Systems Analysis and Design of a Loan Default Prediction System: A Systems Engineering Approach with Chaos Theory

Juan Jose León Gomez
Dept. of Systems Engineering
U. Distrital Francisco José de Caldas
Email: jjleong@udistrital.edu.co

Miguel Angel Hernandez Medina
Dept. of Systems Engineering
U. Distrital Francisco José de Caldas
Email: miguahernandezm@udistrital.edu.co

Juan Pablo Diaz Ricaurte
Dept. of Systems Engineering
U. Distrital Francisco José de Caldas
Email: jpdiazr@udistrital.edu.co

Juan Esteban Avila Trujillo
Dept. of Systems Engineering
U. Distrital Francisco José de Caldas
Email:jeavilat@udistrital.edu.co

***Abstract*—This paper documents the design and analysis process of a predictive system for loan default estimation, based on Systems Engineering principles and Chaos Theory. The problem arises from the Kaggle competition "Loan Default Prediction," which seeks to estimate the probability of loan delinquency. The proposed solution integrates machine learning techniques, modular design, and sensitivity control to manage system complexity and nonlinear behavior. The obtained results demonstrate improved model stability and adaptability to noisy and evolving financial data.**

I. Introduction

The growth of digital credit systems has increased the need for models capable of accurately predicting loan default probabilities. Such models allow financial institutions to reduce risk and optimize resources. However, loan default prediction is inherently a complex system problem—characterized by high dimensionality, incomplete data, and nonlinear behaviors that mirror unpredictable economic and human processes. This work is based on the Kaggle competition "Loan Default Prediction," which poses a multivariate regression challenge with over 200,000 records and nearly 800 features. From a systems engineering perspective, the problem behaves as a dynamic system highly sensitive to initial conditions, where small variations in preprocessing or input data can lead to large differences in model performance. Previous research in credit risk prediction has mostly relied on logistic regression or decision tree models, but few approaches have incorporated feedback mechanisms or sensitivity control. This project proposes a modular, traceable, and robust architecture designed to mitigate chaotic effects stemming from data uncertainty and external economic feedback loops.

II. Methods and Materials

A. Solution Design

The proposed system receives raw financial data as input, applies cleansing, encoding, and feature engineering steps, and trains predictive models to estimate the probability of customer default.The design includes feedback mechanisms that allow the model to be retrained based on new data, ensuring continuous adaptation and monitoring against system sensitivity.

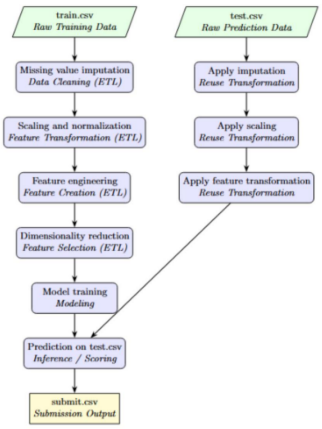

Fig. 1. General representation of the flow of the proposed system.

B. Technologies and Methods

1) Programming Language and Framework: Python is used to implement the learning model due to its versatility in data science, its ecosystem of specialized libraries, and its... accessible learning curve. Python allows for the manipulation of large volumes of data, training of machine learning models, and automation of analysis processes in a controlled and reproducible environment. Furthermore, its extensive community and compatibility with deployment tools make it ideal for collaborative and scalable projects.

2) Design Principles: The system's architecture follows key principles to ensure stability, maintainability, and scalability:

- **Modularity:** Independent modules facilitate debugging, reuse, and updating.
- **Scalability:** Supports distributed training using frameworks such as Dask or Apache Spark.

- **Reproducibility:** Dependency control and environment encapsulation using Docker or virtual environments.
- **Maintainability:** Organized folder structure (data, src, models, reports) and version control via GitHub.
- **Resilience:** Robust exception handling and automated alerts for data anomalies.

3) Libraries: The system integrates a set of Python libraries that support each phase of the modeling process, from data ingestion to model evaluation:

TABLE I
MAIN PYTHON LIBRARIES USED IN SYSTEM DEVELOPMENT.

Library	Main Functionality
Pandas-NumPy	Data manipulation and transformation for numerical and tabular structures.
Scikit-learn	Preprocessing, scaling, imputation, encoding, and baseline models.
LightGBM-XGBoost	High-performance gradient boosting algorithms for prediction.
TensorFlow-Keras	Deep learning architectures and neural network training.
Optuna	Automated hyperparameter optimization for model tuning.
Matplotlib-Seaborn	Exploratory visualization and sensitivity analysis.

These tools allow for the development of a reproducible and controlled workflow, ensuring consistent results and traceability at every stage of the system.

4) Dataset Description: Dataset Description: The dataset used was provided by the Kaggle Loan Default Prediction Competition. It contains customer financial information, including demographic, transactional, and credit characteristics. Each record represents a customer's payment status, allowing supervised learning to estimate the probability of default.

TABLE II
GENERAL DATASET DESCRIPTION.

Characteristic	Description
Number of Records	Approx. 200,000
Number of Variables	50–60 (depending on preprocessing)
Variable Types	Numerical and categorical
Target Variable	Client payment status (default / non-default)
File Format	CSV
Source	Kaggle: Loan Default Prediction Competition

This dataset was selected for its complexity and realism, allowing for the exploration of sensitivity phenomena, nonlinear correlations, and chaotic behavior within financial systems, thus providing a good way to corroborate the quality of the learning model's responses.

III. RESULTS AND DISCUSSION

The results obtained so far correspond to the analysis and design stages of the predictive system. The systematic analysis confirmed that the dataset exhibits chaotic and highly sensitive behavior, where small variations in preprocessing or feature selection can lead to significant changes in the final prediction. This observation reinforced the need to include feedback loops and rigorous preprocessing processes.

During the design stage, the system architecture was divided into modular components: data ingestion, preprocessing, feature engineering, model training, and evaluation. This modularity improves scalability and control of each process while ensuring system feedback.

The selection of Python and its ecosystem of libraries (Scikit-learn, LightGBM, Optuna) allows for the creation of a structured environment for future model experimentation. Preliminary data exploration also revealed the importance of variable scaling, imputation consistency, and outlier handling to avoid unstable model behavior.

In summary, the discussion highlights the chaotic nature of the system and the design decisions made to mitigate instability, ensuring that future implementations are more robust and reproducible.

IV. CONCLUSIONS

The analysis and design phases provided a systemic understanding of the challenges in building predictive models for loan default estimation. The proposed architecture ensures modularity, interpretability, and robustness against chaotic data variations. Future work will focus on full model implementation, hyperparameter tuning, deployment via FastAPI, and retraining mechanisms for continuous learning.

REFERENCES

[1] Kaggle, "Loan Default Prediction Competition." [Online]. Available: https://www.kaggle.com/competitions/loan-default-prediction