

Real Estate Transactions System Enhanced with Machine Learning Algorithms

Andreea Bianca Popa, Paul Stefan Popescu

University of Craiova, Craiova, Romania

E-mail: popa.andreea.j2p@student.ucv.ro, stefan.popescu@edu.ucv.ro

Abstract. Real-estate transactions are interaction-intensive processes that require property discovery, negotiation, and frequent coordination between clients and agents. This paper presents the design and implementation of a web platform that supports end-to-end user interaction around residential listings: role-based authentication (client/agent/admin), advanced search and filtering, wishlist management, internal property-contextual messaging, and meeting requests for on-site visits. To support decision-making, the platform integrates automated price estimation using supervised machine learning. The prediction module is deployed as a Python (Flask) microservice and is consumed by the PHP backend through a REST-style interface, enabling modular evolution of the intelligent component. Several regression algorithms were evaluated using a real-world dataset, and Random Forest achieved the best performance (MAE 41,634.65; RMSE 86,272.47; R^2 0.71). The resulting system demonstrates how interaction features and data-driven intelligence can be integrated into a cohesive, practical solution for digital real estate workflows.

Keywords: user-system interaction; real-estate platform; web application

DOI: 10.37789/ijusi.2024.17.3.2

1. Introduction

Despite widespread online listing portals, real estate transactions still rely heavily on fragmented communication (phone, email, messaging apps) and manual coordination for visits and negotiations.

Real estate interactions involve uncertainty, substantial financial stakes, and repeated coordination among multiple parties. While online portals have improved the early stage of discovery (searching, browsing, and filtering), the later stages - clarifying details, negotiating, and scheduling viewings - often remain fragmented across external channels such as calls, email, or generic messaging applications. Fragmentation creates practical issues: users lose conversational context, it becomes harder to reconstruct what was

agreed, and stakeholders spend additional time switching between tools.

From a user-system interaction perspective, the core challenge is to reduce task switching and preserve context across the transaction lifecycle. Clients typically alternate between listing pages, personal notes, and multiple communication channels; agents manage many concurrent conversations and must track each client's intent, constraints, and preferred properties. When messages are not anchored to the referenced listing, misunderstandings become more frequent, and the time-to-agreement increases. In contrast, contextual messaging linked to a listing supports continuity: users can resume a negotiation by opening the property and immediately seeing the relevant discussion history.

A second challenge concerns price transparency. Asking prices do not always reflect comparable market conditions, and users - especially first-time buyers - may lack a reference point for what is reasonable given a property's attributes (surface, rooms, age, location, furnishing, and parking). Automated valuation models (AVMs) can provide indicative estimates that support decision-making and negotiation, but they must be presented carefully as decision support rather than authoritative appraisal. From an engineering viewpoint, AVMs also require consistent preprocessing at training and inference time, as well as clear versioning for reproducibility.

The platform presented in this paper addresses these challenges by combining interaction-centred features that keep key actions anchored to listings (shortlisting, messaging, and meeting requests), and a data-driven price estimation tool integrated into the same workflow. A central design goal is modularity: the machine-learning component is deployed as a separate microservice, allowing the web platform to evolve independently of the predictive model. This separation supports practical lifecycle operations such as retraining, model replacement, and incremental deployment without disrupting core transaction features.

More specifically, the work focuses on three objectives: supporting the full client-agent interaction loop (search -> shortlist -> communicate -> schedule), providing administrative oversight and traceability of platform activity, and integrating an ML estimation module with consistent preprocessing and a stable API interface suitable for production-like deployment.

We propose a web platform that consolidates key interaction steps in a single environment. The platform targets three roles:

- clients, who search and shortlist properties, communicate with agents, and request visits;
- agents, who publish listings and coordinate requests;
- administrators, who oversee platform content and basic statistics.

Beyond interaction support, the platform includes an automated valuation mechanism to help users understand price ranges for properties with similar attributes.

Because the machine-learning lifecycle (feature preparation, model updates, deployment) differs from standard web development, the valuation component is implemented as a dedicated microservice, which promotes maintainability and supports future scaling.

The contributions of the system presented in this paper are:

1. a modular, role-based real-estate platform emphasizing property-contextual interaction (wishlist, messaging, meeting requests);
2. integration of external services (Maps/Geocoding and SMTP email);
3. an ML price-estimation module evaluated across multiple regressors, integrated via a lightweight API.

2. Related Work

Real-estate portals commonly provide listing discovery and filtering, but negotiation and coordination are frequently handled outside the platform.

International platforms have introduced automated valuation models (AVMs) to provide indicative price estimates, but these depend on large, high-quality datasets and local market signals.

In many regional contexts, platforms prioritize listing exposure rather than integrated interaction workflows. The approach presented here focuses on interaction completeness: discovery, shortlisting, contextual communication, and scheduling are treated as first-class features.

Price estimation is designed as an assistive tool rather than a replacement for professional appraisal, and the predictive component is separated via a microservice so it can evolve independently.

In a related line of work, Baldominos et al. (2018) propose a machine-learning system for identifying real-time real-estate investment opportunities by detecting listings priced substantially below their estimated market value. The authors frame the task as a regression problem that predicts a property's

expected price from features extracted from online advertisements, then flags potential “bargains” when the listed price deviates negatively from the model estimate. Their case study focuses on high-end residential assets in the Salamanca district (Madrid, Spain) collected from Idealista, and emphasizes the importance of feature engineering and comparing multiple ML methods (e.g., tree-based models, k-NN, and neural networks) to balance predictive performance and practical deployment. Published in *Applied Sciences* by MDPI, this work supports the idea that AVM-style estimates can be operationalised not only for price guidance but also for opportunity detection within user-facing platforms.

Pai and Wang (2020) investigate real-estate price forecasting using actual transaction data (Taiwan) and compare four machine-learning regressors—least squares support vector regression (LSSVR), CART, general regression neural networks (GRNN), and backpropagation neural networks (BPNN)—with genetic algorithms used for parameter selection. Their experiments show that LSSVR achieves the best forecasting accuracy among the tested models and yields competitive error levels compared to prior studies, highlighting the value of transaction-based datasets and careful hyperparameter tuning for robust price estimation.

Choy and Ho (2023) provide a broader perspective on how machine learning is being used in real-estate research, emphasizing that data-driven models increasingly outperform traditional hedonic/statistical techniques for property price prediction. Using 24,936 housing transaction records, they compare Extra Trees, k-Nearest Neighbors, and Random Forest against a hedonic price model and report that the ML approaches achieve higher explanatory power and lower prediction errors. They also propose a feature-engineering strategy that combines property age with size (age \times square footage) to better capture depreciation effects. Beyond accuracy, the authors discuss practical and societal implications—such as privacy, fairness, and workforce impacts—arguing that responsible deployment requires standards and governance. This work motivates our choice of tree-based regressors and supports the integration of AVM-style estimates into user-facing real estate platforms as a decision-support component.

A recent study by Cheng-Hong Yang, Borcy Lee, and Yu-Da Lin (2025) proposes a deep-learning framework for analyzing real-estate prices and transactions in Taiwan using public, government-based real-estate data. Their approach combines clustering with an autoencoder-based representation step

to improve clustering accuracy, and then applies a particle-swarm-optimization (PSO)-driven clustering strategy (DBM-PSO) to identify structure in multidimensional market indicators and explain sustained housing-price growth over the past decade. This work is relevant to our study because it highlights (i) the importance of handling high-dimensional real-estate data with representation learning, and (ii) the practical value of integrating advanced analytics into decision-support workflows—paralleling our aim to embed ML-driven valuation support into an end-user real-estate platform.

Breuer and Steininger (2020) examine recent trends in real-estate research by comparing topics in working papers presented at major real-estate conferences (e.g., ARES, AREUEA, ERES) with those appearing in finance venues, using machine-learning-based text analysis of paper titles. They extract frequently co-occurring terms and visualize them as word-pair networks, which provides an overview of how research themes cluster and evolve over time (e.g., the relative prominence of “real estate,” “housing,” “market,” and the comparatively weaker emphasis on “mortgage” in certain conference title networks). This bibliometric, ML-driven perspective is relevant to our work because it highlights the growing role of data-driven methods in the real-estate domain and motivates integrating ML components (such as automated valuation) into applied systems that support real-world real-estate workflows.

Yanliang Yu et al. (2021) propose a real-estate pricing approach that combines data mining and machine-learning techniques with an explicit treatment of time-varying price trends. Their work argues that static pricing models can become inaccurate when market conditions shift, so they introduce trend estimation (based on quadratic exponential smoothing) and incorporate it into a pricing process that adapts over time. In addition, they formulate a time-varying mean-reverting (Ornstein–Uhlenbeck) process by allowing key parameters (e.g., volatility and mean reversion) to change piecewise across market cycles, aiming for improved flexibility and robustness. Their case-study results suggest that integrating trend-aware components can enhance practical pricing performance, which complements our focus on building an ML-driven valuation feature that can later be extended with periodic updates and model lifecycle management within a real-estate platform

3. System Design

The system is based on a set of requirements divided into functional and non-functional requirements.

The functional requirements are:

- Account management: registration, login/logout, and role-based access control.
- Listing browsing: search, filter, sorting, and detailed property view with images and map location.
- Wishlist: add/remove properties and revisit shortlisted items.
- Messaging: internal chat between client and agent linked to a specific listing.
- Meeting requests: propose a visit date/time, accept/decline, and track status.
- Property management: agents create/edit/delete listings and associated media.
- Administration: manage users and listings, handle contact messages, and view platform activity statistics.
- Price estimation: submit property attributes and obtain a model-based estimated price; store prediction history.

The non-functional requirements are:

- Usability: clear navigation and reduced friction for frequent tasks (search, shortlisting, contacting).
- Security: robust authentication, server-side validation, and safe database access.
- Reliability: predictable behavior when external services are unavailable (maps, email, ML API).
- Maintainability: modular separation of web logic and ML logic; support independent updates.
- Performance: responsive browsing with pagination and indexed queries for filters.

A typical client scenario proceeds as follows: the user logs in, searches for apartments within a budget and preferred location, saves a shortlist to the wishlist, opens a listing to inspect details and map position, then initiates a chat with the agent to request clarifications (e.g., furnishing, parking,

neighborhood). After aligning on details, the client may submit a meeting request for a visit. In parallel, the client uses the price-estimation tool to compare the asked price with an indicative estimate derived from similar properties, supporting a more informed negotiation.



Figure 1. The system's architecture

The system’s architecture is presented in Figure 1 and as we can see the platform uses a browser-based client interface that communicates with a PHP backend and persists data in a MySQL database.

The machine-learning module is separated as a Flask API that receives structured features and returns a numeric prediction.

This architecture isolates the ML stack (Python libraries, model serialization) from the web stack (PHP, MySQL), reducing coupling and simplifying updates to either component.

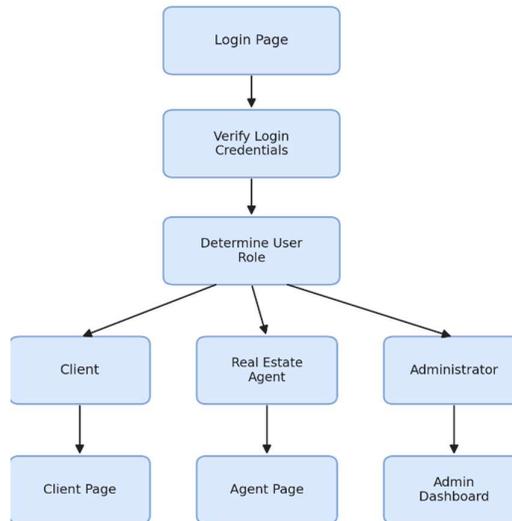


Figure 2. Roles available in the application

Figure 2 presents each role implemented in the application. Role-based dashboards expose the most relevant actions for each user type (e.g., agents manage listings and visit requests; admins manage platform content).

As for the technology stack, the web platform uses PHP for server-side logic and MySQL for persistent storage. The UI uses HTML/CSS with Bootstrap and JavaScript for interactive elements. The price-estimation service is implemented in Python using Flask and standard machine learning libraries for training and inference. This split is motivated by maintainability: PHP handles transactional web workflows efficiently, while Python provides a mature ecosystem for model training and deployment.

Regarding the backend structure, the PHP application is organised around modular pages and server-side controllers that implement the main use cases: authentication, listing management, search/filter, chat, meeting requests, wishlist, administration, and contact workflows. Access control is enforced both at the UI layer (by hiding links/actions) and at the server layer (by applying guard clauses that verify the session role). For database access, PDO prepared statements are used to bind parameters, thereby preventing direct concatenation of SQL queries and reducing the risk of SQL injection.

4. Experimental results

4.1 Dataset

For training and evaluating the machine-learning models, we used a real dataset provided by the Bridge platform, specifically from the source called the ACTRIS Reference Server. This dataset accurately reflects real properties listed on the United States real estate market and provides a setting much closer to real-world applications than generic public datasets. The dataset consists of 5,184 real estate properties (instances).

The data include detailed information about each property, such as: Living area, Number of bedrooms and bathrooms, Year built, City and state, Property type, Furnishing status, and Number of garage spaces.

Before training, the data were preprocessed as follows:

- Categorical attributes were converted to numerical format using LabelEncoder;
- The YearBuilt column was transformed into a numerical variable,

Age, computed as $2025 - \text{YearBuilt}$ to reflect the property's age;

- Incomplete and problematic records were removed.

By using this professional dataset, the trained models benefited from a solid and relevant foundation, improving both prediction accuracy and the overall reliability of the application. Moreover, Bridge's clear and standardized data structure enables potential future automation of dataset updates or even direct API-based integration for real-time prediction. To ensure consistency, the same preprocessing steps used during training are applied at inference time inside the microservice.

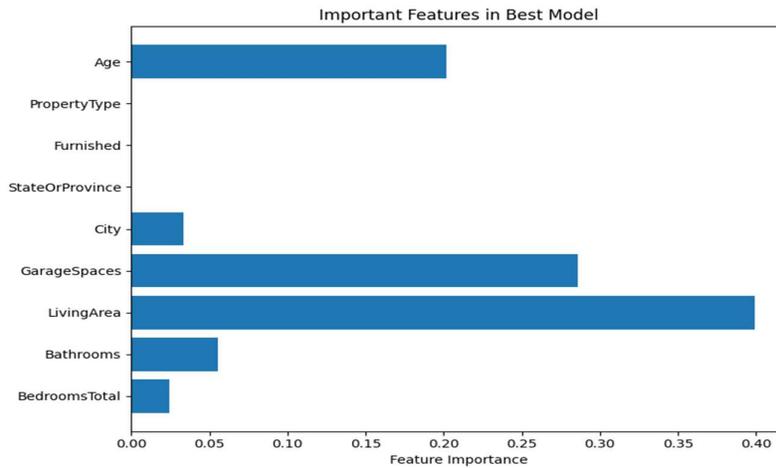


Figure 3. Features' importance

For a better dataset understanding, we computed the features importance using Random Forest as presented in Figure 3. It can be observed that the most relevant variables are *LivingArea*, *GarageSpaces*, and *Age*, while variables such as *Furnished* or *PropertyType* have a reduced influence in the dataset used.

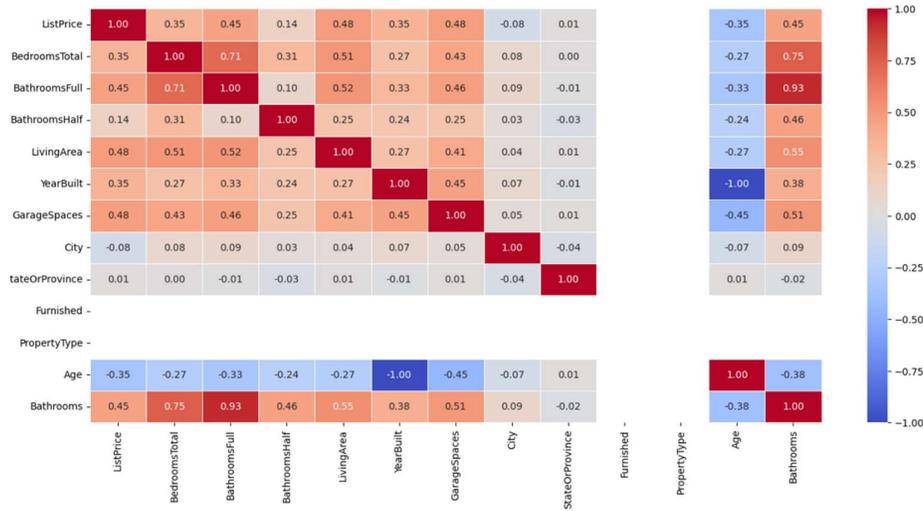


Figure 4. Correlation matrix

To better understand the relationships among the numerical variables in the dataset, we constructed a correlation matrix as presented in Figure 4 using the Pearson coefficient. This highlights how strong the linear relationship is between variables such as living area, number of bathrooms, building age, and property price. Values close to +1 or -1 indicate a strong positive or negative correlation, respectively, whereas values close to 0 indicate a weak or non-existent correlation.

4.2 Results

Table 1. Results obtained using algorithms

Model	MAE	RMSE	R^2
Random Forest	41.634,65	86.272,47	0.71
Gradient Boosting	57.653,55	102.343,46	0.60
Linear Regression	70.148,30	127.980,10	0.37
Decision Tree	45.851,71	110.656,61	0.53
KNN	53.043,73	110.456,29	0.53

As we can see in Table 1 Multiple regression algorithms were compared to balance accuracy, robustness, and interpretability: Linear Regression (Su

et al., 2012), Decision Tree (Charbuty, B., and Abdulazeez, A., 2021), K-Nearest Neighbors, Gradient Boosting, and Random Forest (Schonlau, M., & Zou, R. Y, 2020).

Evaluation uses mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2). Random Forest achieved the best overall performance (MAE 41,634.65; RMSE 86,272.47; R^2 0.71) and was selected for deployment.

5. Conclusions

This paper presented a web-based platform for real-estate interactions that integrates listing discovery and management with wishlist functionality, contextual messaging, meeting scheduling, and machine-learning-based price estimation.

A PHP/MySQL backend is combined with a Flask microservice for predictions, enabling modular development and deployment.

Experimental evaluation across multiple regressors identified Random Forest as the best-performing model on the available dataset.

Future work includes: retraining on region-specific datasets, implementing monitoring for model drift, conducting a formal user study to quantify usability and perceived usefulness of the interaction flows, and adding stronger security hardening for production deployments (HTTPS, CSRF protection, rate limiting).

References

- Baldominos, A., Blanco, I., Moreno, A. J., Iturrarte, R., Bernárdez, Ó., & Afonso, C. (2018). Identifying real estate opportunities using machine learning. *Applied sciences*, 8(11), 2321.
- Pai, P. F., & Wang, W. C. (2020). Using machine learning models and actual transaction data for predicting real estate prices. *Applied Sciences*, 10(17), 5832.
- Choy, L. H., & Ho, W. K. (2023). The use of machine learning in real estate research. *Land*, 12(4), 740.
- Yang, C. H., Lee, B., & Lin, Y. D. (2025). Deep-Learning Approach for An Analysis of Real-estate Prices and Transactions. *IEEE Access*.
- Breuer, W., & Steininger, B. I. (2020). Recent trends in real estate research: a comparison of recent working papers and publications using machine learning algorithms. *Journal of Business Economics*, 90(7), 963-974.

- Yu, Y., Lu, J., Shen, D., & Chen, B. (2021). Research on real estate pricing methods based on data mining and machine learning. *Neural Computing and Applications*, 33(9), 3925-3937.
- Su, X., Yan, X., & Tsai, C. L. (2012). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3), 275-294.
- Charbuty, B., & Abdulazeez, A. (2021). Classification based on a decision tree algorithm for machine learning. *Journal of applied science and technology trends*, 2(01), 20-28.
- Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, 20(1), 3-29.