Systems Analysis and Design Applied to the Kaggle Problem: House Price Prediction

Juan David Zárate Moya (20222020184), Jesús Mateo Munevar Méndez (2023202042),
Kevin David Rincón Valencia (20232020356), Juan David Romero Morales (20222020102)
Systems Engineering
Universidad Distrital Francisco José de Caldas

***Abstract*—This paper presents an applied study on systems analysis and design based on the real Kaggle problem *"House Prices: Advanced Regression Techniques"*, whose objective is to predict the sale price of residential properties in Ames, Iowa. Following the principles of Systems Engineering and Systems Thinking, a modular architecture is proposed that integrates components for data ingestion, processing, modeling, and monitoring, addressing the system's inherent sensitivity, complexity, and chaos. The proposed framework aims to ensure scalability, traceability, and adaptability in dynamic environments, enabling continuous improvement of the predictive model.**

***Index Terms*—Systems analysis, machine learning, Kaggle, modular architecture, systems engineering, price prediction.**

I. Introduction

This work is developed within the context of the Kaggle challenge *House Prices: Advanced Regression Techniques*, whose purpose is to build a machine learning model capable of predicting the sale price of houses in Ames, Iowa, using a structured dataset of descriptive variables.

The dataset provides information on 2,919 properties, divided into two files:

- **train.csv:** 1,460 records with 79 explanatory variables and the target variable `SalePrice`.
- **test.csv:** 1,459 records with the same variables but without the target value.

The features include numerical variables (e.g., lot area, living area, year built), categorical variables (e.g., neighborhood, house type), and ordinal variables (e.g., overall construction quality). The official evaluation metric for this competition is the Root Mean Squared Error (RMSE) between the logarithms of the predicted and true prices:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y_i) - \log(\hat{y}_i))^2}.$$

The problem is approached from the perspective of **systems analysis and design**, considering that both the dataset and predictive model represent an open and dynamic socio-technical system. This system is influenced by structural, economic, and social factors, making it sensitive to small variations and unpredictable to external changes.

This document proposes a systems architecture that combines Systems Engineering principles with modern Data Science techniques applied to the Kaggle case study. The goal is to ensure robustness, traceability, and sustainability throughout the predictive model's lifecycle.

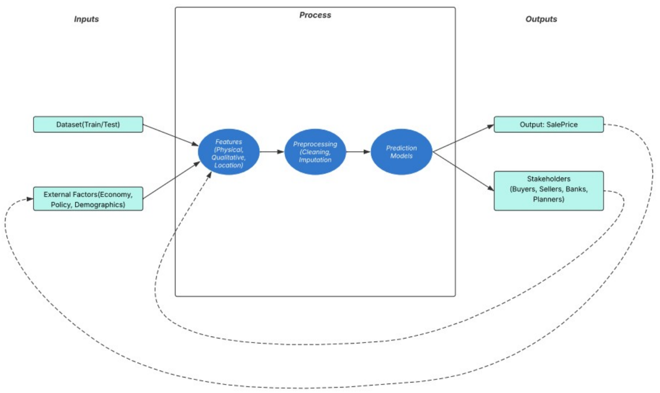

Inputs
Process
Outputs
Dataset(Train/Test)
External Factors(Economy, Policy, Demographics)
Features (Physical, Qualitative, Location)
Preprocessing (Cleaning, Imputation
Prediction Models
Output: SalePrice
Stakeholders (Buyers, Sellers, Banks, Planners)

Fig. 1. Systemic diagram of the housing price prediction process. It shows the inputs (datasets and external factors), analysis and modeling processes, and the outputs for various real-estate stakeholders.

II. Methods and Materials

A. Functional Requirements

TABLE I
Functional Requirements of the Proposed System

Code	Description
FR-01	Ingest and validate datasets obtained from Kaggle.
FR-02	Preprocess data through imputation, encoding, normalization, and scaling.
FR-03	Train regression models using cross-validation and hyperparameter tuning.
FR-04	Evaluate model performance using the RMSE metric.
FR-05	Serialize and deploy the best-performing model for batch or real-time predictions.
FR-06	Monitor model performance and trigger automatic retraining upon degradation.
FR-07	Generate interpretability reports, including feature importance and SHAP plots.

B. Non-Functional Requirements

C. System Architecture

The system follows a modular structure consisting of six main subsystems: **Data Ingestion**, **Processing**, **Modeling**, **Deployment**, **Monitoring**, and **Orchestration**. Each module

TABLE II

MAIN RISKS IDENTIFIED AND MITIGATION STRATEGIES

Risk	Impact	Probability	Mitigation
Data loss or corruption	High	Medium	Backups and dataset versioning
Pipeline failures during training	High	Medium	Checkpoints and unit tests
Data drift in real-world distribution	High	High	Continuous drift monitoring and automatic retraining
Model overfitting	Medium	High	Regularization, CV, and ensemble methods
Integration errors between modules	Medium	Medium	Peer review and integration testing
Delays in schedule	Medium	Medium	Weekly backlog review and adjustments

performs a specific function within the workflow, ensuring coherence and traceability throughout the model's lifecycle.

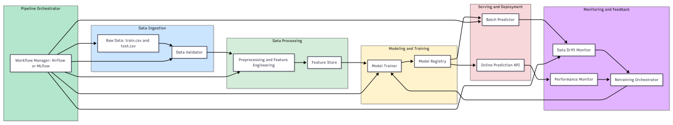


Fig. 2. Overall system architecture proposed for the Kaggle problem.

D. Applied Principles

1) **Holistic Vision:** Integration of all components under a common objective.
2) **Systems Thinking:** Identification of cause–effect relationships and feedback loops.
3) **Modularity:** Separation of responsibilities among subsystems.
4) **Traceability:** Version control for datasets, features, and trained models.
5) **Automation:** Orchestrated retraining upon detection of data drift or degradation.
6) **Sensitivity Management:** Monitoring and mitigation of variability in model performance.

E. Technical Implementation

The project will be developed using **Python 3.11** with the following libraries:

- **pandas**, **numpy**: Data manipulation and numerical analysis.
- **scikit-learn**: Algorithms, pipelines, and cross-validation tools.
- **xgboost**, **lightgbm**: Efficient gradient boosting regression models.
- **matplotlib**, **seaborn**: Data visualization and feature exploration.
- **shap**: Feature contribution and interpretability analysis.

III. PROJECT MANAGEMENT AND SYSTEM ANALYSIS

The development of the housing price prediction system was structured following system analysis and design principles, ensuring clear requirements, traceability, and an architecture capable of adapting to real-world usage scenarios. This section describes the organization of team activities, the workflow modeling, and the architectural refinements derived from the functional and non-functional analysis of the system.

A. Team Roles and Responsibilities

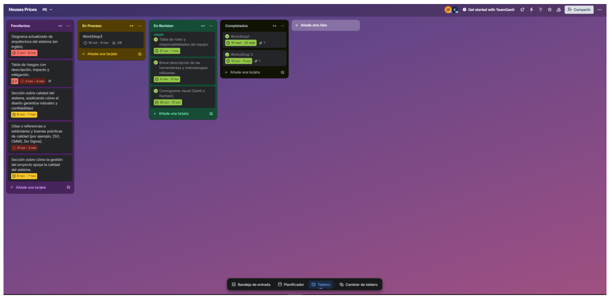
To maintain an organized execution of the project, the team adopted an agile-inspired structure aligned with system analysis practices. Each role was defined to support different stages of the system life cycle:

- **Product Owner:** Identifies system requirements, validates priorities, and ensures alignment with functional objectives.
- **Scrum Master:** Coordinates team communication, facilitates planning, and ensures compliance with the workflow.
- **Development Team:** Designs and implements system components, performs testing, and maintains technical documentation.

This distribution of responsibilities strengthened traceability, decision-making, and communication throughout the project.

B. Planning and Workflow Organization

1) Kanban Workflow: The work organization was modeled through a Kanban board that visualized the system workflow: tasks pending, in progress, and completed. This mechanism facilitated the identification of bottlenecks and the control of progress across the phases of analysis, design, and implementation.



Pendientes

- Diagrama actualizado de arquitectura del sistema (en inglés). 3 nov - 6 nov
- Tabla de riesgos con descripción, impacto y mitigación. 3 nov - 4 nov
- Sección sobre calidad del sistema, explicando cómo el diseño garantiza robustez y confiabilidad. 6 nov - 7 nov
- Citas o referencias a estándares y buenas prácticas de calidad (por ejemplo, ISO, CMMI, Six Sigma). 31 oct - 2 nov
- Sección sobre cómo la gestión del proyecto apoya la calidad del sistema. 6 nov - 7 nov
- Añade una tarjeta

En Proceso

- WorkShop3 30 oct - 8 nov 2/8
- Añade una tarjeta

En Revision

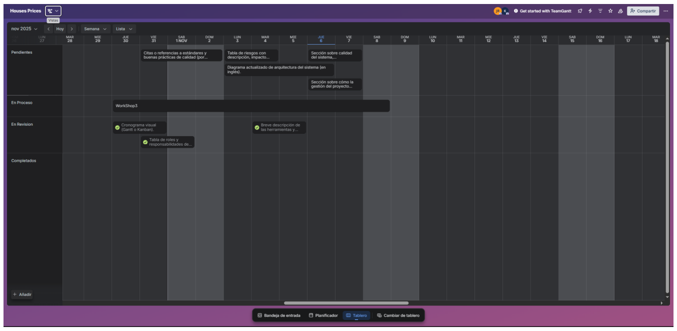
- Tabla de roles y responsabilidades del equipo. 31 oct - 1 nov
- Breve descripción de las herramientas y metodologías utilizadas. 4 nov - 5 nov
- Cronograma visual (Gantt o Kanban). 30 oct - 31 oct
- Añade una tarjeta

Completados

- WorkShop1 19 sept - 20 sept 1
- WorksShop`2 10 oct - 11 oct 1
- Añade una tarjeta

Fig. 3. Kanban board used for managing ongoing tasks.

2) Activity Schedule (Gantt Chart): A Gantt chart was used to structure the system's life cycle, dividing the project into analysis, conceptual modeling, architectural design, implementation, experimentation, validation, and documentation.



Pendientes

Citas o referencias a estándares y buenas prácticas de calidad (por...

Tabla de riesgos con descripción, impacto...

Sección sobre calidad del sistema,...

Diagrama actualizado de arquitectura del sistema (en inglés).

Sección sobre cómo la gestión del proyecto...

En Proceso

WorkShop3

En Revision

Cronograma visual (Gantt o Kanban).

Breve descripción de las herramientas y...

Tabla de roles y responsabilidades de...

Completados

Fig. 4. Gantt chart showing planned activities and dependencies.

C. Quality Assurance Processes

From a systems design perspective, quality was ensured through practices that promote reliability, maintainability, and reproducibility.

1) Version Control: The repository followed a branch-based workflow that enabled:

- consistent version management,
- traceability of system configuration changes,
- tracking updates to models and pipelines,
- maintaining integrity across the project life cycle.

2) Peer Review: Before integrating each module, technical reviews were conducted to validate:

- alignment with system requirements,
- correct implementation of the design,
- cohesion between components,
- adherence to system metrics and standards.

3) Technical Validation: The following validation mechanisms were applied:

- unit testing of preprocessing modules,
- validation of the end-to-end training and inference pipeline,
- K-Fold Cross-Validation to assess system stability under different scenarios.

4) Continuous Improvement: Insights from experimentation and error analysis were incorporated into iterative refinements of preprocessing steps, model selection, and system architecture.

D. Architectural Refinement

Based on functional and non-functional requirements analysis, the system architecture was refined to improve modularity, traceability, and maintainability. The main architectural improvements include:

- clear separation between the **feature store** and the inference module,
- integration of a **data drift monitoring** subsystem,
- implementation of an **automatic retraining orchestrator**,
- redefinition of interfaces between modules to reduce coupling.

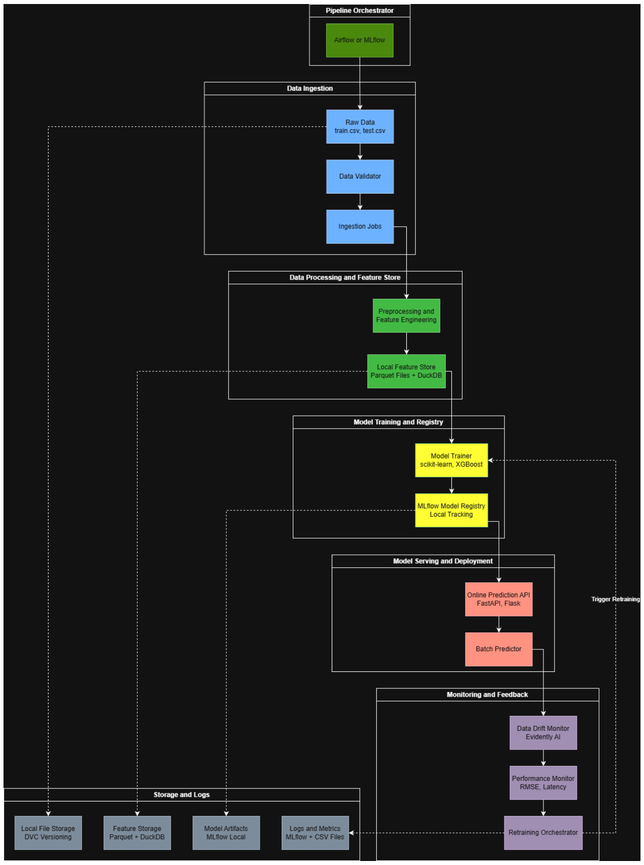


Fig. 5. Refined system architecture including monitoring and retraining mechanisms.

E. Integration with System Development

The mechanisms described integrate requirements analysis, modeling, architectural design, and quality assurance practices. This strengthens the stability of the system under multiple scenarios and supports continuous refinement — fundamental principles of system analysis and design.

IV. Data-Driven Systemic Analysis and Simulation

This stage of the project deepened the understanding of the real estate system through two complementary perspectives: (1) quantitative analysis using Machine Learning, and (2) conceptual simulations that reveal emergent behaviors within the system. Both approaches helped link dataset structure with systemic dynamics, feedback loops, and nonlinear patterns characteristic of complex systems.

A. Systemic Classification of Variables

The dataset variables were categorized according to their role within the Ames housing system, enabling a multidimensional interpretation of structural, spatial, functional, temporal, and administrative components that jointly influence the final sale price. This classification provided a holistic understanding of the system prior to developing predictive models.

B. Dataset Preparation and Cleaning

A detailed cleaning process ensured consistency and stability in the dataset used for modeling. The main decisions included:

- **Variable selection** based on correlation analysis, domain knowledge, and preliminary tests.
- **Imputation** using the median for numerical variables and the mode for categorical ones.
- **Encoding** of ordinal variables using semantic rankings and one-hot encoding for nominal attributes.
- **Derived variables** such as `HouseAge`, `YearsSinceRemodel`, and `TotalBathrooms`, which captured temporal and structural characteristics.

These steps produced a coherent and structured dataset suitable for simulation and modeling tasks.

C. Machine Learning–Based Simulation

Two predictive models were implemented: Linear Regression and Random Forest. These models provided insight into variable influence, system sensitivity, and internal patterns governing price behavior.

1) Linear Regression: The Linear Regression model served as a baseline due to its interpretability.

- RMSE and R^2 showed acceptable performance for a linear model.
- The real-vs-predicted values aligned reasonably with the ideal diagonal.
- Residuals remained centered around zero, with long tails caused by extreme values.

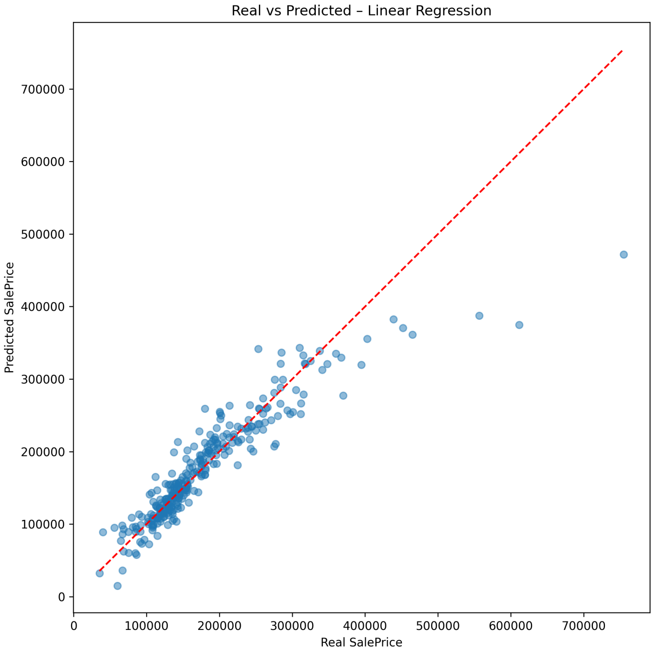


Fig. 6. Real vs. predicted prices using Linear Regression.

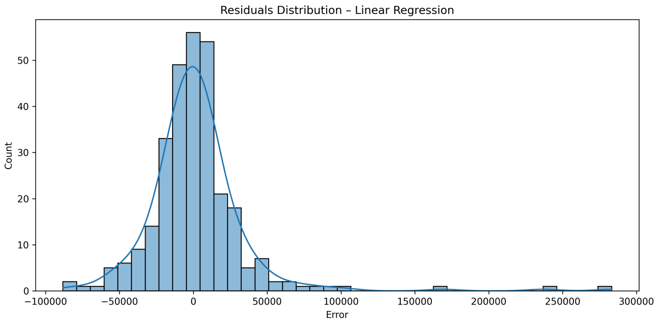


Fig. 7. Residual distribution for the Linear Regression model.

2) Random Forest: The Random Forest model achieved higher performance by capturing nonlinear interactions:

- Significantly lower RMSE compared to Linear Regression.
- Higher R^2, indicating superior explanatory power.
- Feature importance highlighted structural and quality variables as dominant.

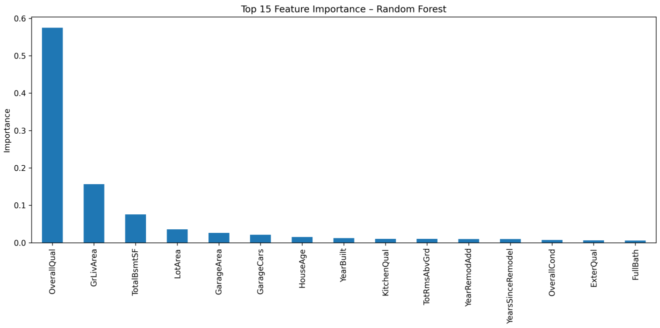


Fig. 8. Feature importance ranking obtained from the Random Forest model.

D. Correlation Analysis

A correlation matrix was used to identify strong variable relationships and verify model behaviors.

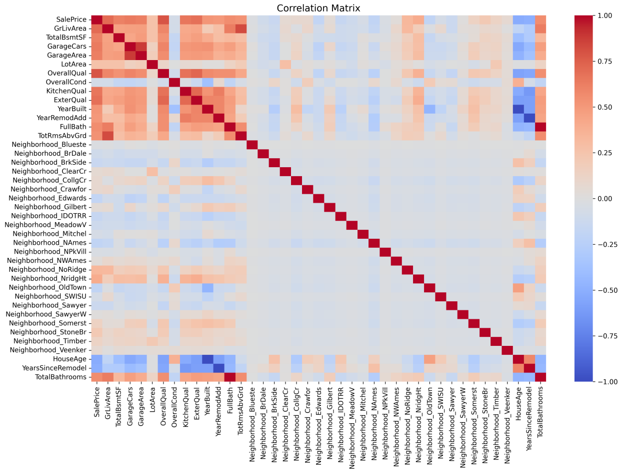


Fig. 9. Correlation matrix illustrating relationships among key variables.

E. Systemic Interpretation Through Causal Loops

From the Machine Learning results, several reinforcing feedback loops characteristic of real estate systems were identified. These loops help explain how changes in certain structural or qualitative attributes propagate through the system and influence long-term behavior:

- **OverallQual** → **SalePrice** → **Remodeling Incentive** → **OverallQual**: Higher overall quality leads to higher prices, which increases incentives for remodeling, further improving quality.
- **GrLivArea** → **SalePrice** → **Expansion Incentive** → **GrLivArea**: Larger living areas raise sale prices, motivating owners to expand living spaces, reinforcing this attribute.
- **GarageCars / TotalBsmtSF** → **Comfort** → **Perceived Value** → **Investment**: Enhanced functional space improves comfort and perceived value, encouraging investment in garage or basement improvements.

Controlled experiments—such as artificially increasing structural quality, modifying surface area, or simulating house aging—allowed the evaluation of leverage points within the system and its sensitivity to controlled perturbations. These analyses connected the quantitative results with qualitative systemic behavior, reinforcing the interpretation of the housing market as a dynamic socio-technical system.

F. Conceptual Simulation via Cellular Automata

To complement the ML analysis, a Cellular Automata (CA) simulation was developed to explore spatial and emergent behaviors of a simplified housing system. The CA operates on a 20x20 grid, where each cell represents a micro-region that evolves according to local interaction rules.

1) States and Transition Rules: Each cell may take one of three states: *Stable (0)*, *Risky (1)*, or *Chaotic/Noise (2)*. The evolution follows:

- Risk contagion based on neighboring conditions.
- Noise influence introducing probabilistic perturbations.
- Recovery of risky regions when surrounded by stable cells.
- Periodic global shocks converting portions of the grid into noise.

2) Emergent Behavior: Across 40 iterations, the automaton displayed:

- formation of clusters,
- expansion and contraction of risky regions,
- strong sensitivity to noise,
- resilience after global shocks.

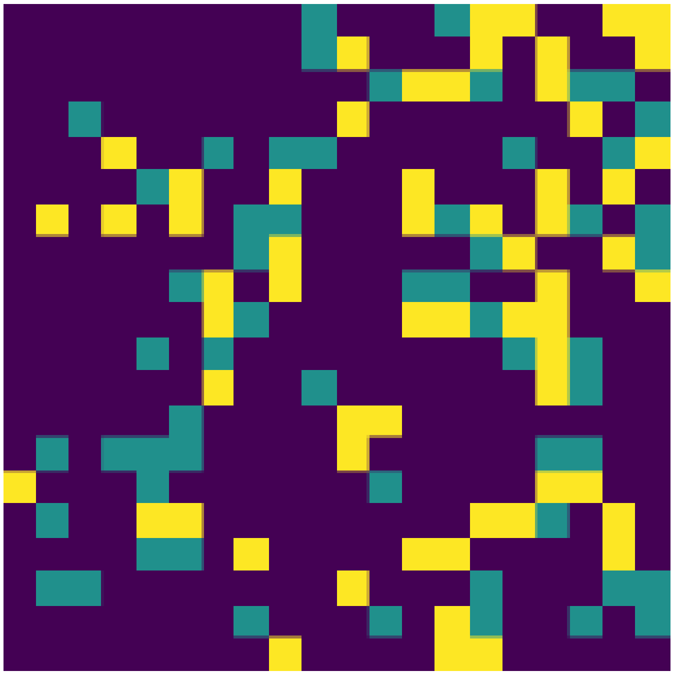


Fig. 10. Initial state distribution of the Cellular Automata (Iteration 1).

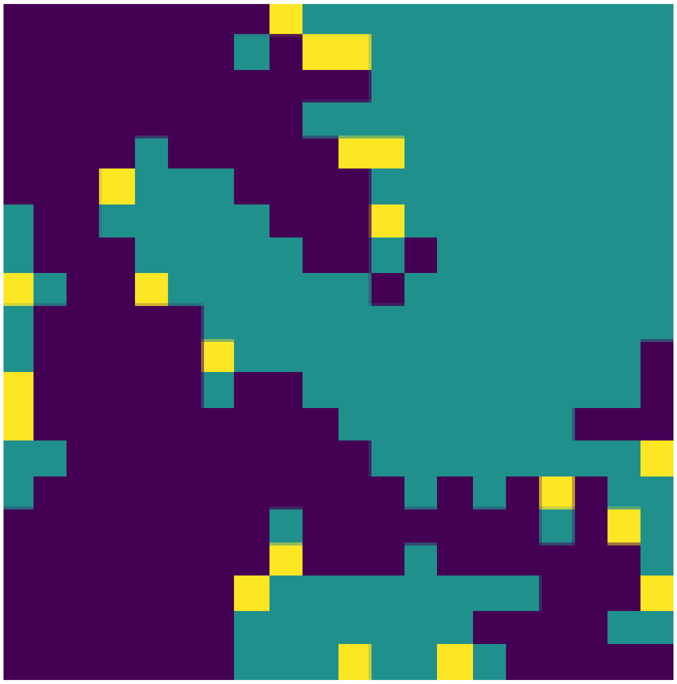


Fig. 11. System evolution after the first global shock (Iteration 10).

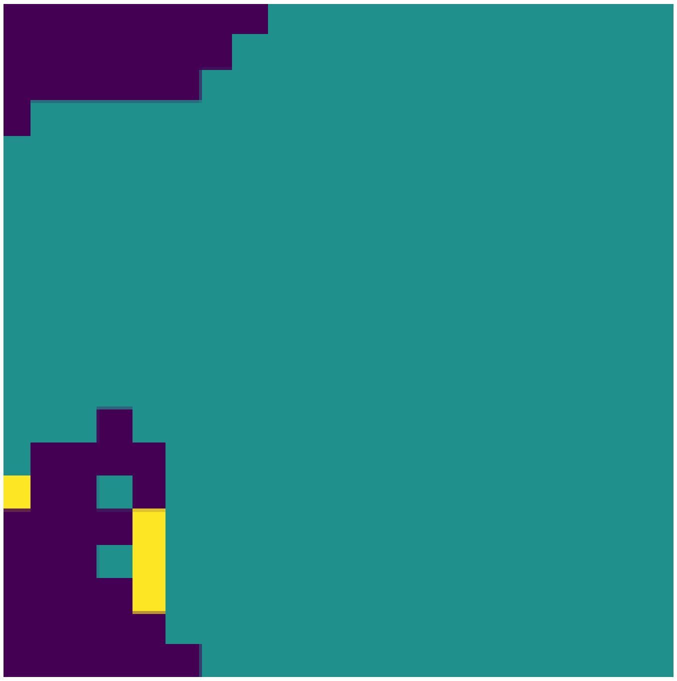


Fig. 12. Clustering and nonlinear transitions at Iteration 20.

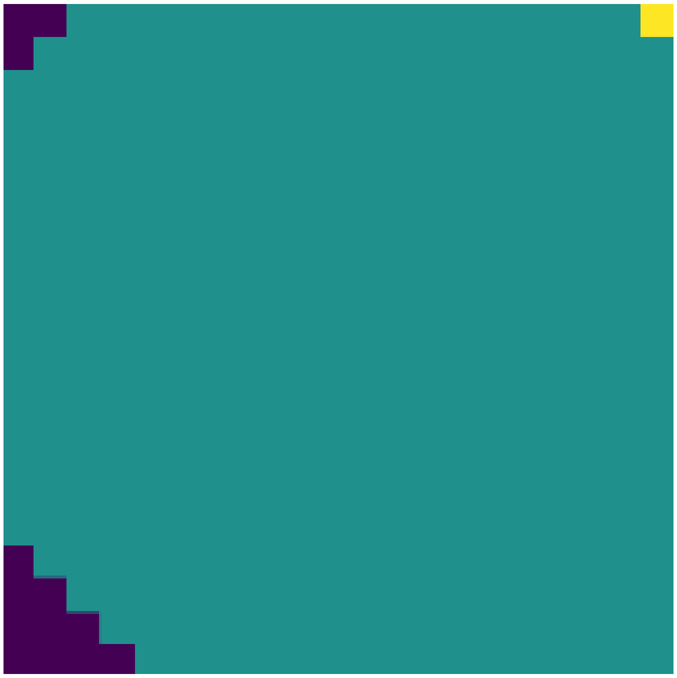


Fig. 13. Final configuration of the system at Iteration 40.

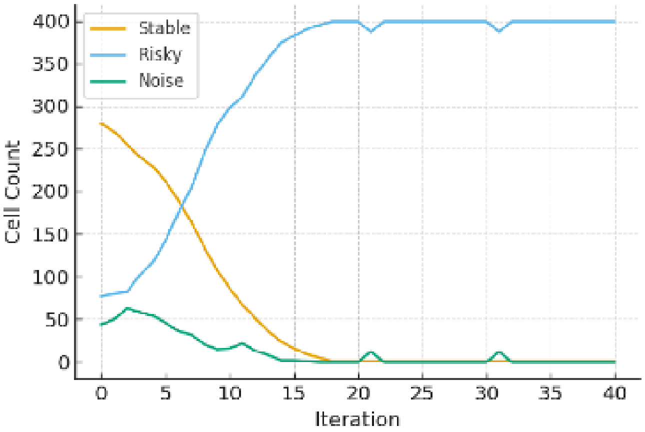


Fig. 14. Temporal evolution of stable, risky, and noise states.

3) Quantitative Evolution and Interpretation: The temporal analysis revealed that:

- risky states oscillate rather than converge,
- noise spikes correlate with shock events,
- stable regions dominate but remain vulnerable to contagion.

This behavior matches real socio-technical systems where micro-interactions generate unpredictable macro-level patterns.

4) Conclusion of the Simulation: The Cellular Automata simulation demonstrated:

- emergence from simple rules,
- sensitivity to perturbations,

- self-organization,
- nonlinear propagation of local effects.

This conceptual simulation complements the ML analysis and reinforces the systemic architecture designed for the predictive environment.

V. Conclusions

This study demonstrates that the problem of housing price prediction extends far beyond a purely statistical or machine–learning exercise and can be effectively interpreted, structured, and solved through systems analysis and systems design principles. By approaching the Kaggle Housing dataset as a socio-technical system, it was possible to identify interactions, feedback loops, risks, constraints, and architectural requirements that shape the behavior of the predictive environment.

The proposed modular architecture—composed of ingestion, processing, modeling, deployment, monitoring, and orchestration subsystems—proved adequate for ensuring traceability, maintainability, and adaptability throughout the model lifecycle. The incorporation of systemic principles such as holism, modularity, sensitivity management, and continuous improvement enabled the construction of a framework capable of operating in dynamic contexts where data distributions, performance demands, and external conditions evolve over time.

The integration of Machine Learning with systemic thinking enriched the understanding of the problem domain. The models provided quantitative evidence of key variables and structural relationships, while the conceptual simulations and causal-loop interpretations highlighted emergent and nonlinear behaviors characteristic of real estate markets. Together, these perspectives validated the need for an architecture designed not only for accuracy but also for robustness and resilience.

Furthermore, the project management practices—Kanban workflow, architectural refinement, quality assurance, and coordinated team roles—ensured coherence between requirements, design decisions, and implementation strategies. This alignment reflects the essence of systems engineering: achieving orderly development of complex systems through disciplined processes.

In conclusion, the work presented establishes a comprehensive and scalable approach for addressing predictive modeling within an engineered system. It shows that combining modern data-driven methods with classical systems thinking results in solutions that are not only technically effective but structurally sustainable.

References

[1] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer, 2009.

[3] D. De Cock, “Ames, Iowa: Alternative to the Boston Housing Dataset,” *Journal of Statistics Education*, vol. 19, no. 3, 2011.

[4] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013.

[5] S. Wolfram, *A New Kind of Science*. Wolfram Media, 2002.

[6] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Kaggle, “House Prices: Advanced Regression Techniques,” 2024. [Online]. Available: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques