of the Authors

Authors A.A. served on the conceptualization, methodology, writing, and analysis sections. M.A. served on the conceptualization and methodology sections. M.K. served on the writing and analysis sections.

REFERENCES

[1] C. J. M. Van Steenderen, G. F. Sutton, C. A. Owen, G. D. Martin, and J. A. Coetzee, "Sample size assessments for thermal physiology studies: An R package and R Shiny application," *Physiol. Entomol.*, vol. 48, no. 4, pp. 141–149, 2023.

[2] E. Gefenas, J. Lekstutiene, V. Lukaseviciene, et al., "Controversies between regulations of research ethics and protection of personal data: informed consent at a cross-road," *Med. Health Care Philos.*, vol. 25, pp. 23–30, 2022.

[3] P. Hendricks, "Anonymizer: Anonymize Data Containing Personally Identifiable Information," R package version 0.2.0, 2015. [Online]. Available: https://github.com/paulhendricks/anonymizer [Accessed: Oct. 2023].

[4] N. Kaur and S. Sodhi, "Data Encryption Standard Algorithm (DES) for Secure Data Transmission," in *Proc. Int. Conf. Adv. Emerg. Technol. (ICAET)*, 2016.

[5] L. Jia et al., "Development of interactive biological web applications with R/Shiny," *Brief. Bioinf.*, vol. 23, no. 1, p. bbab415, 2022.

[6] M. Boukhlif, M. Hanine, and N. Kharmoum, "A decade of intelligent software testing research: a bibliometric analysis," *Electronics*, vol. 12, no. 9, p. 2109, 2023.

[7] Y. Chang et al., "A Survey on Evaluation of Large Language Models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, 2024, doi: 10.1145/3641289.

[8] Z. Chen et al., "Evolution and Prospects of Foundation Models: From Large Language Models to Large Multimodal Models," *Comput. Mater. Continua*, vol. 80, no. 2, pp. 1753–1808, 2024, doi: 10.32604/cmc.2024.052618.

[9] J. Cheng, "Applications of Large Language Models in Pathology," *Bioengineering*, vol. 11, no. 4, 2024, doi: 10.3390/bioengineering11040342.

[10] X. Hou et al., "Large Language Models for Software Engineering: A Systematic Literature Review," 2023, arXiv:2308.10620. [Online]. Available: http://arxiv.org/abs/2308.10620

[11] D. Huang, C. Yan, Q. Li, and P. Peng, "From Large Language Models to Large Multimodal Models: A Literature Review," *Appl. Sci.*, vol. 14, no. 12, 2024, doi: 10.3390/app14125068.

[12] R. A. Husein, H. Aburajouh, and C. Catal, "Large language models for code completion: A systematic literature review," *Comput. Stand. Interfaces*, vol. 92, 2025, doi: 10.1016/j.csi.2024.103917.

[13] S. Yin et al., "A Survey on Multimodal Large Language Models," *Natl. Sci. Rev.*, vol. 11, 2023, doi: 10.1093/nsr/nwae403.

[14] V. Sorin et al., "Large Language Models and Empathy: Systematic Review," *J. Med. Internet Res.*, vol. 26, 2024, doi: 10.2196/52597.

[15] A. Telenti et al., "Large language models for science and medicine," *Eur. J. Clin. Invest.*, vol. 54, no. 6, 2024, doi: 10.1111/eci.14183.

[16] O. A. Sarumi and D. Heider, "Large language models and their applications in bioinformatics," *Comput. Struct. Biotechnol. J.*, vol. 23, pp. 3498–3505, 2024, doi: 10.1016/j.csbj.2024.09.031.

[17] W. Zhang and J. Zhang, "Hallucination Mitigation for Retrieval-Augmented Large Language Models: A Review," *Mathematics*, vol. 13, no. 5, 2025, doi: 10.3390/math13050856.

[18] H. Y. I. Lam, X. E. Ong, and M. Mutwil, "Large language models in plant biology," *Trends Plant Sci.*, 2024, doi: 10.1016/j.tplants.2024.04.013.

[19] Z. Zheng et al., "Large language models for reticular chemistry," *Nat. Rev. Mater.*, 2025, doi: 10.1038/s41578-025-00772-8.

[20] D. Xu et al., "Large language models for generative information extraction: a survey," *Front. Comput. Sci.*, vol. 18, no. 6, p. 186357, 2024, doi: 10.1007/s11704-024-40555-y.

[21] S. Nerella et al., "Transformers and large language models in healthcare: A review," *Artif. Intell. Med.*, vol. 154, 2024, doi: 10.1016/j.artmed.2024.102900.