Real Estate Digital Intelligence: Home Rental Prices in the Province of Alicante Using Machine Learning

Mora-Garcia, Raul-Tomas;
Cespedes-Lopez, Maria-Francisca;
Perez-Sanchez, Vicente-Raul; Perez-Sanchez, Juan-Carlos.

[1] Departamento/Universidad: Edificación y Urbanismo, Universidad de Alicante, rtmg@ua.es, paqui.cespedes@ua.es, raul.perez@ua.es, jc.perez@ua.es

Abstract

The emergence of Artificial Intelligence (AI) has deeply transformed various sectors, including the real estate industry, where its application improves operational efficiency, personalizes services, and facilitates decision-making through predictive analysis. In this context, the present research aims to develop a methodology for the training, optimization, and interpretation of machine learning models focused on estimating the rental price of multifamily housing in the province of Alicante. To achieve this, a large georeferenced rental price database, cross-sectional and grouped, collected between 2019 and 2024, was used. The proposed methodology covers all phases of the modeling cycle: from data preparation and cleaning, feature engineering, model training and hyperparameter optimization, to performance evaluation and result interpretation. The results show that boosting-based machine learning algorithms offer the best trade-off between predictive performance, generalization capability, and computational efficiency. The interpretability analysis of machine learning models, both global and local, has allowed the identification of the most relevant features in predicting rental prices. Together, the

results demonstrate the potential of AI to support automated property valuation processes and facilitate informed decision-making in complex urban environments.

Keywords: Housing Price, Mass Valuation, Machine Learning, Hyperparameters, Alicante.

1. Introduction

In the last decade, Artificial Intelligence (AI) has emerged with great force, transforming various aspects of people's daily lives, providing innovative and efficient solutions to a wide range of tasks. AI has proven to be versatile and highly useful in applications such as virtual assistants and chatbots, pattern recognition and computer vision, natural language processing for translation or text summarization, recommendation systems, medical diagnostics, autonomous driving, and even weather forecasting.

One of the existing problems with the implementation of AI algorithms is that they do not "explain" how they arrive at a particular result or decision. This is particularly evident in machine learning models developed by private companies, where this information is intentionally hidden as trade secrets to protect intellectual property and/or maintain a competitive edge in the market.

AI has begun to play a significant role in the real estate market, introducing innovations that improve efficiency, personalization, and decision-making for users. Examples of these include real estate platforms that use AI in property search and recommendation, virtual property viewing assistants, predictive market analysis, risk assessment, credit or financial fraud evaluation, among other applications.

In Spain, there is a situation of rising property sales and rental prices, which is causing significant problems in accessing housing, whether for purchase or rent. The increase in prices has occurred unevenly across different areas of Spain, particularly affecting large cities and tourist areas. This has raised concerns about housing accessibility, especially for young people and those with lower incomes. Many circumstances contribute to this price escalation,

such as housing shortages, the impact of tourism and short-term rentals, foreign investments, rising mortgage costs due to interest rate increases, inflation, rising construction material prices, and recent public policies affecting landlords and sellers.

Currently, the general public often has limited knowledge about the real estate market, its dynamics and evolution, and how new technologies can help better understand the issue. For this reason, it is considered necessary to foster a "digital citizenship" that learns to use digital tools responsibly to address their doubts and questions about this topic, thereby promoting greater "digital intelligence" in our society.

In this context, the authors of this research believe that providing quality information to future buyers and renters of properties can efficiently assist in housing searches and user decision-making. Additionally, professionals involved in property sales, real estate appraisers, housing administration technicians, property developers, as well as small and medium-sized investors, can also benefit from this new source of information and technology about the real estate market proposed in this research.

The main objective of this work is to develop a methodology for training and optimizing machine learning algorithms aimed at estimating the rental price of multifamily housing in the province of Alicante. This methodology is implemented in an open-access web application, designed to provide updated information about the residential rental market in large and medium-sized municipalities in the province. The application aims to facilitate access to this data even for users with limited technological skills, promoting an accessible and efficient use of artificial intelligence-based tools for understanding and analyzing the local real estate market.

The automated estimation of housing prices has received growing attention in the scientific literature, driven by the availability of large volumes of structured and georeferenced data, as well as advances in AI techniques. In particular, machine learning algorithms have demonstrated significant potential for capturing complex relationships between multiple variables, overcoming linearity limitations, and offering high levels of performance in predicting real estate prices.

A first widely documented approach involves comparing the predictive behavior of traditional regression models, especially hedonic price models (HPM), with various machine learning algorithms (1). Empirical evidence gathered in numerous studies (2-12) suggests that classic models, while useful for the economic interpretation of price determinants, have lower predictive capacity compared to machine learning algorithms, especially in contexts with high dimensionality and non-linear relationships between variables.

A second approach in the literature focuses on identifying the most suitable machine learning algorithms for predicting property prices in different geographic and temporal contexts (1, 13-17). These studies comparatively evaluate models such as Random Forest (RF), Gradient Boosting (GBR), Extreme Gradient Boosting (XGBoost), LightGBM, Support Vector Machines (SVM), and deep neural networks (DNN), applying metrics such as MAE, RMSE, and R^2 on different data subsets. The results tend to agree on the superiority of boosting-based algorithms, particularly in their ability to model non-linear relationships and handle variable interactions without requiring prior functional specification.

This document is organized as follows. Section 2 describes the materials and methods, detailing the sources used and the generated database. Section 3 presents the results of the training, optimization, and validation process of the machine learning models, along with their interpretation. Section 4 provides a summary of the conclusions.

2. Materials and Method

2.1. Information Sources and Database

The information on property offer prices was extracted from a real estate web portal, collecting asking prices as well as the characteristics of both the properties and the buildings. Monthly downloads were made over six years, from 2019 to 2024. The collected data includes rental asking prices, property characteristics (type, built area, number of bedrooms and bathrooms, etc.), building attributes (elevator, parking space, swimming pool, etc.), and geographical location (geographic coordinates).

After the initial raw data download, a review was conducted to detect inconsistencies and outliers. A missing data analysis was then performed, identifying and discarding properties without price, geographic coordinates, or essential features. Variables with low variability or a high percentage of missing values were also removed. Once this stage was completed, a second cleaning process was carried out to eliminate duplicate properties with identical characteristics.

To model the effect of time, a categorical feature representing the temporal presence of each property in the market was incorporated, identifying the specific quarter in which it was listed. Finally, the information was structured into a pooled cross-section dataset. The multifamily rental property database consists of 59,875 unique properties. The complete temporal data sample includes 101,724 observations, spanning from 2019 to 2024. Table 1 lists the features used in the project, categorized into four groups.

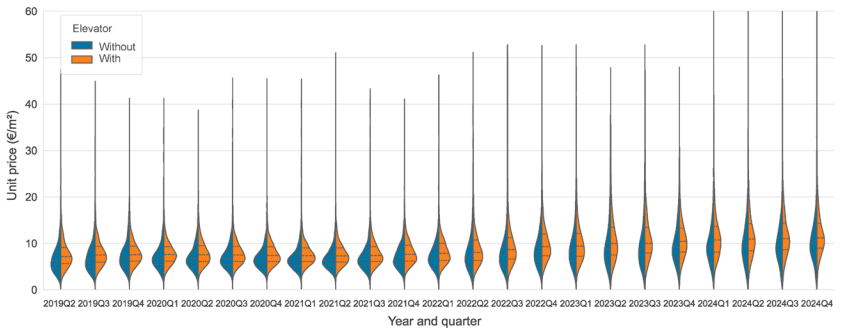
Table 1: Characteristics of the variable set to be used.

Category	Characteristic	Value Type	Description
Dwelling characteristics	Typology	Categorical	Categorical feature identifying the dwelling typology (flat, penthouse, duplex/triplex, studio/loft, and ground floor)
	Area m2	Numerical	Build area of the dwelling (m^2)
	Bedrooms	Numerical	Number of bedrooms in the dwelling
	Bathrooms	Numerical	Number of bathrooms in the dwelling
	Toilets	Numerical	Number of toilets in the dwelling
	Air conditioning	Dichotomous	Air conditioning availability
	Heating	Dichotomous	Heating availability
	New construction	Dichotomous	Indicates whether the dwelling is a new build
Building characteristics	Elevator	Dichotomous	Elevator availability
	Garage	Dichotomous	Parking space availability
	Storage room	Dichotomous	Storage room availability
	Pool	Dichotomous	Swimming pool availability
	Terrace	Dichotomous	Open terrace availability

Category	Characteristic	Value Type	Description
Location characteristics	Longitude	Numerical	Geographic coordinates of the spatial location (in decimal degrees), WGS84 datum
	Latitude		
	Municipality	Categorical	Categorical feature identifying the municipality in which each property is located
Temporal characteristics	Year-Quarter	Categorical	Categorical feature for modeling the time factor (23 quarters, from 2019Q2 to 2024Q4)
Dependent feature	Price	Numerical	The asking rental price of the multi-family dwelling in euros

Figure 1 shows the distribution of unit rental prices based on the presence or absence of an elevator. It can be observed that the price distribution for properties with an elevator is shifted and more stretched toward higher price ranges compared to those without an elevator. Additionally, the price range has widened in recent years (between 2022 and 2024), in contrast to the period from 2019 to 2021, when prices were more concentrated within narrower ranges. The average unit rental price in the province of Alicante has increased from €7.67/m² (*SD*=3.94) in the second quarter of 2019 to €11.70/m² (*SD*=5.15) in the fourth quarter of 2024.

Figure 1: Violin chart showing the distribution of unit prices (€/m²) for rental housing with and without an elevator, according to the quarter of supply.



2.2. Methodology

In this research, various ensemble learning algorithms have been implemented and evaluated with the aim of estimating housing rental prices based on large datasets. The ensemble learning algorithms used include boosting-based methods: Gradient Boosting Regressor (GBR), Extreme Gradient Boosting (XGBM), and Light Gradient Boosting Machine (LGBM); as well as bagging-based methods: Random Forest (RF) and Extra Trees Regressor (ET).

The Python programming language was used, employing the pandas and numpy libraries for data processing. For the implementation of machine learning algorithms, the scikit-learn and scikit-optimize libraries were used, along with lightgbm and xgboost. For plotting, the matplotlib, seaborn, and yellowbrick libraries were employed. For model interpretation, the scikit-learn, shap, and eli5 packages were used.

All the steps typically involved in a standard machine learning workflow were addressed: data collection, preparation and exploratory analysis, preprocessing and feature engineering, model training and hyperparameter optimization, model evaluation and selection, model interpretability, and finally, web deployment and monitoring. A more detailed description of these steps is provided below:

Data Preparation: An exploratory data analysis is performed, followed by a data cleaning and transformation process prior to conducting data mining tasks. This process involves formatting the data (grouping, discretization, binarization), enriching it with information from other sources, identifying and handling outliers and missing values, or transforming variables (standardization, normalization).

Feature Engineering: Techniques are used to extract features from raw data through data mining techniques (feature creation, feature extraction, and feature selection). These techniques enhance the performance of machine learning algorithms. Based on several baseline models, the possibility of encoding categorical variables is evaluated; new features are generated through aggregation, transformation, or combination of existing ones; new features are created to summarize original variables (principal and discriminant component analysis, auto-encoding); and the most relevant features are selected using

recursive selection methods (sequential feature selection, recursive feature elimination).

Training and Model Selection: Multiple candidate models are trained to estimate their predictive power and determine which algorithms perform best. Model selection is based on error metrics (MAE, RMSE, MAPE, etc.), goodness of fit (R^2), or other selection indices that consider both fit and model complexity (BIC and AIC).

Hyperparameter Optimization: This phase aims to improve the model's fit and/or minimize prediction errors. Once the best algorithm(s) have been identified, the impact of hyperparameters is assessed and fine-tuned using cross-validation techniques. A subset of the training data is used to apply a variety of hyperparameter combinations and evaluate their performance on another validation subset.

Model Evaluation: In this phase, the models are evaluated to identify which one performs best in predicting the dependent variable. The potential for overfitting, detected in the previous phase through cross-validation and the use of training and test subsets, is also assessed.

Model Interpretation: Tools are used to identify the most important features in the model using global and local approaches, as well as visual techniques to interpret the predictions. These include Partial Dependence Plots (PDP), Individual Conditional Expectation (ICE) plots, Accumulated Local Effects (ALE) plots, and SHAP (SHapley Additive exPlanations) values, among others.

Model Deployment: In this phase, a web platform is developed to deploy the model into production, allowing predictions to be made based on initial input data.

3. Results

3.1. Model Training and Optimization

The goal of hyperparameter optimization is to improve model fit and minimize prediction errors as much as possible. This process was carried out exclusively on the training partition, which corresponds to 70% of the original dataset. This partition was used in cross-validation

with a three-fold scheme, generating separate subsets: CV-training and CV-validation. All data splits (training and test; and CV-training and CV-validation) were done in order to separate the datasets based on a group identifier, using the "GroupShuffleSplit" method from the scikit-learn library, with the property identifier used as the grouping attribute.

Once the dataset was prepared and the training pipeline defined, each algorithm was trained using two hyperparameter search strategies: random search and Bayesian search, with a maximum of 20 iterations in each case. This limit was chosen due to the computational complexity inherent in some algorithms, such as RF and ET, which have high processing times during this stage.

Table 2 presents the performance results from the training and optimization phase of the machine learning algorithms. Figure 2 shows a box plot with the distribution of the values obtained during cross-validation (CV-training and CV-validation), considering both the random and Bayesian search strategies for each of the evaluated algorithms.

The best results were achieved with the boosting-based algorithms, in the following order: LGBM, GBR, and XGBM. In contrast, the bagging-based algorithms, such as RF and ET, did not achieve competitive results compared to their boosting counterparts during hyperparameter optimization.

Table 2: Hyperparameter tuning results using cross-validation (R^2 values in CV-validation).

Model	**Name**	**Hyperparameter Optimization (CV-validation)**		
		Random search	**Bayesian search**	**Best fit**
Random Forest Regressor	RF	**0.5639** (0.0081)	0.5601 (0.0062)	Random Search
Extra Trees Regressor	ET	**0.5384** (0.0031)	0.5285 (0.0011)	Random Search
Gradient Boosting Regressor	GBR	0.5857 (0.0082)	**0.6134** (0.0032)	Bayesian Search
Extreme Gradient Boosting	XGBM	**0.6099** (0.0059)	0.6000 (0.0061)	Random Search

Model	Name	Hyperparameter Optimization (CV-validation)		
		Random search	Bayesian search	Best fit
Light Gradient Boosting Machine	LGBM	0.6077 (0.0065)	**0.6143** (0.0031)	Bayesian Search

Note: R^2 value and standard deviation (SD) in parentheses.

Figure 2: Performance results of hyperparameter tuning using cross-validation with random and Bayesian search strategies (R^2 values for CV-training and CV-validation).

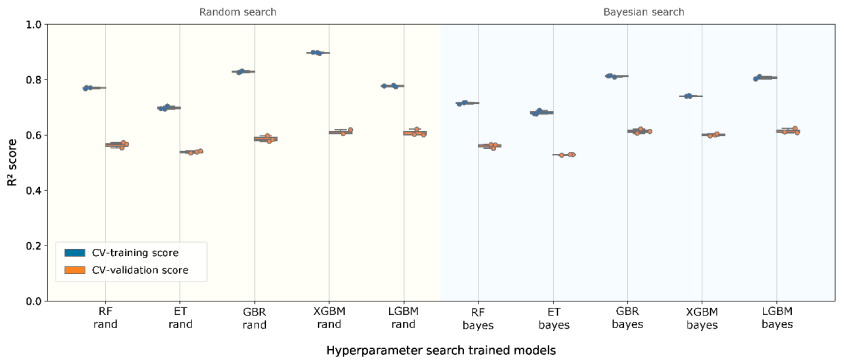
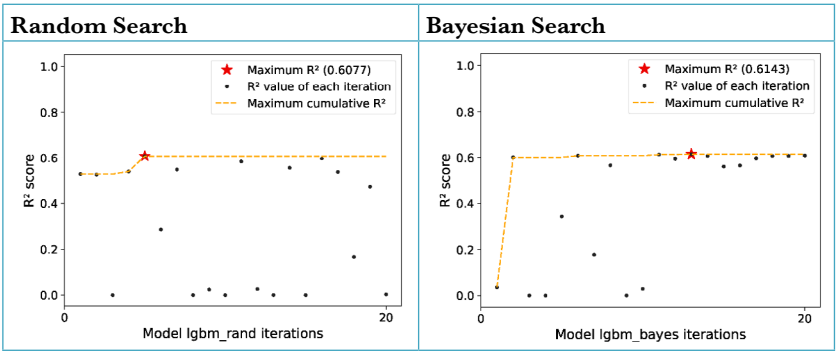


Figure 3 shows the learning curves of hyperparameter search for the LGBM algorithm. The results indicate that neither of the two hyperparameter search strategies demonstrated significantly superior performance over the other. While the Bayesian search required longer processing times, the random search produced a greater number of suboptimal hyperparameter combinations. This highlights the trade-off between processing time and efficiency in generating optimal configurations, an important factor to consider in future developments and applications of the model.

Figure 3: Learning curves for hyperparameter search of the LGBM algorithm.



★ *Maximum R², • R² value for each iteration, --- Cumulative maximum R².*

3.2. Model Evaluation and Selection

At this stage, the candidate algorithms are trained in order to evaluate their predictive capabilities and determine which models perform best on the test data subset (30% of the original dataset). Model selection is carried out using the error and goodness-of-fit metrics previously described in the methodology section.

Once the best hyperparameters were identified in the previous phase, the corresponding models were trained, and performance metrics were extracted for both the training and test sets. The values presented in the cross-validation column (CV-validation) reflect the performance of the selected models during the prior stage and are included here for comparative and illustrative purposes. The overfitting column was calculated as the percentage difference between performance on the test and training sets, providing an estimate of each algorithm's generalization capability.

Table 3 presents the performance and overfitting values for the different evaluated algorithms. Among the best-performing models (LGBM, GBR, and XGBM), the XGBM algorithm shows a considerable degree of overfitting (+40.6%), in contrast with the more moderate levels of LGBM (+23.5%) and GBR (+27.6%). Overall, the results are stable and consistent, as demonstrated by the small

discrepancy between the R^2 values on the test set and those obtained during cross-validation (CV-validation), with differences ranging from 0.013 to 0.032.

Table 3: Performance results (R^2 values) of the trained algorithms for the rental price prediction models.

Model	Name	CV-validation R^2 values (*SD*)	R^2 values		
			Training set (70%)	Test set (30%)	Overfitting (%)
Linear Regression	LR	-	0.4426	0.4290	-
Random Forest Regressor	RF	0.5601 (0.0062)	0.7394	0.5925	+24.8
Extra Trees Regressor	ET	0.5384 (0.0031)	0.6876	0.5569	+23.5
Gradient Boosting Regressor	GBR	0.6134 (0.0032)	0.8018	0.6286	+27.6
Extreme Gradient Boosting	XGBM	0.6099 (0.0059)	0.8753	0.6227	+40.6
Light Gradient Boosting Machine	LGBM	0.6143 (0.0031)	0.7810	0.6324	+23.5

Note: Training subset $N = 71{,}019$; Test subset $N = 30{,}705$.

Table 4 presents the error metrics for each algorithm, specifically MAE, MSE, and RMSE. As these metrics are error-oriented, higher values clearly indicate worse performance. In this regard, the models based on boosting techniques (GBR, XGBM, and LGBM) continue to stand out as the best overall performers.

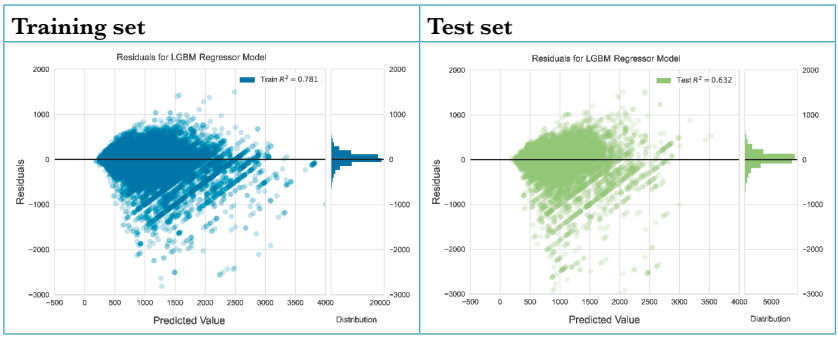
Table 4: Performance results (error values) of the trained algorithms on the test subset.

Model	Name	Performance metrics		
		MAE	MSE	RMSE
Linear Regression	LR	211	118,524	344
Random Forest Regressor	RF	164	84,577	290
Extra Trees Regressor	ET	173	91,970	303
Gradient Boosting Regressor	GBR	152	77,087	277
Extreme Gradient Boosting	XGBM	152	78,310	279
Light Gradient Boosting Machine	LGBM	152	76,297	276

Note: MAE = Mean Absolute Error; MSE = Mean Squared Error; RMSE = Root Mean Squared Error.

Residual plots provide a visual and qualitative evaluation that complements the quantitative performance metrics of the model. Figure 4 illustrates the behavior of the LGBM algorithm on both the training and test sets, allowing for an assessment of its generalization capability. The residual distribution appears random and slightly centered around zero, without the presence of systematic patterns or discernible structures. This behavior suggests that the model effectively captures the underlying relationship between the independent variables and the dependent variable. Likewise, the similarity in the residual distribution across both the training and test sets indicates a low incidence of overfitting, reinforcing the model's robustness and its ability to generalize to unseen data.

Figure 4: Residual plots of the trained LGBM algorithm.



Regarding training time, XGBM and LGBM are positioned as the most computationally efficient algorithms, with XGBM being particularly fast across all evaluated sample sizes. In contrast, the RF and ET algorithms exhibit significantly longer computation times, which can become prohibitive for large datasets.

For the final evaluation of model performance, the algorithms were retrained using the entire dataset (100% of the data) and the best hyperparameters previously obtained. The corresponding results are presented in Table 5, showing high R^2 values for XGBM, GBR, and LGBM. However, the final model selection should consider not only predictive performance but also other factors such as computational efficiency and generalization capability. Although XGBM achieves the best fit, its high level of overfitting compromises its ability to generalize to new data. Consequently, GBR and LGBM emerge as

the most robust options, combining strong performance with lower overfitting risk and shorter computation times, making them more suitable candidates for future applications involving previously unseen data.

Table 5: Error metrics and goodness-of-fit measures of the algorithms on the entire dataset (multiple metrics).

Algorithm name	Model performance for the entire data sample			
	MAE	MSE	RMSE	R^2
LR	207.5	109,097.6	330.3	0.4380
RF	137.3	52,531.8	229.2	0.7294
ET	151.3	63,541.7	252.1	0.6727
GBR	117.7	40,102.2	200.2	0.7934
XGBM	99.7	27,917.7	167.1	0.8562
LGBM	128.9	46,564.5	215.8	0.7601

Note: MAE = Mean Absolute Error; MSE = Mean Square Error; RMSE = Root Mean Squared Error; R^2 = Coefficient of Determination.

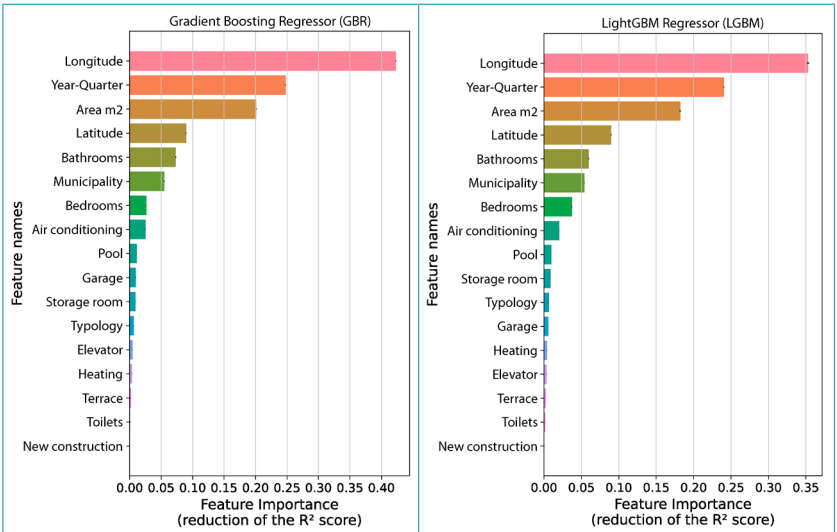
3.3. Model Interpretation

Figure 5 presents the results of the feature importance evaluation using permutation, applied to the two best-performing algorithms in the previous stages of the study (GBR and LGBM). In both models, variables related to the geographic location of the property stand out, with longitude being particularly relevant. This indicates significant variation in prices based on east-west location within the province of Alicante.

The quarter feature (year and quarter) also appears among the most influential, reflecting a marked increase in rental prices over the studied time period.

Regarding the intrinsic characteristics of the property, built area, number of bathrooms, and number of bedrooms are consolidated as the most decisive attributes in price estimation. In contrast, the presence of an elevator does not appear to be a particularly relevant variable in the rental market.

Figure 5: Relative importance of the most relevant features according to the algorithms GBR y LGBM.

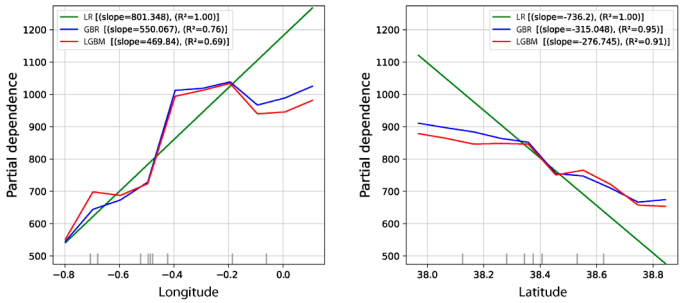


a) GBR algorithm b) LGBM algorithm

Figure 6 shows the partial dependence plots corresponding to the location features. These plots reveal that rental prices tend to be higher in the eastern areas of the region, while they progressively decrease toward the west.

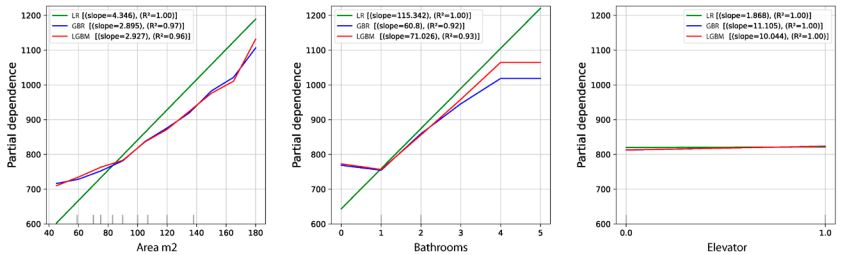
The GBR and LGBM models capture more complex relationships than the linear regression (LR) model, although they maintain a consistent trend with it, particularly in the analysis of the latitude coordinate. The only notable discrepancy lies in the behavior of the longitude coordinate, where the nonlinear models introduce greater variability in the shape of the curve.

Figure 6: Unidirectional partial dependence plots for the longitude and latitude coordinates with the LR, GBR, and LGBM algorithms.



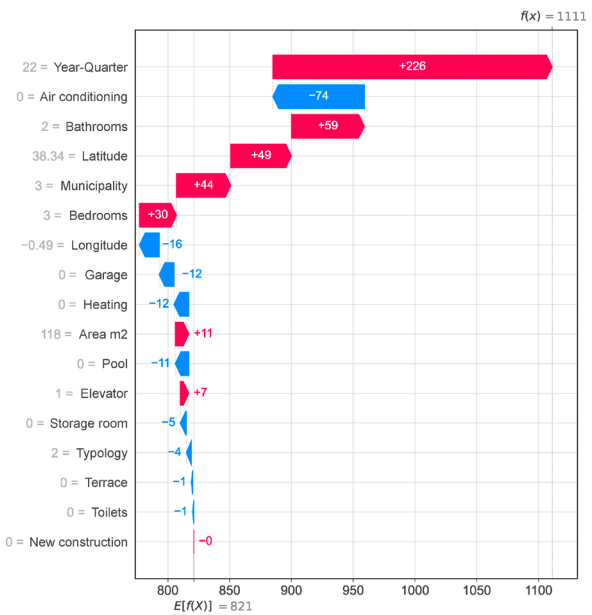
To illustrate the behavior of other features, three representative features have been selected, which are shown in the PDP plots in Figure 7: two continuous features (constructed area and number of bathrooms) and one binary feature (presence of an elevator). The interpretation of these plots has been complemented with the calculation of the slopes of the best-fit lines, which allow estimating the average impact of each variable on rental price. Thus, for every additional square meter of area, the monthly rent increases by an average of approximately 3 euros. In the case of the number of bathrooms, an additional bathroom results in an estimated increase of between 61 and 71 euros per month. The presence of an elevator is associated with an average increase of between 10 and 11 euros compared to a similar property that lacks this feature.

Figure 7: Unidirectional partial dependence plots for the features area, number of bathrooms, and elevator obtained with the LR, GBR, and LGBM algorithms.



Regarding the local interpretation of the model, Shapley values (SHAP) have been used, a technique based on cooperative game theory that allows decomposing the result of a prediction into the individual contributions of each feature. Figure 8 presents a waterfall plot that shows the SHAP estimation for a specific case: a property offered for rent in the city of Alicante during the fourth quarter of 2024. In this plot, the value E[f(x)] represents the average model prediction across the entire sample (821 €), which acts as the base value. From this value, the contributions of each feature are added or subtracted until the final model prediction, f(x) = 1,111 €, is reached. Features that increase the price are shown in red, while those that decrease it are represented in blue. In this example, the year-quarter feature has the highest positive impact on the price, contributing +226 €, while the absence of air conditioning has a negative impact of −74 € on the final estimate. This type of explanation provides a transparent and understandable view of the model's functioning at an individual level, facilitating its practical application and boosting confidence in its predictions.

Figure 8: Waterfall plot for a rental database observation, estimation performed using the LGBM algorithm.



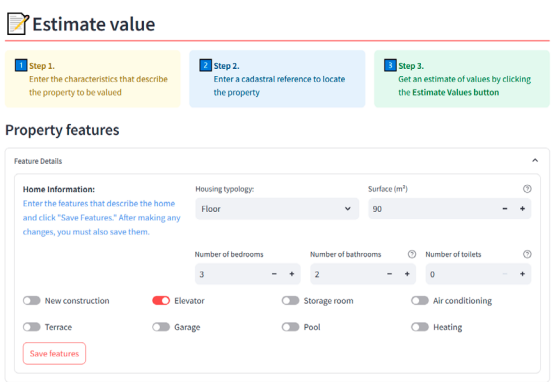
3.4. Model Deployment

In this phase of the research, a web application has been developed to put the predictive model into production, allowing estimations based on data manually entered by the user. The tool is available with open access, aimed at facilitating its use by anyone interested, including real estate professionals and potential investors who wish to run simulations and obtain automated estimates.

One of the fundamental requirements for development was to ensure that both server maintenance and the use of the virtual machine were free of charge, additionally promoting the exclusive use of open-source software throughout the implementation. Python programming language was used for developing the application, integrating the Streamlit package, which allows the deployment of interactive web applications with the possibility of free hosting (https://preciosdevivienda.streamlit.app/).

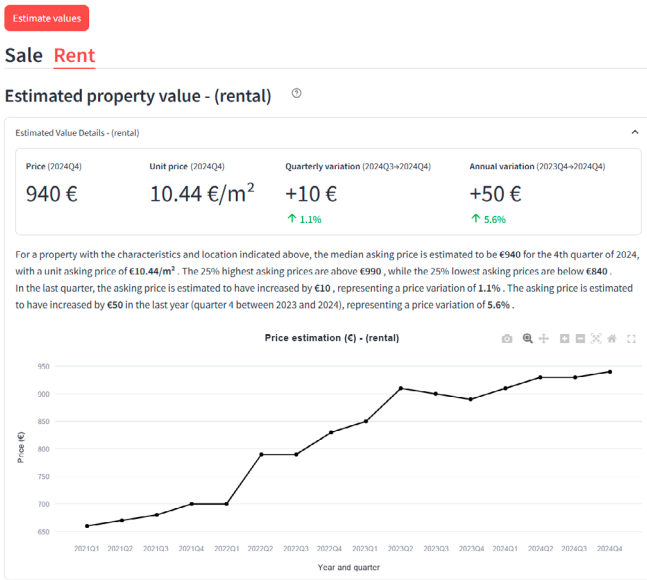
The data entry interface is organized into three main sections: 1) Property characteristics (Figure 9); 2) Geographic location; and 3) Estimation results (Figure 10). In the first section, the user can specify the property type, built area, number of bedrooms, bathrooms, and toilets, as well as other relevant additional features. In the location section, the user is asked to provide a valid cadastral reference, either 14 or 20 characters long. The system performs an automatic check to verify the existence of the property, providing an approximate address along with its geographical coordinates.

Figure 9: Data entry screen for property characteristics.



In the final section, the application displays the estimation results. A summary of the total estimated value (in euros) and the unit value (euros per square meter) is provided, along with the price variation during the last quarter and the past year. To facilitate the interpretation of the results, a brief textual explanation is included, highlighting the most relevant elements of the analysis. Additionally, a graph showing the historical evolution of the property's estimated price is provided, allowing the user to contextualize the price trends over time.

Figure 10: Estimation results screen, summary, and estimation graph.



Additionally, a waterfall chart is included, representing the SHAP (SHapley Additive exPlanations) values, which illustrates the process by which the model adjusts the estimated price starting from the average value (see Figure 8). This chart allows users to identify the features that contribute positively to the increase in the property's price, as well as those that have a negative effect, reducing the estimated value. This tool facilitates a better understanding of the model's behavior and provides relevant information about the relative influence of each feature on the final prediction.

4. Conclusions

This study has allowed the development, optimization, and interpretation of a set of machine learning models aimed at estimating housing rental prices. Through a systematic process of training, validation, and evaluation, it has been demonstrated that boosting-based algorithms—particularly LGBM and GBR—offer the best balance between predictive performance, generalization capability, and computational efficiency.

The optimization of hyperparameters through cross-validation has highlighted the usefulness of combining random and Bayesian search strategies, although with limited differences in their final performance. The LGBM and GBR models have shown consistent results across various error metrics (MAE, MSE, RMSE) and the coefficient of determination (R^2), as well as a lower tendency for overfitting compared to XGBM, which, despite achieving the best fit on the data, has a more limited generalization capability.

The interpretability analysis of the machine learning models, both global and local, has identified the most relevant features in predicting rental prices. Among these, geographic location (especially longitude), the temporal year-quarter, built area, and the number of bathrooms stand out, while other features, such as the presence of an elevator, have a marginal influence in this market.

From an applied perspective, a publicly accessible web application has been developed, allowing any user—whether individuals, professionals, or investors—to estimate the current and historical rental value of a property based on its characteristics. Additionally, this tool provides detailed explanations through SHAP values, contributing to a better understanding and increased confidence in the predictions generated.

Overall, this work not only proposes a rigorous and replicable approach for rental price estimation using artificial intelligence but also offers an interpretable and accessible system with practical application potential in contexts such as automated valuation, urban planning, and decision-making in the real estate market.

Funding

Research Project with reference GRE23-05A, "Grants for research projects and networks, Category A (Annex X)", funded by the University of Alicante (BOUA 10/05/2023). An initial prototype of the project was developed within the framework of the Research Project with reference CENID2024/12, funded by the Digital Intelligence Center of the University of Alicante (CENID).

5. Bibliography

[1] Park B, Bae JK. Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications [Internet]. 2015;42(6):2928–34. Available from: https://doi.org/10.1016/j.eswa.2014.11.040

[2] Antipov EA, Pokryshevskaya EB. Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. Expert Systems with Applications [Internet]. 2012;39(2):1772–8. Available from: https://doi.org/10.1016/j.eswa.2011.08.077

[3] Čeh M, Kilibarda M, Lisec A, Bajat B. Estimating the Performance of Random Forest versus Multiple Regression for Predicting Prices of the Apartments. ISPRS International Journal of Geo-Information [Internet]. 2018;7(5):168. Available from: https://doi.org/10.3390/ijgi7050168

[4] Embaye WT, Zereyesus YA, Chen B. Predicting the rental value of houses in household surveys in Tanzania, Uganda and Malawi: Evaluations of hedonic pricing and machine learning approaches. PLOS ONE [Internet]. 2021;16(2):e0244953. Available from: https://doi.org/10.1371/journal.pone.0244953

[5] Gnat S. Property Mass Valuation on Small Markets. Land [Internet]. 2021;10(4):388. Available from: https://doi.org/10.3390/land10040388

[6] Hong J. An Application of XGBoost, LightGBM, CatBoost Algorithms on House Price Appraisal System. Housing Finance Research [Internet]. 2020;4:33–64. Available from: https://doi.org/10.52344/hfr.2020.4.0.33

[7] Hong J, Choi H, Kim W-S. A house price valuation based on the random forest approach: the mass appraisal of residential property in South Korea. International Journal of Strategic Property Management [Internet]. 2020;24(3):140–52. Available from: https://doi.org/10.3846/ijspm.2020.11544

[8] Kok N, Koponen E-L, Martínez-Barbosa CA. Big Data in Real Estate? From Manual Appraisal to Automated Valuation. Journal of Portfolio Management [Internet]. 2017;43(6):202–11. Available from: https://doi.org/10.3905/jpm.2017.43.6.202

[9] Rico-Juan JR, Taltavull de La Paz P. Machine learning with explainability or spatial hedonics tools? An analysis of the asking prices in the housing market in

Alicante, Spain. Expert Systems with Applications [Internet]. 2021;171:114590. Available from: https://doi.org/10.1016/j.eswa.2021.114590

[10] Xu L, Li Z. A New Appraisal Model of Second-Hand Housing Prices in China's First-Tier Cities Based on Machine Learning Algorithms. Computational Economics [Internet]. 2021;57(2):617–37. Available from: https://doi.org/10.1007/s10614-020-09973-5

[11] Yilmazer S, Kocaman S. A mass appraisal assessment study using machine learning based on multiple regression and random forest. Land Use Policy [Internet]. 2020;99:104889. Available from: https://doi.org/10.1016/j.landusepol.2020.104889

[12] Mora-Garcia RT, Cespedes-Lopez MF, Perez-Sanchez VR. Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times. Land [Internet]. 2022;11(11):2100. Available from: https://doi.org/10.3390/land11112100

[13] Alfaro-Navarro J-L, Cano EL, Alfaro-Cortés E, García N, Gámez M, Larraz B. A Fully Automated Adjustment of Ensemble Methods in Machine Learning for Modeling Complex Real Estate Systems. Complexity [Internet]. 2020;2020(1):5287263. Available from: https://doi.org/10.1155/2020/5287263

[14] Canaz Sevgen S, Aliefendioğlu Y. Mass Apprasial With A Machine Learning Algorithm: Random Forest Regression. Bilişim Teknolojileri Dergisi [Internet]. 2020;13(3):301–11. Available from: https://doi.org/10.17671/gazibtd.555784

[15] Ho WKO, Tang B-S, Wong SW. Predicting property prices with machine learning algorithms. Journal of Property Research [Internet]. 2021;38(1):48–70. Available from: https://doi.org/10.1080/09599916.2020.1832558

[16] Hu L, He S, Han Z, Xiao H, Su S, Weng M, et al. Monitoring housing rental prices based on social media: An integrated approach of machine-learning algorithms and hedonic modeling to inform equitable housing policies. Land Use Policy [Internet]. 2019;82:657–73. Available from: https://doi.org/10.1016/j.landusepol.2018.12.030

[17] Pai P-F, Wang W-C. Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Prices. Applied Sciences [Internet]. 2020;10(17):5832. Available from: https://doi.org/10.3390/app10175832