Technical Report Final

Kaggle Systems Engineering Analysis

Loan Default Prediction

Team #9

Juan Esteban Avila Trujillo – 20251020054
Juan Jose León Gomez – 20212020055
Juan Pablo Díaz Ricaurte – 20222020076
Miguel Angel Hernández Medina – 20222020035

Institution:

Universidad Distrital Francisco José de Caldas
Systems Engineering, Bogotá, Colombia

Classification: Technical Research Report

1 Summary

This report presents the development of a loan default prediction and financial loss estimation system, following a rigorous Systems Engineering approach. The main objective is to design a robust and scalable system capable of addressing the dual challenge of binary classification (identifying loans that will default) and loss regression (estimating the monetary magnitude of the loss) on a massive, highly dimensional dataset from Kaggle's Loan Default Prediction competition. The initial analysis revealed that the prediction task is inherently complex and susceptible to chaotic effects due to data-quality issues, non-linear interactions among variables, and feedback loops typical of financial systems. Based on these findings, critical functional and non-functional requirements were defined (including scalability and resiliency), and a modular, traceable, and reproducible high-level architecture was proposed. Key design decisions were implemented—such as version-controlled preprocessing pipelines, model ensembles, and specialized monitoring modules—to mitigate sensitivity to perturbations and ensure stable performance under the primary evaluation metric, Mean Absolute Error (MAE). Experimental results demonstrated the accuracy paradox in imbalanced data (high overall accuracy but zero detection of the minority class) and confirmed the system's technical resilience under noisy data, while also exposing the need to improve sensitivity to credit risk. This work provides a structured framework, guided by engineering principles, for developing reliable machine learning solutions in volatile financial environments.

2 Introduction

In the financial industry, predicting loan default is a critical problem traditionally addressed as a binary classification task (determining whether a borrower will be "good" or "bad") in order to reduce losses by avoiding defaults. However, Kaggle's "Loan Default Prediction" competition (Imperial College London) poses an additional challenge: besides predicting whether a default will occur, it requires estimating the loss amount associated with that default for each loan. In other words, it combines a classification problem with a loss regression problem, adding complexity to both analysis and performance evaluation. This dual objective requires a comprehensive Systems Engineering approach to build a robust and scalable solution, since the system must balance accuracy in detecting defaults with accuracy in quantifying losses.

The dataset provided by the competition contains more than 200,000 observations with around 800 variables, which are anonymized and preprocessed. Anonymization means there is no semantic context for the features, creating a technical challenge: participants must rely on statistical techniques and feature engineering rather than specific business knowledge to extract reliable predictive signals. Additionally, the dataset shows a strong class imbalance (approximately 91% solvent loans vs. 9% default loans), which anticipates difficulties for traditional models that may achieve high overall accuracy by ignoring the minority class. The competition provides separate files (train.csv and test.csv) and requires generating a submission file with loss predictions for each loan in the test set. Performance is evaluated using Mean Absolute Error (MAE) on estimated losses. This metric measures the average absolute error in the numeric loss estimate, encouraging models that minimize error magnitude regardless of direction.

Under these conditions, small changes in the quality or distribution of input data can significantly affect predictions, showing sensitivity to initial conditions and complex behavior. Therefore, a systemic perspective supported by Chaos and Complexity Theory principles was adopted, assuming that even minimal perturbations could propagate and alter results if the system lacks robustness. To address this challenge, this work details the implemented methodology, the technical architecture of the designed system, and the strategies used to mitigate the dataset's inherent challenges. In particular, emphasis was placed on building a modular and reproducible system capable of maintaining consistent predictive performance under the main evaluation metric (MAE), while exploring techniques to handle data imbalance and reduce sensitivity to perturbations. Next, a review of relevant literature approaches is presented, followed by the project context and scope, the proposed methodology, the results obtained, and the conclusions and recommendations derived from this work.

3 Literature Review

Previous studies and recognized approaches in credit risk and loan default prediction have explored various techniques and considerations that provide the foundation for this competition:

- **Baseline models (Logistic Regression, Decision Trees):** Typically used as starting points due to simplicity and high interpretability. However, they often have lower predictive power compared to more advanced methods, especially in problems with many attributes.

- **Ensemble models (Random Forest, Gradient Boosting Machines):** These are the most effective and prevalent techniques for tabular data. Boosting algorithms such as LightGBM or XGBoost have shown high accuracy and the ability to capture complex, non-linear interactions among variables, generally outperforming simple individual models.

- **Deep models (Neural Networks):** Useful for capturing extremely complex relationships between features. While neural networks can model high-level interactions, they require much larger data volumes and more delicate tuning; in tabular competitions, they often underperform tree ensembles unless the pattern complexity truly warrants them.

- **Feature engineering:** Widely recognized as one of the most critical processes in this domain. It involves creating new highly predictive variables (e.g., financial ratios like debt/income, aggregated temporal variables, differences between highly correlated features, etc.) to incorporate business knowledge or hidden patterns. Effective feature engineering can substantially improve model performance, especially when original variables are anonymized or do not directly explain borrower behavior.

- **Evaluation metrics (ROC-AUC, Precision-Recall, MAE):** Essential tools to measure and balance performance, especially under class imbalance. ROC-AUC is commonly used to evaluate a binary classifier's ability to rank predictions independently of a threshold; however, in highly imbalanced datasets it can be misleading because it is dominated by majority-class performance. Precision-Recall curves and the F1 metric provide more faithful insight into minority-class performance by emphasizing detection capability (recall) and alert accuracy (precision). In this competition, MAE is the main metric for loss regression. Since a trivial strategy that always predicts "zero loss" could achieve a seemingly low MAE but detect no defaults, it is critical to balance regression metrics with classification metrics to ensure practical value.

- **Competition-specific constraints (Anonymized features):** All features have been transformed and anonymized, forcing reliance on statistical feature selection, automated methods, and algorithmic robustness. This adds uncertainty, since generic variable names (f1, f2, . . .) may hide valuable information that must be discovered through exploratory analysis, dimensionality reduction, and experimentation.

In summary, the literature and prior experience suggest that a hybrid model may be appropriate: combining advanced learning methods (ensembles) with careful feature engineering and using appropriate metrics to deal with imbalance. They also highlight the need for a robust system capable of handling missing context and the potentially chaotic nature of financial data.

4 Background and Context

Traditional credit risk management has historically focused on binary default prediction, classifying borrowers as "good" or "bad" to mitigate financial risk. This classic approach aims to minimize expected losses by reducing exposure to loans with high default probability, typically using metrics such as delinquency rates and credit scoring techniques to support decision-making. However, in a modern financial optimization context, this view is insufficient. It is not enough to predict who will default; it is equally crucial to quantify the severity of the loss in case of default, in order to estimate capital provisions and recovery strategies.

This competition provides a large historical loan dataset with preprocessed and anonymized variables and proposes a dual challenge of classification + regression. Additionally, MAE-based evaluation implies that a model can achieve a low average error even by ignoring default cases if defaults are few. This phenomenon, known as the *accuracy paradox*, warns that optimizing only the global metric can produce solutions that are not useful in practice.

Furthermore, chaos engineering principles were considered in the data science context. In financial systems, small input fluctuations can cause significant output variations due to non-linear interdependencies. This project incorporates that concept through robustness tests in which random perturbations are introduced into the data to evaluate model stability. A well-designed system must be resilient to noisy or incomplete data so that performance does not degrade catastrophically in adverse scenarios.

5 Project Scope

Based on competition requirements and team objectives, the system scope was defined clearly:

5.1 Dual prediction per record

For each loan in the test set, the system must generate two interrelated outputs:

- The binary probability of default, i.e., the estimated probability that the loan will default.
- The estimated financial loss in case of default, expressed as a continuous amount (regression).

5.2 Complete end-to-end pipeline

The system constitutes a full end-to-end machine learning solution, covering all stages from raw data ingestion and cleaning to model training and generation of the final prediction file (submit.csv).

5.3 Evaluation of advanced modeling techniques

The solution focuses on tree-based ensemble methods (e.g., LightGBM, XGBoost). No external data sources are incorporated, in compliance with Kaggle rules and to preserve reproducibility.

6 Assumptions

The following assumptions were adopted:

- The provided train/test data are clean aside from missing values.
- Anonymized variables maintain consistent statistical relationships over time.
- The test distribution is similar to the train distribution (no strong concept drift).
- Kaggle evaluation depends solely on the submitted predictions (submit.csv).

7 Limitations

Key limitations include:

- **Feature interpretability:** anonymization and high dimensionality prevent attribute-level semantic interpretation.
- **No external data:** only competition-provided data are used.
- **Metric limitation:** MAE does not capture asymmetric business costs (underestimation vs. overestimation).

8 Methodology

The methodology followed a structured Systems Engineering approach focused on robustness, reproducibility, and mitigation of credit risk complexity. Development was divided into: modular architecture design, data preparation, modeling/validation, and inference generation.

8.1 System Architecture and Modular Design

To ensure maintainability and robustness, the system was implemented modularly. Each main pipeline function was encapsulated in independent modules.

Module	**Role and key engineering decision**
`ingestion.py`	Loads raw data (train/test), detects target (loss), removes IDs to prevent leakage, controls data types. Key decision: forced string conversion for mixed-type categorical columns (fail-fast typing).
`preprocessing.py`	Builds scikit-learn transformation pipeline; imputes, scales, encodes categoricals. Key decision: replaced one-hot encoding with ordinal encoding to reduce dimensionality and avoid out-of-memory failures.
`model.py`	Trains the main model; Random Forest used for simulation baseline. Key decision: fixed global seed (`random_state=42`) for reproducibility.
`chaos.py`	Introduces controlled perturbations to simulate noisy/corrupted inputs and measure sensitivity. Key decision: supports robustness and stress testing via sensitivity analysis.
`main.py`	Orchestrates end-to-end flow: ingestion, preprocessing, training, evaluation, and chaos tests. Key decision: centralized coordination improves traceability and integration guarantees.

Table 1: System modules and key engineering decisions.

8.2 Data Preprocessing

The data processing pipeline emphasized consistent transformations and reproducibility:

- **Missing values:** numerical median imputation; categorical "Unknown" or frequency-based imputation. Training-derived parameters were reused on test.
- **Normalization and scaling:** applied standardization and other transformations as appropriate to prevent dominance by high-magnitude features.
- **Feature engineering:** created ratios/combinations, performed redundancy reduction via correlation filtering, variance checks, and evaluated dimensionality reduction approaches.

8.3 Modeling and Validation

Modeling focused on robust generalization:

- **Model selection:** tree ensembles (LightGBM, XGBoost) for final competition; Random Forest as simulation reference.
- **Stratified k-fold cross-validation:** to obtain stable performance estimates while preserving class imbalance ratios.
- **Fixed randomness:** controlled seeds across the pipeline to ensure repeatability.
- **Regularization and early stopping:** to reduce overfitting in boosting models and control complexity in Random Forest.
- **MAE vs. business cost awareness:** explored weighting to reduce false negatives in defaults (e.g., `class_weight='balanced'`), while keeping MAE as the main objective.

8.4 Inference and Result Generation

Inference applied the same preprocessing pipeline to test data and generated final loss predictions:

- The trained transformations were applied to `test.csv` to produce the final feature matrix.
- Predictions were generated using a trained model or ensemble. In a two-stage approach, the workflow is:

  $$P(default) \;\; \rightarrow \;\; loss \mid \widehat{default}$$

  For example, loss modeling may use a transformation such as:

  $$\log(1 + loss)$$

  followed by an inverse transformation back to the original scale.
- A `submit.csv` file was generated in the required format (ID and predicted loss).

9 Simulation Results and Analysis

An experimental simulation was conducted using a sample of 50,000 training records.

9.1 Technical performance and stability

The system processed 50,000 records with 769 features without memory issues or major delays. Ordinal encoding effectively reduced dimensionality and enabled local training under limited resources. No severe I/O bottlenecks or memory errors were observed.

9.2 Predictive performance: the accuracy paradox

The Random Forest achieved 91.22% overall accuracy, but recall for the default class was 0.00. The model effectively predicted “no default” for all loans, exploiting the strong class imbalance and yielding misleadingly high accuracy and a competitive MAE. This illustrates the accuracy paradox: high global performance with no operational value for detecting risky loans.

9.3 Robustness and chaos analysis

Using `chaos.py`, noise was injected into a critical feature (f1). Accuracy decreased only from 91.22% to 91.21% (a 0.01% drop). While this suggests technical resilience to noise, it also indicates model insensitivity: the classifier was already biased toward the majority class, so small perturbations did not change its predictions toward default.

10 Recommendations and Future Steps

Based on the findings, the following improvements are recommended:

- **Class rebalancing:** apply SMOTE and/or cost-sensitive learning (e.g., class weights) to force attention to the minority class.
- **Internal metric adjustment:** prioritize recall, F1, and PR-AUC for default detection rather than relying mainly on accuracy/MAE.
- **Refined two-stage model:** first-stage classifier for $P(default)$ optimized for high recall, second-stage regression specialized on likely defaults.
- **Advanced feature exploration:** generate “golden features” and apply stronger feature selection to reduce noise and improve generalization.
- **Additional stress testing:** expand chaos tests to simulate broader distribution shifts and monitor training curves continuously.

11 Conclusions

This report documented the complete design, implementation, and evaluation process of a prediction system for Kaggle’s Loan Default Prediction competition. A modular and reproducible end-to-end pipeline was built and demonstrated technical stability and resilience to perturbations. However, experiments showed that optimizing only global metrics (MAE/accuracy) can yield misleading solutions under severe class imbalance. The baseline model exhibited the accuracy paradox, failing to detect any defaults, highlighting the need for imbalance-aware training and evaluation. The proposed roadmap (rebalancing, metric shifts, two-stage modeling, improved feature engineering, and stronger stress testing) is expected to significantly improve default detection while maintaining competitive MAE. Ultimately, this work reinforces that successful machine learning in volatile financial settings requires not only strong algorithms, but also robust engineering design, reproducibility, and alignment with domain objectives.