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
Department of Computer Engineering
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
{saljureb, cabarrigag, dsaldanag, jdalvarezc}@udistrital.edu.co

The Sweet Regression competition challenges participants to develop a predictive system capable of estimating chocolate sales as the company expands into new regional markets. The problem centers on modeling the relationship between advertising strategies, regional conditions, and consumer behavior using the provided training datasets. The proposed solution involves implementing a regression-based predictive model in R, whose performance is evaluated through accuracy metrics on unseen test data to identify the most effective predictive configuration.

***Index Terms*—System analysis, predictive modeling, regression systems, reproducibility, systems engineering, software design, educational projects**

I. Introduction

Predictive modeling has become a cornerstone of data-driven decision-making in both academic and industrial contexts. According to Hastie, Tibshirani, and Friedman, predictive models enable structured inference from data while balancing bias, variance, and interpretability in real-world applications [1]. Competitions such as the Sweet Regression Competition serve as pedagogical strategies to apply theoretical concepts of systems analysis, software design, and data science within controlled experimental environments. Similar data-driven challenges have proven effective in promoting experiential learning and reproducible research practices in engineering education [2].

In this specific case, participants are tasked with predicting chocolate sales for a fictional company, Chocolates 4U, using regression techniques implemented in R. The challenge involves processing marketing exposure variables, regional socio-economic factors, and consumer behavior indicators to estimate future sales.

From a systems engineering perspective, such a competition represents a complex and dynamic system characterized by multiple interacting components and feedback loops. Workshop 1 focused on understanding the competition as a system defined by its inputs, processes, outputs, and constraints. The analysis revealed high sensitivity to data variations, algorithmic randomness, and human decision-making in preprocessing and model selection. These factors contribute to chaotic behavior, where small perturbations in input data or methodological choices can significantly alter predictive outcomes.

Several systemic constraints further intensify this complexity: the mandatory use of the R programming language, a strict limit of ten submission attempts, and relative performance evaluation based on peer comparison. These restrictions highlight the importance of efficiency, reproducibility, and methodological rigor when designing analytical workflows.

In response, Workshop 2 transitioned from analysis to design, proposing a modular architecture that incorporates systems engineering principles such as modularity, traceability, feedback control, and scalability. The resulting framework emphasizes controlled experimentation, reproducible preprocessing, and transparent model evaluation, even under constrained testing conditions. This evolution from systemic understanding to architectural specification demonstrates how analytical insights can directly inform the engineering design of predictive systems in educational settings.

The present paper consolidates the findings of both workshops, describing the conceptual analysis and the proposed system design for the Sweet Regression Competition. It does not include model execution or experimental results; rather, it documents the methodological approach and design foundations required to support future implementation and evaluation phases.

We used Python for data analysis due to its extensive ecosystem of specialized libraries. Tools such as Pandas, Matplotlib, Statsmodels, Scikit-learn, and NumPy enable advanced statistical analyses to be performed efficiently and reproducibly [3]. In addition, Python offers high readability and flexibility, despite not being as fast as some other programming languages. Its comprehensive documentation and broad community support also make it easier to explore, compare, and identify the most effective analytical and modeling techniques for the project.

Among the various computational techniques employed during the development of the competition, the ANalysis Of VAriance (ANOVA) was included as a fundamental tool to contrast and validate the results obtained by the predictive models. Alongside this, the Bellman rule was applied to determine the optimal intervals between data points, allowing the identification of the most sensitive variables and thus optimizing the process. This facilitated a more precise understanding of the influence of each factor on sales predictions, providing robustness to the modeling process and enhancing

the system's ability to generalize results.

In addition, this competency demonstrates the value of integrating systems engineering methodologies into data science education. By framing predictive modeling as a system-of-systems problem, participants are encouraged to think beyond code and algorithms, considering structure, interdependence, and process optimization. This approach fosters a deeper understanding of how analytical systems can be designed, validated, and evolved within constrained environments, preparing participants for the multifaceted challenges of data-driven engineering in the real world.

Similarly, the project promotes the development of essential soft skills for modern professional practice, such as teamwork, uncertainty management, decision-making under pressure, and effective communication. The combination of theory, simulation, and guided application not only enhances the internalization of technical concepts but also encourages critical reflection on the ethical and operational boundaries of predictive model use in real-world scenarios. In this sense, the competition stands as an integrative experience that bridges scientific rigor with engineering applicability.

II. Methods and Materials

The methodological approach adopted in the *Sweet Regression Competition* project was divided into two complementary phases. The first phase (Workshop #1) applied a systemic perspective to analyze the competition as a complex system composed of interacting components and sensitive feedback loops. The second phase (Workshop #2) transformed this analytical understanding into a structured system design, defining architecture, requirements, and implementation strategies to ensure reproducibility and stability in subsequent modeling tasks.

A. Systemic Analysis (Workshop #1)

The competition was conceptualized as a system defined by four primary components: **inputs**, **processes**, **outputs**, and **constraints**.

- **Inputs:** Training and test datasets containing 27 variables and 750 records; competition rules (mandatory use of R, ten submission limit, one final graded submission); and the participants' prior analytical knowledge.
- **Processes:** Data preprocessing, regression model training, validation, and prediction generation. Each process was identified as highly sensitive to data variability and preprocessing decisions.
- **Outputs:** Prediction files in `.csv` format submitted to a competitive platform, evaluated through the Mean Absolute Error (MAE) metric.
- **Constraints:** Limited attempts, computational complexity, and relative evaluation against peers rather than absolute benchmarks.

The analysis revealed that these elements interact dynamically, creating systemic instability. Small perturbations in variables (e.g., marketing indicators or socio-economic indices) could produce large deviations in predicted sales. In addition, randomness inherent to stochastic algorithms and human variability in analytical decisions amplify system sensitivity, making reproducibility a central design concern.

B. Design Phase (Workshop #2)

Building upon the findings of Workshop #1, the second phase focused on the architectural design of a predictive system that integrates mechanisms to control sensitivity and chaotic behavior. The proposed solution follows systems engineering principles—*modularity, traceability, feedback control, and scalability*—to ensure robustness and transparency.

The system architecture is composed of six core modules (Fig. **??**):

- **M1: Data Ingestion Module:** Imports training and test datasets in `.csv` format, validates schema consistency, and reports missing or inconsistent data.
- **M2: Preprocessing and Transformation Module:** Cleans and encodes categorical variables (e.g., *Tone of Ad*, *Weather*, *Coffee Consumption*), normalizes quantitative attributes (e.g., GRPs, temperature), and detects outliers. Produces a standardized dataset for subsequent analysis.
- **M3: Feature Analysis and Selection Module:** Performs correlation analysis and statistical tests such as ANOVA to identify relevant predictors of sales. Results are visualized through correlation heatmaps and feature-importance plots.
- **M4: Modeling Engine:** Designed to compare multiple regression techniques—linear, regularized, and ensemble-based—evaluated with the MAE metric. Though models were not implemented at this stage, the architecture defines the process flow and evaluation logic.
- **M5: Validation and Control Unit:** Integrates cross-validation and early-stopping procedures to detect overfitting and maintain generalization. It serves as a feedback mechanism to adjust preprocessing or model parameters.
- **M6: Output and Reporting Module:** Exports prediction files in the competition's required format and generates interpretable visual reports (e.g., residual plots, feature importance charts).

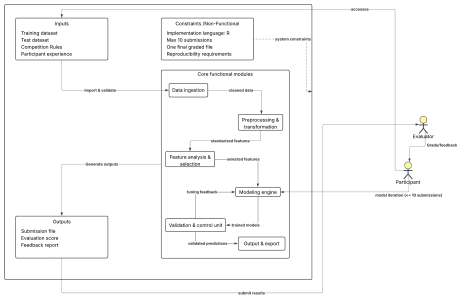

Fig. 1. System's diagram

C. Requirements Definition

To operationalize the design, a series of functional (FR) and non-functional (NFR) requirements were formalized as user stories. These requirements ensure that the system remains usable, efficient, and reproducible. Key examples include:

- **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. Tone_of_Ad) 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% reproducible.
- **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 (NFRs) emphasize performance, interpretability, and security:

- **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.

D. Technical Stack and Workflow

The proposed implementation adopts a hybrid software stack:

- **Python** for analytical exploration, visualization, and statistical testing, employing libraries such as `pandas`, `numpy`, `matplotlib`, `scikit-learn`, and `xgboost`.
- **R** for regression modeling and validation, using `caret`, `glmnet`, `xgboost`, and `ggplot2`.

Version control and documentation are maintained through GitHub, ensuring that preprocessing scripts, configurations, and reports remain traceable and reproducible.

Overall, the methodological framework prioritizes reproducibility, transparency, and stability over performance, laying the conceptual groundwork for subsequent implementation and experimentation phases.

III. Results & Discussion

The results of the project are conceptual rather than experimental, as the work conducted up to this stage focused on the analytical understanding and system design of the *Sweet Regression Competition*. The findings derived from Workshops #1 and #2 establish a theoretical and structural foundation for future model implementation and evaluation.

A. Conceptual Outcomes from Systemic Analysis

Workshop #1 identified the key sources of instability and sensitivity within the competition system. These findings are summarized in Table I. The analysis demonstrated that even small variations in data, modeling choices, or evaluation context can lead to large differences in performance, a phenomenon consistent with chaotic behavior in complex systems.

TABLE I
SUMMARY OF SENSITIVITY SOURCES IDENTIFIED IN WORKSHOP #1

Source of Sensitivity	Effect on System Behavior
Data fluctuations (e.g., GRP variations)	Significant deviation in predicted sales values.
Algorithmic randomness	Different results from identical data without fixed seeds.
Preprocessing variability	Divergent outputs depending on scaling or encoding methods.
Evaluation dependency	Relative scoring leads to inconsistent performance comparison.
Human decision-making	Analytical choices introduce non-reproducible effects.

These systemic observations justified the inclusion of control mechanisms in the system design—such as fixed random seeds, reproducible preprocessing, and consistent validation procedures—to mitigate chaotic behaviors.

B. Architectural Results from Workshop #2

Workshop #2 transformed the theoretical findings into a structured system architecture. The final design, shown conceptually in Figure 2, integrates sequential modules connected through a controlled data flow and feedback mechanisms. Each module performs a well-defined task while maintaining traceability across stages.

The modular approach ensures:

- **Traceability:** Every data transformation is logged and reproducible.

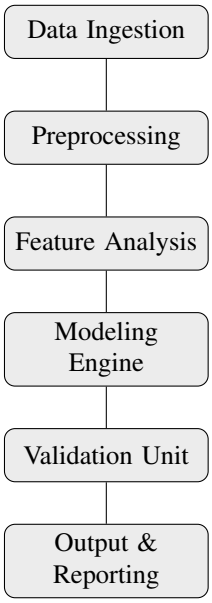

Fig. 2. Sequential module flow of the predictive system architecture.

- **Feedback control:** Validation metrics guide manual adjustments.
- **Scalability:** Modules can be replaced or expanded without altering the overall workflow.

This design reflects the principle of separation of concerns—data handling, feature engineering, and modeling are functionally isolated—allowing future developers to focus on improving one component without disrupting others.

C. Sensitivity Control Mechanisms

To address chaotic tendencies identified earlier, Workshop #2 proposed a series of stabilization techniques summarized in Table II. These mechanisms are preventive in nature, ensuring that future implementations remain stable and reproducible even when model complexity increases.

TABLE II
SENSITIVITY AND CHAOS CONTROL MECHANISMS IN THE SYSTEM DESIGN

Mechanism	**Intended Effect**
Fixed random seeds	Ensure reproducibility across runs.
Regularization (Lasso, Ridge)	Reduce the impact of correlated predictors.
Standardized preprocessing	Guarantee consistent feature scaling and encoding.
Early-stopping	Prevent model overfitting during training.
Outlier detection	Limit distortion in regression coefficients.
Error logging and rollback	Preserve stability during execution failures.

Although no models were executed, these mechanisms form the theoretical foundation for maintaining equilibrium between predictive power and systemic stability once implementation begins.

D. Expected Design Benefits

The proposed architecture and requirements yield several anticipated benefits:

- **Reproducibility:** Fixed random seeds and logged configurations ensure identical results for identical inputs.
- **Interpretability:** Planned visual outputs (correlation maps, feature importance plots, residual charts) promote transparency in model evaluation.
- **Efficiency:** The system is structured to complete model training in under five minutes on standard hardware, meeting performance requirements.
- **Security:** All data processing is conducted locally, ensuring confidentiality and compliance with privacy policies.

E. Discussion

The integration of systems analysis with software design principles provides a methodological bridge between theoretical understanding and practical system development. While Workshop #1 highlighted the complexity and chaotic nature of the competition, Workshop #2 demonstrated that disciplined architectural design—grounded in modularity, feedback, and traceability—can convert instability into a manageable and structured process.

These conceptual results do not measure predictive accuracy but rather demonstrate preparedness for a controlled implementation phase. The framework thus serves as a blueprint for future work, where empirical validation and statistical modeling will complete the engineering cycle envisioned in the initial systemic analysis.

IV. CONCLUSIONS

The *Sweet Regression Competition* project established a methodological foundation that links systems thinking with predictive modeling design. Through the systemic characterization of the competition, the study identified the main sources of variability and instability affecting data-driven decision processes. These findings informed the formulation of an architectural framework grounded in modularity, feedback control, and traceability, ensuring that future implementations can maintain methodological rigor under uncertainty and data sensitivity.

Rather than measuring predictive accuracy, this stage emphasized the engineering discipline required for reproducibility and transparent workflow design. The resulting framework provides a structured path for subsequent implementation and validation, where the proposed modules and control mechanisms can be empirically tested. In doing so, the project advances from conceptual design toward a reproducible predictive system capable of integrating analytical precision with systemic stability.

ACKNOWLEDGMENTS

The authors express their gratitude to the Department of Computer Engineering at Universidad Distrital Francisco José de Caldas for providing the academic framework and guidance necessary to develop this project. Special thanks are extended

to the Systems Analysis and Design course instructor for their continuous support and for fostering a methodological approach that integrates engineering principles with data-driven problem solving.

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 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.