UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

District University Francisco José de Caldas

Department of engineering

Systemic Analysis and Design of a Predictive Framework for the Sweet Regression Competition

Samuel Aljure Bernal
Carlos Alberto Barriga Gámez
David Santiago Aldana González

Juan Diego Álvarez Cristancho

Supervisor: Carlos Andrés Sierra

A report submitted in partial fulfilment of the requirements of the District University Francisco José de Caldas for the degree of Systems Analysis and Design in *Systems Engineering*

October 25, 2025

Declaration

We, Samuel Aljure Bernal, Carlos Alberto Barriga Gámez, David Santiago Aldana González, and Juan Diego Álvarez Cristancho, of the Department of engineering, District University Francisco José de Caldas, confirm that this is our own work and that all figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. We understand that failure to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

We give consent to a copy of our report being shared with future students as an exemplar.

We also consent for our work to be made available more widely to members of the District University Francisco José de Caldas and the public with interest in teaching, learning, and research.

Samuel Aljure Bernal, Carlos Alberto Barriga Gámez, David Santiago Aldana González, and
Juan Diego Álvarez Cristancho
October 25, 2025

Abstract

This report documents the systemic analysis and a proposed modular architectural design for a predictive system aimed at estimating sales of the fictional company Chocolates 4U in new regions. The proposed solution involves implementing a regression-based predictive model in R, relating variables such as advertising strategies, regional conditions, and consumer behavior. The report describes the inputs, processes, outputs, constraints and sensitivity control mechanisms (randomness and chaos), functional and non-functional requirements, and a technical implementation outline combining Python (exploratory analysis) and R (modeling and validation). This report serves as a design guide prior to the experimental implementation phase.

Keywords: System analysis, predictive modeling, regression systems, reproducibility, systems engineering, software design, educational projects.

Acknowledgements

We would like to thank the Systems Engineering Department of the Francisco José de Caldas District University, the course instructor Carlos Andres Sierra, and the participating colleagues for their contributions in data, documentation, and discussions that facilitated this project.

Contents

List of Figures

List of Tables

List of Abbreviations

MAE	Mean Absolute Error
ANOVA	Analysis of Variance

Chapter 1

Introduction

1.1 Background

The Sweet Regression competition challenges the prediction of Chocolates 4U sales in new regions using historical advertising campaign data, regional variables, and demographics. The dataset includes GRP (advertising exposure per channel) variables, socioeconomic characteristics, and campaign characteristics, totaling 27 variables and approximately 750 observations. The competition imposes specific constraints (limited deliveries, delivery format, and agreed-upon metrics) that determine the experimentation strategy and the need for reproducible procedures.

1.2 Problem statement

Design and document a predictive system that allows: (1) preprocessing and transforming input data in a consistent manner; (2) performing feature selection and comparative training of regression models; (3) generating predictions ready for submission to the competing platform (Kaggle), minimizing the MAE. The Sweet Regression challenge must be addressed by respecting the platform's rules (use of R in submission, maximum number of submissions) and controlling the sensitivity associated with stochastic processes and preprocessing decisions.

1.3 Aims and objectives

Aims: Develop a modular system that allows for forecasting Chocolates 4U sales in new regions based on historical data.

Objectives:

- Define the system structure (inputs, processes, and outputs).
- Determine the system's functional and non-functional requirements.
- Define data ingestion and validation procedures.
- Implement variable selection criteria (ANOVA, correlation, regularization).
- Train and compare regression models using MAE as the primary metric for sales prediction.

1.4 Solution approach

The proposed solution includes two stages. (1) Design stage: focused on specifying the system architecture (inputs, processes, outputs), defining requirements, system constraints, and statistical analysis of sales-related variables to determine practices for controlling system sensitivity. (2) Experimentation: focused on system implementation, running controlled experiments, and final model selection for submission. The implementation will combine Python for exploration and reproducible transformation, and R for final modeling and generation of the CSV for submission.

1.5 Scope

The report covers the system analysis and design phases: specification of functional and non-functional requirements, modular system design, recommendations for reproducibility tools and practices, and proposed diagrams (system architecture diagram, data flow diagram).

1.6 Assumptions

- The data provided is representative of the prediction problem (there is no sampling bias).
- There are no massively missing values; preprocessing will require specific cleaning and coding.
- The team has access to runtime environments with reasonable resources to train models (sufficient CPU and memory).
- The advertising variables (GRP) are related to the target and provide a predictive signal.

1.7 Limitations

- Platform restrictions (maximum 10 submissions) limit iterations in external evaluation and require greater reliance on internal validation.
- Dependence on the quality and representativeness of the dataset: undetected anomalies could bias models.
- Possible collinearity between advertising variables reduces the interpretability of linear models.

Chapter 2

Literature Review

2.1 State-of-the-Art

Predictive modeling has become fundamental to data-driven decision-making, both in academia and industry. As described by Hastie, Tibshirani, and Friedman , predictive models enable structured inference from data by balancing bias, variance, and interpretability [1]. These approaches are fundamental to modern analytics and artificial intelligence, supporting trend prediction, process optimization, and strategic planning.

Competitions such as Sweet Regression allow participants to apply concepts from systems analysis, software design, and data science to real-world problems. In this specific case, participants must predict chocolate sales for a fictional company, Chocolates 4U, using regression techniques in R. The challenge involves analyzing marketing exposure, socioeconomic indicators, and behavioral data, demonstrating the complexity of applied predictive modeling.

2.2 The Project in the Context of Literature and Existing Systems

From a systems engineering perspective, the Sweet Regression Competition can be considered a dynamic system composed of inputs, processes, outputs, and constraints. Workshop 1 identified the system's sensitivity to data variability, algorithmic randomness, and human decision-making. Workshop 2 addressed these issues by proposing a modular framework that integrates engineering principles such as modularity, scalability, traceability, and feedback control.

Python was used for exploratory analysis thanks to its extensive ecosystem of libraries such as Pandas, Scikit-learn, and Matplotlib, which facilitate efficient data exploration and validation before final implementation in R.

2.3 Critique of Existing Work and Relevance of the Project

Existing data-driven frameworks and competitions typically prioritize performance metrics over methodological rigor. These approaches, while effective for short-term accuracy, often result in analytical workflows that are difficult to replicate. The Sweet Regression framework seeks to overcome these limitations by integrating system reproducibility and traceability as fundamental design principles.

Techniques such as ANOVA and Bellman's rule were applied to refine feature selection and assess model stability. This strengthens the system's ability to generalize results and maintain robustness under variable data conditions.

The importance of the project lies in connecting two disciplines: systems engineering and data science. By integrating control, feedback, and reproducibility mechanisms into the modeling process, it not only improves analytical reliability but also strengthens uncertainty management in complex systems.

Chapter 3

Methodology

The methodological structure of the Sweet Regression Competition project was organized into two complementary phases designed to move from system understanding to system design. The first stage (Workshop 1) approached the challenge as a complex dynamic system composed of multiple interacting elements, while the second stage (Workshop 2) transformed that analytical understanding into a concrete system architecture, complete with defined modules, requirements, and implementation strategies aimed at reproducibility and stability.

3.1 Systemic Analysis (Workshop 1)

The initial phase conceptualized the competition as a system consisting of inputs, processes, outputs, and constraints.

Inputs: The system receives two datasets (training and testing) with 27 attributes and approximately 750 records, the official competition guidelines (mandatory use of R, 10 submission attempts, one graded submission), and the analytical expertise of the participants.

Processes: The workflow involves preprocessing data, training regression models, validating results, and generating predictions. Each of these stages was identified as sensitive to both data variability and methodological choices.

Outputs: The competition requires the generation of .csv prediction files evaluated through the Mean Absolute Error (MAE) metric.

Constraints: The framework operates under computational and procedural limitations (a fixed number of attempts, a performance evaluation relative to other teams, and strict adherence to the platform's format).

This systemic assessment revealed that even minimal changes in certain input variables (such as advertising exposure or socioeconomic indices) can produce substantial variations in predicted sales. Furthermore, randomness inherent in machine learning algorithms and differences in human decision-making introduce additional volatility. As a result, reproducibility emerged as a critical design goal to mitigate systemic instability and chaotic behavior.

3.2 Design Phase (Workshop 2)

Based on the findings from the first workshop, the second phase focused on the **architectural design** of the predictive system, integrating mechanisms to control sensitivity and chaotic behavior. The proposed solution is grounded in systems engineering principles: **modularity**, **traceability**, **feedback**, and **scalability**, aimed at ensuring robustness and transparency of the workflow.

The architecture consists of six main modules:

- **M1 – Data Ingestion Module:** Imports training and test files in `.csv` format, validates schema consistency, and reports missing or inconsistent values.
- **M2 – Preprocessing and Transformation Module:** Performs data cleaning, encoding of categorical variables (e.g., *Tone of Ad*, *Weather*, *Coffee Consumption*), normalization of quantitative variables (e.g., GRP or Temperature), and detection of outliers. Produces a standardized dataset for subsequent analysis.
- **M3 – Feature Analysis and Selection Module:** Conducts correlation analysis and statistical tests such as ANOVA to identify relevant predictors of sales. Results are visualized through heatmaps and variable importance plots.
- **M4 – Modeling Engine:** Designed to compare different regression techniques—linear, regularized, and ensemble—evaluated using the MAE metric. Although models have not yet been implemented, the workflow and evaluation logic are fully defined.
- **M5 – Validation and Control Unit:** Integrates cross-validation and early stopping mechanisms to prevent overfitting and acts as a feedback loop for adjusting preprocessing or modeling parameters.
- **M6 – Output and Reporting Module:** Generates prediction files in the required competition format and produces visual and statistical reports (residual plots, variable importance, etc.).

  The system architecture diagram is shown below:

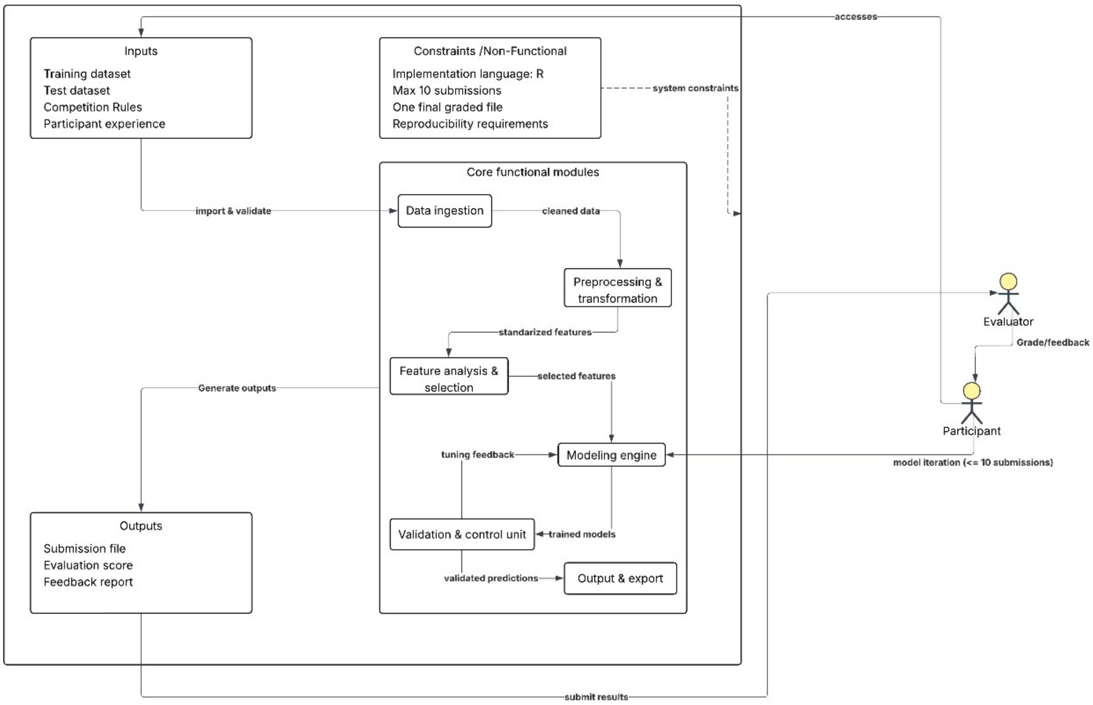

Figure 3.1: High-Level Architecture of the Chocolates 4U Predictive System

Similarly, the data flow diagram in the system is shown below, taking into account the modules named above:

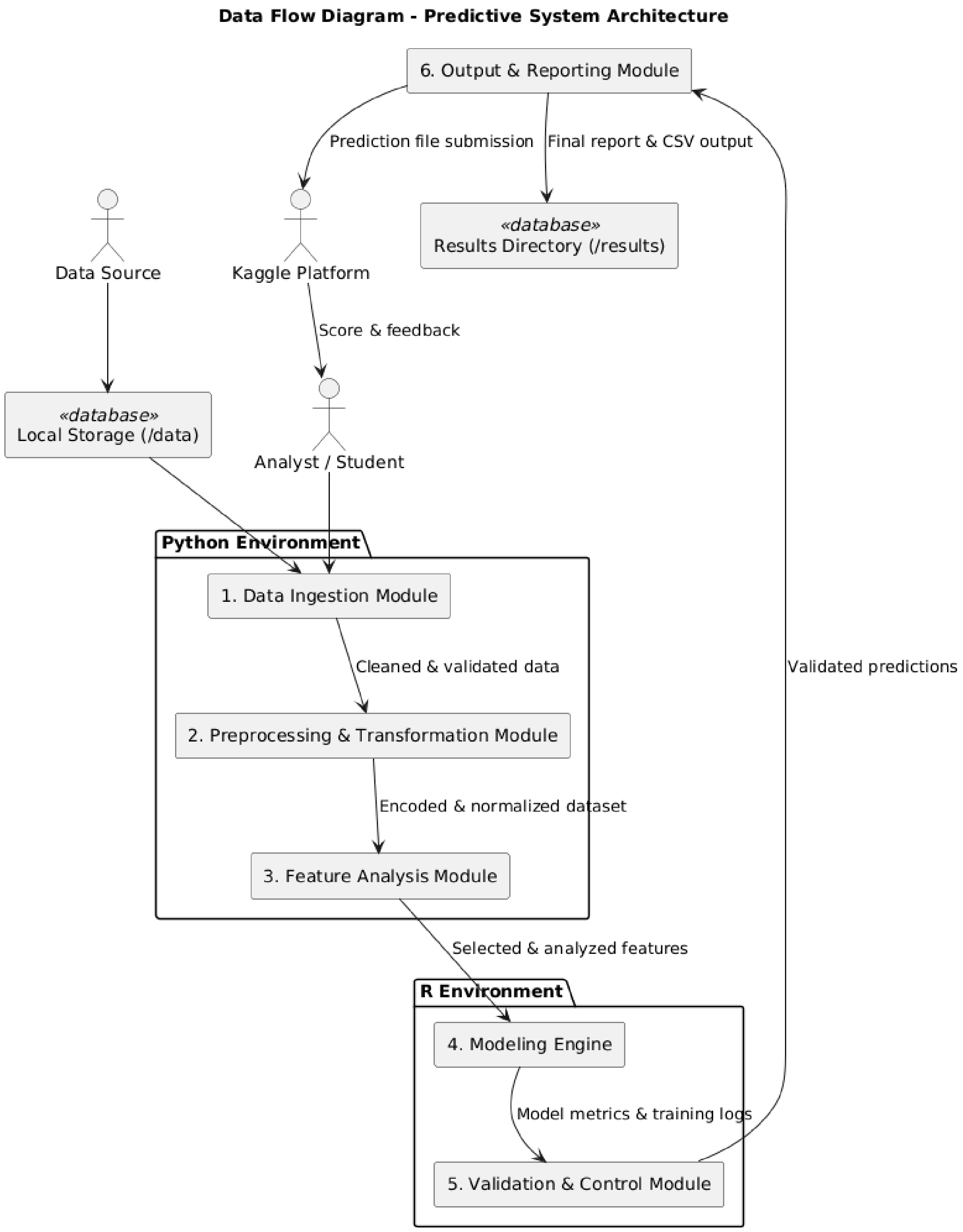
Data Flow Diagram - Predictive System Architecture
6. Output & Reporting Module
Prediction file submission
Final report & CSV output
«database»
Results Directory (/results)
Data Source
Kaggle Platform
Score & feedback
«database»
Local Storage (/data)
Analyst / Student
Python Environment
1. Data Ingestion Module
Cleaned & validated data
2. Preprocessing & Transformation Module
Encoded & normalized dataset
3. Feature Analysis Module
Validated predictions
Selected & analyzed features
R Environment
4. Modeling Engine
Model metrics & training logs
5. Validation & Control Module

Figure 3.2: Flow Diagram of the Chocolates 4U Predictive System

3.3 Requirements Definition

To operationalize the design, a set of **Functional Requirements (FR)** and **Non-Functional Requirements (NFR)** were formalized as user stories, ensuring the system's usability, efficiency, and reproducibility.

Functional Requirements (FR):

- FR1: As an Analyst, I want to upload the training and test .csv data files, so that the system can automatically check for schema consistency, data types, and completeness.
- FR2: As an Analyst, I want to apply normalization routines, categorical variable coding (e.g. ToneofAd) and correlation analysis, so that I can reduce noise and multicollinearity of the variables.
- FR3: As an Analyst, I want to train and compare multiple regression techniques (linear, regularized, ensemble), so that I can internally evaluate which one has the lowest Mean Absolute Error (MAE).
- FR4: As an Analyst, I want the system to use early-stopping mechanisms, so that it can prevent overfitting and ensure that the model generalizes well to new data.
- FR5: As an Analyst, I want the system to record the model configurations (random seeds and validation metrics), so that I can ensure that my results are consistent and 100
- FR6: As an Analyst, I want to export the predictions from my selected final model, so that I can generate a .csv file that complies with the submission format required by the competitor.

Non-Functional Requirements (NFR):

- NFR1: As an analyst, I want each training iteration of a model to complete quickly, so that I can experiment with multiple configurations efficiently.
- NFR2: As an analyst, I want the system to run processes seamlessly and automatically handle data in unexpected formats, so that I can ensure workflow continuity and reliability.
- NFR3: As a Competitive Analyst, I want the architecture to support data sets at least twice the current size without requiring structural modifications, so that I can maintain efficiency in the face of data growth.
- NFR4: As a Competitive Analyst, I want to generate variable importance visualizations and residuals plots, so that I can understand how key predictors influence model predictions.
- NFR5: As an Analyst (or a new team member), I want all procedures to be modular and clearly documented, so that the workflow can be executed without having to modify the source code.
- NFR6: As a Competitive Analyst, I want all data to be processed locally without being sent to external servers, so that I can ensure confidentiality and compliance with security policies.

3.4 Technology Stack and Workflow

The proposed implementation adopts a **hybrid Python–R approach**:

- **Python** is used for exploratory data analysis, visualization, and statistical testing, employing libraries such as `pandas`, `numpy`, `matplotlib`, `scikit-learn`, and `xgboost`.
- **R** is used for regression modeling and validation, utilizing `caret`, `glmnet`, `xgboost`, and `ggplot2`.

Version control and documentation are managed through GitHub to ensure traceability of scripts, configurations, and reports. Overall, the methodological framework emphasizes **reproducibility, transparency, and stability** over raw performance, laying a solid foundation for future experimental and implementation phases.

Chapter 4

Results

The results obtained at this stage are primarily **conceptual**, as the work focused on the **systemic understanding and design** of the predictive framework for the Sweet Regression Competition. The findings derived from Workshops #1 and #2 provide the theoretical and structural basis for subsequent implementation and evaluation phases.

4.1 Systemic Analysis Results

Workshop #1 identified the main **sources of instability and sensitivity** in the system. It showed that small variations in the data, methodological decisions, or evaluation context can lead to substantial differences in performance—a phenomenon characteristic of chaotic behavior in complex systems. These findings justified incorporating **control mechanisms** into the system design, such as fixed random seeds, reproducible preprocessing, and consistent validation procedures.

4.2 Architectural Results from Workshop #2

The second workshop translated theoretical insights into a **structured architecture**, where modules are connected sequentially and communicate through controlled data flows and feedback mechanisms. Each module has a well-defined purpose, maintaining full traceability across the process. This modular approach ensures:

- **Traceability:** All data transformations are logged and reproducible.
- **Feedback:** Validation metrics guide manual and automatic adjustments.
- **Scalability:** Modules can be replaced or extended without affecting the overall flow.

This design reflects the principle of **separation of concerns**, allowing independent improvement of components without compromising the global system.

4.3 Sensitivity Control Mechanisms

To mitigate chaotic tendencies, several stabilization mechanisms were defined:

- Fixed random seeds to guarantee consistent results across runs.
- Regularization techniques (Lasso, Ridge) to minimize multicollinearity effects.

- Standardized preprocessing to ensure consistent encoding and scaling.
- Early stopping to prevent overfitting during training.
- Outlier detection to reduce distortion in regression coefficients.
- Error logging and rollback procedures to maintain operational stability.

These mechanisms constitute the theoretical foundation for balancing predictive performance and systemic stability during implementation.

Chapter 5

Discussion and Analysis

The Sweet Regression Competition project aimed to design a reproducible and modular predictive system capable of generating consistent sales predictions under strict competition constraints. These results highlight the integra- tion of systems thinking with software modeling principles and illustrate how an engineering perspective can be applied to analytical competitions. The analysis revealed that the competition behaves as a highly sensitive and complex sys- tem, where small variations in data, preprocessing techniques, or modeling configurations can produce large deviations in output predictions. The architectural framework proposed mitigates these sources of instability through modular design, version control, and reproducibility mechanisms.

5.1 Significance of the Findings

The significance of this study lies in its systematic approach to reproducibility and sensitivity control. This project establishes a structure where traceability, feedback, and control are embedded into every stage of the workflow. The modular architecture enables flexibility: individual components (such as feature se- lection or regression modeling) can be replaced or extended without redesigning the entire pipeline. The explicit inclusion of random seed control, configuration logging, and standard- ized preprocessing steps ensures that every experiment can be replicated precisely, an essential factor for academic integrity and real-world deployment. Although numerical metrics such as MAE were not yet computed, the conceptual foundation directly contributes to future stages where reproducible experimentation will lead to measurable performance improvements.

5.2 Summary

In summary, this chapter analyzed the conceptual results of the Sweet Regression framework and discussed their implications. The findings highlight the importance of reproducibility, traceability, and stability as design priorities within predictive modeling. While current limitations relate to the absence of empirical testing, the theoretical foundation and modular architecture provide a strong basis for controlled implementation and quantitative evaluation in the next project phase.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This project investigated the design of a predictive and reproducible system for the Sweet Regression Competition. A structured, traceable, and scalable workflow was established, capable of producing consistent predictions under conditions of uncertainty and with a limited number of participants. Applying the principles of systems analysis, the study decomposed the competition into interdependent components (inputs, processes, outputs, and constraints), revealing the chaotic and sensitive nature of the task.

The subsequent architectural design introduced six key modules encompassing data ingestion, preprocessing, feature selection, modeling, validation, and reporting. Each module was defined with clear functional and non-functional requirements, ensuring that the system can be developed, tested, and modified in a controlled manner.

Overall, this work connects systems engineering with data science, illustrating how principles such as modularity, feedback, and version control can transform analytical processes into reproducible systems.

6.2 Future Work

Future work will build on the conceptual design to implement, test, and evaluate the predictive framework in practice. Immediate next steps include:

- **Pipeline Implementation:** Develop the six defined modules in Python and R, ensuring full integration through configuration files and seed control.

- **Model Training and Validation:** Apply and compare multiple regression algorithms (Lasso, Ridge, Random Forest, XGBoost) using MAE as the primary metric, supported by cross-validation.

- **Performance Benchmarking:** Measure computational efficiency, scalability, and reproducibility under different dataset sizes and system environments.

- **Sensitivity Experiments:** Introduce controlled perturbations to input variables to evaluate the system's stability and sensitivity to noise.

- **Documentation and Deployment:** Prepare full documentation and release the project on GitHub.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[2] J. D. Hunter, F. Pérez, and B. E. Granger, “Interactive computing and reproducible research using Python: Lessons from data-driven competitions,” *Computing in Science amd Engineering*, vol. 13, no. 2, pp. 45–51, 2011.

[3] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.