A Systems Analysis Approach to Predictive Modeling: A Case Study on the Titanic Dataset

Andres C. Ramos Rojas
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: ancramosr@udistrital.edu.co

Andres M. Cepeda Villanueva
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: amcepedav@udistrital.edu.co

Jhonatan D. Moreno Barragán
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: jdmorenob@udistrital.edu.co

Julian Carvajal Garnica
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: jcarvajalg@udistrital.edu.co

***Abstract*—This paper presents the application of systems analysis and design principles to a predictive modeling problem, utilizing a well-known Kaggle competition as a case study. The primary objective is to deconstruct the system of factors influencing a specific outcome and refine a baseline predictive model. Initial analysis reveals that the model's predictive accuracy is compromised, a deficiency attributed to data integrity issues and suboptimal parameter weighting. The proposed methodology involves a sensitivity analysis to identify the system's most influential components and a rigorous data validation process. This approach aims to re-evaluate the significance of each system variable, leading to a more robust and accurate predictive model.**

***Index Terms*—Systems Analysis, Predictive Modeling, Machine Learning, Data Integrity, Sensitivity Analysis, Feature Engineering.**

I. Introduction

The principles of Systems Analysis and Design [1] offer a structured framework for comprehending complex systems by identifying their components, interactions, and environmental influences. This study applies this framework to the domain of predictive modeling, using the historical dataset of the RMS Titanic disaster as a practical case study. This dataset, provided through a Kaggle competition, serves as a controlled environment for developing and evaluating models that predict passenger survival.

The system under investigation is defined by a set of variables—suchas passenger demographics, socio-economic status, and travel details—and their intricate interrelationships, which collectively determine a binary outcome: survival or non-survival. The competition formalizes this problem by providing two primary data artifacts: a training set (‘train.csv‘), where passenger features are correlated with the known survival outcome, and a test set (‘test.csv‘), which lacks this ground truth. This data partitioning establishes the central predictive task: to develop a model based on the training data that can accurately generalize to the unseen test data.

To facilitate this task, the Kaggle platform provides a comprehensive environment beyond the datasets. This includes interactive computational 'Notebooks' [2] (in-browser IDEs with pre-installed data science libraries), extensive community-driven tutorials and discussions [3], and a baseline submission file (‘gender_submission.csv‘). This baseline, which posits that all females survive and all males perish, serves as a simplistic initial model, establishing a benchmark for performance. The competition explicitly defines the system's primary evaluation metric as classification accuracy—the percentage of correct predictions.

A critical component of the competition environment is the evaluation mechanism, which consists of a public and a private leaderboard. Submissions are scored against a fixed subset of the test data (the public leaderboard), providing real-time feedback. However, the final system ranking is determined by performance on the remaining, unseen portion of the test data (the private leaderboard). This structure itself is a complex system dynamic, introducing a feedback loop where models that are over-fitted to the public leaderboard may fail in the final evaluation.

The core objective of this study is to deconstruct this entire predictive system—comprising the data artifacts, the computational environment, and the evaluation feedback loop—to analyze its constituent elements and their relative importance. A key focus is on assessing the system's sensitivity to its input parameters (features), particularly in the context of incomplete or inconsistent data. Our approach treats the competition framework not merely as a contest, but as a laboratory for the practical application and validation of systems analysis methodologies.

II. Methods and Materials

In this section, you will find which resources were used to carry out the analysis of the competition and understand the environment proposed by it as a complex system of elements that includes all kinds of relationships, in order to arrive at a more accurate prediction.

A. System Definition and Architecture

The first approach was to identify what makes up the Kaggle platform, what can be interacted with, and how the system will react to different interactions. With this in mind, we started with a diagram of the architecture it presents.

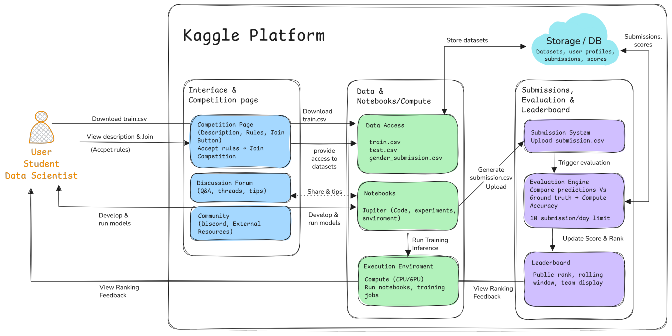

Fig. 1. Architecture of the Kaggle competition environment.

Figure 1 illustrates the architecture of the Kaggle competition environment. Competitors access the competition page, download datasets, train models in notebooks, and submit predictions for evaluation. The system computes accuracy, updates leaderboards, and integrates community interaction via forums and repositories.

User: The user will be an ordinary person regardless of their occupation; in reality, they only need to have an interest in the world of systems analysis, with access to practically the entire system except for certain interactions that will be carried out internally by the system.

Interface & competition page: This module allows the user to interact directly with the competency environment, meaning that they are allowed to access the competency and view all the necessary information to understand the environment. Additionally, there are community environments that allow users to ask questions about the competencies.

Data & Notebooks/Compute: This module allows the user to dive into the competition, accessing the data that will be used, the testing environment, and resources such as the notebooks provided by Kaggle for a better understanding of the competition.

Storage/DB: This module is not directly accessible to the user but interacts with them indirectly by sending information that will be stored for later viewing, as in the leaderboard or in the different saved files.

Submissions, Evolution & Leaderboard: This module allows the user to finalize their attempt in the competition by submitting the presented model to improve the prediction algorithm, as well as assigning a ranking by comparing it with the other submitted solutions.

B. Data Preprocessing and Integrity Analysis

A foundational principle of systems analysis dictates that unexamined assumptions or logical flaws within a system's initial state can propagate and amplify errors, leading to significant output degradation. In the context of predictive modeling, these logical flaws often manifest as data integrity deficiencies.

Our preliminary audit of the dataset confirms the presence of such deficiencies, including missing values (e.g., in the 'Age' and 'Cabin' features) and potential outliers, which compromise the integrity of the input data. These data voids are not benign; they introduce ambiguity and force the predictive algorithm to operate on an incomplete and potentially skewed representation of the system.

Furthermore, the system's behavior includes elements initially classified as stochastic or "random." A rigorous systems approach mandates that we challenge this assumption. We, therefore, propose a deeper investigation into these seemingly random events. The objective is to determine if they are truly stochastic or if they represent complex, non-linear interactions between variables that have not yet been modeled. By identifying and mapping these latent relationships, we can begin to transform these elements from unpredictable noise into deterministic or probabilistic components, thereby enhancing the overall fidelity of the model.

C. Sensitivity Analysis and Feature Engineering

The predictive model is a dynamic system where the output (survival) is not equally dependent on all inputs. A critical flaw in a baseline model is the failure to correctly weight the influence of each component. Sensitivity analysis is the formal methodology employed to systematically quantify these differential impacts. This process involves perturbing individual input variables and measuring the corresponding change in the model's output, allowing us to identify the features to which the system is most "sensitive."

Our analysis focuses on validating whether the machine learning algorithm's internal reaction mechanisms—its learned parameters or feature importances—align with the empirically-determined sensitivity of the system. A discrepancy implies a logical flaw: the model may be "over-reacting" to noise (overfitting) or "under-reacting" to a critical predictor variable. This validation is essential for guiding subsequent feature engineering, where we will refine, combine, or construct new features that more accurately represent the true dynamics of the system, ultimately leading to a more robust and accurate predictive algorithm.

III. Results & Discussion

IV. Conclusions

Acknowledgments

References

[1] https://sebokwiki.org/wiki/System_Analysis#Principles_Governing_System_Analysis.

[2] Kaggle, "Titanic - Machine Learning from Disaster," https://www.kaggle.com/competitions/titanic/code, accessed: 2025-10-24.

[3] ——, "Titanic - Machine Learning from Disaster," https://www.kaggle.com/c/titanic, accessed: 2025-10-24.