Analyza (Data Analysis & Visualization Tool)

A

Project Report

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Name	**Roll No.**
Dhuruv Kumar	R2142221199
Khushi Chauhan	R2142221203
Shruti Srivastava	R2142220658

Under the guidance of

Dr. Deepak Kumar Sharma

School of Computer Science, UPES

Bidholi, Via Prem Nagar, Dehradun, Uttarakhand

May – 2025

CANDIDATE'S DECLARATION

We hereby certify that the project work entitled **"Analyza (Data Analysis & Visualization Tool)"**, submitted in partial fulfillment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** with specialization in **BIG DATA**, to the Data Science Cluster, School of Computer Science, UPES, Dehradun, is an authentic record of our work carried out during the period from **January 2025** to **May 2025**, under the supervision of **Dr. Deepak Sharma, Assistant Professor (SG), Systems Cluster, SoCS**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Dhuruv Kumar
500107769
Khushi Chauhan
500102244
Shruti Srivastava
500105401

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: May 2025

Dr. Deepak Sharma
Asst. Professor (SG)
System Cluster, SoCS
Project Guide

Acknowledgement

We wish to express our deep gratitude to our **Dr. Deepak Sharma**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions. We sincerely thanks to our respected **Dr. Virender Kadyan, Cluster Head of Data Science Cluster**, for his great support in doing our project in **Analyza (Data Analysis & Visualization Tool)**. We are also grateful to Dean SoCS UPES for giving us the necessary facilities to carry out our project work successfully. We also thanks to our Course Coordinator, **Dr. Prabhat Ranjan Singh** and our Activity Coordinator **Dr. Sachin Chaudhary** for providing timely support and information during the completion of this project. We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Dhuruv Kumar
500107769
Khushi Chauhan
500102244
Shruti Srivastava
500105401

Abstract

"Analyza" is a data analysis and visualization tool designed to simplify complex data processing and enhance decision-making. It integrates machine learning (ML) techniques for predictive and diagnostic analysis, while leveraging large language models (LLMs) for descriptive analysis, enabling users to gain valuable insights from their data. The platform supports scalable batch processing, ensuring efficient handling of large datasets, and provides customizable visualizations to help users interpret data effectively.

Users can upload CSV (Comma Separated Values) files as the primary data source, making it easy to work with structured datasets. Designed for both technical and non-technical users, Analyza features an intuitive and interactive interface, allowing users to explore and analyze data without requiring programming expertise. The tool supports descriptive analysis for summarizing datasets, predictive analysis for forecasting trends, and diagnostic analysis for identifying root causes of patterns and anomalies. By offering a structured and interactive approach to data analysis, Analyza empowers businesses, researchers, financial analysts, educators, and government organizations to derive actionable insights and make informed, data-driven decisions efficiently.

Contents

1 Introduction **1**
- 1.1 Purpose of the Project . . . 1
- 1.2 Motivation . . . 1
- 1.3 Objective . . . 2
- 1.4 Target Beneficiary . . . 2
- 1.5 Project Scope . . . 3
  - 1.5.1 Web-Based Interface . . . 3
  - 1.5.2 Integration with Google's Gemini API . . . 4
  - 1.5.3 Predictive Analysis . . . 4
  - 1.5.4 Diagnostic Analysis . . . 4
  - 1.5.5 Custom Visualization . . . 4
  - 1.5.6 Scalable and Secure Infrastructure . . . 4
- 1.6 Area of Application . . . 5
- 1.7 PERT Chart . . . 6

2 Project Description **7**
- 2.1 Reference Algorithm . . . 7
  - 2.1.1 Descriptive Analysis . . . 7
  - 2.1.2 Predictive Analysis . . . 8
  - 2.1.3 Diagnostic Analysis . . . 10
- 2.2 Data/Data Structure . . . 11
  - 2.2.1 Data Sources . . . 11
  - 2.2.2 Data Structure . . . 12
- 2.3 SWOT Analysis . . . 13
  - 2.3.1 Strengths . . . 13
  - 2.3.2 Weaknesses . . . 14
  - 2.3.3 Opportunities . . . 14
  - 2.3.4 Threats . . . 14
- 2.4 Project Features . . . 14
- 2.5 User Classes and Characteristics . . . 15
  - 2.5.1 Data Analysts & Data Scientists . . . 15
  - 2.5.2 Business Professionals & Decision Makers . . . 16
  - 2.5.3 Educators & Researchers . . . 16
  - 2.5.4 Students & Beginners . . . 16
  - 2.5.5 IT & Software Developers . . . 17
  - 2.5.6 General Users & Non-Technical Users . . . 17
- 2.6 Design and Implementation Constraints . . . 17
  - 2.6.1 Data Handling Constraints . . . 17

2.6.2 Performance & Scalability Constraints . . . 18
2.6.3 User Experience & Accessibility Constraints . . . 18
2.6.4 Machine Learning & Statistical Constraint . . . 18
2.7 Design Diagrams . . . 19
2.7.1 Use Case Diagram . . . 19
2.7.2 Work Flow Diagram . . . 20
2.7.3 Data Flow Diagram . . . 21
2.7.4 Swimlane Diagram (Activity Diagram) . . . 23
2.7.5 State Diagram . . . 24
2.7.6 Sequence Diagram . . . 25
2.8 Assumptions and Dependencies . . . 27
2.8.1 Assumptions . . . 27
2.8.2 Dependencies . . . 27

3 System Requirements 28
3.1 User Interface . . . 28
3.1.1 Dashboard (Main Screen) . . . 28
3.1.2 LLM Analysis Interface . . . 28
3.1.3 Custom Analysis Interface . . . 29
3.1.4 Custom Visualization Interface . . . 29
3.2 Software Interface . . . 29
3.2.1 Frontend-Backend Communication . . . 30
3.2.2 LLM Integration . . . 30
3.2.3 Data Processing & Machine Learning . . . 30
3.2.4 Visualization & Reporting . . . 31
3.3 Protocols . . . 31
3.3.1 Communication Protocols . . . 31
3.3.2 Real-Time Data & Future Enhancements . . . 32

4 Non-functional Requirements 33
4.1 Performance Requirements . . . 33
4.1.1 Dataset Size Handling . . . 33
4.1.2 Response Time . . . 33
4.1.3 Optimization Techniques . . . 34
4.1.4 Scalability . . . 34
4.2 Security Requirements . . . 34
4.2.1 Data Storage Policy . . . 34
4.2.2 Data Encryption . . . 34
4.2.3 Data Integrity and Validation . . . 35
4.3 Software Quality Attributes . . . 35
4.3.1 Reliability . . . 35
4.3.2 Usability . . . 35
4.3.3 Maintainability . . . 36

5 Other Requirements 37
5.1 Browser Compatibility . . . 37
5.1.1 Cross-Browser Testing . . . 37
5.2 System Requirements . . . 38
5.2.1 Client-Side Requirements . . . 38

5.2.2 Server-Side Setup 38
5.3 Security and Performance Considerations 39

6 Project Progress and Implementation Status **40**
6.1 Landing Page 40
6.2 LLM Module 41
6.3 Custom Analysis Module 43
6.3.1 Predictive Analysis 43
6.3.2 Diagnostic Analysis Module 53
6.4 Custom Visualization Module 57
6.5 Documentation Tab 60
6.6 About Us 62

7 Conclusion **64**
7.1 Future Enhancements 65

A Glossary **67**

B Analysis Model **69**

C Issues List **71**

List of Figures

1.1 Area of Application 5
1.2 PERT Chart 6

2.1 Predictive analysis 8
2.2 Diagnostic analysis 10
2.3 Data/Data Structure 11
2.4 SWOT analysis 13
2.5 Use Case Diagram for Analyza 19
2.6 Work Flow Diagram for Analyza 20
2.7 Level 0 Data Flow Diagram for Analyza 21
2.8 Level 1 Data Flow Diagram for Analyza 22
2.9 Swimlane Diagram for Analyza 23
2.10 State Diagram for Analyza 24
2.11 Sequence Diagram for Analyza 26

6.1 Landing Page Top View 41
6.2 Landing Page Bottom View 41
6.3 Step 1: Uploading dataset and preview 42
6.4 Step 2: Submitting a natural language query for analysis 42
6.5 Step 3: Viewing predicted results and insights with visualization 42
6.6 Predictive Analysis Dashboard: Uploading data and applying AI suggestions for features and target variable 44
6.7 Model Selection & Model Performance: XGBoost 45
6.8 Model Selection & Model Performance: Gradient Boost 45
6.9 Model Selection & Model Performance: AdaBoost 46
6.10 Model Selection & Model Performance: Naive Bayes 46
6.11 Model Selection & Model Performance: KNN 47
6.12 Model Selection & Model Performance: Random Forest 47
6.13 Model Selection & Model Performance: Linear Regression 48
6.14 Model Selection & Model Performance: Polynomial Regression 49
6.15 Model Selection & Model Performance: KNN Regression 49
6.16 Model Selection & Model Performance: Random Forest Regression 50
6.17 Model Selection & Model Performance: AdaBoost Regression 50
6.18 Model Selection & Model Performance: Gradient Boosting Regression . . 51
6.19 Model Selection & Model Performance: XGBoost Regression 52
6.20 Classification Prediction Result 53
6.21 Regression Prediction Result 53
6.22 Diagnostic Analysis Dashboard 54
6.23 Selecting Target Variable for Root Cause Analysis 55

6.24 Correlation Matrix: Identifying Strong Relationships Between Variables . . 56
6.25 Anomalies Detected and Root Cause Analysis Results 56
6.26 Custom Visualization Dashboard 57
6.27 Chart Type Selection Interface 58
6.28 Line Plot and Histogram Visualization 59
6.29 Pie Chart Visualization 59
6.30 Box Plot and Violin Plot Visualization 60
6.31 Documentation Overview 61
6.32 Documentation User Guide LLM Analysis 61
6.33 Documentation User Guide Custom Analysis and Visualization 62
6.34 About us: Developers 62
6.35 About us: Mentor 63

Chapter 1

Introduction

1.1 Purpose of the Project

The purpose of the project is to develop a web-based data analysis tool called **Analyza**. This tool is designed to simplify data analysis for users by providing an intuitive interface for uploading datasets, performing various types of analysis, and generating visualizations. Analyza caters to both non-technical users and data scientists, offering a range of features from basic data previews to advanced predictive modeling. By integrating machine learning for predictive and diagnostic insights, as well as leveraging LLMs for descriptive analysis, Analyza aims to bridge the gap between raw data and actionable insights. The platform supports scalable batch processing, customizable visualizations, and seamless integration with various data sources, ensuring an interactive and user-friendly experience.

Analyza is designed to assist users in making data-driven decisions by providing interactive dashboards, machine learning-based insights, and customizable visualizations. The tool enables users to explore and interpret their data with ease, regardless of their technical expertise. By identifying trends, patterns, and anomalies within datasets, Analyza facilitates better decision-making and enhances data exploration. The integration of advanced processing techniques ensures a seamless analytical experience, making it a powerful tool for individuals and organizations seeking to derive meaningful insights from their data.

1.2 Motivation

In today's data-driven world, the ability to analyze and interpret vast amounts of information has become essential for decision-making across all domains — from businesses and governments to education and healthcare. However, the tools available for data analysis often require technical expertise, limiting accessibility for non-technical users and creating barriers to data exploration.

Analyza was conceived as a response to this challenge. The motivation behind developing Analyza lies in the growing need for a user-friendly, intelligent, and scalable platform that democratizes data analysis. By integrating machine learning techniques and LLMs, Analyza simplifies complex analytical tasks and presents insights in an intuitive format,

making data analytics accessible to everyone — regardless of their background in programming or statistics.

With support for natural language queries, interactive dashboards, and customizable visualizations, Analyza empowers users to derive actionable insights from structured data with ease. Whether it's a researcher validating a hypothesis, a business manager identifying trends, or a student learning data science concepts, Analyza provides the necessary tools to make informed, data-driven decisions.

The project is driven by the vision to bridge the gap between raw data and meaningful understanding, and to create a seamless, inclusive, and impactful analytical experience for all users.

1.3 Objective

The primary objective of the Analyza project is to develop a unified, web-based platform that simplifies and enhances the process of data analysis and visualization for a wide range of users. The key objectives of this project include:

- **Descriptive Analysis using LLMs:**
  Integrate advanced LLMs (such as Google's Gemini API) to automatically generate natural language summaries and insights from uploaded datasets, enabling users to understand patterns and trends without requiring statistical expertise.
- **Custom Analysis Module (Predictive and Diagnostic):**
  Implement predictive analysis using machine learning models like Linear Regression, Logistic Regression, Random Forest and all other ML models to forecast trends and classify data. Additionally, provide diagnostic analysis through correlation matrices, anomaly detection, and root cause analysis to uncover meaningful relationships and detect irregularities in the dataset.
- **Custom Visualization Module:**
  Offer a variety of interactive and customizable charts — including bar charts, pie charts, scatter plots, line graphs, box plots & violin plots — allowing users to effectively visualize and interpret their data.
- **Unified Web-Based Platform:**
  Combine all the above functionalities into a single, intuitive web application with a clean interface built using React.js and FastAPI, ensuring seamless navigation and accessibility for both technical and non-technical users.

1.4 Target Beneficiary

The primary beneficiaries of Analyza include businesses, researchers, and professionals who rely on data-driven insights for decision-making. It serves data analysts, financial experts, students, educators, and organizations by providing an intuitive and scalable platform for processing and visualizing data. Analyza is designed to make data analysis

accessible, even for users with minimal programming experience. Analyza serves a wide range of users across multiple domains, helping them leverage data for better decision-making and insights. The primary beneficiaries of this tool include:

- **Businesses**
  Businesses can leverage these tools to gain critical insights into market trends, understand customer behavior more deeply, and significantly improve operational efficiency. By analyzing data patterns, companies can make informed decisions that drive growth and improve their competitive edge.

- **Researchers**
  Researchers benefit from the ability to analyze large datasets, identify complex patterns, and validate hypotheses more effectively. These capabilities accelerate the pace of discovery and enable groundbreaking research across various fields.

- **Financial Experts**
  Financial experts can visualize financial data, assess risks, and predict market trends with greater accuracy. These tools provide the insights needed to make informed investment decisions and manage financial resources effectively.

- **Students and Educators**
  Students and educators find vaulue in facilitating academic research, tracking student performance, and enhancing learning outcomes. These resources support a more engaging and effective educational environment.

- **Healthcare Professionals**
  Healthcare professionals can monitor patient data, predict health trends, and optimize hospital resources, leading to improved patient care and more efficient healthcare delivery.

- **Government Agencies**
  Government agencies can analyze public data, improve policy decisions, track economic trends, and enhance governance. By leveraging data-driven insights, agencies can make informed decisions that benefit society as a whole.

1.5 Project Scope

The scope of the **Analyza** project encompasses the development of a comprehensive web-based data analysis platform designed to cater to a wide range of users, from non-technical individuals to data scientists. The platform will provide an intuitive interface for uploading datasets, performing advanced data analysis, and generating insightful visualizations. Below is a detailed breakdown of the project scope:

1.5.1 Web-Based Interface

The project will focus on developing a user-friendly web-based interface that allows users to upload datasets in CSV format. The interface will include features such as data preview, column selection, and analysis options. Users will be able to interact with the platform seamlessly, making it accessible to individuals with varying levels of technical expertise. The interface will also support drag-and-drop functionality for file uploads, ensuring a smooth user experience.

1.5.2 Integration with Google's Gemini API

One of the key components of the project is the integration with Google's Gemini API, which will enable advanced data analysis capabilities. This integration will allow users to perform natural language queries on their datasets, making it easier to extract insights without requiring deep technical knowledge. The API will also support the generation of Python code for visualizations, which will be executed on the backend to produce graphical representations of the data.

1.5.3 Predictive Analysis

The platform will support predictive analysis using machine learning models such as Linear Regression, Logistic Regression, Random Forest, Boosting etc. These models will be implemented to help users predict future trends, classify data, and identify patterns. The system will automatically handle data preprocessing tasks, such as handling missing values and encoding categorical variables, to ensure accurate predictions. Users will be able to select target variables and features or use AI suggested target and features, and the platform will generate performance metrics such as accuracy, mean squared error, and R^2 scores.

1.5.4 Diagnostic Analysis

In addition to predictive analysis, the platform will offer diagnostic analysis capabilities. This will include identifying anomalies in datasets using statistical methods such as Z-score analysis. The system will also perform root cause analysis by examining correlations between variables, helping users understand the underlying factors influencing their data. These features will be particularly useful for identifying outliers and understanding the relationships between different data points.

1.5.5 Custom Visualization

The platform will provide robust visualization capabilities, allowing users to create a variety of charts and graphs, including bar charts, scatter plots, line graphs, histograms, box plots, violin plots, and pie charts. Users will be able to customize these visualizations by selecting axes, colors, and other parameters. The visualizations will be interactive, enabling users to explore their data in greater detail.

1.5.6 Scalable and Secure Infrastructure

The project will ensure that the platform is built on a secure and scalable infrastructure. Data processing will be performed in memory, and user-uploaded data will not be stored permanently on the server, ensuring data privacy and security. The system will be designed to handle large datasets efficiently, with support for batch processing to manage data of varying sizes. The backend will be built using FastAPI, a modern and high-performance web framework, while the frontend will be developed using React.js for a responsive and dynamic user interface.

In summary, the scope of the **Analyza** project is to create a powerful, user-friendly, and scalable data analysis platform that empowers users to explore, analyze, and visualize

their data effectively. The platform will combine advanced machine learning techniques, natural language processing, and interactive visualization tools to provide a comprehensive solution for data-driven decision-making.

1.6 Area of Application

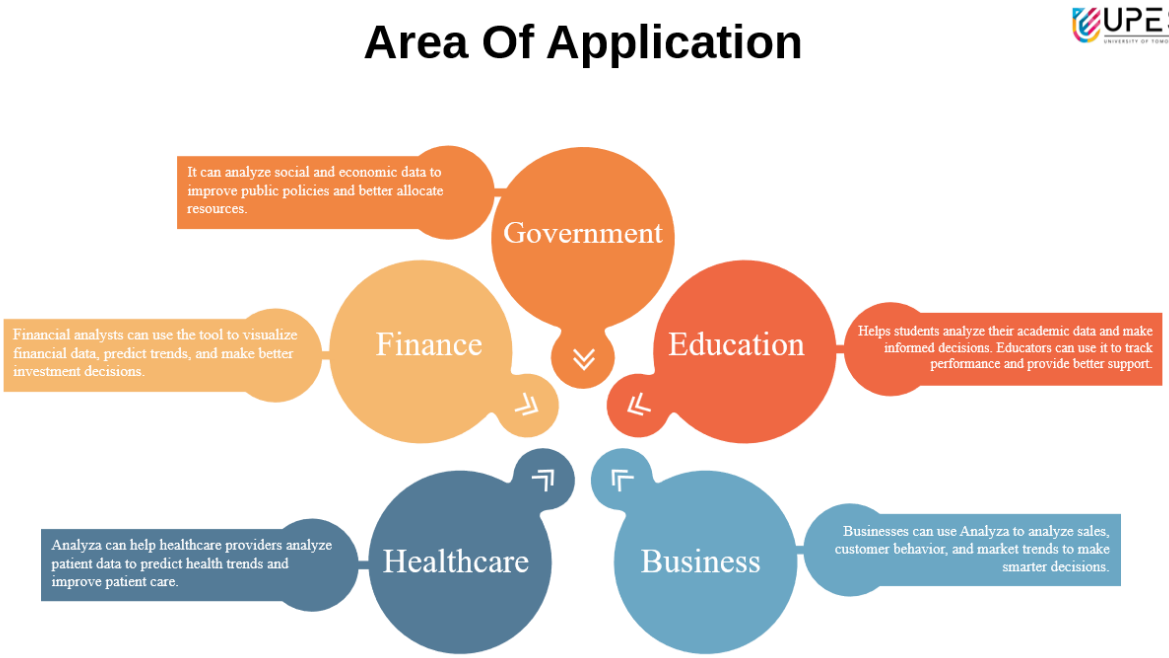


Figure 1.1: Area of Application

Analyza is a versatile data analytics tool that empowers users across various domains to derive insights and make data-driven decisions, as shown in Figure 1.1.

- **Education:** Helps students analyze their academic data and make informed decisions. Educators can use it to track student performance, identify learning gaps, and provide better support to improve overall educational outcomes.
- **Business:** Businesses can utilize Analyza to analyze sales trends, customer behavior, and market patterns. It helps companies optimize their strategies, improve operational efficiency, and make data-driven decisions to enhance profitability.
- **Healthcare:** Analyza can support healthcare professionals in analyzing patient data, predicting health trends, and improving patient care. It can be used for identifying disease patterns, monitoring patient recovery, and optimizing hospital resource allocation.
- **Finance:** Financial analysts can leverage Analyza to visualize financial data, predict market trends, and make informed investment decisions. The tool can assist in portfolio analysis, risk assessment, and fraud detection, improving overall financial planning.

- **Government:** Governments can analyze social and economic data to improve public policies and better allocate resources. It can be used for studying population trends, monitoring economic growth, and assessing the impact of various initiatives on society.
- **Research and Development:** Researchers can use Analyza to process large datasets, identify patterns, and validate hypotheses. It supports data-driven discoveries in fields such as environmental science, social studies, and artificial intelligence research.

1.7 PERT Chart

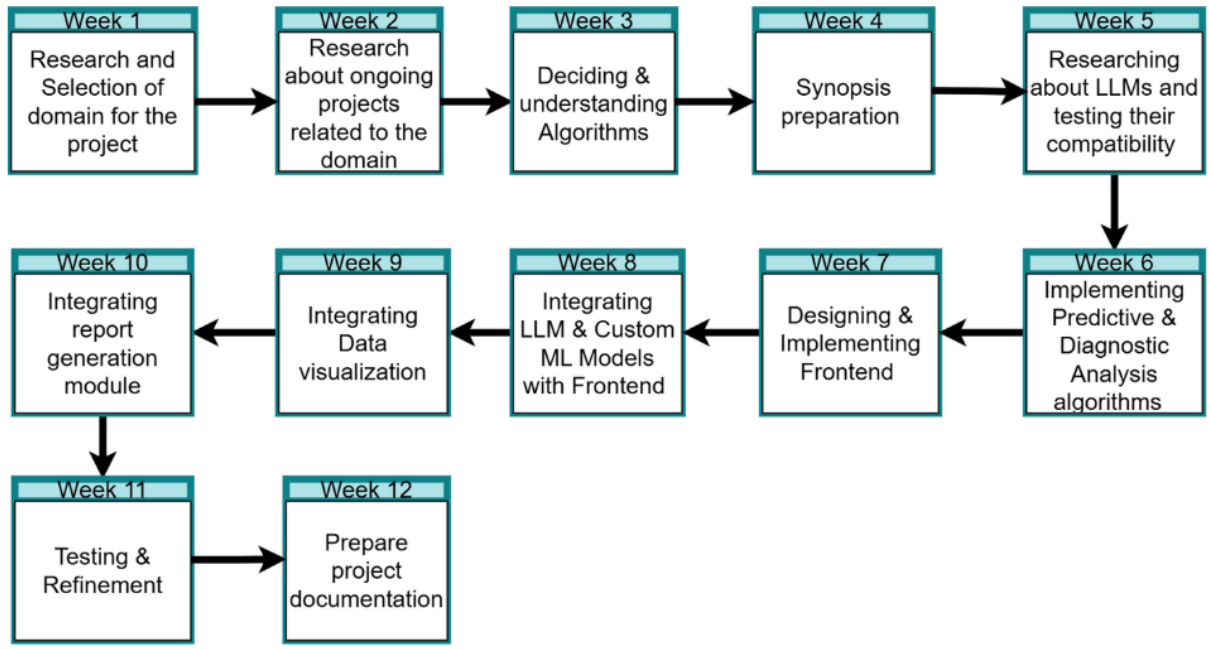


Figure 1.2: PERT Chart

In 1.2, this chart shows a 12-week project plan. It starts with picking a project focus in Week 1, then researching similar projects in Week 2. By Week 5, the team is looking into LLMs, then building analysis tools in Week 6. Weeks 7-10 cover building the user interface, connecting the AI models to it, adding data visuals, and creating reports. The project wraps up with testing in Week 11 and documentation in Week 12. The arrows show how each week's work leads to the next step.

Chapter 2

Project Description

2.1 Reference Algorithm

Analyza employs a structured combination of analytical techniques to extract meaningful insights from data. These techniques are categorized into **Descriptive**, **Predictive**, and **Diagnostic** analyses, each serving a specific purpose within the data analysis workflow. The following subsections outline the core algorithms and methods used in each type of analysis, enabling users to explore, forecast, and interpret data effectively.

2.1.1 Descriptive Analysis

Descriptive analysis focuses on summarizing and interpreting raw data to identify meaningful patterns, trends, and relationships. It provides a clear and concise representation of data through statistical measures and visualizations, making complex datasets more accessible and actionable.

Large Language Models (LLMs): Analyza integrates LLMs to provide natural language-based insights and summaries, helping users understand patterns and trends without deep statistical knowledge. LLMs enhance data interpretation by generating human-readable explanations of complex analytical results.

2.1.2 Predictive Analysis

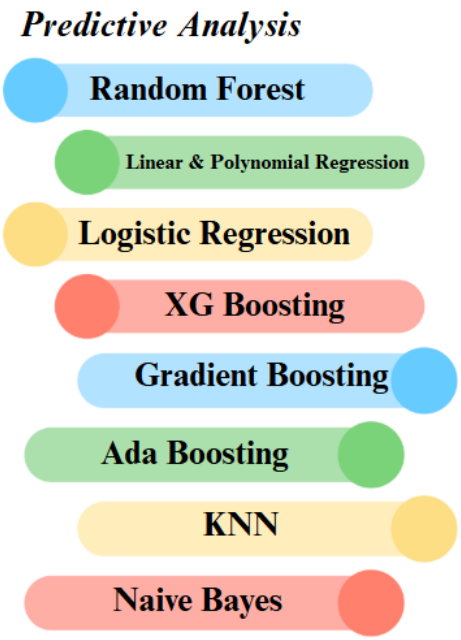


Figure 2.1: Predictive analysis

Predictive analysis in Analyza is designed to forecast future outcomes and trends based on historical data, as illustrated in Figure 2.1.

Linear and Polynomial Regression

Linear and Polynomial Regression are techniques used to model the relationship between variables. Linear Regression fits a straight line to the data, ideal for simple relationships. Polynomial Regression, on the other hand, fits a curve to capture more complex, non-linear patterns. These methods are commonly applied in forecasting trends, estimating values, and evaluating continuous metrics like pricing and temperature variations.

Logistic Regression

Logistic Regression is a statistical method used for binary classification. It predicts the probability of an outcome based on one or more predictor variables. Logistic Regression is often utilized in scenarios like fraud detection, medical diagnoses, and marketing response prediction, where the outcome is a binary value (e.g., yes/no, true/false).

Random Forest

Random Forest is a powerful ensemble learning method that combines multiple decision trees to create a robust model for both classification and regression tasks. By averaging the results from many trees, Random Forest reduces overfitting and improves predictive accuracy. It is widely used in tasks such as feature selection, classification, and prediction in complex datasets.

AdaBoost

AdaBoost (Adaptive Boosting) is an ensemble learning method that combines weak classifiers to form a strong one. It adjusts the weights of incorrectly predicted instances so subsequent models focus more on difficult cases. AdaBoost is widely used for face detection, text classification, and spam filtering.

Gradient Boosting

Gradient Boosting is a machine learning technique that builds models sequentially by minimizing errors made by previous models. Each new model corrects the residuals (errors) of its predecessor, leading to improved performance. It's commonly applied in areas such as risk modeling, sales forecasting, and web search ranking.

XGBoost

XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient boosting algorithms. Known for its speed and performance, XGBoost builds models in a sequential manner, correcting previous errors at each step. It's particularly useful in data science competitions and real-world applications like ranking and recommendation systems.

K-Nearest Neighbors (KNN)

KNN is a simple, instance-based learning algorithm used for both classification and regression. It works by finding the 'k' closest data points in the training set to a given input and predicting the output based on majority voting (for classification) or averaging (for regression). KNN is often applied in recommendation engines, handwriting recognition, and pattern detection.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of feature independence. Despite its simplicity, it performs well in high-dimensional data and is widely used in spam detection, sentiment analysis, and document classification.

2.1.3 Diagnostic Analysis

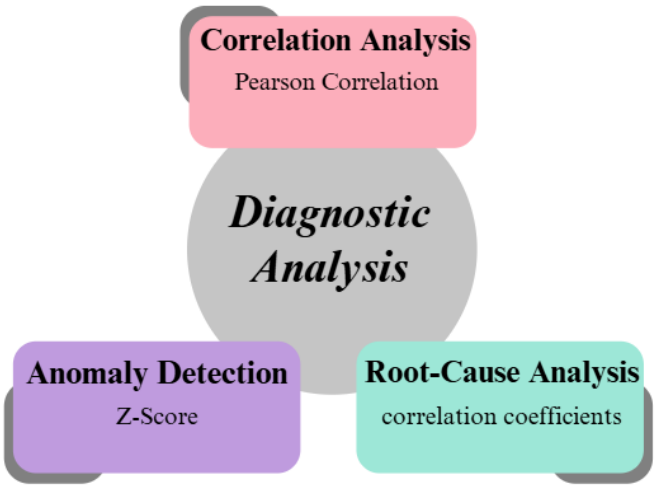


Figure 2.2: Diagnostic analysis

Diagnostic analysis in Analyza helps users uncover the causes behind observed patterns, anomalies, or trends in their data. It identifies key relationships between variables, detects unusual data points, and determines significant factors influencing specific outcomes. This module consists of correlation analysis, anomaly detection, and root-cause analysis, along with visualizations to enhance data interpretation, as shown in Figure 2.2.

Correlation Analysis

Correlation analysis measures the strength and direction of relationships between numerical variables. It helps identify whether changes in one variable correspond to changes in another, which is useful in various fields such as finance, healthcare, and business analytics. A strong positive correlation means both variables move in the same direction, while a strong negative correlation indicates they move in opposite directions.

Analyza calculates Pearson correlation coefficients, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). A correlation matrix is generated, highlighting significant relationships among variables. The results are visualized using a correlation heatmap, where deep red represents strong positive correlations, deep blue indicates strong negative correlations, and neutral colors signify weak or no correlation.

Anomaly Detection

Anomalies, or outliers, are data points that deviate significantly from expected values, often indicating errors or rare events. Detecting anomalies is essential for identifying fraudulent transactions, security threats, or operational inefficiencies.

Analyza uses the Z-score method, which measures how many standard deviations a data point is from the mean. By default, the system flags anomalies with an absolute Z-score greater than 3, as this threshold captures values that fall outside approximately 99.7% of a normal distribution. However, users can modify this threshold to make the anomaly detection more or less sensitive.

Root-Cause Analysis

Root-cause analysis identifies the most influential factors affecting a target variable. This method helps users understand what drives specific outcomes, such as customer behavior, market trends, or system failures.

Analyza computes correlation coefficients between the target variable and all other numerical variables, ranking them by correlation strength. The results are presented in tabular format and visualized for easier interpretation, helping users pinpoint key drivers behind trends and anomalies.

Visualization – Enhancing Interpretability

To improve the usability of diagnostic analysis, Analyza provides intuitive visualizations. One of the key visual tools is the correlation heatmap, a color-coded matrix that visually represents the relationships between variables. In this heatmap, deep red indicates strong positive correlations, deep blue represents strong negative correlations, and neutral colors signify weak or no correlations. Numeric values are displayed on the heatmap to provide precise correlation information.

2.2 Data/Data Structure

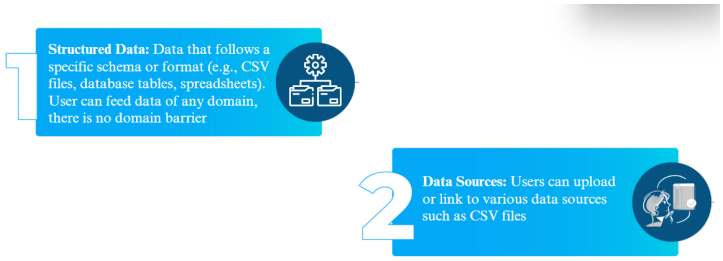


Figure 2.3: Data/Data Structure

2.2.1 Data Sources

Analyza allows users to seamlessly upload or connect to various data sources, ensuring flexibility in data analysis. The platform currently supports:

- **User-Uploaded Files**: Users can upload structured data files in CSV format, which are directly processed and analyzed within the system. This format is widely supported and allows for seamless integration into Analyza's processing and visualization modules.

While the current version of Analyza is focused on CSV-based data ingestion for simplicity and performance, the architecture is designed to be extensible. Future updates may introduce support for additional data sources, such as real-time data streams or API-based imports, enhancing the platform's versatility.

By supporting direct user uploads in a standard format, Analyza ensures a smooth and efficient workflow, allowing users to focus on extracting insights without complex setup

or configuration. This approach empowers users to work with a wide range of datasets, promoting personalized and meaningful data analysis.

By supporting multiple data sources, Analyza ensures users can work with diverse datasets, making data analysis more comprehensive and efficient.

2.2.2 Data Structure

The tool is optimized for handling structured data. Structured data refers to information that follows a specific schema or format, making it easily searchable and analyzable. Examples include CSV files, database tables, and spreadsheets. Analyza is designed to process structured data efficiently, allowing users to extract meaningful insights without domain-specific restrictions. This flexibility enables users from various industries—including finance, healthcare, education, and research—to analyze data without requiring specialized data formats. The tool is optimized for handling structured data,as shown in Figure 2.3 where:

- Each row represents an individual data point or record. Example: A student's performance record in an education dataset.
- Each column holds a specific attribute or feature. Example: Attributes like "age," "income," "purchase history" in a business dataset.
- Supports both numerical and categorical data. Numerical values can be used for statistical computations, while categorical labels aid in classification and segmentation.
- **Data Cleaning & Preprocessing**: Analyza provides built-in tools to handle missing values, remove duplicates, and standardize data formats to ensure high-quality analysis.

2.3 SWOT Analysis

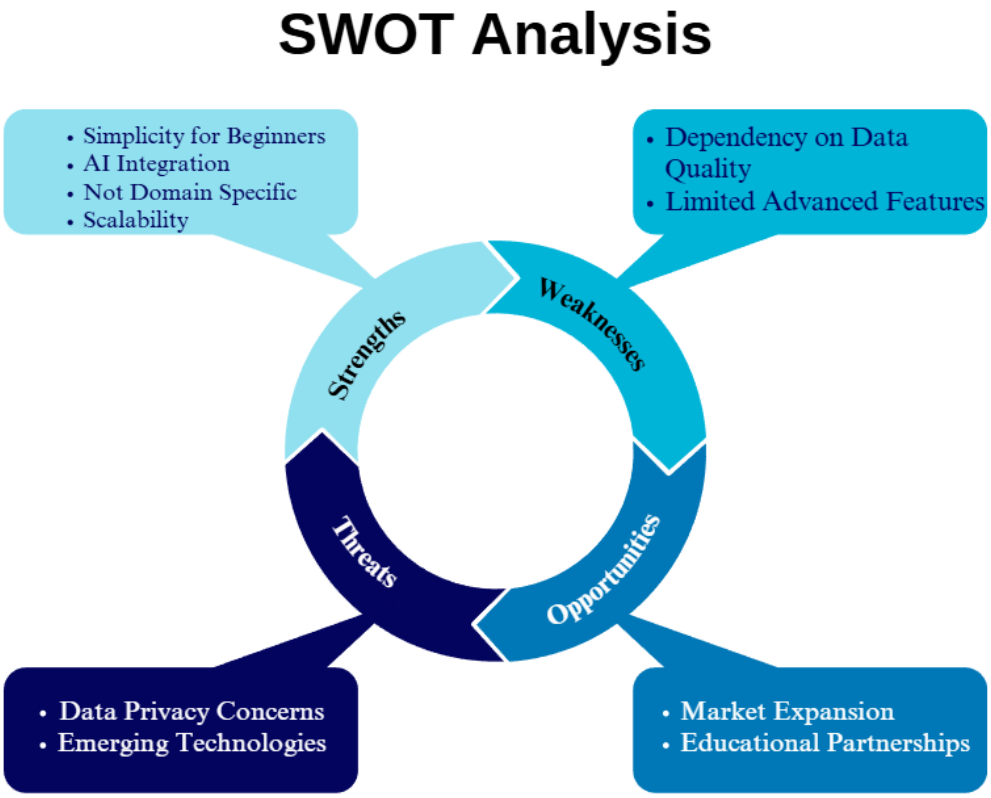


Figure 2.4: SWOT analysis

2.3.1 Strengths

- **Simplicity for Beginners** – Analyza is designed to be easy to use, making data analysis accessible to users without advanced technical expertise. Its interactive interface ensures that even beginners can explore, visualize, and interpret data with minimal effort.
- **AI Integration** – The platform uses machine learning and LLMs to enhance data analysis. This integration allows for predictive and diagnostic insights, automating complex tasks and making data-driven decision-making more efficient.
- **Custom Visualization Support** – Users can generate and personalize a variety of visualizations to better interpret their data. The system allows flexibility in graph types, color schemes, and variable selection, making data exploration more interactive and insightful.
- **Multi-Domain Compatibility** – Analyza is adaptable across industries such as education, healthcare, finance, and marketing, providing insights tailored to different domains.

2.3.2 Weaknesses

- **Dependency on Data Quality** – Analyza is designed to be easy to use, making data analysis accessible to users without advanced technical expertise. Its interactive interface ensures that even beginners can explore, visualize, and interpret data with minimal effort.
- **Limited Advanced Features** – While Analyza is beginner-friendly, it may lack some advanced functionalities required by data science professionals, such as deep statistical modelling, custom scripting, or specialized machine learning workflows.

2.3.3 Opportunities

- **Market Expansion** – With the increasing demand for data-driven insights across industries, Analyza has the potential to expand into new markets, including business intelligence, healthcare, and finance, where data analysis plays a critical role.
- **Educational Partnerships** – Collaborating with universities and training institutions can position Analyza as a valuable learning tool. By providing student-friendly analytics features, the platform can support academic research and data literacy programmers.
- **Cloud and API Integrations** – Expanding support for cloud-based data processing and third-party API integrations can enhance accessibility and usability for enterprises.

2.3.4 Threats

- **Data Privacy Concerns** – Handling sensitive data raises privacy and security challenges. Users may hesitate to use Analyza if they perceive risks related to data breaches, regulatory compliance, or unauthorized access.
- **Emerging Technologies** – Rapid advancements in AI and data analytics could lead to the development of more sophisticated competitors. New tools with superior automation, real-time processing, or advanced AI capabilities may challenge Analyza's market position.

2.4 Project Features

Analyza is designed with a comprehensive set of features that enable seamless data analysis, visualization, and insight generation. These features enhance usability, efficiency, and accuracy, catering to both beginners and advanced users. To support diverse analytical needs, Analyza includes the following key features:

- **Interactive Dashboards** – It offers dynamic, customizable visualizations such as bar charts, histograms, scatter plots, and violin plot, allowing users to explore trends, patterns, and key metrics in an intuitive manner. Users can filter, drill down, and compare different data points interactively.

- **Batch Processing** – It optimized for handling large datasets efficiently, enabling users to process, analyse, and transform bulk data without performance bottlenecks. This feature supports parallel computing and optimized memory management to accelerate data processing.
- **Machine Learning Integration** – It incorporates descriptive, predictive, and diagnostic analytics using machine learning models to uncover hidden patterns, detect anomalies, and forecast trends. Analyza automates model selection and parameter tuning to enhance accuracy
- **Data Pre-processing and Cleaning** – It automates essential pre-processing tasks such as handling missing values, detecting and correcting outliers, standardizing formats, and transforming categorical variables into numerical representations for smoother analysis.
- **Scalability and Performance Optimization** – It designed to scale efficiently across datasets of varying sizes, from small-scale academic projects to enterprise-level data processing. Advanced optimization techniques ensure quick query execution and minimal latency.
- **User-Friendly Interface** – It provides an intuitive, guided workflow that simplifies complex analytical tasks for both technical and non-technical users. Interactive tooltips, predefined templates, and AI-powered recommendations assist users at every stage of data exploration.
- **Customizable Analysis** – It allows users to tailor statistical and machine-learning-based analyses according to specific requirements, including selecting preferred models, adjusting parameters, and defining custom data transformations for domain-specific applications.
- **Custom Visualization** – It enables users to generate tailored visualizations based on the structure and attributes of their uploaded CSV data. Users can choose from multiple chart types, define axes, apply color coding, and interact with visual elements to enhance data interpretation

2.5 User Classes and Characteristics

Analyza is designed to serve a diverse range of users with varying levels of expertise and analytical needs. The platform ensures accessibility, efficiency, and ease of use, enabling seamless data exploration, analysis, and visualization for all users.

2.5.1 Data Analysts & Data Scientists

These users require advanced data querying, visualization, and AI-driven insights to analyze trends, identify patterns, and generate detailed reports.

Characteristics:

- They are proficient in data processing and visualization techniques.

- They need flexible querying options for both real-time and batch data analysis.
- They require API and database integrations to handle large-scale datasets efficiently.
- They prefer customizable dashboards for in-depth analysis and reporting.
- They may use LLM-powered automated insights and natural language queries to simplify complex analyses.

2.5.2 Business Professionals & Decision Makers

Business users leverage Analyza to gain actionable insights from organizational data, track key performance indicators (KPIs), and make informed decisions without requiring advanced data science expertise.

Characteristics:

- They require interactive dashboards for real-time decision-making.
- They need AI-powered recommendations and trend forecasting to support strategic planning.
- They prefer simple and visually intuitive reports over complex statistical models.
- They seek seamless integration with existing business tools and data sources for a unified workflow.

2.5.3 Educators & Researchers

Academics and researchers use Analyza to analyze datasets, predict trends, and present insights using structured visualizations and statistical tools.

Characteristics:

- They require statistical tools such as correlation analysis and hypothesis testing.
- They need clean, structured visualizations for presenting findings in research papers and presentations.
- They prefer customizable reports tailored to their research publications.
- They may integrate Analyza with external research databases for advanced analysis and data validation.

2.5.4 Students & Beginners

Students and beginners use Analyza to learn data analysis concepts interactively, gaining hands-on experience with data-driven insights in a user-friendly environment.

Characteristics:

- They need a user-friendly interface with guided workflows and tooltips to assist them in navigating the platform.

- They require basic statistical tools such as mean, median, variance, and frequency distribution for foundational analysis.
- They prefer interactive charts and visual summaries for easy understanding of data patterns.
- They may use AI-generated explanations to interpret complex results and enhance their learning experience.

2.5.5 IT & Software Developers

Developers integrate Analyza with other applications, build custom solutions, and enhance analytics capabilities through APIs and scripting.

Characteristics:

- They require API access for embedding analytics into their applications.
- They need scalability options to handle large datasets efficiently.
- They prefer flexible querying systems to support a wide range of use cases.
- They may integrate LLM-powered analysis for automated insights and natural language processing.

2.5.6 General Users & Non-Technical Users

Casual users or non-technical professionals use Analyza to quickly explore and understand data without requiring technical expertise.

Characteristics:

- They require simple data uploads and automated insights for quick analysis.
- They prefer natural language queries over complex SQL-based querying for ease of use.
- They need intuitive drag-and-drop features for basic data exploration and visualization.
- They prefer pre-built templates for quick analysis and reporting.

2.6 Design and Implementation Constraints

While designing and implementing Analyza, several constraints must be considered to ensure performance, scalability, and security. These constraints help define the scope and feasibility of the system while addressing technological and user-related limitations.

2.6.1 Data Handling Constraints

Supports structured data formats like CSV and excel entries; unstructured data (text, images) is not currently supported.

2.6.2 Performance & Scalability Constraints

- Processing large datasets requires optimized batch processing and memory management
- Machine learning computations must balance speed and accuracy.
- Scalable architecture is necessary to accommodate varying user workloads without performance degradation

2.6.3 User Experience & Accessibility Constraints

- The platform should be intuitive for both beginners and experienced users.
- AI-powered recommendations must not override user-defined analytical preferences.
- Cross-platform compatibility (desktop, mobile) should be considered.

2.6.4 Machine Learning & Statistical Constraint

- Machine learning models should be interpretable and explainable to non-technical users
- Statistical accuracy depends on data quality; garbage-in, garbage-out scenarios must be mitigated.
- Customization should be balanced with automated insights for ease of use.

2.7 Design Diagrams

2.7.1 Use Case Diagram

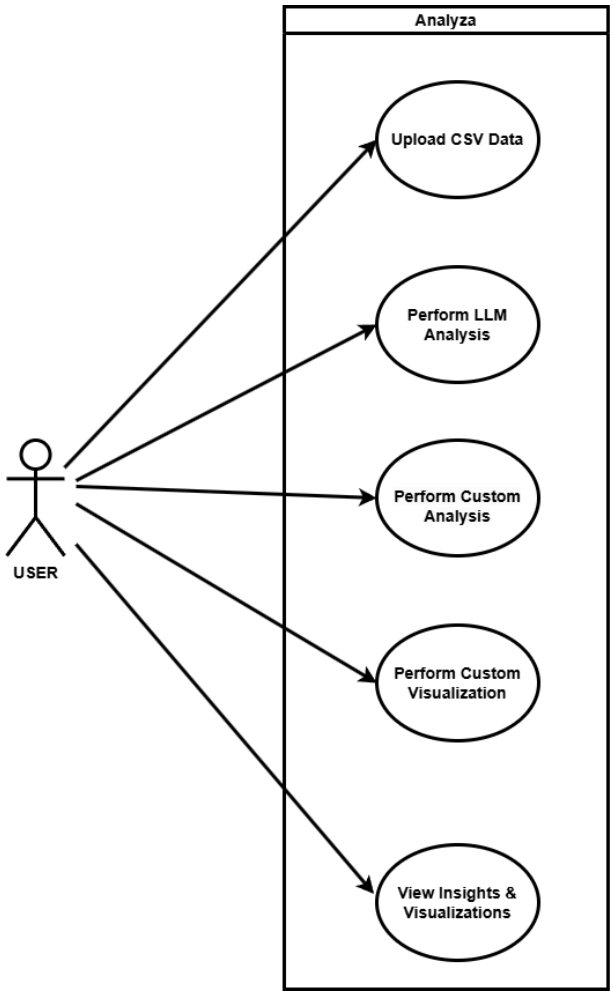


Figure 2.5: Use Case Diagram for Analyza

Figure 2.5 shows a **use case diagram for analyza analysis system**. The diagram displays a single user actor on the left who can perform five different actions within the Analyza system. These actions include: uploading CSV data, performing LLM analysis, performing custom analysis, creating custom visualizations, and viewing insights and visualizations. This diagram illustrates the core functionalities available to users of the analysis platform and shows how they can interact with different data processing and visualization features.

2.7.2 Work Flow Diagram

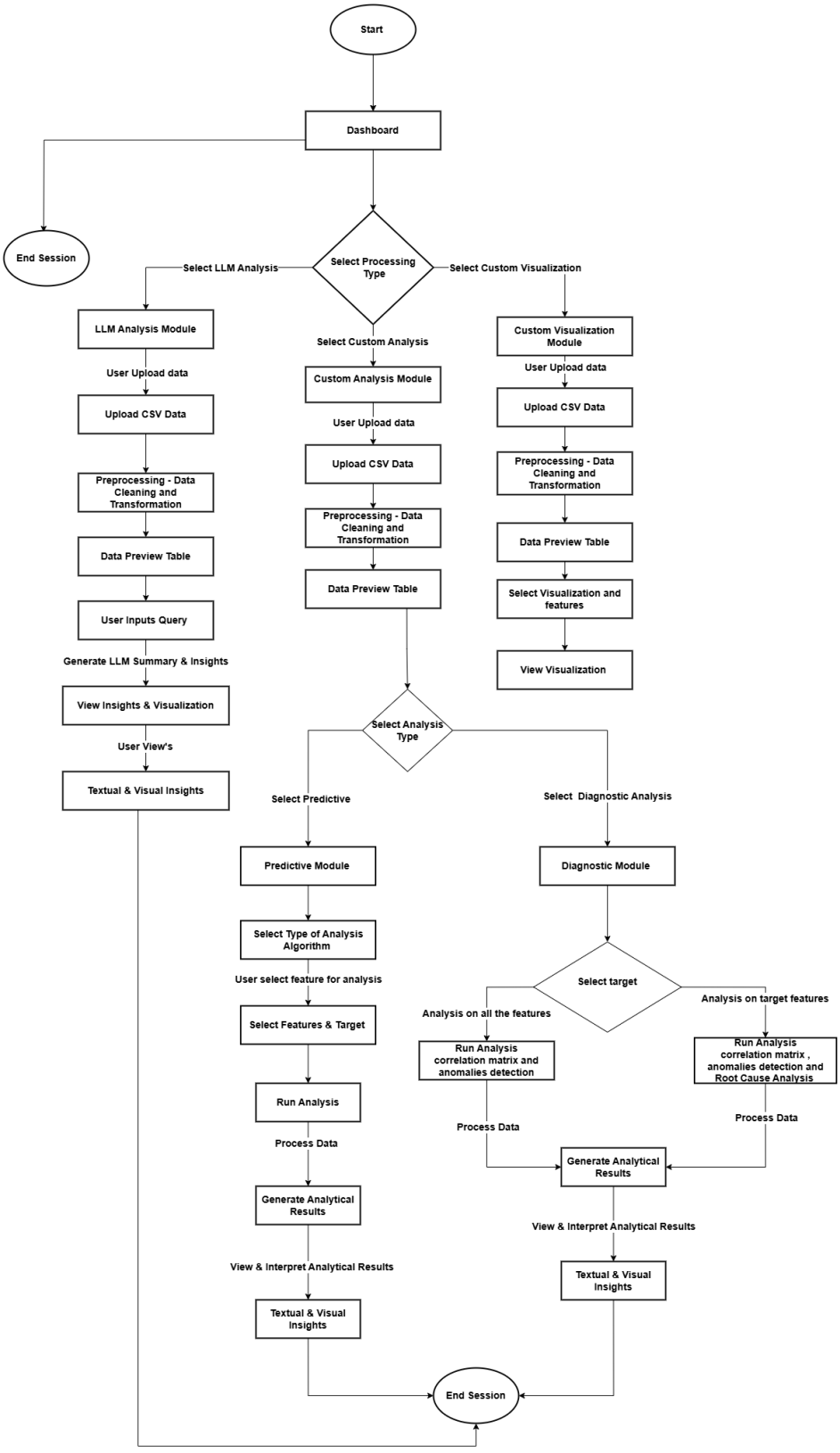


Figure 2.6: Work Flow Diagram for Analyza

Figure 2.6 presents the **Work Flow Diagram for Analyza**, detailing the complete end-to-end process from the user's perspective, beginning at the dashboard and concluding with analytical or visual insights.

The flow begins with the Dashboard, from where the user chooses a processing type—**LLM Analysis**, **Custom Analysis**, or **Custom Visualization**.

- **LLM Analysis Module:** The user uploads a CSV file. The system performs data cleaning and displays a preview table. After the user submits a query, the system generates LLM-based summaries and visual insights, which are displayed to the user.
- **Custom Analysis Module:** The user uploads and previews data, then selects between **Predictive** and **Diagnostic Analysis**.
  - Predictive Module: The user selects an algorithm and sets features and targets. The system runs the analysis, processes the data, and produces analytical results with insights.
  - Diagnostic Module: The user can either analyze all features or specify a target. The system performs correlation, anomaly, and root cause analysis before generating and displaying results.
- **Custom Visualization Module:** The user uploads data, previews it, selects visualization types and features, and views the generated visualizations.

Each path concludes with the display of **Textual and Visual Insights**, after which the session ends.

2.7.3 Data Flow Diagram

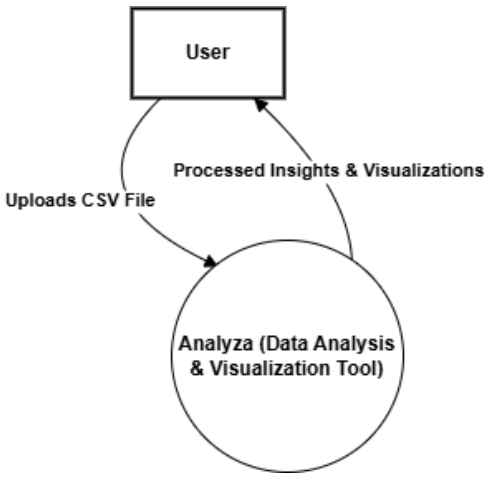


Figure 2.7: Level 0 Data Flow Diagram for Analyza

Figure 2.7 illustrates the **Level 0 Data Flow Diagram for Analyza**, providing a high-level overview of the system. It highlights the basic interaction between the user and the system. The user uploads a CSV file to the Analyza platform, which then processes the data and returns meaningful insights and visualizations. This abstraction gives a broad view of how the data flows from the user to the system and back.

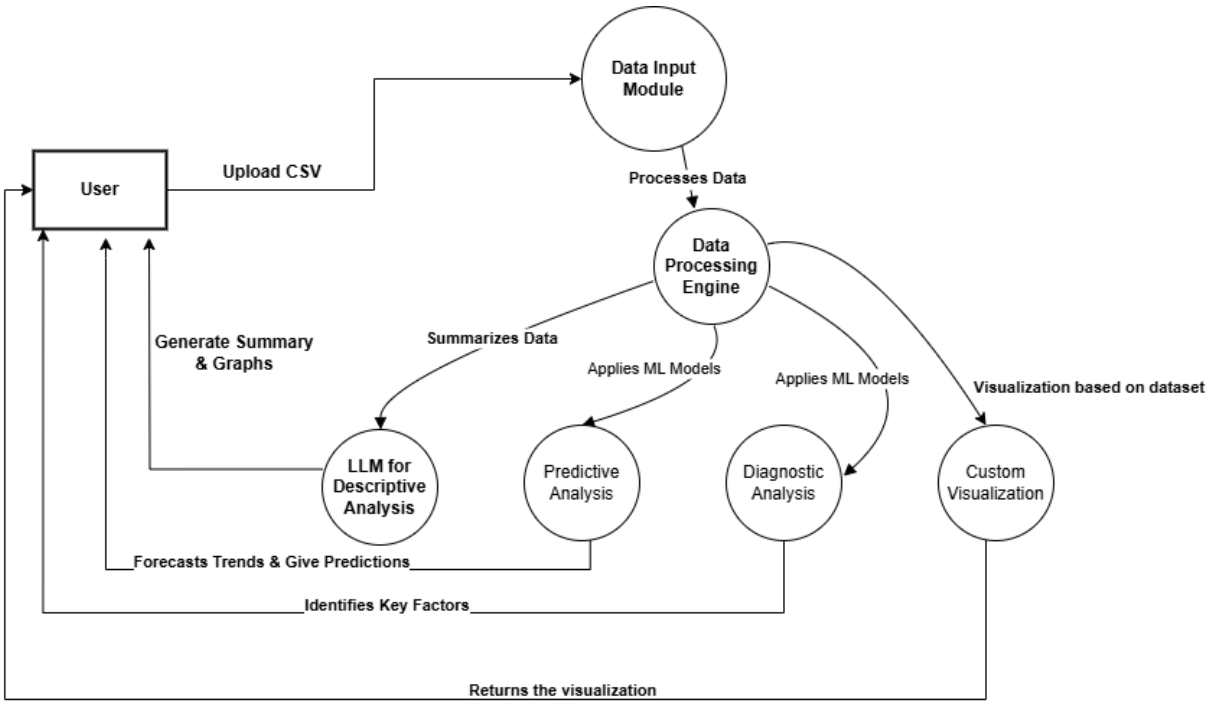


Figure 2.8: Level 1 Data Flow Diagram for Analyza

Figure 2.8 presents the **Level 1 Data Flow Diagram**, offering a more detailed breakdown of Analyza's internal processes. The user begins by uploading a CSV file, which is handled by the Data Input Module. This module forwards the information to the Data Processing Engine, the core component that applies machine learning models for both Predictive and Diagnostic Analysis.

The LLM for Descriptive Analysis module interacts with the processing engine to generate summaries, trends, and key insights. The outcomes from these analyses are then used to produce Custom Visualizations based on the dataset. Finally, processed insights, visualizations, and summaries are returned to the user. This detailed flow illustrates how different modules in Analyza collaborate to transform raw data into actionable insights.

2.7.4 Swimlane Diagram (Activity Diagram)

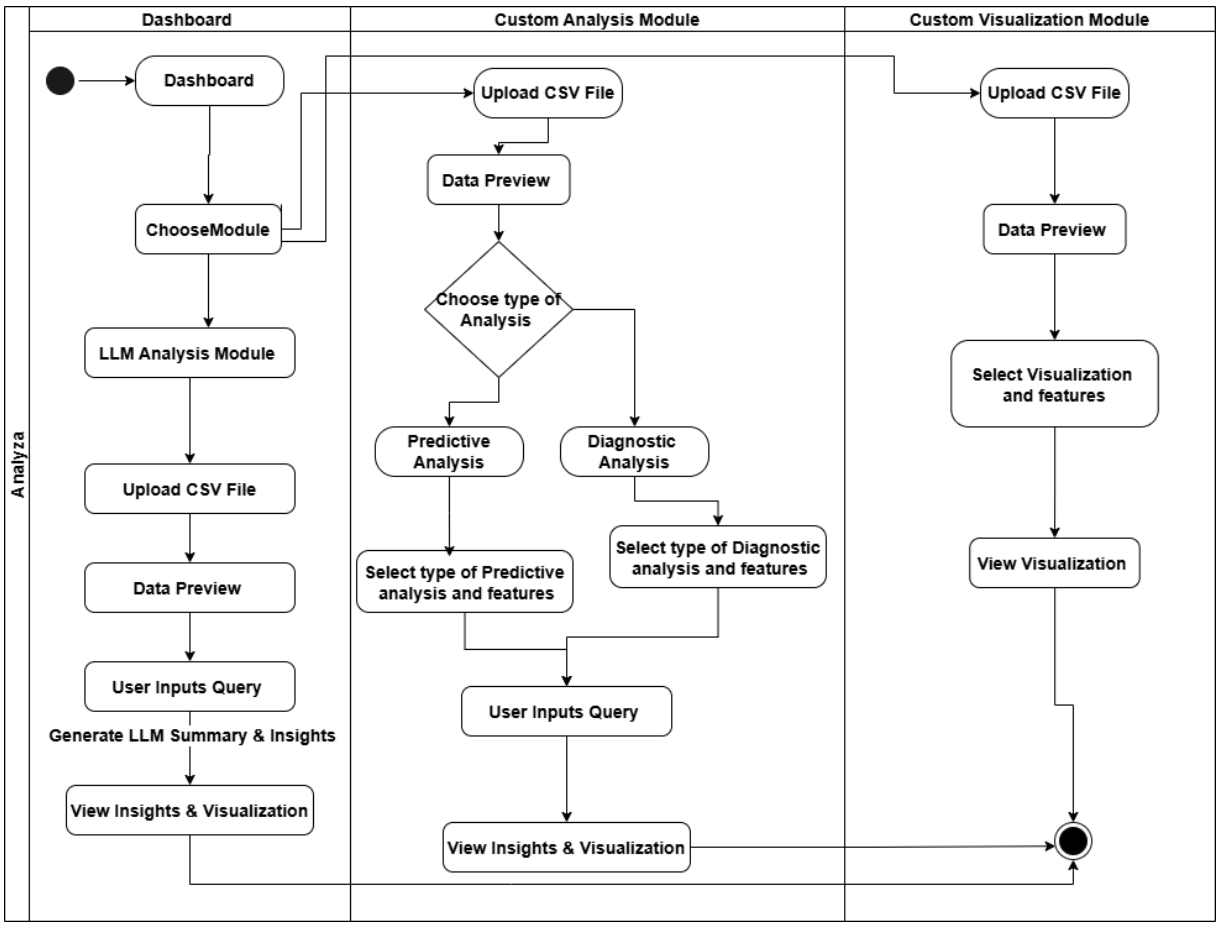


Figure 2.9: Swimlane Diagram for Analyza

Figure 2.9 depicts the **Swimlane Diagram for Analyza**, which clearly outlines the sequence of user actions and system processes across three major functional modules: Dashboard, Custom Analysis Module, and Custom Visualization Module.

In the Dashboard lane, the user begins by accessing the main dashboard and selecting the desired module. If the user chooses the LLM Analysis Module, they proceed to upload a CSV file, preview the data, input their query, and receive LLM-generated summaries and insights. These insights are then visualized within the same flow.

In the Custom Analysis Module lane, users first upload a CSV file and preview the data. They are then prompted to choose between Predictive Analysis and Diagnostic Analysis. Based on their choice, users select the appropriate analysis type and relevant features. After submitting their query, they receive insights and visualizations tailored to the chosen analysis type.

The Custom Visualization Module This starts similarly with a CSV upload and data preview. Users then select the type of visualization and features they wish to explore. Once selections are made, the system returns a customized visualization.

The diagram effectively captures the flow of interactions between the user and system components, illustrating how Analyza provides a modular and user-driven approach to data analysis and visualization.

2.7.5 State Diagram

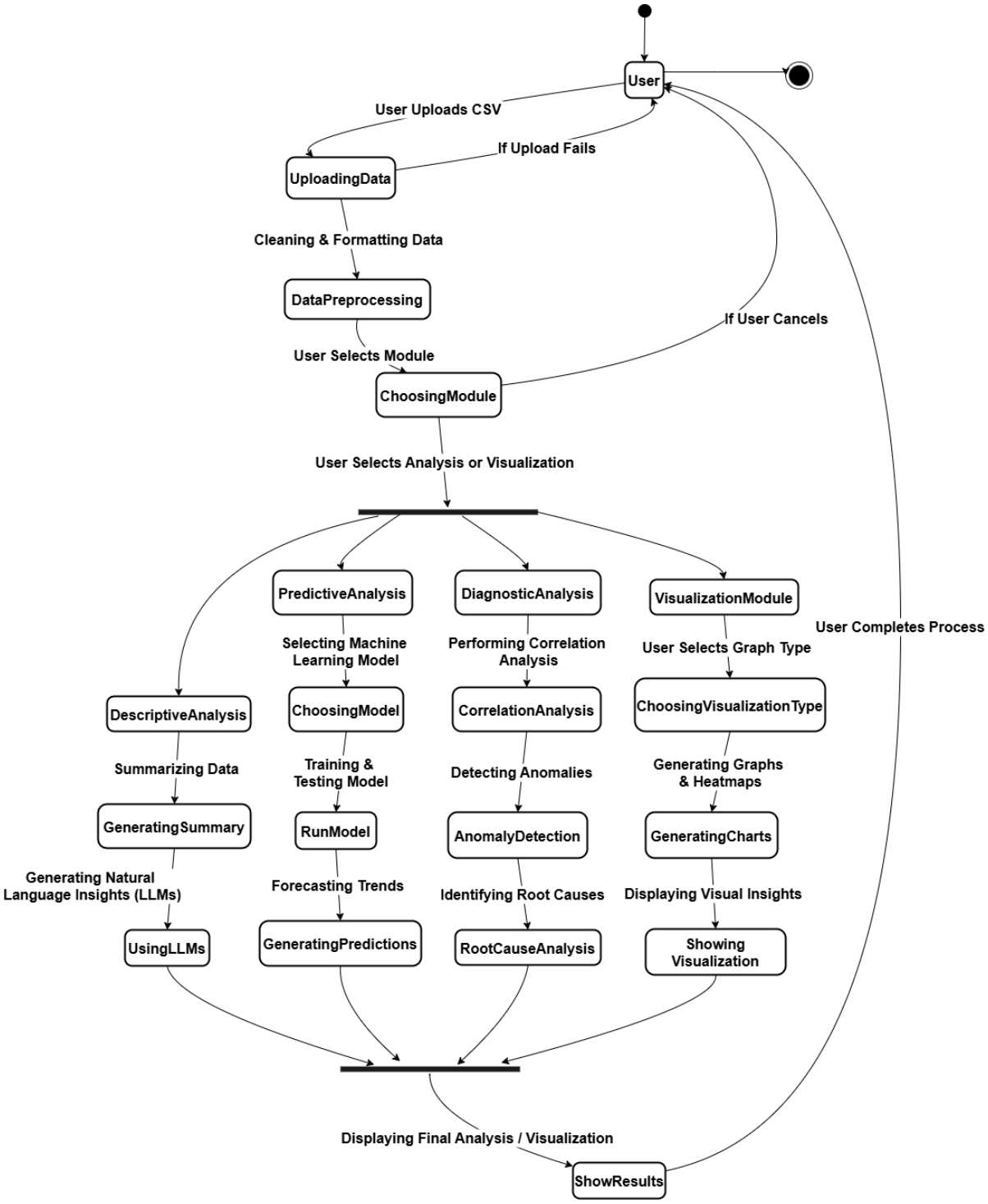


Figure 2.10: State Diagram for Analyza

Figure 2.10 illustrates the **State Diagram for Analyza**, which represents the various states the system transitions through based on user actions and system responses during the data analysis and visualization process.

The process begins when the User uploads a CSV file. If the upload is unsuccessful, the system may either prompt a retry or allow the user to cancel the process. Upon a successful upload, the system transitions into the Uploading Data state, followed by Data Preprocessing, where the uploaded file undergoes cleaning and formatting. Next, the Choosing Module state allows the user to select the desired module—either for analysis or visualization.

The process diverges into three parallel modules:

- **Predictive Analysis Path:** The user selects Predictive Analysis, chooses a machine learning model (Choosing Model), and proceeds with training and testing (Run Model). This leads to Generating Predictions and optionally Generating Summary or Using LLMs for descriptive insights.
- **Diagnostic Analysis Path:** Begins with Diagnostic Analysis, followed by Correlation Analysis, Anomaly Detection, and Root Cause Analysis. These stages help in identifying data relationships, detecting outliers, and determining underlying issues.
- **Visualization Module Path:** The user enters the Visualization Module, selects the graph type (Choosing Visualization Type), and proceeds with Generating Charts and Showing Visualization.

After any of the above paths, the system transitions to the Show Results state where the final output—either analysis or visualization—is displayed to the user.

This diagram also accounts for early exits, such as the user cancelling the operation, and loops back to the main state when the process completes successfully.

2.7.6 Sequence Diagram

Figure 2.11 illustrates the **Sequence Diagram for Analyza**, capturing the interaction between the user interface, backend modules, and external services. The diagram outlines the end-to-end flow for data upload, processing, analysis, and result delivery.

Initially, the User uploads a CSV file using the React Frontend, which transmits the file to the FastAPI Backend via the API Controller. The backend validates and parses the data using the Data Handler, followed by preprocessing through the Data Preprocessor.

Upon analysis type selection, the flow diverges into different modules:

- **Predictive Analysis:** The system trains a model and returns predictions.
- **Diagnostic Analysis:** Anomaly detection and root cause analysis are performed.
- **Descriptive (LLM) Analysis:** The backend requests a summary from the LLM Manager, which uses the Google Gemini API to fetch and return descriptive insights.
- **Visualization:** Visual outputs are generated and returned using the Visualization Module.

All generated results are forwarded back to the frontend to update the dashboard and display textual or visual insights to the user.

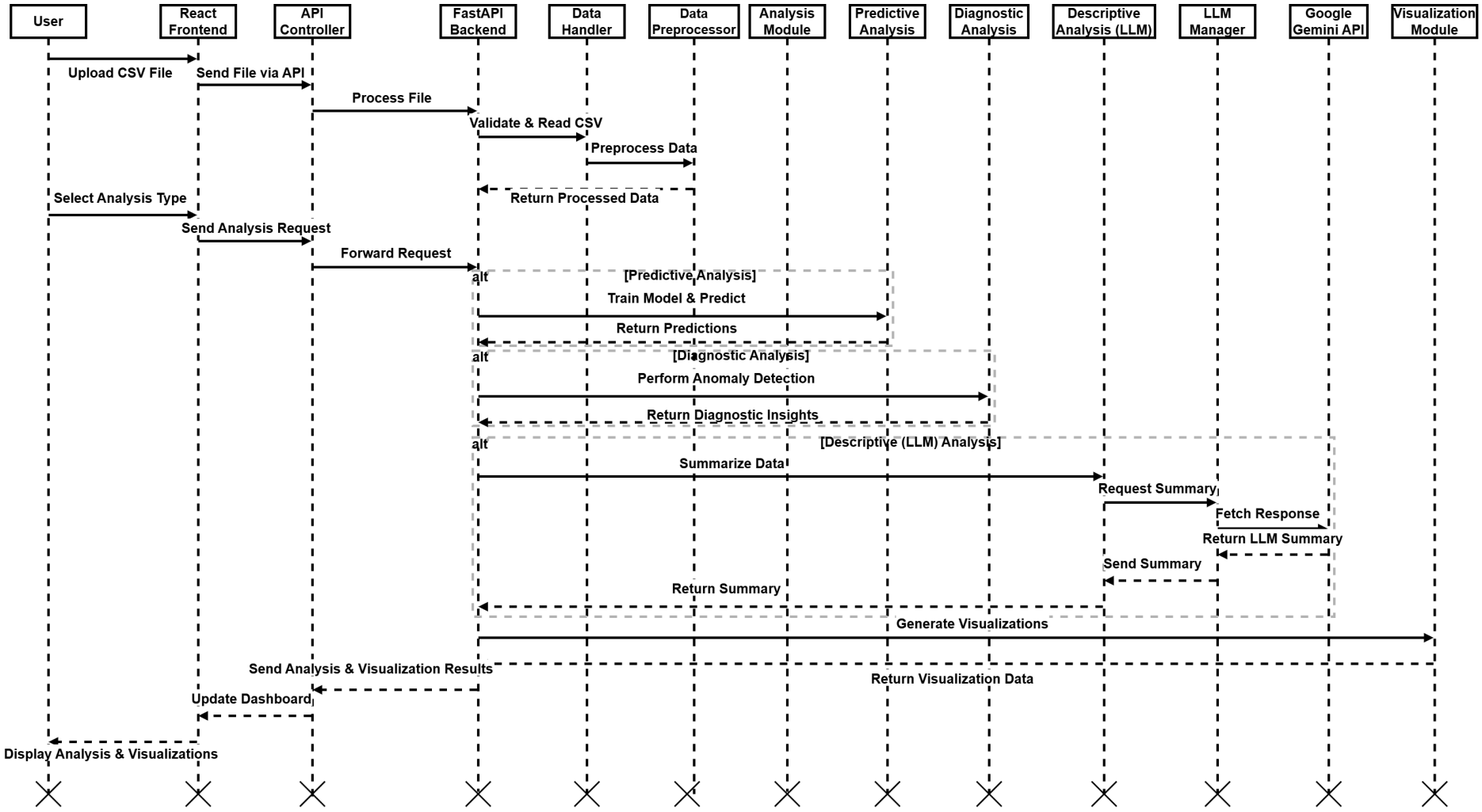


Figure 2.11: Sequence Diagram for Analyza

2.8 Assumptions and Dependencies

The design and functionality of Analyza rely on certain assumptions and dependencies that influence its development and operational feasibility. Assumptions define the expected conditions under which the system will function effectively, while dependencies highlight external factors and technologies required for smooth execution.

2.8.1 Assumptions

- Users will provide structured data in CSV format or connect to supported databases.
- The input data will be clean or pre-processed to some extent before analysis.
- Internet connectivity will be available for cloud-based processing and real-time analytics.
- Analyza will be used within ethical and legal data-sharing boundaries.
- Hardware resources will be sufficient for large-scale computations.

2.8.2 Dependencies

- **Programming Environment**: Analyza relies on Python for machine learning and statistical processing
- **Libraries & Frameworks**: Uses libraries like Pandas, NumPy, Matplotlib, Scikit-learn, and LLM APIs for data analysis and visualization.
- **Machine Learning Models**: External ML models and LLMs are integrated for predictive and diagnostic insights
- **Cloud & Hosting Services**: Requires cloud infrastructure or a local server for large-scale processing and multi-user access.
- **Regulatory Compliance**: Compliance with data privacy regulations (GDPR, HIPAA) may be required depending on the domain.

Chapter 3

System Requirements

3.1 User Interface

Analyza's user interface is designed with simplicity and functionality in mind, ensuring both technical and non-technical users can interact with the platform effortlessly. The UI is broken into distinct modules, each tailored to perform a specific function, such as analysis, visualization, or documentation reference. This modular approach ensures smooth navigation and an optimal user experience across different workflows.

3.1.1 Dashboard (Main Screen)

The Dashboard serves as the primary entry point into Analyza and provides an overview of the platform's capabilities. It introduces the user to Analyza's vision and gives direct access to its major features. The interface is clean, engaging, and highlights the tool's core functionalities upfront.

Components

- Top Navigation Bar: Offers seamless navigation across Analyza's main functionalities:

    1. Home (Default View) – Displays the tool introduction and highlights core features.
    2. LLM Analysis – Leverages LLMs for data-driven summaries.
    3. Custom Analysis – Enables predictive and diagnostic data modeling.
    4. Custom Visualization – For building tailored, dynamic charts.
    5. Documentation – Contains user guides and technical references.
    6. About – Information about the project, purpose, and development team.

3.1.2 LLM Analysis Interface

This section leverages LLMs to generate automatic insights and summaries from the uploaded dataset.

Components

- Data Upload Panel: Users can upload only CSV files for analysis.
- Preview Table: Displays the uploaded dataset's first few rows.
- LLM Analysis Button: Generates a textual summary and key observations.

3.1.3 Custom Analysis Interface

Allows users to apply Predictive and Diagnostic analysis (Accessible via Top Tab in Dashboard).

Components

- Analysis Type Selection: Dropdown for selecting predictive or diagnostic analysis.
- Data Upload Panel: Users can upload only CSV files for analysis.
- Target, Features & Model selection in predictive analysis dashboard
- Target variable selection (optional) for diagnostic analysis dashboard
- Run Analysis Button: Executes the chosen analysis based on conditions.
- Results Panel: Displays trends, patterns, and insights derived from the conditions.

3.1.4 Custom Visualization Interface

Custom Visualization Interface (Accessible via Top Tab in Dashboard)

Components

- Chart Type Selection: Drop down menu with options like bar chart, scatter plot, heatmap, etc.
- Axis Selection Fields: Users define the X and Y axes for the chart.
- Color Coding Options: Adds colors to highlight key data aspects.
- Generate Visualization Button: Creates and displays the chart dynamically.

3.2 Software Interface

Analyza interacts with multiple software components, including backend services, databases, APIs, and third-party libraries, to provide a seamless and efficient user experience. This section outlines the key software interactions that enable the platform's core functionalities.

3.2.1 Frontend-Backend Communication

The frontend, built using React.js, communicates with the backend, developed using FastAPI, through RESTful APIs. This communication facilitates the processing of user inputs, retrieval of data, and execution of analytical tasks.

Interfaces & Interactions:

- **File Upload**: Processes CSV file uploads from users and prepares the data for analysis. This interface ensures that uploaded files are validated and stored securely for further processing.
- **Analysis** – Executes LLM-based descriptive analysis based on user queries, generating insights and summaries. This interface allows users to interact with the platform using natural language queries and receive meaningful insights. Additionally, Analyza supports custom predictive and diagnostic analysis, enabling users to select features and target variables to train machine learning models for forecasting trends, detecting anomalies, and uncovering key relationships within their data.
- **Visualization**: Generates and retrieves custom visualizations based on user inputs, ensuring an interactive and intuitive experience. This interface enables users to explore data through charts, graphs, and other visual representations.

3.2.2 LLM Integration

The system integrates a (LLM) to provide automated data insights, enabling users to interact with the platform using natural language queries. This integration enhances the platform's usability by making complex data analysis accessible to non-technical users.

Interfaces & Interactions:

- **LLM Query API**: Sends structured queries derived from user inputs to the LLM for processing, ensuring accurate and relevant responses. This interface translates user queries into a format that the LLM can understand and process.
- **Response Processing Module**: Extracts key insights, generates summaries, and creates visualizations based on the LLM's output. This interface ensures that the LLM's responses are presented in a user-friendly and actionable format.

3.2.3 Data Processing & Machine Learning

Analyza integrates machine learning models for predictive and diagnostic analysis using libraries such as Scikit-learn. These models enable users to forecast trends, classify data, and identify patterns in their datasets.

Interfaces & Interactions:

- **Predictive Analysis Engine**: Runs regression and classification models based on user-selected features. This interface allows users to predict outcomes such as sales, customer behavior, or binary classifications (e.g., yes/no).

- **Diagnostic Analysis Engine**: Detects trends, anomalies, and patterns in data using statistical methods. This interface helps users understand the underlying causes of observed data patterns and identify outliers or unusual behavior.

3.2.4 Visualization & Reporting

The system provides interactive data visualizations using libraries such as Matplotlib, Seaborn, and NumPy. These visualizations help users explore and interpret their data effectively.

Interfaces & Interactions:

- **Dashboard Integration**: Ensures seamless embedding of generated visuals into the user dashboard. This interface allows users to view and interact with visualizations directly within the platform.
- **Dynamic Visualization Generation**: Generates and renders visualizations dynamically based on user inputs. This interface ensures that visualizations are updated in real-time as users modify their data or analysis parameters.

3.3 Protocols

Analyza follows modern communication protocols to ensure secure, efficient, and reliable interaction between system components. These protocols govern data exchange, authentication, and system security, ensuring a seamless user experience while maintaining data integrity and confidentiality.

3.3.1 Communication Protocols

These protocols define how data is transmitted between the frontend (React.js), backend (FastAPI) to ensure reliable performance and security.

HTTP/HTTPS (Hypertext Transfer Protocol Secure)

- **Purpose**: Used for communication between the frontend and backend.
- **HTTPS Encryption**: Encrypts all transmitted data, preventing unauthorized access and ensuring data privacy.
- **Security Benefits**: Protects against man-in-the-middle attacks and eavesdropping.
- **Implementation**: All API endpoints utilize HTTPS to maintain data security during transmission.

RESTful API Architecture

- **Design**: The backend follows a RESTful API design, providing standardized methods such as GET, POST, PUT, and DELETE.
- **Scalability**: Ensures scalability by enabling easy integration with various clients and systems.

- **Stateless Communication**: Simplifies data retrieval and processing by maintaining stateless communication, where each request is independent and self-contained.

3.3.2 Real-Time Data & Future Enhancements

To improve real-time interactivity and enhance user experience, Analyza considers implementing additional protocols.

WebSockets (Future Consideration for Real-Time Updates)

- **Purpose**: Enables real-time data streaming between the frontend and backend.
- **Use Cases**: Can be used for live notifications and dynamic updates in dashboards.
- **Advantages**:
  - Provides a persistent connection, reducing request-response overhead compared to traditional HTTP.
  - Enables real-time progress tracking for data analysis tasks.
- **Future Implementation**: Future versions of Analyza may include WebSocket support for enhanced real-time interactivity.

Chapter 4

Non-functional Requirements

4.1 Performance Requirements

Analyza is designed to deliver efficient and responsive performance, ensuring that users can analyze and visualize data without significant delays. The system is optimized to handle large datasets and provide timely results, even for complex analytical tasks. Below are the detailed performance requirements for the platform.

4.1.1 Dataset Size Handling

Analyza is capable of handling datasets of up to 200MB in size. This ensures that users can work with moderately large datasets without encountering performance bottlenecks. The system is optimized to process structured data formats such as CSV files efficiently, enabling users to upload and analyze their data seamlessly.

4.1.2 Response Time

The system is designed to provide a response time of under 10 seconds for most analytical operations. This includes tasks such as:

- **Data Upload and Preprocessing**: The system should validate, clean, and prepare the dataset for analysis within the specified time frame.
- **Descriptive Analysis**: Generating summaries, statistics, and insights using LLM-based analysis should be completed within 10 seconds.
- **Predictive Analysis** – Supports a wide range of machine learning algorithms including Linear Regression, Polynomial Regression, Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, AdaBoost, Gradient Boosting, and XGBoost. These models are optimized to deliver accurate and timely predictions within the specified response time, enabling users to forecast outcomes and trends effectively.
- **Diagnostic Analysis**: Identifying trends, anomalies, and correlations in the data should be completed promptly.
- **Visualization Generation**: Creating charts, graphs, and other visual representations of the data should be performed efficiently to ensure a smooth user experience.

4.1.3 Optimization Techniques

To meet these performance requirements, Analyza employs several optimization techniques:

- **Data Preprocessing**: Leveraging vectorized operations (e.g., Pandas/Numpy) to reduce time complexity.
- **Analysis Models**: Machine learning and pattern detection models are carefully selected based on dataset type and size to ensure minimal computation overhead.
- **Visualizations**: Lazy rendering and optimized charting libraries (e.g., Altair, Pydeck) are used to efficiently handle large visual payloads.

4.1.4 Scalability

Analyza is designed to scale efficiently with increasing workloads. The system can handle multiple users and datasets simultaneously without significant performance degradation. This scalability ensures that the platform remains responsive and reliable, even during peak usage periods.

4.2 Security Requirements

Analyza prioritizes the security and privacy of user data, ensuring that sensitive information is protected throughout the data analysis process. The system is designed to minimize data retention and prevent unauthorized access, providing users with a secure and trustworthy platform.

4.2.1 Data Storage Policy

To ensure user privacy and data security, Analyza adheres to a strict data storage policy:

- **No Permanent Storage**: User-uploaded data is not stored permanently on the server. Once the analysis is complete, the data is discarded to prevent unauthorized access or misuse.
- **In-Memory Processing**: All data is processed in memory, ensuring that it is not written to disk or stored in persistent storage. This approach minimizes the risk of data breaches or leaks.
- **Data Retention**: Temporary data is retained only for the duration of the analysis session. After the session ends, all data is automatically deleted from the system.

4.2.2 Data Encryption

To further enhance security, Analyza employs robust encryption techniques:

In-Transit Encryption: All data transmitted between the frontend and backend is encrypted using HTTPS (Hypertext Transfer Protocol Secure). This ensures that data cannot be intercepted or tampered with during transmission.

4.2.3 Data Integrity and Validation

To ensure the integrity and accuracy of user data, Analyza implements the following measures:

- **Data Validation**: All uploaded data is validated to ensure it meets the required format and structure. Invalid or corrupted files are rejected to prevent processing errors.
- **Error Handling**: The system includes robust error handling mechanisms to detect and respond to potential security threats, such as invalid inputs or unauthorized access attempts.

4.3 Software Quality Attributes

Analyza is designed to meet high standards of software quality, ensuring that the platform is reliable, usable, and maintainable. These attributes are critical to delivering a robust and user-friendly data analysis tool that meets the needs of diverse users.

4.3.1 Reliability

Analyza is built to provide accurate and consistent analysis results, ensuring that users can trust the insights generated by the platform. The system incorporates the following measures to ensure reliability:

- **Data Validation**: All uploaded datasets are validated to ensure they meet the required format and structure. Invalid or corrupted files are rejected to prevent errors during analysis.
- **Robust Algorithms**: The platform uses well-tested and optimized algorithms for data preprocessing, analysis, and visualization, ensuring accurate results.
- **Error Handling**: The system includes comprehensive error handling mechanisms to detect and respond to issues during data processing or analysis, minimizing the risk of incorrect results.
- **Testing and Validation**: Analyza undergoes rigorous testing, including unit tests, integration tests, and user acceptance tests, to ensure the accuracy and reliability of its features.

4.3.2 Usability

Analyza is designed to be user-friendly, catering to both technical and non-technical users. The platform provides an intuitive interface and features that simplify data analysis:

- **Intuitive Interface**: The user interface is designed to be clean, simple, and easy to navigate, allowing users to perform tasks with minimal effort.
- **Guided Workflows**: The platform includes guided workflows and tooltips to assist users in uploading data, performing analysis, and generating visualizations.

- **Natural Language Queries**: Users can interact with the system using natural language queries, making it accessible to those without technical expertise.
- **Customizable Visualizations**: Users can create and customize charts and graphs to suit their needs, enhancing the usability of the platform.

4.3.3 Maintainability

Analyza is designed to be easy to maintain and update, ensuring that the platform can evolve with user needs and technological advancements. The following practices are implemented to ensure maintainability:

- **Modular Codebase**: The system is built using a modular architecture, with clearly defined components and interfaces. This makes it easier to update or replace individual modules without affecting the entire system.
- **Well-Documented Code**: The codebase is thoroughly documented, with clear explanations of functions, modules, and workflows. This ensures that developers can easily understand and modify the code.
- **Version Control**: The platform uses version control systems (e.g., Git) to track changes, manage updates, and collaborate on development.
- **Automated Testing**: Automated tests are integrated into the development process to ensure that updates do not introduce new bugs or issues.

Chapter 5

Other Requirements

Analyza is designed to be a robust, efficient, and platform-independent web-based data analytics tool. To ensure a smooth and high-performance experience for all users, this chapter outlines the technical, compatibility, and operational requirements for both client-side and server-side environments. These requirements help guarantee the platform functions reliably across devices and browsers, delivering a seamless experience for data-driven decision-making.

5.1 Browser Compatibility

Analyza supports full functionality across modern, widely-used web browsers. This ensures accessibility, responsiveness, and consistent user experience regardless of browser preference.

Google Chrome

Optimized for Chrome with fast rendering, low latency, and minimal resource consumption. It supports all visualization modules, interactive elements, and ML-driven insights with full stability.

Arc

Designed to work smoothly on Arc, delivering seamless transitions, fast data uploads, and responsive interactions. All visual components render correctly with maintained performance and usability.

Microsoft Edge

Fully functional on Microsoft Edge, ensuring users can access every feature including file uploads, real-time visualizations, and reports. Performance and responsiveness match those on other modern browsers.

5.1.1 Cross-Browser Testing

To ensure consistent user experience, the platform undergoes a detailed browser testing process. This testing guarantees reliability, correct rendering, and smooth functionality

regardless of the browser in use.

Functional Testing

Validates the complete functionality of all modules such as descriptive insights, charting tools, and predictive analysis. It ensures consistent behavior of features including drag-and-drop upload, visualization interactivity, and report exports across all supported browsers.

Performance Testing

Measures load times, CPU and memory usage, and responsiveness under typical and heavy usage. It ensures the system remains fast and efficient even when handling large datasets or concurrent users.

5.2 System Requirements

To provide an optimized experience, Analyza requires specific client-side and server-side system configurations. These requirements are kept minimal to maintain broad accessibility while supporting advanced processing capabilities.

5.2.1 Client-Side Requirements

These requirements ensure the user's device can fully interact with the platform without performance issues or feature limitations.

- **Modern HTML5-compliant browser**
  Required for rendering dynamic content, handling JavaScript-based interactions, and displaying responsive visualizations correctly. Ensures compatibility with React-based components and modern data processing workflows.
- **Minimum 2 GB RAM and dual-core processor**
  Ensures that front-end operations such as chart rendering, file uploads, and real-time data navigation run without lag or crashes. Prevents memory overflows and supports smooth user interface performance.
- **Internet connection with at least 2 Mbps bandwidth**
  Necessary for real-time interactions, uploading datasets, and receiving model outputs and charts. Ensures low-latency communication with the server for a responsive analytics experience.

5.2.2 Server-Side Setup

The server-side setup defines the core infrastructure needed to run and scale Analyza's services, especially those involving ML processing and visualization generation.

- **Backend: FastAPI with Uvicorn**
  FastAPI provides a high-performance, asynchronous web framework that is ideal for handling RESTful API requests efficiently. Uvicorn serves as the ASGI server, enabling fast execution of ML tasks and data analysis routines.

- **Frontend: React.js**
  Enables creation of a highly interactive and modular UI. React helps manage component states effectively, ensuring that charts update dynamically, filters work instantly, and the UI remains responsive.

- **Core Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn**
  These libraries are essential for numerical computation, data manipulation, statistical analysis, and machine learning. They power Analyza's backend analytical engine for both basic and advanced data operations.

- **Language Models: Integration with LLMs (e.g., Gemini API) for descriptive insights**
  Enables users to receive natural-language interpretations of their data. This integration helps generate textual insights and summaries, providing clarity and context to raw numerical outputs.

- **Deployment: Supports both local and cloud hosting using a lightweight and scalable setup**
  The backend architecture supports Docker-based deployment, container orchestration (e.g., Kubernetes), and virtual machine-based deployment. This flexibility allows institutions to run Analyza in private environments or scale it in cloud-based ecosystems with minimal configuration.

5.3 Security and Performance Considerations

Analyza incorporates robust performance tuning and security protocols to protect user data, maintain system integrity, and ensure responsive interaction across all modules.

- **HTTPS communication for all client-server data exchanges**
  All communication between users and the server is secured via encrypted HTTPS protocols, preventing interception of sensitive data such as uploaded datasets and generated results.

- **Efficient memory management during data analysis and model execution**
  Backend processes handle data in chunks or batches when necessary and use memory-efficient structures to avoid overload, especially during machine learning or visualization operations.

- **In-memory processing of uploaded datasets for fast analysis and low latency**
  Uploaded files are temporarily processed in RAM to eliminate disk latency, enabling real-time insights and near-instant results without saving intermediate states unless explicitly requested.

- **Response time optimization using asynchronous API calls and ML model tuning**
  FastAPI's async features are leveraged to process multiple requests in parallel. Machine learning models are optimized and pre-loaded where applicable, ensuring rapid generation of outputs with minimal delay.

Chapter 6

Project Progress and Implementation Status

We have successfully implemented key components of our project, focusing on various types of data analysis and visualization. Our descriptive analysis is powered by the Gemini API, which enables LLM-based insights from data using natural language processing. Additionally, we have developed a custom analysis module that provides users with flexible data exploration options.

For predictive analysis, we have integrated Linear Regression,Polynomial Regression, Random Forest, XG boost, Gradient Boost, Adaboost, KNN, Naive Bayes and Logistic Regression models to generate accurate forecasts based on user datasets. In diagnostic analysis, we have implemented correlation analysis, anomaly detection, and root cause analysis to help users identify patterns, inconsistencies, and key influencing factors. Furthermore, our custom visualization module allows users to upload datasets and generate various interactive charts, such as bar graphs and scatter plots, with customizable attributes. These features together create a powerful, user-friendly, and interactive data analysis experience.

6.1 Landing Page

The landing page of our website, Analyza, showcases a powerful data analytics platform offering descriptive (LLM-based), predictive, and diagnostic analysis, along with custom visualizations. As shown in Figure 6.1 and Figure 6.2, it emphasizes transforming raw data into actionable insights through interactive modules, with a clean and modern design that invites users to get started easily.

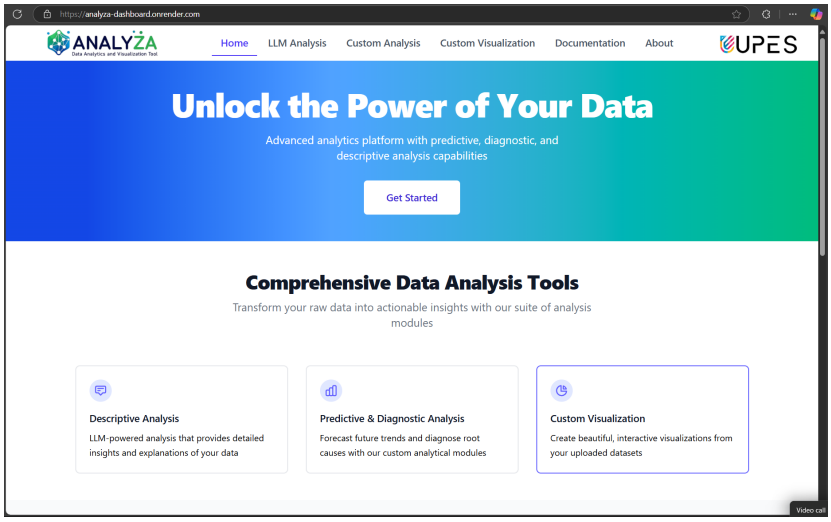


Figure 6.1: Landing Page Top View

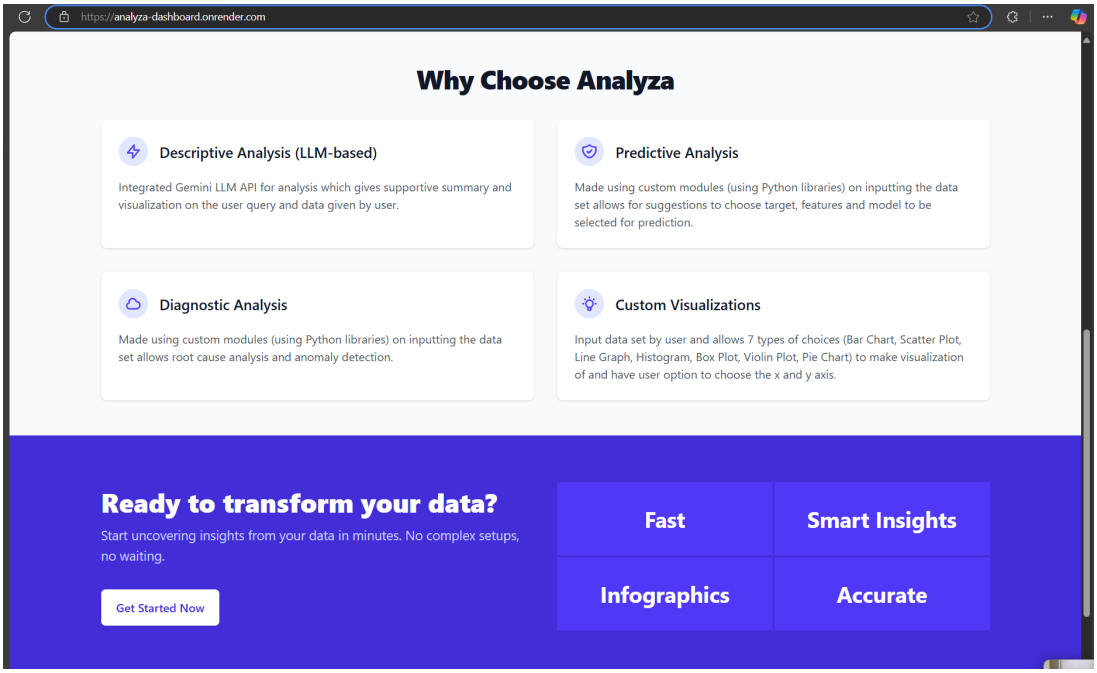


Figure 6.2: Landing Page Bottom View

6.2 LLM Module

The LLM Analysis Module in Analyza uses Gemini Flash 2.0 Model to generate automated insights and summaries from user-uploaded datasets. It provides a natural language interface for users to query their data and receive meaningful insights along with relevant visualizations.

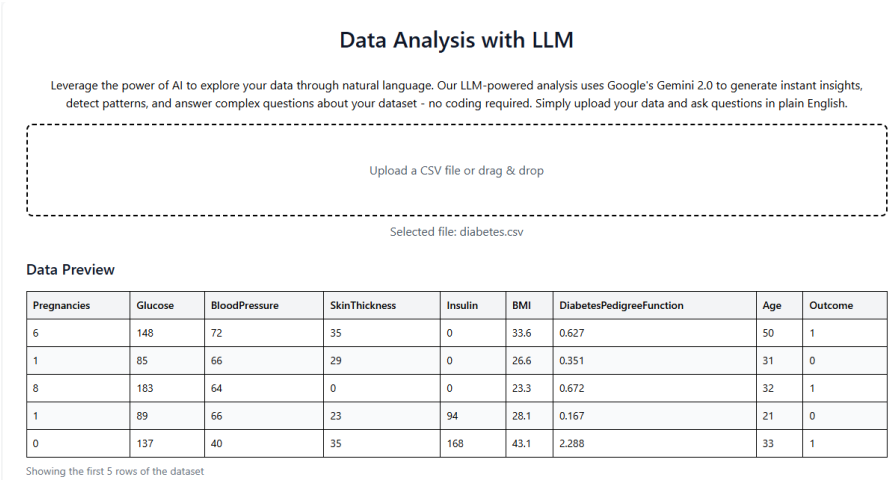


Figure 6.3: Step 1: Uploading dataset and preview

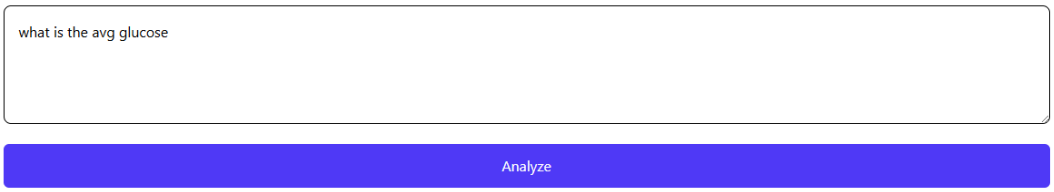


Figure 6.4: Step 2: Submitting a natural language query for analysis

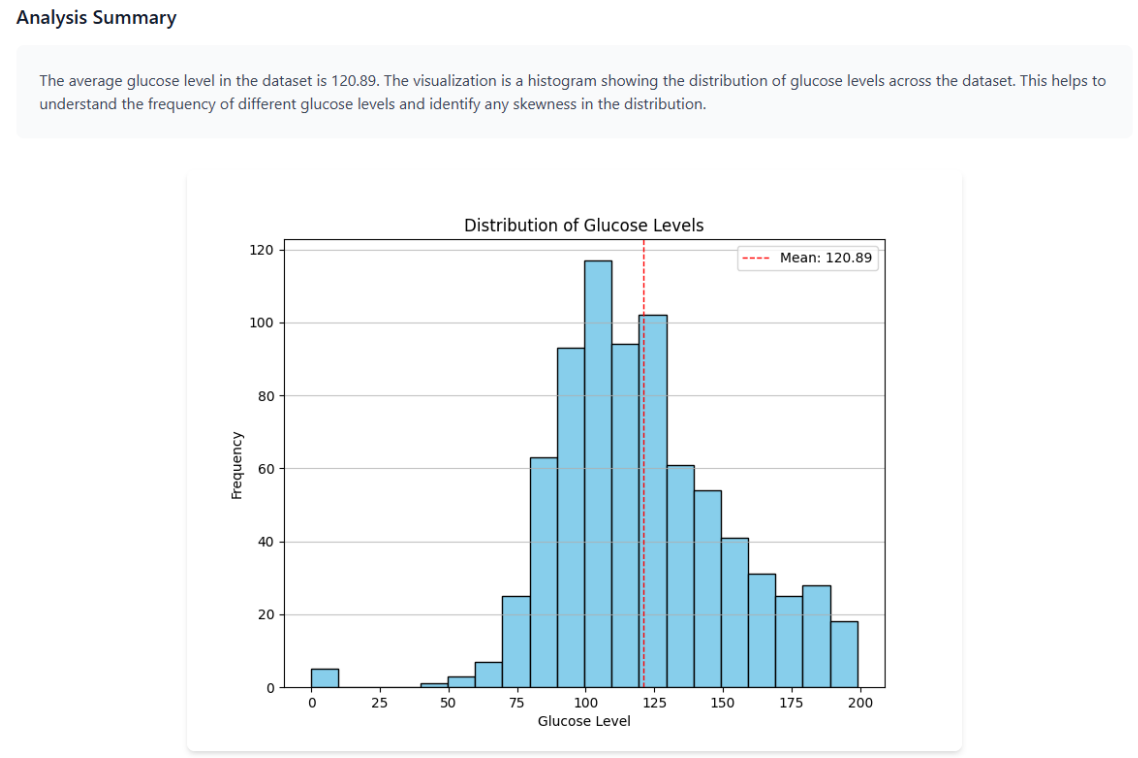


Figure 6.5: Step 3: Viewing predicted results and insights with visualization

The LLM-powered analysis module in Analyza, enhanced with Google's Gemini 2.0, enables deep and intuitive interaction with data, as demonstrated in Figures 6.3 through 6.5. This module consists of the following three key stages:

- **File Upload and Preview (Figure 6.3):** Users can upload a CSV file via drag-and-drop or file selection. Once uploaded (e.g., `diabetes.csv`), the dashboard shows a preview of the dataset, including key columns such as Pregnancies, Glucose, and BloodPressure.
- **Natural Language Query Input (Figure 6.4):** A plain text input field allows users to type queries like "what is the avg glucose". The LLM can interpret queries even with grammatical errors, showcasing its intelligent language processing.
- **Insight Summary and Visualization (Figure 6.5):** After processing the query, the system provides a statistical summary—such as the average glucose level (e.g., 120.89)—and generates a corresponding histogram. The visualization includes annotations (like the mean line) to aid in pattern recognition and interpretation.

6.3 Custom Analysis Module

The Custom Analysis Module in Analyza enables users to perform both predictive and diagnostic analysis on their datasets. In predictive analysis, users can choose from Linear Regression, Logistic Regression, Polynomial Regression, Ada Boosting, Gradient Boosting , XG Boosting, KNN(K- Nearest Neighbors), Naive Bayes and Random Forest to forecast trends and outcomes. Diagnostic analysis provides tools like the correlation matrix, anomaly detection, and root cause analysis to uncover key relationships and insights. Users can specify target variables and features, ensuring a tailored approach to data exploration.

6.3.1 Predictive Analysis

Predictive analysis involves applying machine learning techniques to forecast outcomes based on existing data. The following steps describe the generalized workflow implemented in the system, supported by AI-driven recommendations and visual dashboards.

Step 1: Select Analysis Type

Users choose between:

- **Predictive Analysis:** For forecasting outcomes using machine learning.
- **Diagnostic Analysis:** For identifying correlations, patterns, or anomalies in the data.

Step 2: Upload Dataset

Users begin by uploading a dataset in `.csv` format. Upon successful upload, the system previews the first five rows for verification (Figure 6.6).

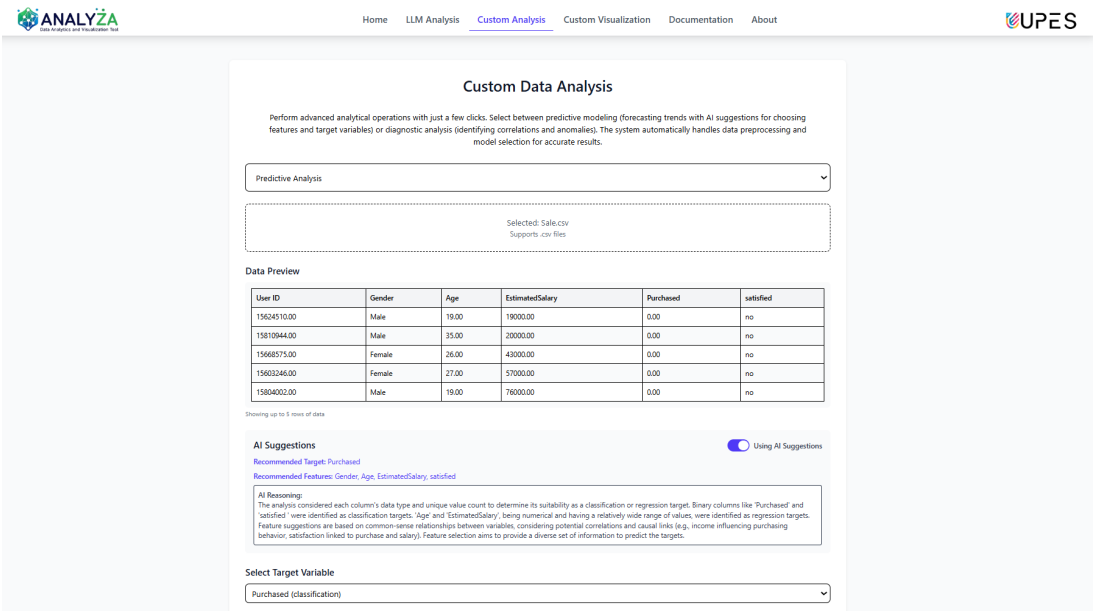


Figure 6.6: Predictive Analysis Dashboard: Uploading data and applying AI suggestions for features and target variable

Step 3: Choose Target Variable

The system provides:

- **AI Suggestion:** Automatically selects an appropriate target based on data type and distribution.
- **Manual Selection:** Users can manually override the suggestion.

Step 4: Select Input Features

Relevant features are identified using statistical methods like correlation analysis. Users can accept AI-recommended features or choose them manually (Figure 6.6).

Step 5: Select Machine Learning Model

The system suggests models based on the type of problem (classification or regression).

Classification options include:

- XGBoost (Figure 6.7)
- Gradient Boosting (Figure 6.8)
- AdaBoost (Figure 6.9)
- Naive Bayes (Figure 6.10)
- K-Nearest Neighbors (KNN) (Figure 6.11)
- Random Forest (Figure 6.12)

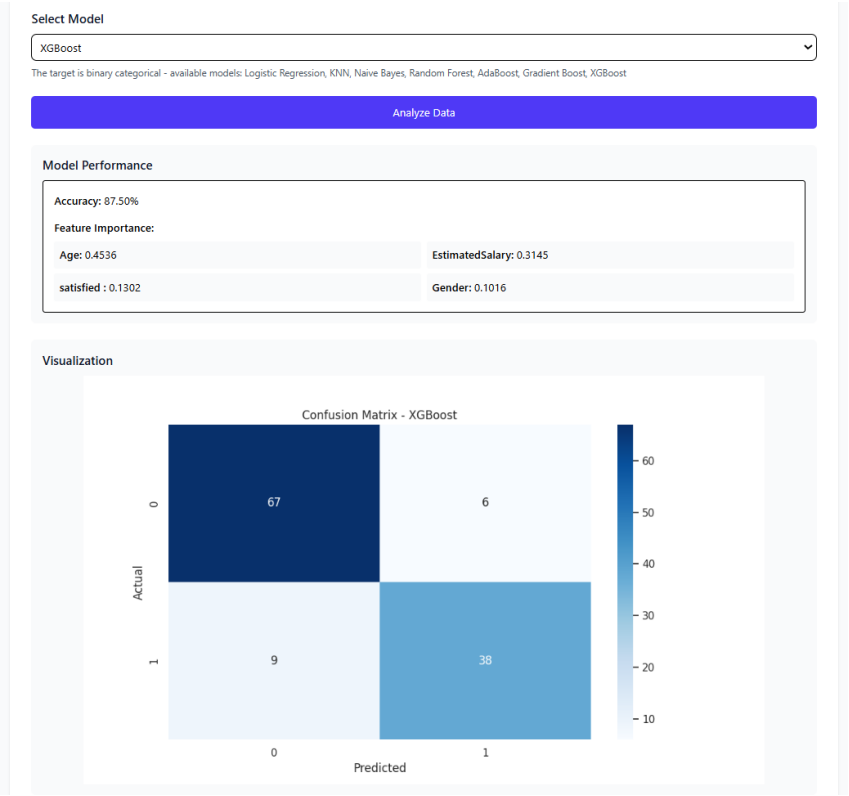


Figure 6.7: Model Selection & Model Performance: XGBoost

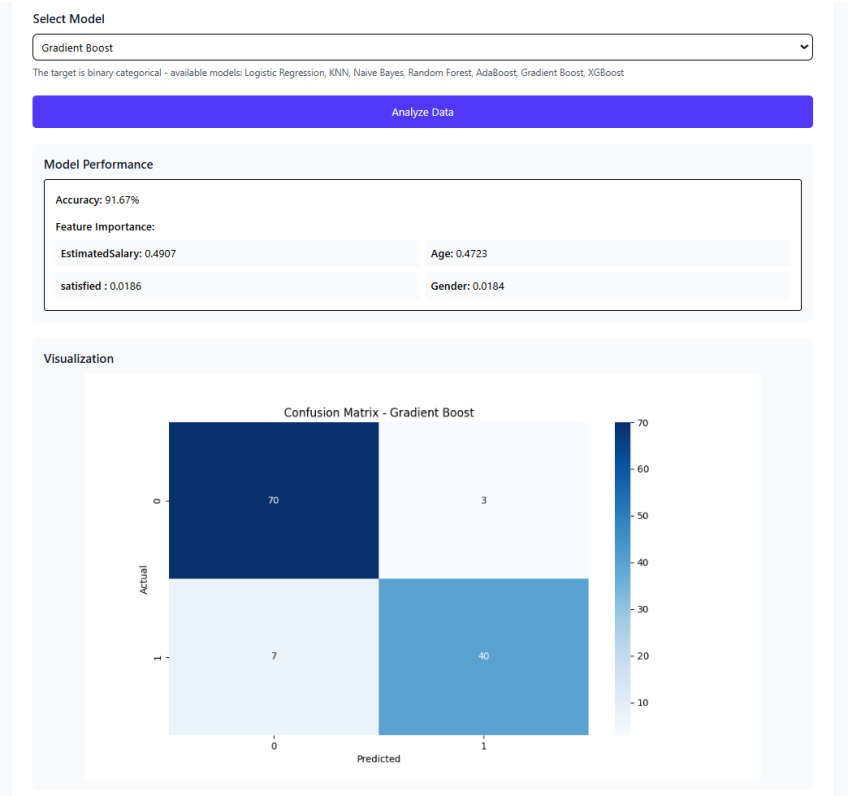


Figure 6.8: Model Selection & Model Performance: Gradient Boost

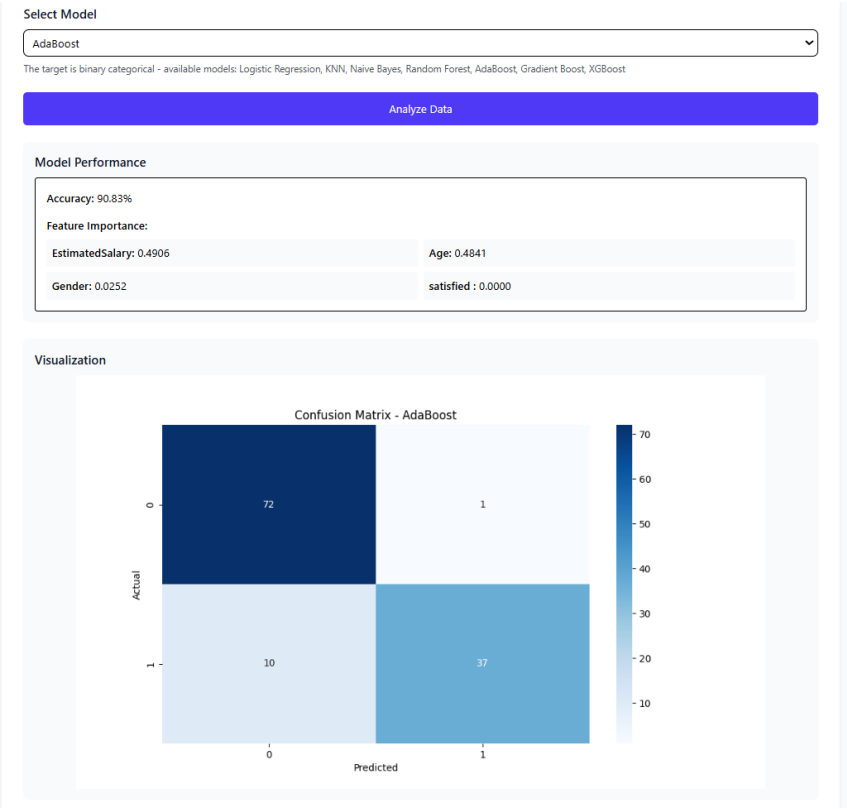


Figure 6.9: Model Selection & Model Performance: AdaBoost

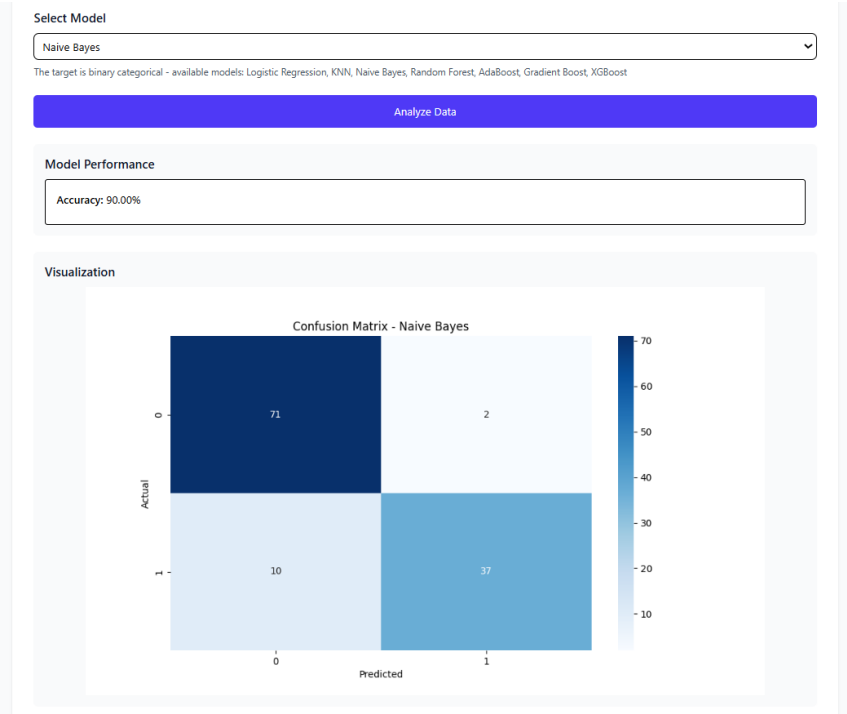


Figure 6.10: Model Selection & Model Performance: Naive Bayes

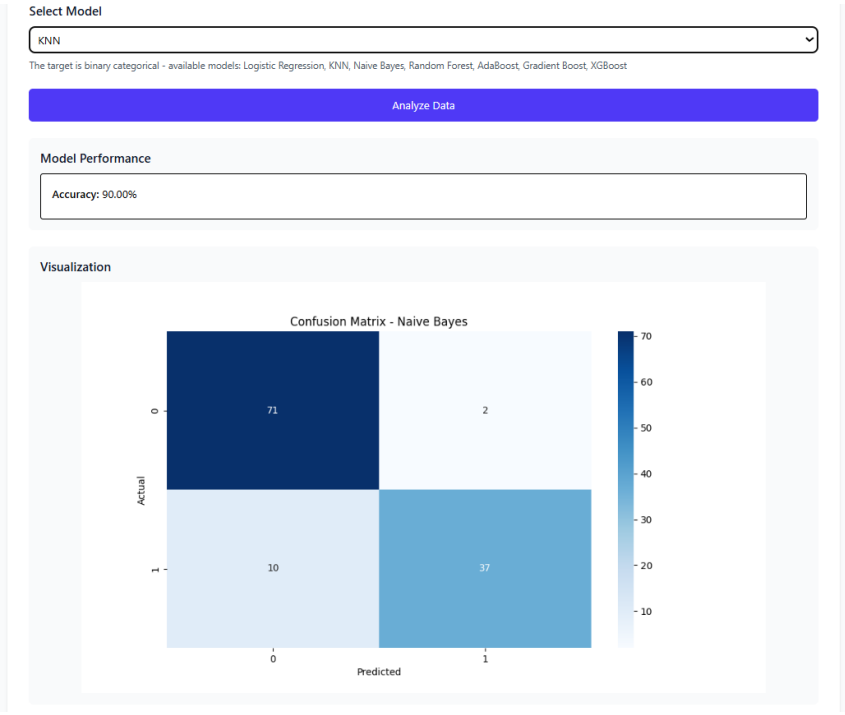


Figure 6.11: Model Selection & Model Performance: KNN

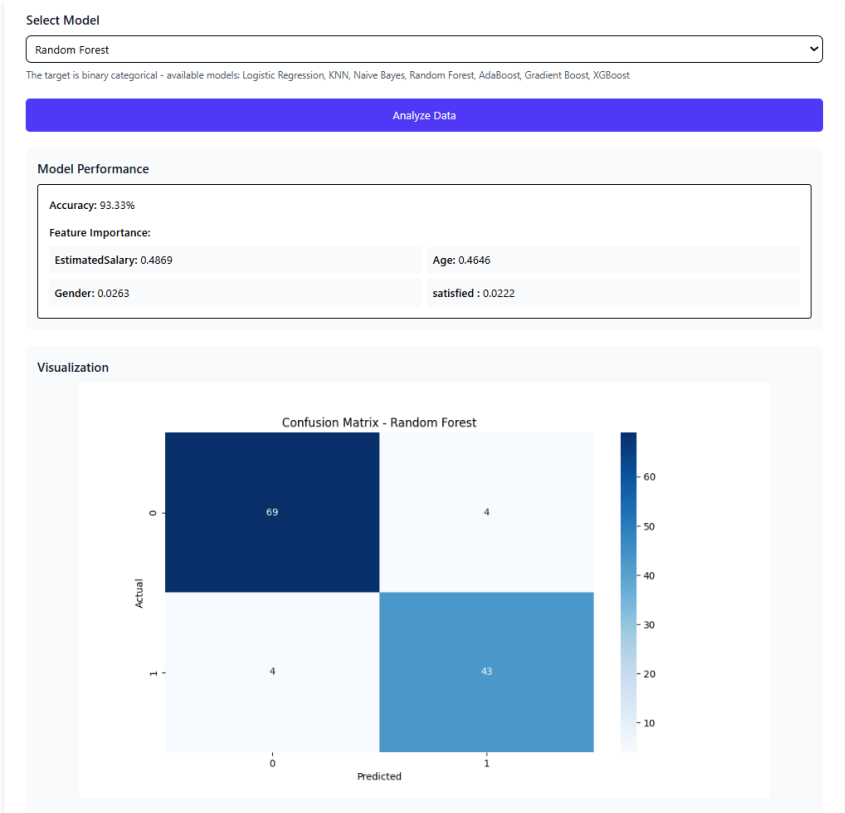


Figure 6.12: Model Selection & Model Performance: Random Forest

Regression options include:

- Linear Regression (Figure 6.13)
- Polynomial Regression (Figure 6.14)
- K-Nearest Neighbors (KNN) Regression (Figure 6.15)
- Random Forest Regression (Figure 6.16)
- AdaBoost Regression (Figure 6.17)
- Gradient Boosting Regression (Figure 6.18)
- XGBoost Regression (Figure 6.19)

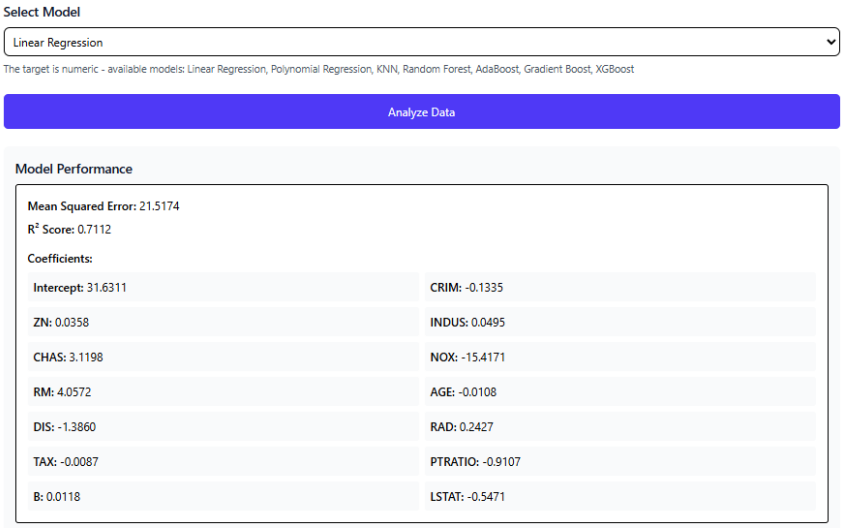


Figure 6.13: Model Selection & Model Performance: Linear Regression

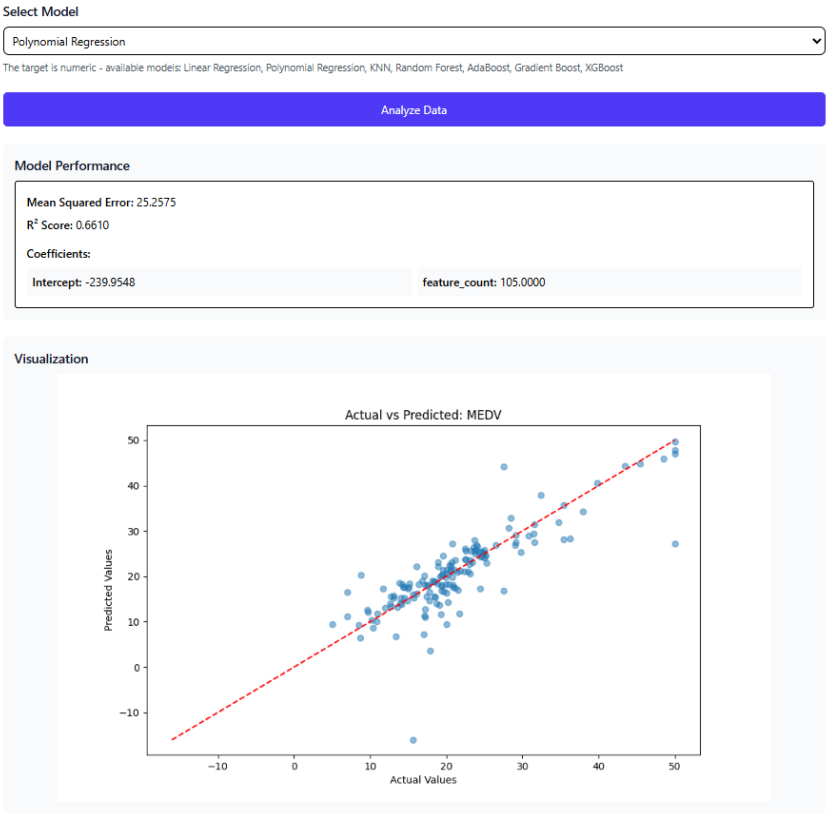


Figure 6.14: Model Selection & Model Performance: Polynomial Regression

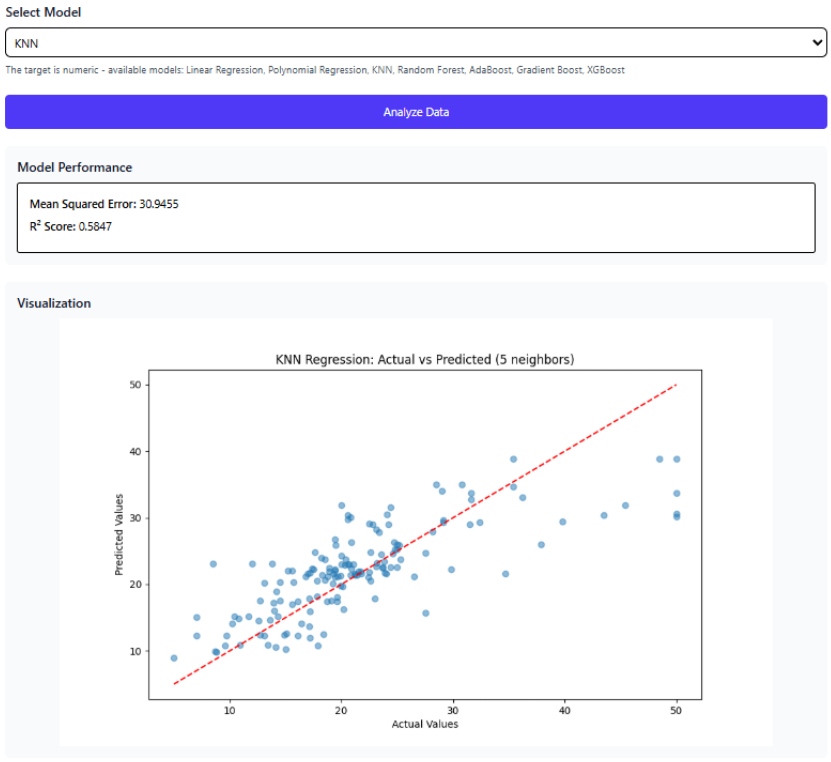


Figure 6.15: Model Selection & Model Performance: KNN Regression

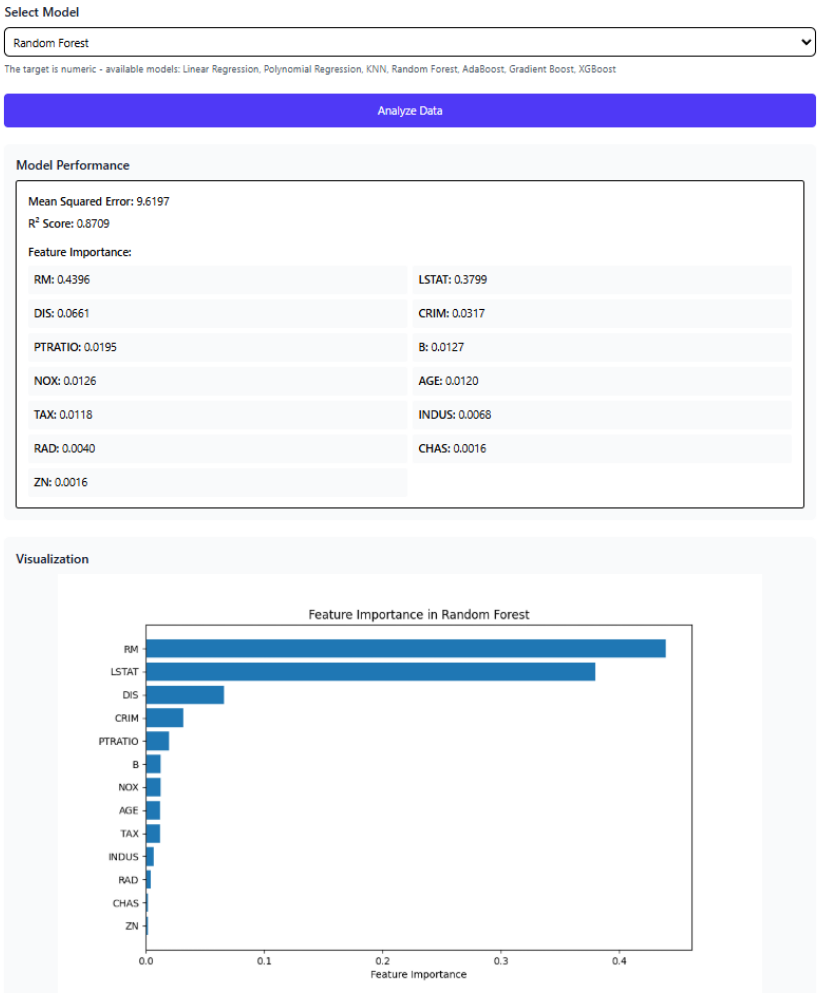


Figure 6.16: Model Selection & Model Performance: Random Forest Regression

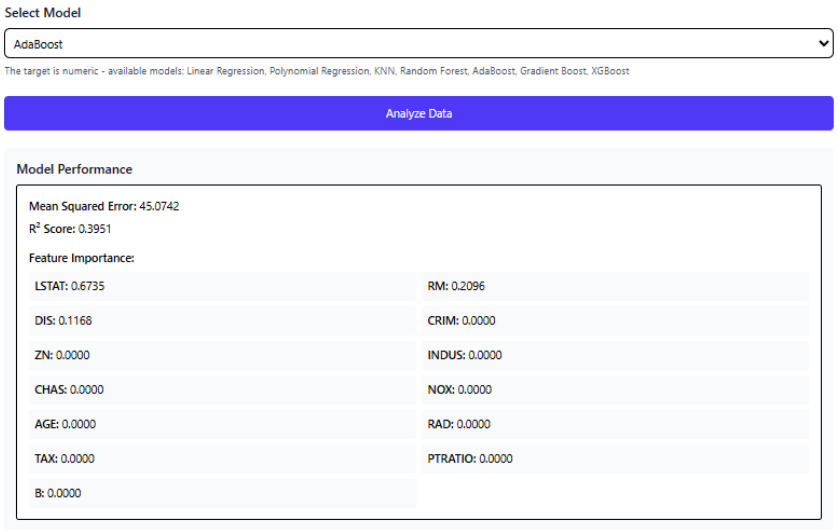


Figure 6.17: Model Selection & Model Performance: AdaBoost Regression

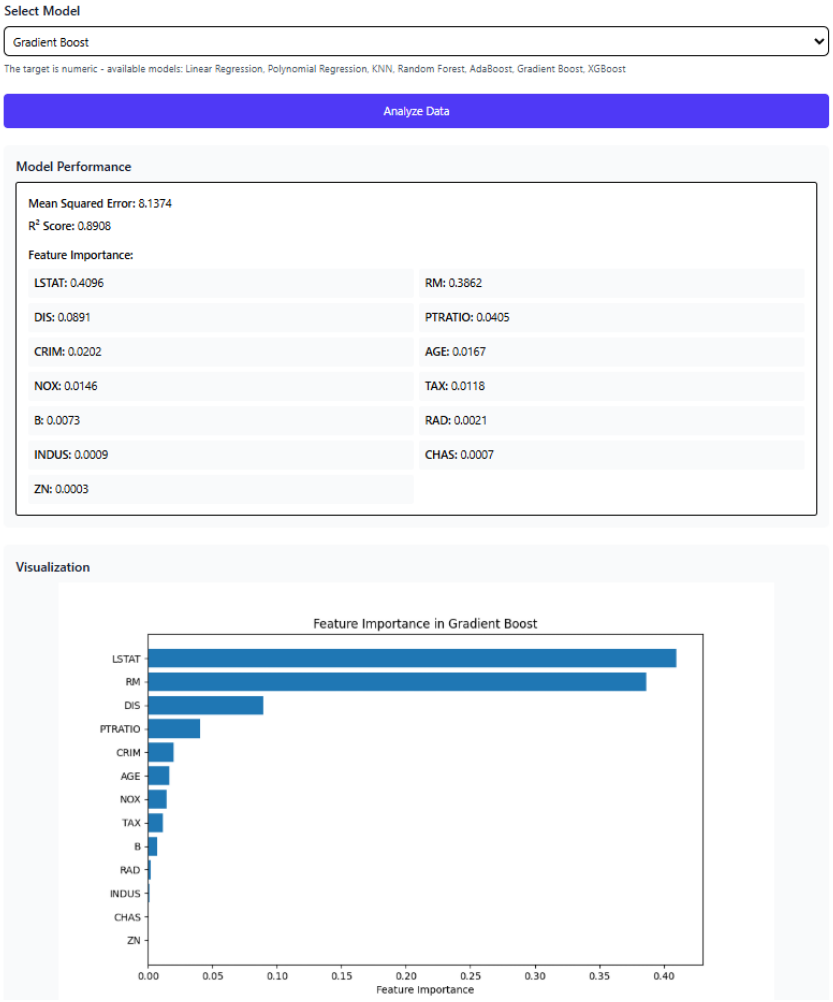


Figure 6.18: Model Selection & Model Performance: Gradient Boosting Regression

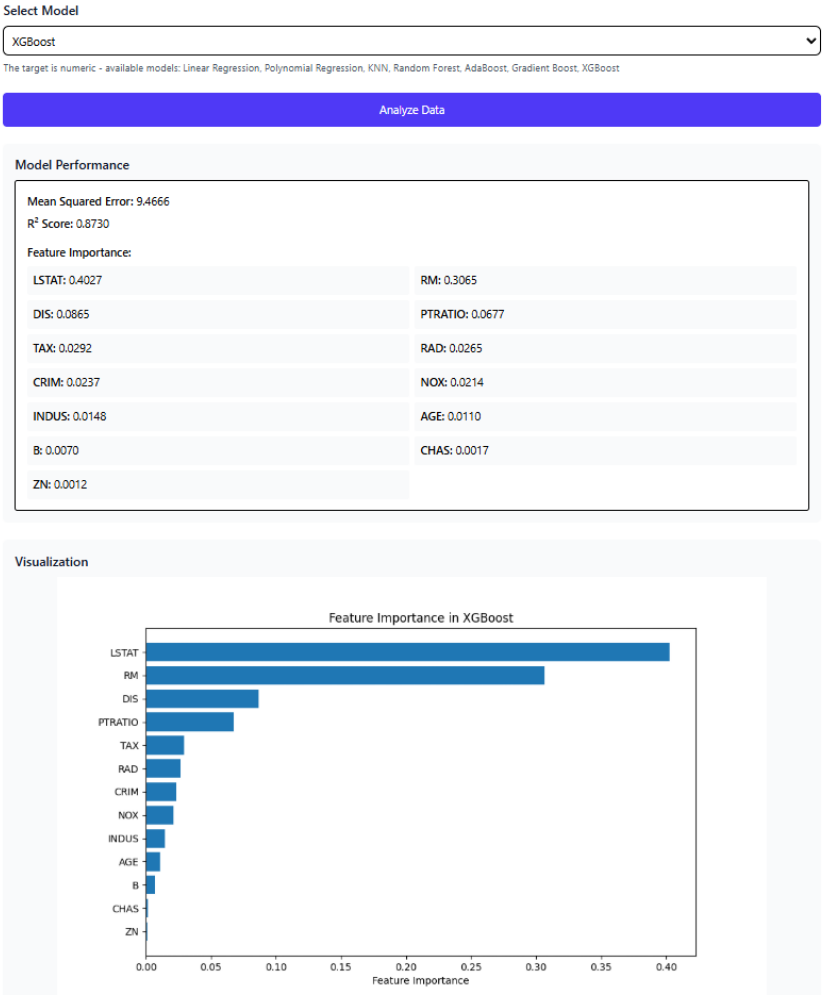


Figure 6.19: Model Selection & Model Performance: XGBoost Regression

Step 6: Train and Evaluate Model

The system handles preprocessing, including missing value imputation and normalization. It splits the dataset into training and test sets, trains the model, and displays performance metrics such as:

- Accuracy, Precision, Recall, and F1-Score (Classification)
- Confusion Matrix
- Feature Importance
- RMSE, MAE, R^2 Score (Regression, if applicable)

Step 7: Make Predictions

Users input custom feature values to generate predictions. The model outputs:

- **Classification:** Predicted class with probability/confidence.

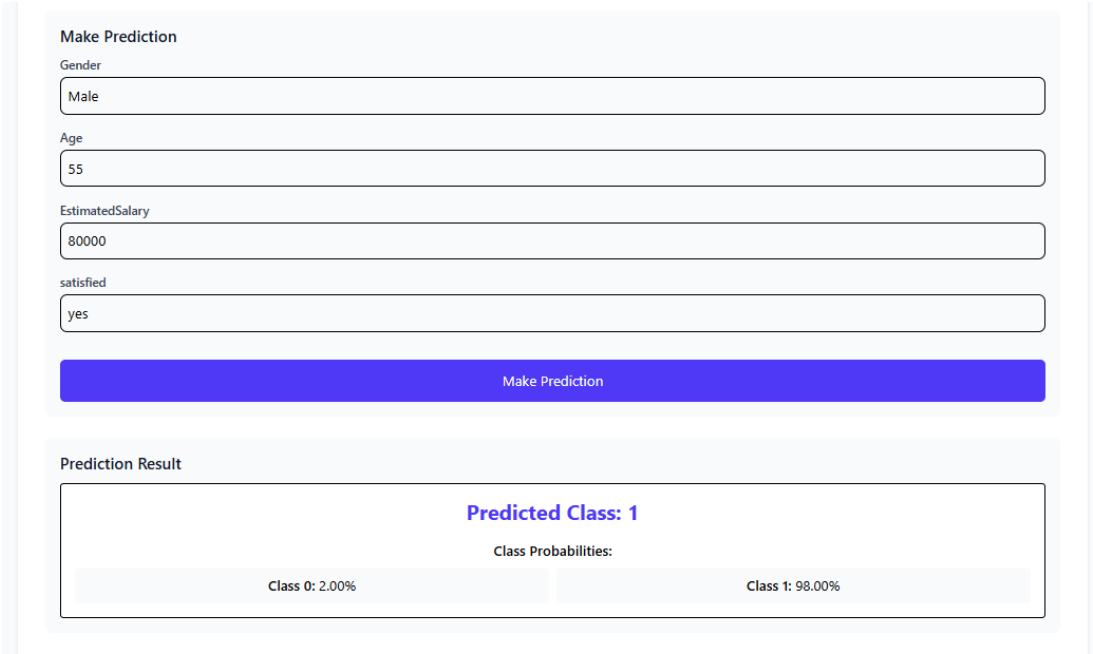


Figure 6.20: Classification Prediction Result

- **Regression:** Predicted numeric value.

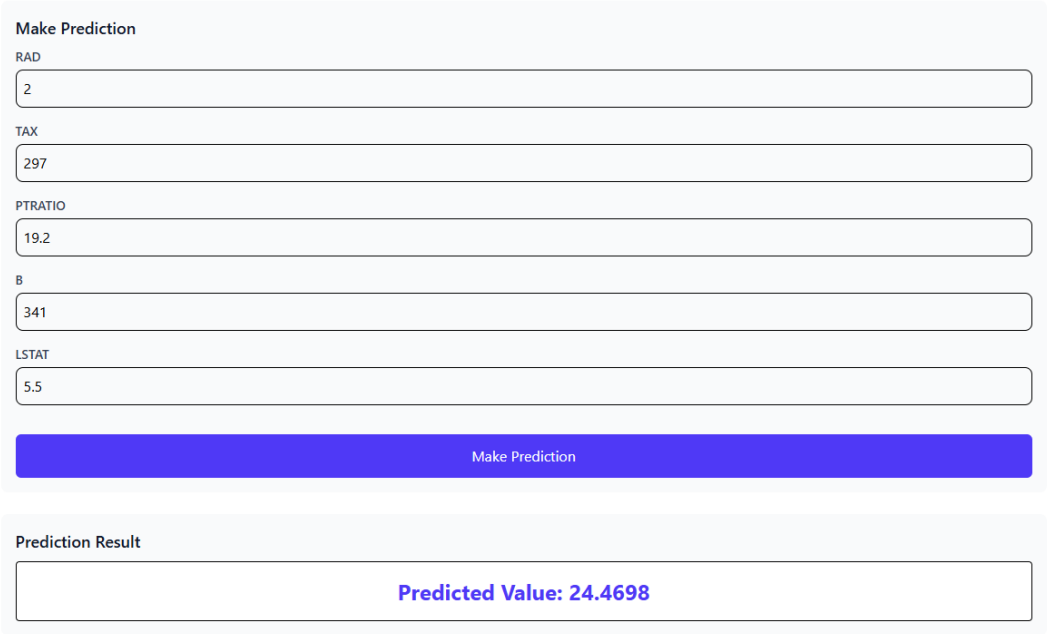


Figure 6.21: Regression Prediction Result

6.3.2 Diagnostic Analysis Module

The Diagnostic Analysis module enables users to uncover patterns, correlations, anomalies, and potential root causes within the data set. This module helps to understand why certain patterns exist, helping users make informed decisions based on the behavior of the underlying data.

Steps for Diagnostic Analysis

1. **Select Diagnostic Analysis**
   From the Custom Analysis tab, the user chooses the "Diagnostic Analysis" option to initiate the diagnostic workflow. This selection opens the dashboard for uploading the dataset and configuring the analysis parameters, as shown in Figure 6.22.

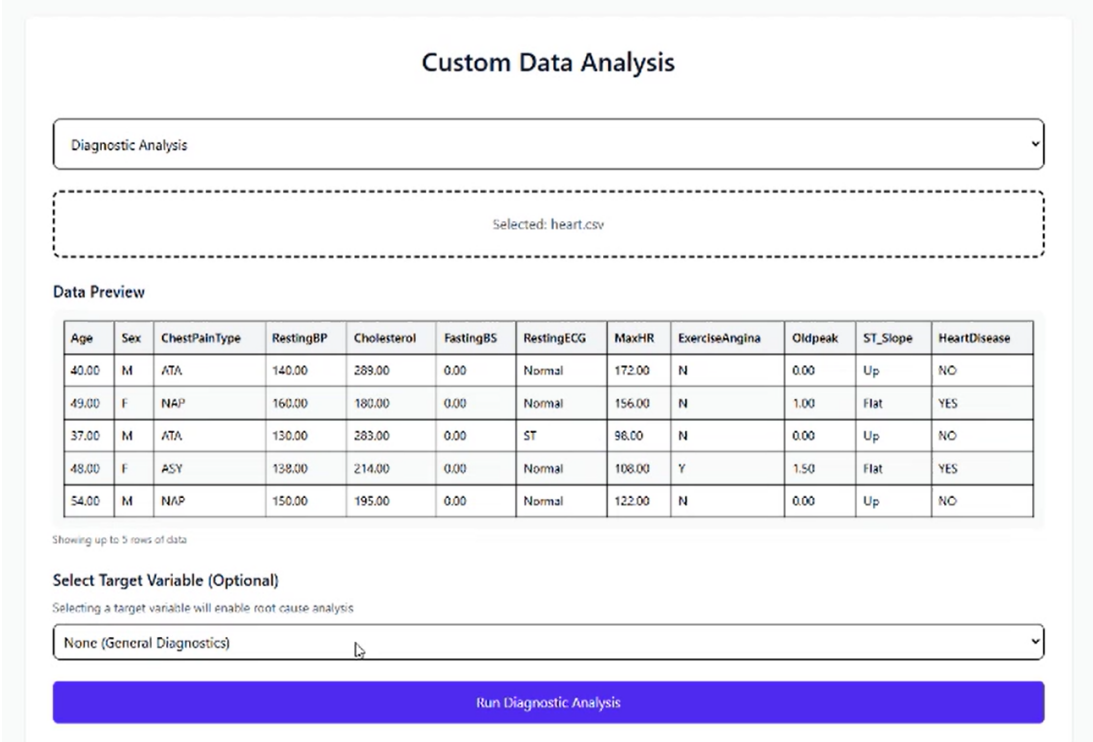


Figure 6.22: Diagnostic Analysis Dashboard

2. **Upload Dataset**
   The user uploads a dataset in `.csv` format. Upon upload, the system previews the data to ensure correct formatting.

3. **Select Target Variable (Optional)**
   For root cause analysis, the user can optionally select a target variable that represents the primary outcome or metric of interest. This step allows the system to identify the most influential features that contribute to changes in the selected target. Figure 6.23 illustrates this selection process.

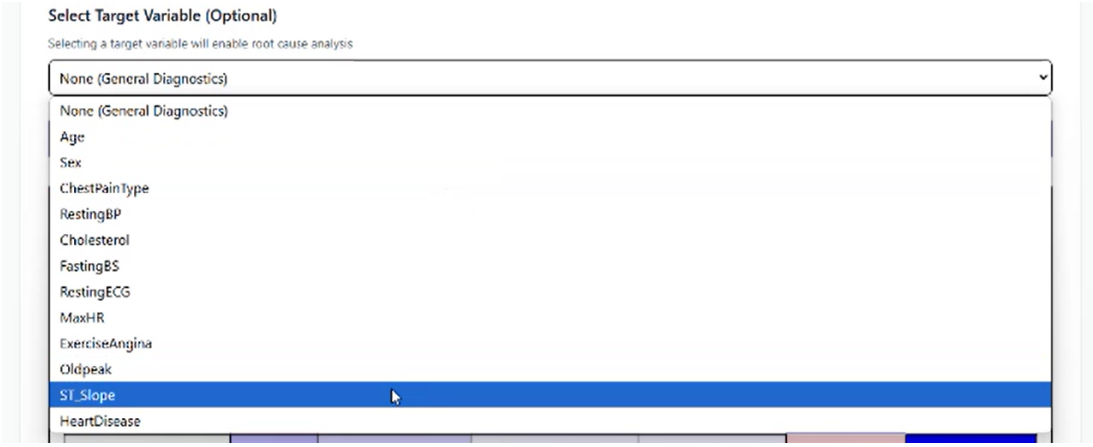
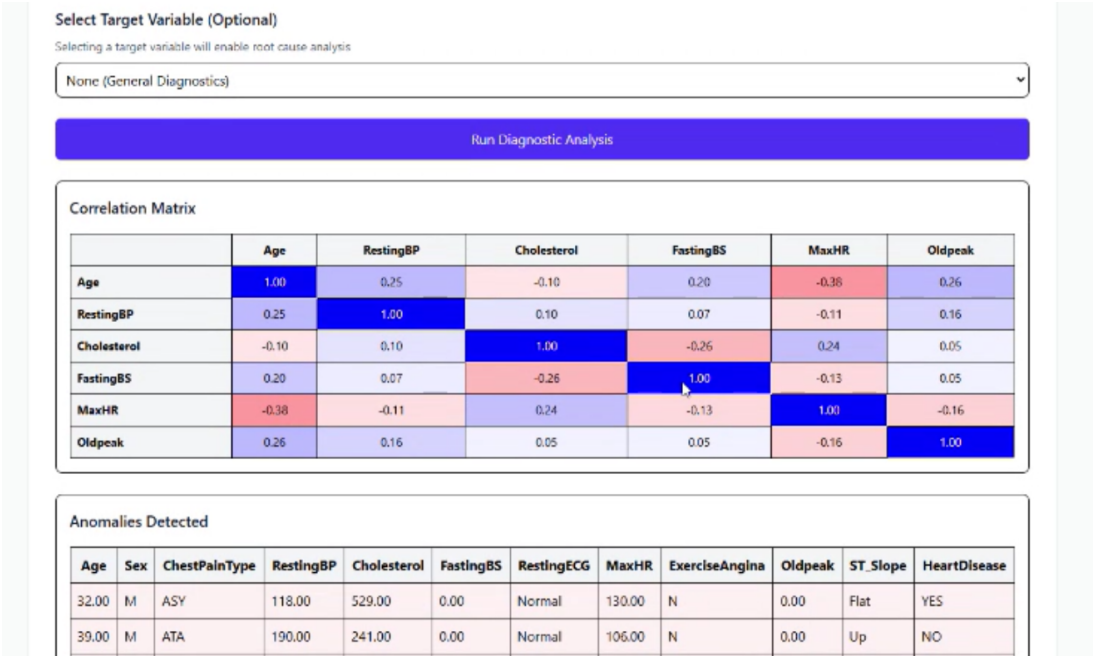


Figure 6.23: Selecting Target Variable for Root Cause Analysis

4. **Types of Diagnostic Analysis**
   The system dynamically determines the available diagnostic techniques based on whether a target variable is selected:

   - **If a Target Variable is Selected:**
     All three diagnostic methods are performed automatically:
     (a) **Correlation Matrix** — Visualizes the strength and direction of relationships between all variables (Figure 6.24).
     (b) **Anomaly Detection** — Identifies outliers and abnormal patterns in the data.
     (c) **Root Cause Analysis** — Determines which features most strongly influence variations in the selected target variable (Figure 6.25).
   - **If No Target Variable is Selected:**
     Only the Correlation Matrix and Anomaly Detection are performed, as Root Cause Analysis requires a target for comparison.

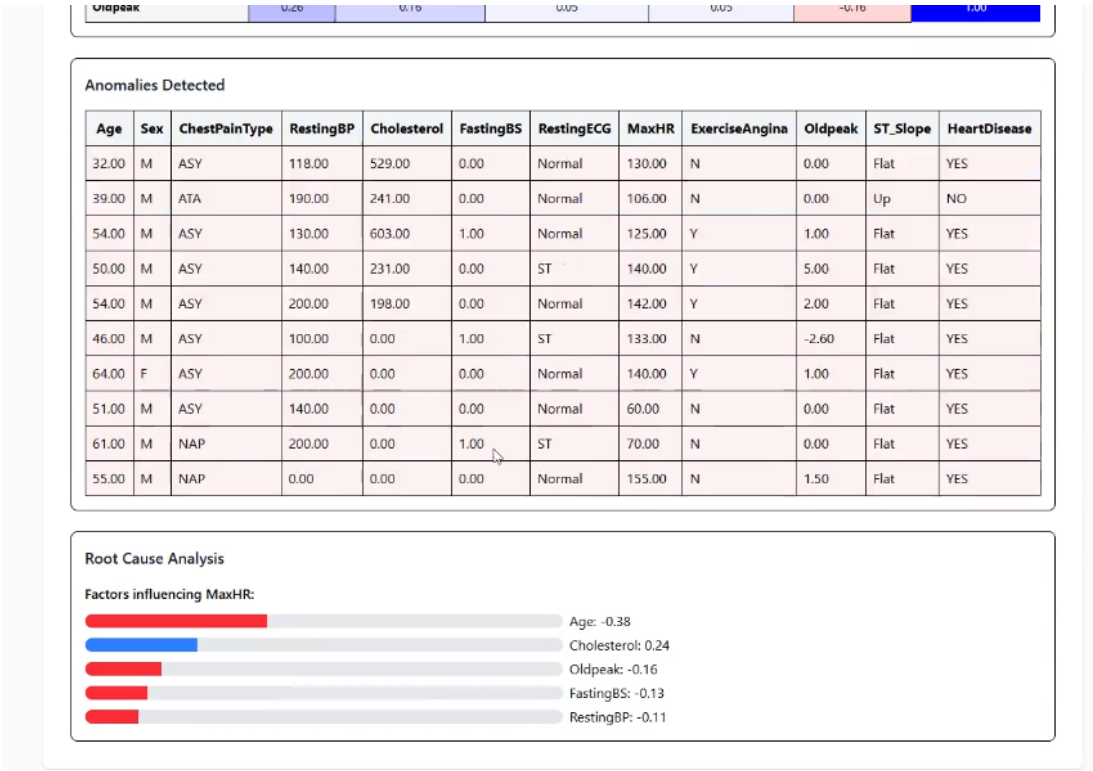


	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak
Age	1.00	0.25	-0.10	0.20	-0.38	0.26
RestingBP	0.25	1.00	0.10	0.07	-0.11	0.16
Cholesterol	-0.10	0.10	1.00	-0.26	0.24	0.05
FastingBS	0.20	0.07	-0.26	1.00	-0.13	0.05
MaxHR	-0.38	-0.11	0.24	-0.13	1.00	-0.16
Oldpeak	0.26	0.16	0.05	0.05	-0.16	1.00

Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
32.00	M	ASY	118.00	529.00	0.00	Normal	130.00	N	0.00	Flat	YES
39.00	M	ATA	190.00	241.00	0.00	Normal	106.00	N	0.00	Up	NO

Figure 6.24: Correlation Matrix: Identifying Strong Relationships Between Variables

5. **Run the Analysis and View Results**
   Upon clicking the "Run Analysis" button, the system performs the selected diagnostic operations and visualizes the results. The output includes highlighted anomalies, correlation heatmaps, and root cause summaries with feature contributions. An example of anomalies and root causes is shown in Figure 6.25.



Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
32.00	M	ASY	118.00	529.00	0.00	Normal	130.00	N	0.00	Flat	YES
39.00	M	ATA	190.00	241.00	0.00	Normal	106.00	N	0.00	Up	NO
54.00	M	ASY	130.00	603.00	1.00	Normal	125.00	Y	1.00	Flat	YES
50.00	M	ASY	140.00	231.00	0.00	ST	140.00	Y	5.00	Flat	YES
54.00	M	ASY	200.00	198.00	0.00	Normal	142.00	Y	2.00	Flat	YES
46.00	M	ASY	100.00	0.00	1.00	ST	133.00	N	-2.60	Flat	YES
64.00	F	ASY	200.00	0.00	0.00	Normal	140.00	Y	1.00	Flat	YES
51.00	M	ASY	140.00	0.00	0.00	Normal	60.00	N	0.00	Flat	YES
61.00	M	NAP	200.00	0.00	1.00	ST	70.00	N	0.00	Flat	YES
55.00	M	NAP	0.00	0.00	0.00	Normal	155.00	N	1.50	Flat	YES

Figure 6.25: Anomalies Detected and Root Cause Analysis Results

This module empowers users to explore beyond predictive outputs by diving deeper into the factors influencing their data trends and anomalies.

6.4 Custom Visualization Module

The **Custom Visualization Module** in Analyza empowers users to explore data sets visually using interactive and customizable charts. It is designed to help users—especially beginners—to gain intuitive insights from data without having to write code. By selecting chart types and adjusting visual parameters such as color, axis labels, and data mapping, users can quickly visualize patterns, trends, and distributions.

Features and Capabilities

Users can select from a variety of chart types, including bar charts, scatter plots, line plots, pie charts, histograms, box plots, violin plots, and heatmaps. Each visualization is customizable and interactive, enabling dynamic exploration of the uploaded dataset.

Steps for Custom Visualization

1. **Open the Custom Visualization Tab**
   From the main dashboard, navigate to the "Custom Visualization" section to access the visualization workspace & upload your csv data, as shown in Figure 6.26.

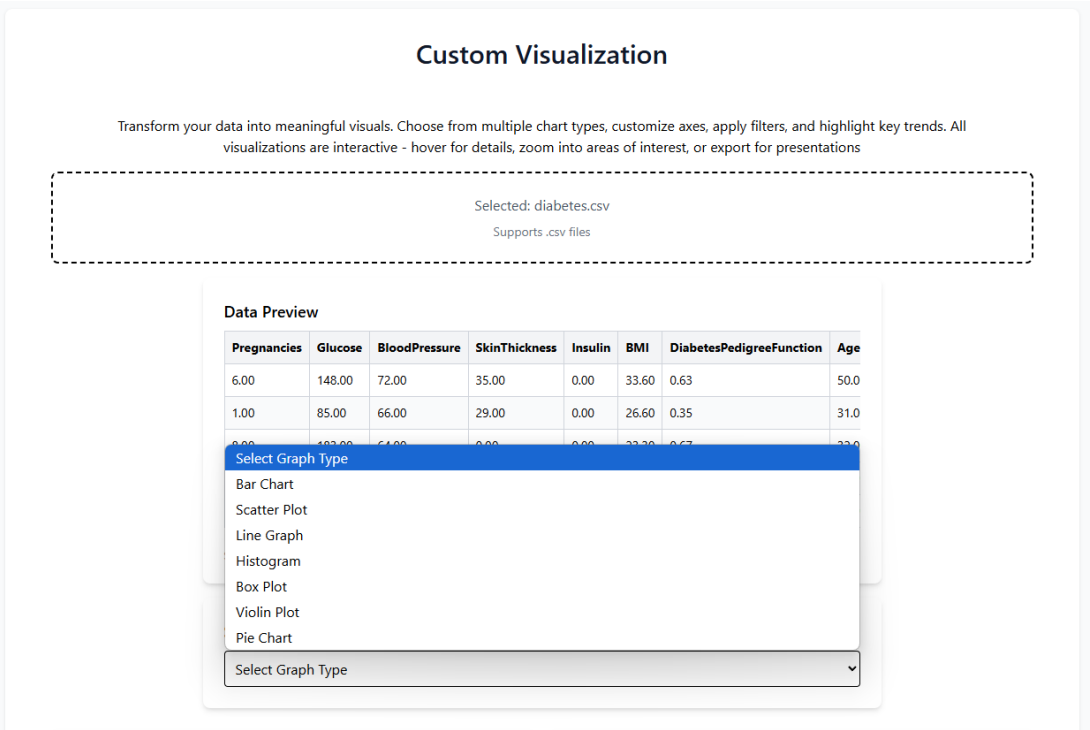


Figure 6.26: Custom Visualization Dashboard

2. **Select a Chart Type**
   Choose a desired chart type from a dropdown list which includes bar chart, line plot, pie chart, histogram, scatter plot, box plot, and violin plot (Figure 6.27).

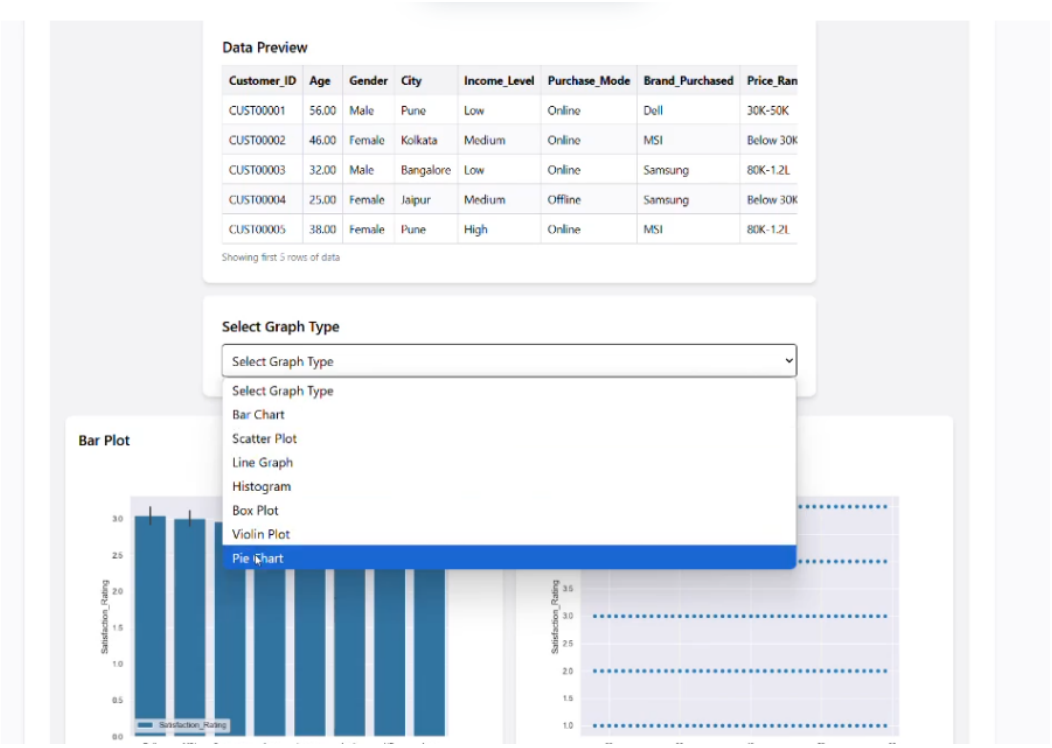


Figure 6.27: Chart Type Selection Interface

3. **Choose Variables for Axes (if applicable)**
   For most plots like scatter, line, and bar charts, users select the X and Y axes variables. For other chart types, such as pie or box plots, only one variable may be required depending on the context.

4. **Customize the Appearance**
   Customize elements like:

   - Axis titles and labels
   - Chart colors
   - Legend positions
   - Grid and background themes

5. **Generate the Visualization**
   Once configuration is complete, users click "Generate Visualization" to display the chart.

6. **Interact with the Visualization**
   The charts are interactive — users can hover to view values, zoom in, and export if needed.

7. **Repeat for Multiple Visuals**
   Users can generate and view multiple charts on the same dashboard for deeper analysis and comparison.

Sample Visualizations

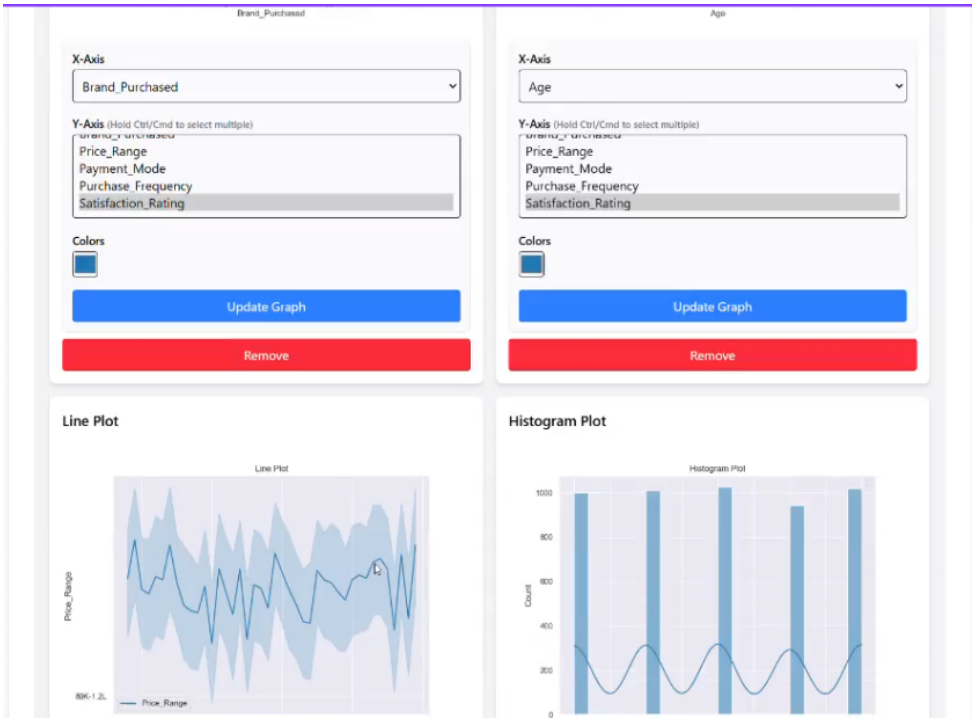


Figure 6.28: Line Plot and Histogram Visualization

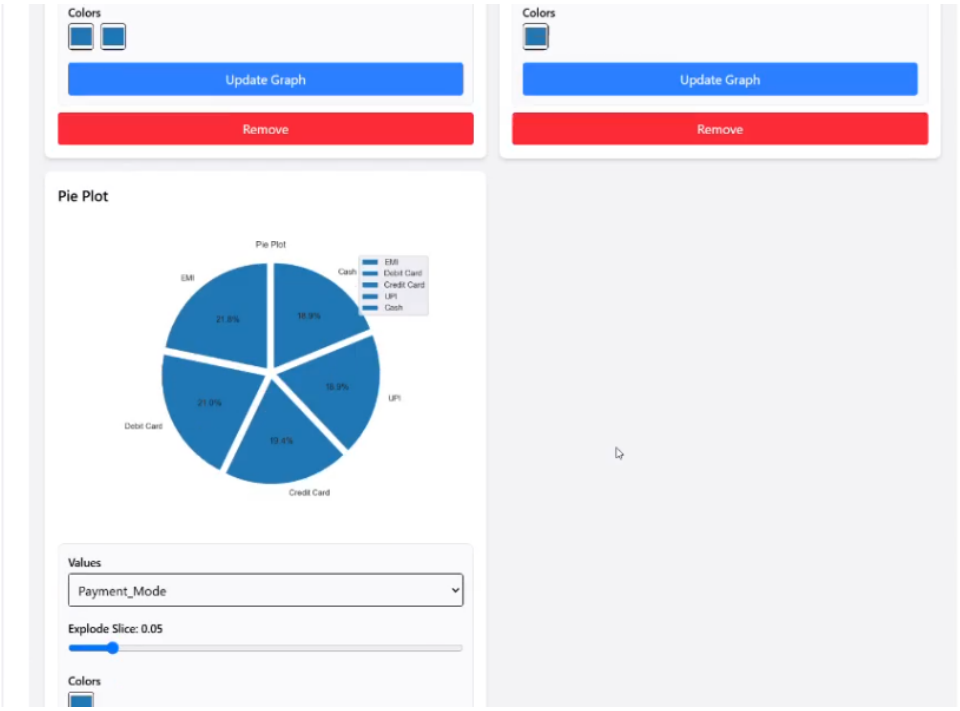


Figure 6.29: Pie Chart Visualization

Figure 6.30: Box Plot and Violin Plot Visualization

This module bridges the gap between raw data and insightful storytelling, offering users a seamless way to visualize and interpret complex datasets with minimal effort.

6.5 Documentation Tab

Overall Structure The Left Navigation Panel provides a clear and concise menu for navigating the documentation, including:

- **Overview**: Likely an introductory section
- **Key Features**: Highlights the main functionalities
- **Modules**: Details the different modules within Analyza
- **Technology Stack**: Specifies the technologies used to build Analyza
- **User Guide**: The currently viewed section
- **Limitations**: Outlines any known limitations or restrictions

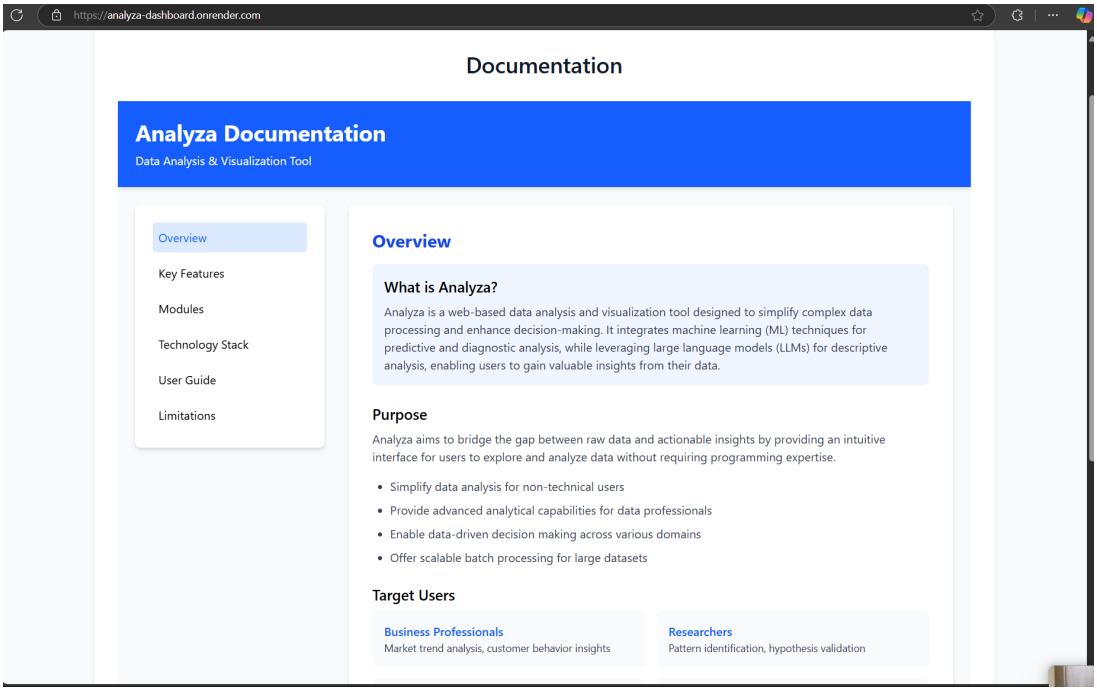


Figure 6.31: Documentation Overview

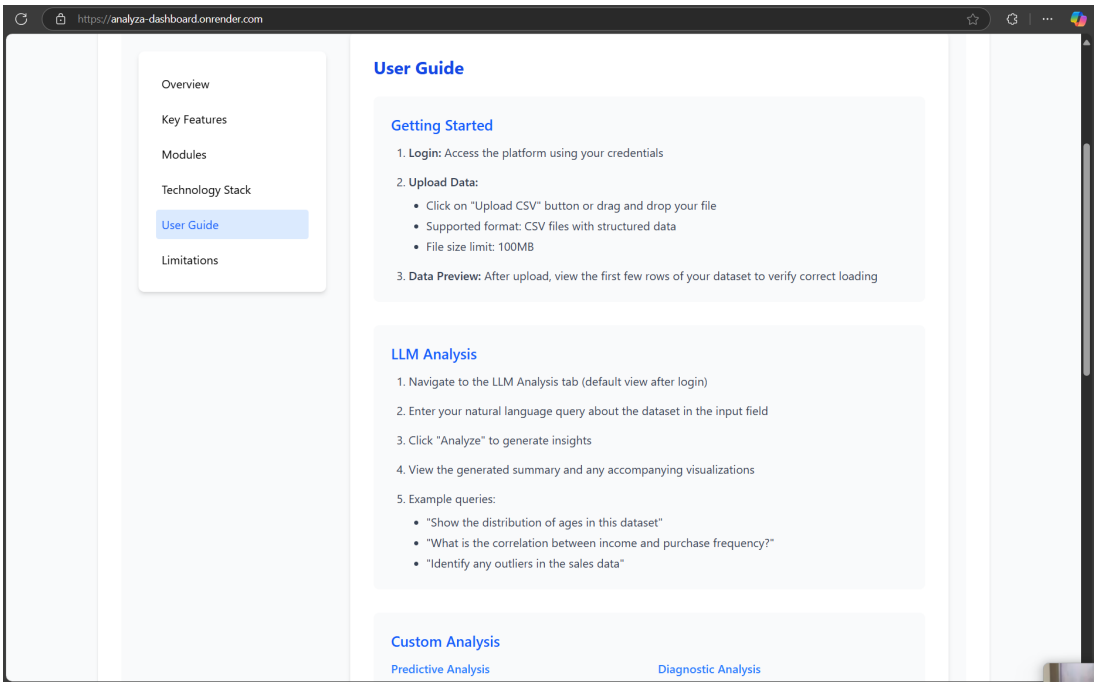


Figure 6.32: Documentation User Guide LLM Analysis

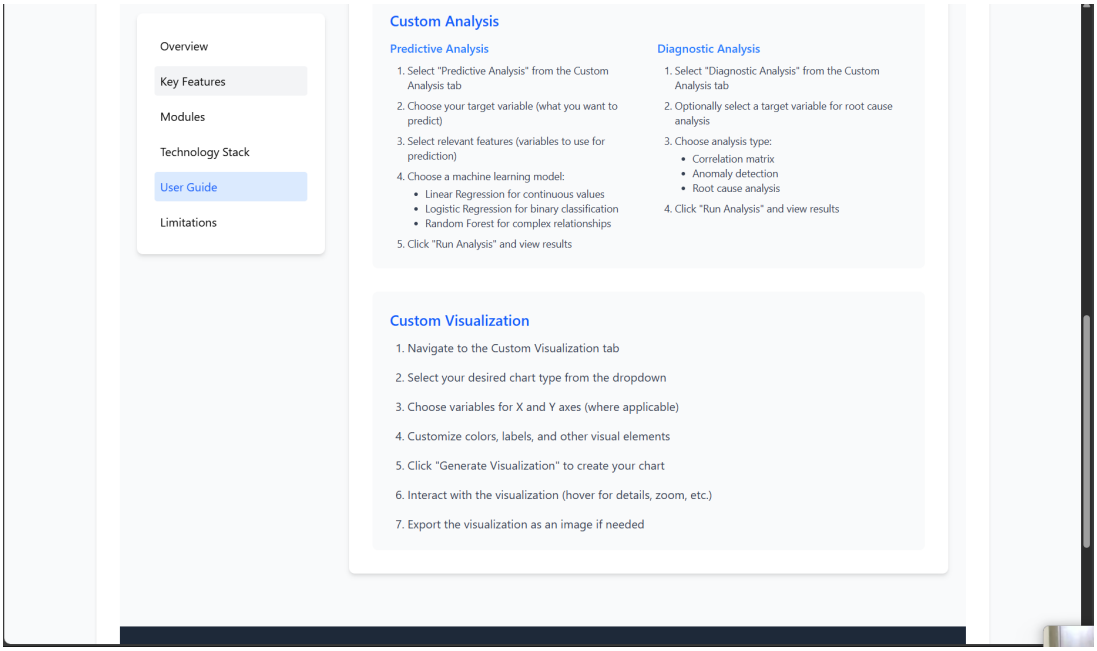


Figure 6.33: Documentation User Guide Custom Analysis and Visualization

6.6 About Us

This tab shows the developers and mentor associated with this project & their roles and contribution is depicted.

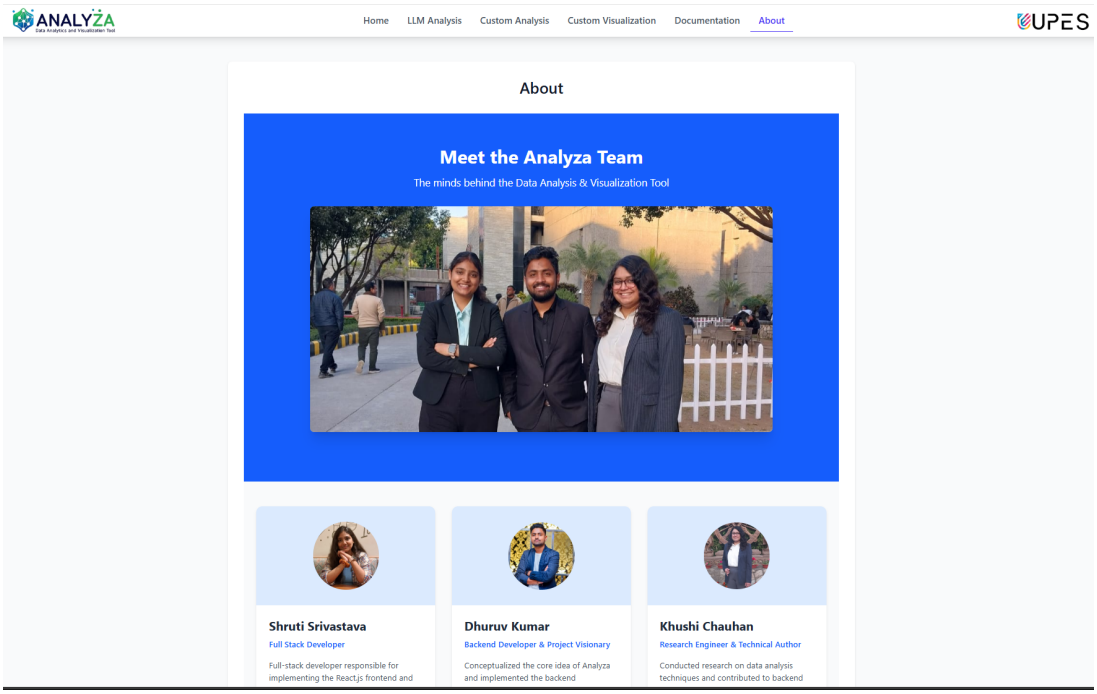


Figure 6.34: About us: Developers

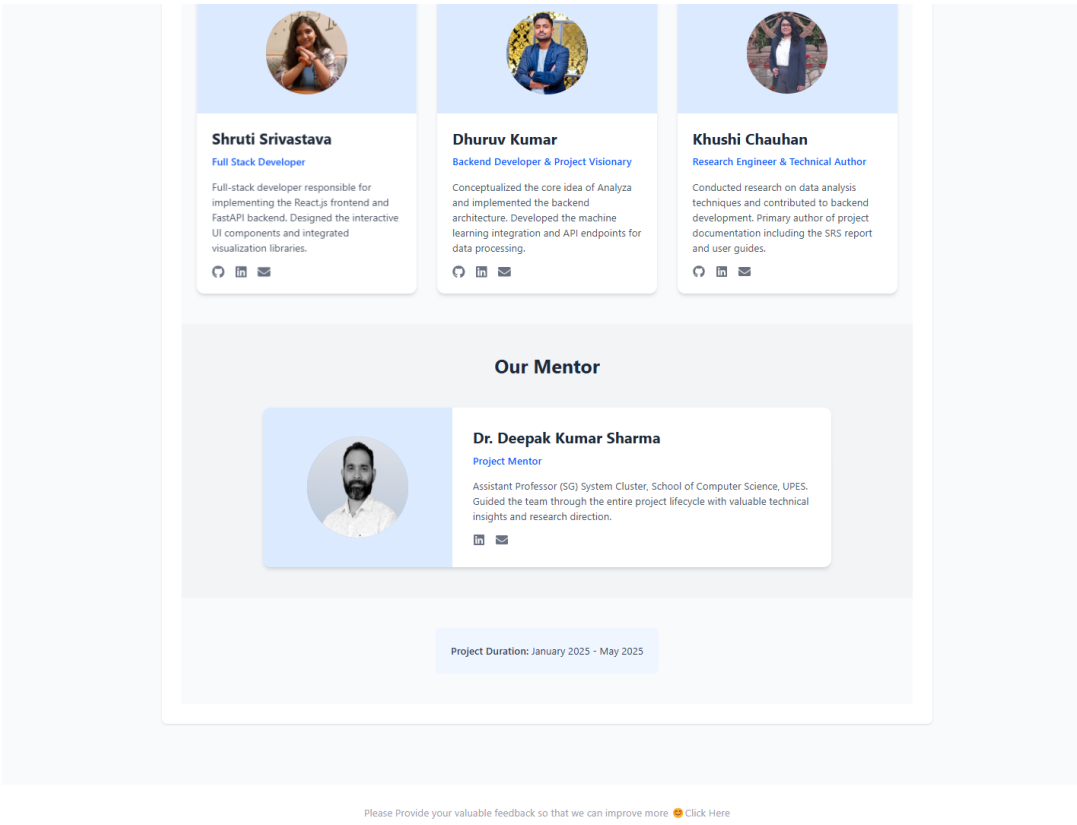


Figure 6.35: About us: Mentor

Chapter 7

Conclusion

The project **Analyza** was developed as a user-friendly web-based tool to make data analysis and visualization easy and accessible for everyone. The goal was to help both technical and non-technical users understand and work with data without needing programming skills. With the increasing importance of data in all fields, we wanted to create a tool that simplifies data processing and helps people gain meaningful insights using just a few simple steps.

In this project, we built a complete web platform where users can upload CSV files, perform different types of analysis, and generate visual reports. The platform includes descriptive analysis using (LLMs) to give easy-to-read summaries, predictive analysis to forecast future trends using machine learning models, and diagnostic analysis to find patterns, anomalies, and root causes in the data. It also supports custom data visualization, allowing users to choose from a variety of charts like bar charts, scatter plots, pie charts, histograms, and heatmaps.

To build Analyza, we used a modern tech stack. The **frontend** of the application was developed using **React.js**, which provided a clean and interactive interface. The **backend** was created using **FastAPI**, a high-performance web framework for building APIs in Python. For data processing and machine learning, we used popular Python libraries like **Pandas**, **NumPy**, **Matplotlib**, **Seaborn**, and **Scikit-learn**. We also integrated **LLMs** to allow natural language queries, so users could interact with their data using plain English.

The system was designed to handle large datasets through batch processing, and care was taken to maintain a fast response time and good performance. We implemented features like **data cleaning**, **model selection**, **automatic feature handling**, and **in-memory processing** to make the analysis smooth and secure. The platform does not store user data permanently, ensuring data privacy and security.

As a result of our work, Analyza turned out to be a robust and efficient tool for data analysis. It can be used in a variety of fields such as business, finance, education, healthcare, research, and government. Businesses can use it to study customer behavior, students can learn data science using real examples, and researchers can find trends and patterns in large datasets. The tool has shown that it can process data quickly, produce accurate insights, and present results in a clear and easy way.

In conclusion, Analyza successfully achieved its goal of making data analysis simple, powerful, and accessible. It brings together the power of machine learning, natural language processing, and interactive visualization into one platform. The project not only helped us learn valuable technical and teamwork skills, but it also created a solution that has real-world value and can be further improved in the future.

7.1 Future Enhancements

While **Analyza** already provides powerful features for data analysis and visualization, there are several ways in which it can be further improved and expanded in the future. These enhancements can help make the platform more advanced, flexible, and suitable for a wider range of users and industries.

1. **Real-Time Data Analysis**
   Currently, Analyza supports batch processing using CSV files. In the future, it can be enhanced to handle real-time data streams from sources like APIs, IoT devices, or social media. This would allow users to analyze live data and get instant updates and insights as data flows in.

2. **Support for Big Data**
   Analyza can be enhanced to handle big data by integrating distributed processing frameworks like Apache Hadoop or Apache Spark. This would enable users to process and analyze large-scale datasets efficiently, providing more powerful insights without performance bottlenecks.

3. **Support for More Data Formats**
   At present, Analyza works with structured data in CSV format. Future versions could support Excel, JSON, SQL databases, NoSQL databases (like MongoDB), and even unstructured data such as text files or logs, making the platform more flexible and powerful.

4. **Advanced Machine Learning and Deep Learning Models**
   While the current version uses common ML models like Random Forest and Logistic Regression, future upgrades can include deep learning models (such as neural networks) and time series forecasting models like LSTM or ARIMA to handle more complex prediction tasks.

5. **Improved Visualization and Dashboarding**
   More advanced and interactive visualizations like live charts, geo maps, dashboard templates, and drag-and-drop builders can be added. Users could also have the ability to save and export dashboards in different formats (PDF, PNG, etc.).

6. **Collaboration Features**
   A future version could include multi-user access, where teams can collaborate on datasets, share insights, and comment on visualizations or analysis results in real time.

7. **Integration with Cloud Services**
   Analyza can be integrated with cloud platforms like AWS, Azure, or Google Cloud to support cloud-based storage, processing, and scalability. It could also support data from Google Sheets, Google Drive, or OneDrive for more seamless access.

8. **Enhanced Security and User Management**
   Future versions can introduce user roles, permissions, multi-factor authentication (MFA), and audit logs to make the system more secure and enterprise-ready.

9. **Offline Mode or Desktop App**
   An offline version or a desktop application could be built using tools like Electron.js or PyInstaller so users can perform analysis without needing an internet connection.

10. **Mobile Compatibility**
    Enhancing the interface for mobile devices would allow users to upload data and view results from their smartphones or tablets, increasing convenience and accessibility.

11. **Custom Scripting & Plugin Support**
    Allowing users to write custom Python or SQL scripts or add plugins would give power users the flexibility to extend Analyza's capabilities according to their specific needs.

Appendix A

Glossary

CSV (Comma Separated Values)

CSV is a widely used file format for storing tabular data in plain text. In a CSV file, each row corresponds to a record, and columns are separated by commas. This format is commonly used for data storage and exchange due to its simplicity and compatibility with various applications, including spreadsheet software, databases, and programming languages.

In the context of Analyza, CSV files serve as the primary input format for users to upload datasets. The system processes these files to perform data analysis, visualization, and machine learning tasks. Users can select features and target variables from CSV files, allowing

Analyza to generate insights, train predictive models, and display customized visualizations.

LLM (Large Language Model)

A LLM is an advanced artificial intelligence system trained on massive amounts of text data to understand and generate human-like language. LLMs use deep learning techniques, specifically transformer architectures, to analyze and predict text-based outputs.

In Analyza, LLMs are integrated to assist users in descriptive data analysis. They help by:

- Summarizing datasets by identifying key statistical properties.
- Generating insights and explanations about patterns, trends, and anomalies.
- Assisting users with natural language queries related to their data.

This enhances user experience by making complex data analysis more accessible, even for beginners without extensive statistical or machine learning knowledge.

FastAPI

FastAPI is a modern, high-performance web framework for building APIs with Python. It is designed for speed and efficiency, leveraging asynchronous programming to handle multiple requests simultaneously. FastAPI also provides built-in validation, automatic

interactive documentation, and seamless integration with machine learning models.

In Analyza, FastAPI serves as the backend framework responsible for:

- Handling user requests, such as uploading CSV files and selecting analysis options.
- Managing interactions between the frontend (React.js) and backend services.
- Processing data using machine learning models and generating predictions.
- Serving visualization requests by returning processed data in a format suitable for rendering.

By using FastAPI, Analyza ensures fast response times and efficient data processing, enhancing user interaction and experience.

React.js

React.js is a JavaScript library used for building dynamic and interactive user interfaces. It follows a component-based architecture, allowing developers to create reusable UI elements for web applications. React.js is known for its virtual DOM, which improves performance by minimizing direct manipulations of the actual DOM.

In Analyza, React.js is utilized to develop an intuitive and interactive frontend, enabling users to:

- Upload datasets and interact with different analysis modules.
- Customize visualizations by modifying colors, axes, chart types, and dataset attributes.
- View real-time updates on data insights and machine learning model outputs.
- Organize multiple visualizations on a dashboard for comparative analysis.

By leveraging React.js, Analyza provides a seamless and user-friendly experience for data analysis and visualization.

Appendix B

Analysis Model

The analysis model in Analyza follows a structured workflow that integrates machine learning and LLM-based descriptive analysis to provide comprehensive insights. The key steps involved are as follows:

1. Data Upload and Validation

The first step in the analysis model is to allow users to upload datasets in CSV format. This stage includes:

- File Format Checking: Ensuring that the uploaded file adheres to the CSV format.
- Schema Validation: Checking whether the dataset contains valid column names, correct data types, and a proper structure.
- Duplicate and Corrupted Data Detection: Identifying and handling duplicate or corrupted entries that might affect the analysis.

This step ensures that only clean and structured data is passed to the next phase.

2. Data Preprocessing

Once the data is uploaded, preprocessing is performed to clean and transform it into a format suitable for analysis. The key preprocessing tasks include:

- Handling Missing Values: Imputing missing values using statistical methods (mean, median, mode) or removing incomplete records if necessary.
- Encoding Categorical Variables: Converting categorical data into numerical form using encoding techniques such as one-hot encoding or label encoding.
- Data Normalization and Scaling: Standardizing numerical values to ensure consistency, especially for machine learning models.
- Feature Selection: Allowing users to select relevant features for analysis, improving model efficiency and accuracy.

Preprocessing ensures that the dataset is cleaned and optimized for accurate insights and predictions.

3. Analysis Phase

This stage focuses on applying analytical techniques to extract meaningful insights from the data. Users can choose from different types of analyses:

- Predictive Analysis: Utilizing machine learning models such as Linear Regression, Logistic Regression, and Random Forest to forecast trends and outcomes.
- Diagnostic Analysis: Identifying key factors influencing trends, detecting anomalies, and explaining patterns in the data.
- Descriptive Analysis using LLMs: Leveraging LLMs to provide natural language summaries, insights, and explanations of the dataset. The LLM can:
  - Summarize key statistical properties such as mean, median, and standard deviation.
  - Identify and describe patterns, trends, and anomalies in the data.
  - Provide explanations of relationships between different variables in simple language.
- Custom Visualization: Allowing users to generate tailored visual representations of their data, selecting variables and visualization types such as bar charts, scatter plots, and heatmaps.

The integration of machine learning and LLMs enhances both simple and advanced data-driven decision-making.

4. Visualization and Result Display

The final step involves presenting the analyzed data in an interactive and user-friendly manner. This includes:

- Graphical Representations: Displaying insights using various visualization techniques, including histograms, line charts, and pie charts.
- Customizable Dashboard: Allowing users to modify visualization attributes such as colors, axes, labels, and dataset values.
- Insight Generation via LLMs: Providing natural language explanations of trends and patterns, making complex data insights accessible to beginners.

This step enhances the usability of the tool by ensuring that insights are presented clearly and effectively.

By following these structured steps, Analyza provides a seamless experience for users, integrating machine learning and LLM-based descriptive analysis for enhanced data interpretation.

Appendix C

Issues List

While Analyza is designed to provide a seamless and user-friendly experience for data analysis, the following limitations and challenges exist:

1. Limited Support for File Formats Other than CSV

Currently, Analyza primarily supports datasets in CSV format. The limitations of this approach include:

- Lack of Support for Other Formats: Users who work with Excel files (.xlsx), JSON, or database connections may face difficulties importing their data.
- Manual Conversion Required: Users must manually convert non-CSV datasets into CSV format before analysis, which can be inconvenient.
- Potential Data Loss: Converting structured data (e.g., multi-sheet Excel files or hierarchical JSON structures) to CSV may lead to loss of metadata or relationships between fields.

Future enhancements could include support for additional file formats such as Excel, JSON, and direct database connections.

2. Dependency on Third-Party APIs (Google's Gemini)

Analyza leverages LLMs, specifically Google's Gemini API, to generate descriptive analysis and natural language summaries. The dependency on external APIs introduces several challenges:

- Reliability Concerns: The availability and performance of third-party APIs depend on external service providers, which may experience downtime or rate limits.
- Cost Considerations: API usage may incur costs, especially when processing large datasets or making frequent requests.
- Privacy and Security Risks: Since data is sent to an external API for processing, users may have concerns regarding data confidentiality and compliance with data protection regulations.

Future updates could explore on-premise LLM models or alternative APIs to mitigate these dependencies.

3. Potential Performance Issues with Large Datasets

Handling large datasets efficiently is a key challenge in data analysis. The current implementation of Analyza may face performance bottlenecks in the following areas:

- Memory Consumption: Processing large datasets in memory can lead to slow performance or crashes, especially when dealing with high-dimensional data.
- Longer Processing Times: Computationally intensive tasks such as machine learning training, predictive modeling, or visualization rendering may take longer with large datasets.
- Scalability Limitations: The current backend, based on FastAPI and Python, may require optimizations or distributed computing techniques (e.g., Spark) to scale effectively.

Potential improvements include optimized data handling techniques, batch processing, and parallel computation support.

These issues highlight areas for future development and enhancement in Analyza, ensuring improved performance, flexibility, and user experience.

Bibliography

[1] Weng, Luoxuan, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, Haozhe Feng, Danqing Huang, and Wei Chen. *InsightLens: Augmenting LLM-Powered Data Analysis with Interactive Insight Management and Navigation*. arXiv preprint arXiv:2404.01644 (2024).

[2] Zhou, Xuanhe, Xinyang Zhao, and Guoliang Li. *LLM-Enhanced Data Management*. arXiv preprint arXiv:2402.02643 (2024).

[3] Wang, Qianwen, Zhutian Chen, Yong Wang, and Huamin Qu. *A survey on ML4VIS: Applying machine learning advances to data visualization*. IEEE transactions on visualization and computer graphics 28, no. 12 (2021): 5134-5153.

[4] Bonthu, Sridevi, and K. Hima Bindu. *Review of leading data analytics tools*. International Journal of Engineering & Technology 7, no. 3.31 (2017): 10-15.

[5] Nejjar, Mohamed, Luca Zacharias, Fabian Stiehle, and Ingo Weber. *LLMs for science: Usage for code generation and data analysis*. Journal of Software: Evolution and Process 37, no. 1 (2025): e2723.

[6] Ali, Jehad, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. *Random forests and decision trees*. International Journal of Computer Science Issues (IJCSI) 9, no. 5 (2012): 272.

[7] Ridzuan, Fakhitah, and Wan Mohd Nazmee Wan Zainon. *Diagnostic analysis for outlier detection in big data analytics*. Procedia Computer Science 197 (2022): 685–692.

[8] Kumar, Vaibhav, and M. L. Garg. *Predictive analytics: a review of trends and techniques*. International Journal of Computer Applications 182, no. 1 (2018): 31–37.

[9] Lavanya, Addepalli, Lokhande Gaurav, Sakinam Sindhuja, Hussain Seam, Mookerjee Joydeep, Vamsi Uppalapati, Waqas Ali, and Vidya Sagar SD. *Assessing the performance of Python data visualization libraries: a review*. International Journal of Computer Engineering Research Trends (IJCERT) 10, no. 1 (2023): 28–39.

[10] Wu, Xiaomin. *FastAPI as a backend framework*. (2024).

[11] Fan, Cheng, Meiling Chen, Xinghua Wang, Jiayuan Wang, and Bufu Huang. *A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data*. Frontiers in Energy Research 9 (2021): 652801.

[12] Lazuardy, Mochammad Fariz Syah, and Dyah Anggraini. *Modern front end web architectures with React.js and Next.js*. Research Journal of Advanced Engineering and Science 7, no. 1 (2022): 132–141.