Housing-Price Prediction in Colombia using Machine Learning

Predicción de Precios de Viviendas en Colombia usando Machine Learning

Miguel Ángel Correa Manrique (1)
Omar Becerra Sierra (2)
Daniel Otero Gómez (3)
Henry Laniado (4)
Rafael Mateus Carrión (5)
David Andres Romero Millan (6)

(1)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: macorream@eafit.edu.co
(2)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: oabecerras@eafit.edu.co
(3)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: doterog@eafit.edu.co
(4)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: hlaniado@eafit.edu.co
(5)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: rmateusc@eafit.edu.co
(6)Universidad EAFIT. Mathematical Science Department. Medellín, Colombia. E-mail: dromero1@eafit.edu.co

*Correspondence author: macorream@eafit.edu.co

ABSTRACT

It is a common practice to price a house without proper evaluation studies being performed for assurance. That is why the purpose of this study provide an explanatory model by establishing parameters for accuracy in interpretation and projection of housing prices. In addition, it is intentioned to establish proper data preprocessing practices in order to increase the accuracy of machine learning algorithms. Indeed, according to our literature review, there are few articles and reports on the use of Machine Learning tools for the prediction of property prices in Colombia. The dataset in which the research is built upon was provided by an existing real estate company. It contains near 940,000 items (housing advertisements) posted on the platform from the year 2018 to 2020. The database was enriched using statistical imputation techniques. Housing prices prediction was performed using Decision Tree Regressors and LightGBM methods, thus deriving in better alternatives for house price prediction in Colombia. Moreover, to measure the accuracy of the proposed models, the Root Mean Squared Logarithmic Error (RMSLE) statistical indicator was used. The best cross validation results obtained were 0.25354 ± 0.00699 for the LightGBM, 0.25296 ± 0.00511 for the Bagging Regressor, and 0.25312 ± 0.00559 for the ExtraTree Regressor with Bagging Regressor, and it was not found a statistical difference between their performances.

Keywords: Machine learning, housing-price prediction, decision-tree regression, LightGBM.

RESUMEN

Es común determinar el precio de un domicilio sin llevar a cabo algún estudio para garantizar su certeza. Por ello, el propósito de este estudio es proveer un modelo explicativo mediante el establecimiento de parámetros para la precisión en la interpretación y proyección de los precios de vivienda. Adicionalmente, se quieren establecer prácticas de preprocesamiento de datos apropiadas que incrementen la precisión de los algoritmos. En efecto, de acuerdo con nuestra revisión de literatura, existen pocos artículos y reportes en el uso de herramientas de Machine Learning para la predicción de los precios de vivienda en Colombia. El conjunto de datos en el cual el estudio se basó fue proporcionado por una empresa inmobiliaria existente. Contiene 940,000 ítems con respecto a anuncios de viviendas publicadas en la plataforma desde el año 2018 hasta el 2020. El conjunto de datos se enriqueció usando técnicas de imputación estadística. La predicción de los precios de las viviendas fue realizada usando métodos de Árbol de Regresión Multivariable y LightGBM, derivando así mejores alternativas en la predicción de los precios de vivienda en Colombia. Adicionalmente, para medir la exactitud de nuestros modelos, la Raíz Cuadrada del Error Cuadrático Medio (RMSLE por sus siglas en inglés) fue utilizada como evaluador estadistico. Los mejores resultados obtenidos en la validaciónón cruzada fueron 0.25354±0.00699 para el LightGBM, 0.25296 ±0.00511 para el Bagging Regressor y 0.25312±0.00559 para el ExtraTree Regressor con Bagging Regressor, y no fue encontrada diferencia estadísticamente significativa entre estas.

Palabras Clave: Aprendizaje máquina, predicción de precios de viviendas, arboles de regresión, LightGBM.

INTRODUCTION

Predicting housing prices is, undoubtedly, a matter of interest in the industry, and particularly for employees in the real estate business, as it helps improve their efficiency and profits. Accelerated urban and population growth has increased the use of real estate appraisal [(1)]. In addition, housing valuation use has incremented recently, since it is required for processes like taxation, requesting a loan, buying and selling properties, etc. [(1)]. The necessity of developing a tool that resolves this issue has raised an interest in the study of this topic. Due to all of this, the prediction of housing prices is a common field of study, and it has received substantial attention. Indeed, various models and techniques have been developed for this subject.

As per the literature, the housing market has been analyzed on research through two principal lenses: hedonic demand theory approaches and machine learning techniques for housing price-prediction models. The first one has been used for several decades, and it bases on the theory that a good is a set of individual characteristics, which have their own price, and when a good is purchased these attributed are who determine the value of the utility. Nonetheless, it possesses limitations regarding model assumptions and estimations such as market disequilibrium, the selection of independent variables, and market segmentation. As a result, recent studies have focused on price prediction performance comparison between these two [2, 3].

The purpose of this study is to establish proper data preprocessing practices in order to increase the accuracy of machine learning algorithms and provide an explanatory model by establishing parameters for accuracy in interpretation and projection of housing prices. To carry out the study a dataset consisting initially of almost 940,000 properties regarding various Colombian properties was used. These data were obtained through Properati*,* a platform dedicated to selling and renting properties. By using Bagging and Decision Tree Regressors, and LightGBM methods, the data was processed to predict housing prices. The main advantages of implementing these methods are the adaptability to any data type, and the identification of the relevant attributes in a dataset, which allows to adjust for a diverse number of conditions in the problem.

The remainder of this article is structured as follows: Literature regarding prediction of housing prices is reviewed in Related Work; the dataset structure, the metric used to evaluate the models, and the theoretical background of each algorithm is contained in Materials and Methods; information concerning the preprocessing techniques implemented can be found in Data Preprocessing; a documentation of how the experimentation was conducted and an overview of the final results is provided in Experimentation and Results; conclusions and future work are discussed in Conclusions; finally, in Acknowledgements, several members of the team are recognized for their contributions to this study.

RELATED WORK

A wide number of studies have been dedicated to determining the most suitable algorithm to perform price prediction within this context. Since a huge number of housing property ads can be found on the internet, plenty of studies can be carried out through their exploitation. Industries provide a great advantage for developers through the tabulation and processing of this data, which can be ultimately used for forecasting trends and assisting in a favorable decision-making process. While many articles focus on highlighting the advantages and limitations of certain approaches, others compare the performance of different methods and indicate which is the best. Algorithms vary from simple techniques such as several types of regression and decision tree or bagging based methods, to more complex ones, such as ensemble methods and neural network implementation.

Manjula et al [4] studied the use of univariate and multivariate linear and polynomial regression, finding that such approaches are rather ineffective to perform the task while used independently. They affirmed that high order polynomial regression tends to overfit the data and linear regression models tend to underfit it, while proposing that a mixed model would have better results than the ones implemented individually.

Discussion approaches based on decision trees provide a tool for statistical pattern recognition in the analysis of the relationship between housing characteristics and housing prices. Hong et al. [3] performed a study that compared classical hedonic models with the random forest algorithm. They work with 16,000 items regarding a developed are of South Korea, which represented 40% of the transactions performed at the area. Hong et al. found that the algorithm had an outstanding performance compared to what they called OLS models and listed the advantages that such methods possessed.

In Perez et al. [5] several algorithms are compared in order to find which one has the best performance. They worked with Linear Regression, Regression Trees, Random Forest and Bagging Regressor and affirmed that the Random Forest and Bagging Regression worked better than the Regression Trees and Linear Regression. In addition, the dataset contained a description feature, containing the description found in the website from where the data was collected, and Natural Language Processing was performed in order to extract attributes of the properties, such as whether the property has a pool, parking space, administration payment, kitchen features, etc. Moreover, feature engineering was performed in order to determine which are the most relevant features, finding that the area, number of bathrooms and age of the property are some of the most important features.

Certain authors have chosen to take a more complex approach to the subject. Klus et al. [6] built an ensemble model which processed housing attributes, non-structured text description and images independently using random forest and recurrent neural networks, ending with outstanding results. In addition, they proposed a study of the relationships of the variables with the prediction of house prices (specifically in Brazil) through the use of decision tree sets and Deep Learning tools.

Despite the broad research developed around the topic, little has been studied regarding data preprocessing, especially regarding imputation of missing values. The present work intends to differ from all those mentioned above, since classic machine learning and differentiated imputation techniques have been employed to predict housing prices, and the study is directed from a national perspective. Exploration and analysis have been also developed in order to understand how the acquired data does behave so that the performance of the models can be enhanced.

MATERIALS AND METHODS

The former section presents the details regarding the data, metrics, and the algorithm used in the research. It is crucial to consider certain factors such as missing values, scales, balances, and structure of the dataset to choose the most suitable algorithm. The study utilizes the LightGBM [7], Bagging Regressor [8], and Bagging Regressor with ExtraTree Regressor [9] framework to operate the housing price predicting model.

The Dataset

The data used in the study was obtained through Properati, a public platform dedicated to property selling and renting, containing data from several countries in Latin America such as Argentina, Colombia, Ecuador, Peru, and Uruguay. For the construction of the model, the dataset corresponding to Colombian properties was used. It initially contained nearly 940,000 items. The features that the dataset contains can be found in Table 1 below:

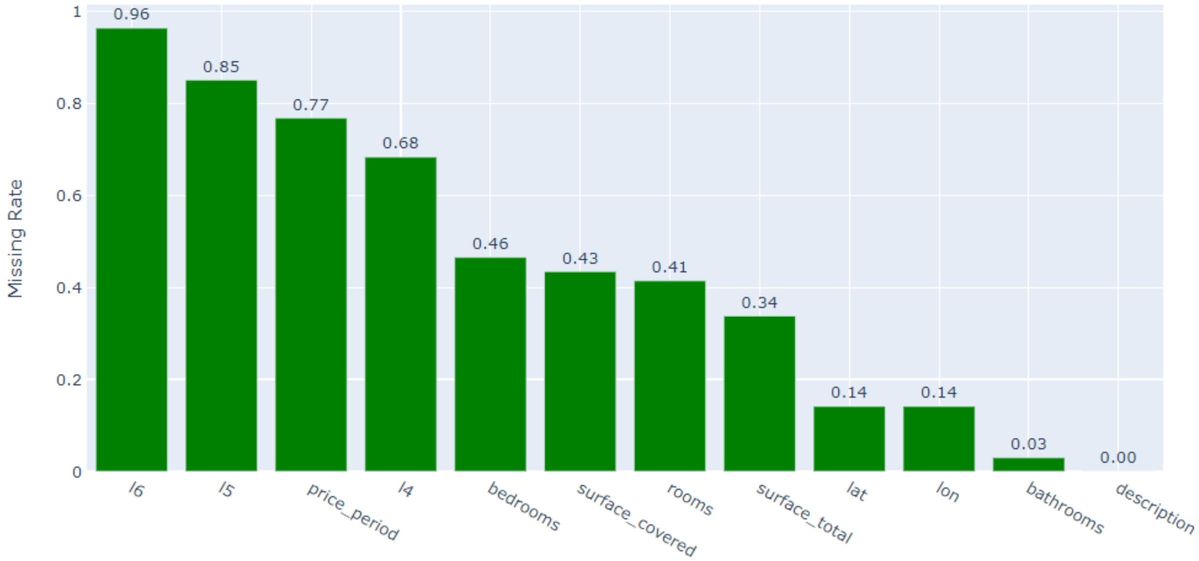
Table 1. Feature Table.

Feature	Data type
id, ad_type, start_date, end_date, created_on, l1, l2, l3, l4, l5, l6, currency, title, description, property_type, operation_type, price_period	0. String
lat, lon, surface_total, surface_covered, price	Float
rooms, bedrooms, bathrooms	0. Integer

The dataset contains the 25 features specified above. “id” is the identifier of the user who posted the ad. “ad_type” is a redundant feature in the case of study, it indicates what is the ad announcing, and in the utilized dataset property is its unique value. “start_date” and “end_date” correspond to the dates in which the ad was posted and when it was retired from the platform. “created_on” is a redundant feature since it contains the same information provided by “start_date”. “l1”, “l2”, “l3”, “l4”, “l5” and “l6” correspond to the specific location of the property. “l1” indicates the country, “l2” the state, “l3” the city, “l4” the municipality, “l5” the neighborhood and “l6” the sub-neighborhood. “currency” is the type of currency in which the property is priced. “title” and “description” correspond to the ones posted in the original ad. “property_type” refers to the type of property. This study only considered properties cataloged as apartments or houses. “operation_type” indicates if the ad corresponds to sale or rent. “price_period” refers to the frequency in which the payment must be done. “lat” and “lon” are the geographical coordinates of the property. “surface_total” is the total surface inside the property, and “surface_covered” relates to the total surface that the property occupies. In the case of apartments, “surface_total” and “surface_covered” are the same. However, in houses, both indicators differ as “surface_covered” refers to the surface of the lot in which the house is built upon. “price” indicates the amount of money that the property costs. Finally, “rooms”, “bedrooms” and “bathrooms” indicates the amount of each of these features that the property possess.

Since it is not required to fill every field of the form when making an ad, the dataset contains a fair amount of missing values. Figure 1 shows the different missing rates found in several features. Features corresponding to the location (such as “l4”, “l5” and “l6”) and “price_period” possess large amounts of missing data. “surface_covered” holds a higher missing rate than “surface_total” since this feature is often not included in apartment cases. Additionally, 14% of the data lacks its coordinates. Finally, “description” is included in the graph to demonstrate that it does present a few missing values, however, it only has 15 missing values.

Figure 1. Missing Rate Bar Graph.



Irrelevant variables were dismissed in order to feed the models with the most relevant information. Location-related variables different to “l3” were not considered due to their high missing rates, as for “l4”, “l5”, and “l6”, and “l2” due to its generality. “title” and “description” were also not considered due to the necessity of applying Natural Language Processing in order to transform the information contained into relevant structured data. “rooms” was not considered either since it is considered to be an irrelevant variable, users rather focus in categories such as “bedrooms”. “surface_covered” was eliminated because of its high missing rate and redundancy regarding apartment cataloged data. Categoric features such as “id”, “ad_type”, “start_date”, “end_date”, “created_on” were also removed. “currency” and “operation_type” were dismissed after filtering the data within the scope of the study. Finally, “lat”, “lon”, “l2”, bedrooms”, “bathrooms”, “surface_total”, “property_type”, and “price” are the variables that are were worked with.

Evaluation

Finding a suitable metric to measure the accuracy of the model is key. Even though the Mean Square Error (MAE) is the most common metric used over regression models, the Root Mean Squared Logarithmic Error was used, denoted by:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i}^{N}\left[\left(\log {}^{y_i}/_{\widehat{y_i}}\right)\right]^2} \qquad (1)$$

Equation 1. RMSLE Equation.

In equation (1) y_i represents the true value, $\widehat{y_i}$ the predicted one, and N the number of samples. The RMSLE metric only cares about the percentage difference between the real value and the predicted one. It presents an asymmetry into de error curve, since it punishes more underestimates. In this case, the logarithmic transformation is further justified given the fact that it makes it so that the relations between the price and some independent variables appear closer to linear (6). If the RMSE is considered, then the presence of an outlier will severely increase the error to a high value. However, when RMSLE is used, the logarithmic scale nullifies outliers' effect on the metric. In addition, it is common to find the use of such a metric in different studies found in the literature (2)(6)(8), which opens the possibility of comparing different models' performances.

Algorithms

Gradient-based sampling is an adaptive sampling technique that considers both input and output data for practical least-square problem-solving. Observations are selected by considering their gradient estimations (2). The adaptive behavior can be described as follows: given an initial estimate for the least-square problem, first, computes the gradient of each data point to assess its importance in the dataset. Then, performs the sampling process considering that the observations with high gradient values are more likely to be selected, and finally minimizes the loss function with the previously sampled data points (8). Given an initial "*good-enough*" guess β_0, the gradient of the loss function for the i-th observation (l_i) is

$$g_i = \frac{\partial l_i(\beta_0)}{\partial \beta_0} = x_i(y_i - x_i^T \beta_0) \qquad (2)$$

Equation 2. Loss Function.

The sampling approach estimates the sampling probabilities as follows:

$$\pi_i = \frac{\| g_i \|}{\sum_{i=1}^{n} \| g_i \|} \qquad (3)$$

Equation 3. Sampling Probability Function.

Therefore, each observation holds a sampling probability that is proportional to its gradient value. It is trivial that all sampling probabilities should sum up to 1. Once all sampling probabilities are computed, an array of independent Bernoulli random variates with success probabilities $p_i = r\pi_i$is generated, where r is the expected subsample size (11). Each element in the Bernoulli array corresponds to the individual observation of the original dataset of the same index. An observation i is sampled if the element s_i of the array is equal to one.

The aforementioned process returns a set of r sampled observations. These data points are now used to solve the linear regression problem and therefore estimate a vector of $\hat{\beta}$ coefficients. One of the advantages of this method is its computational complexity which is $O(nd)$ where n is the number of observations, and d corresponds to the dimension of the dataset (11).

The *Dropouts meet Multiple Additive Regression Trees* (DART), denominated as new GBDT or DART, is a GBDT improved method construed by using the *Multiple Additive Regression Trees* algorithm (MART), that can be estimated as a gradient descent algorithm (12). In this method, each iteration is computed with the derivative of the loss function for the current predictions and adds a regression tree that fits the inverse of these derivatives to the ensemble. The choice of loss makes the algorithm applicable to a variety of learning tasks.

DART diverges from MART in two scenarios. Firstly, when computing the gradient that the next tree will fit, only a random subset of the existing ensemble is considered. Secondly, DART diverges from MART when adding the new tree to the ensemble since DART performs a normalization step. DART also reduces the problem of over-specialization. Therefore, it can be viewed as a regularization where the number of trees dropped controls the amount of regularization. On one extreme, if no tree is dropped then DART would not differ from MART, whereas all trees being dropped would imply no difference in relation to a Random Forest. Conclusively, the size of the dropped-set allows DART to vary between an “aggressive” MART mode to a “conservative” Random-Forest mode (12).

The data input of the algorithm includes a set of points and their labels (x, y), where points are in some space X and label y is in a label space. By using the Loss Generating Function and the labels, the algorithm can define the loss for every point X. L_x: $Y \rightarrow \mathbb{R}$, where Y is the prediction space, and is typically interpreted in real numbers. The Loss Generating Function does change depending on the problem which is aimed to be solved.

At every iteration, let the current model be denoted by $M : X \rightarrow Y$ and $M(x)$ MART creates an intermediate dataset in which a new label, $-L'_x(M(x))$, is associated with every point x in the training data. A tree is trained to predict this inverse derivative and added to the ensemble as a step in the inverse direction of the derivative (in order to minimize the loss). The choice of the loss makes the MART algorithm applicable to a variety of learning tasks as discussed earlier (12).

Exclusive Feature Bundling (EFB) is a feature-reduction technique that assumes that in many cases, in real Machine Learning applications, the feature space is considerably sparse (7). This means that there are possibly many nearly mutually exclusive features. Exclusive features are those that infrequently take non-zero values at the same time.

Exclusive features can be bundled into a smaller set of features that complies with the histogram-based algorithm for GBDT (*Gradient Boosting Decision Tree*). The benefit of this approach is the training time reduction for GBDT without significantly affecting its accuracy.

There are two main concerns when working with EFB. First, identifying mutually exclusive features, and second, determining the bundling approach for the selected features. For the first problem, a graph coloring approach is used, taking features as vertices and adding edges between not mutually exclusive features. A greedy algorithm graph coloring procedure is implemented to select the subset of features to reduce. For the second problem, a feature merging algorithm is applied to reduce the feature space. This merging strategy is based on summing offsets to the original features so that the resulting feature incorporates all the containing features' values [7].

LightGBM

This is a framework for gradient boosted machines [7]. By default, LightGBM usually trains GBDT, but also supports Random Forest *Dropouts meet Multiple Additive Regression Trees* (DART) and *Gradient-Based One-Side Sampling* (GOSS). The new GBDT algorithm is called with GOSS and EFB LightGBM. For GOSS, some data instances are excluded with small gradient values and only use the rest to evaluate the information gain. The reason for the feasibility of this is that having a data set with a large gradient value plays a more important role in the calculation of information gain. GOSS can obtain more accurate information and gain evaluation results on small datasets. EFB bundles mutually exclusive features to reduce the number of features. However, it is found that the optimal mutually exclusive feature is an NP-hard problem, and the greedy algorithm can obtain an approximate score, thus effectively reducing the feature value without compromising the accuracy of the direction of the split point [7].

Bagging

This method is based on Bootstrap sampling that is used to generate multiple versions of a predictor and use them in order to get an improved predictor. Usually, the algorithm works for unstable procedures (trees, neural nets). The evidence, both experimental and theoretical, is that bagging can push a good but unstable procedure a significant step towards optimality. On the other hand, it can slightly degrade the performance of stable procedures.

To gain an understanding of why to employ the bagging method, consider the problem of predicting the value of a numerical response variable Y_x given a set of input x [8]. Suppose that $\phi(x)$ is the prediction that results from using a particular method. Letting μ_{ϕ} denote $E\big(\phi(x)\big)$ where the expectation is for the distribution underlying the learning sample and not x, it has that:

$$E([Y_x - \phi(x)]^2) \text{ (4)}$$

$$= E(\big[(Y_x - \mu_{\phi}) + (\mu_{\phi} - \phi(x))\big]^2$$

$$= E[(Y_x - \mu_\phi)^2] + 2E(Y_x - \mu_\phi)E(\mu_\phi - \phi(x)) + E[(\mu_\phi - \phi(x))^2]$$

$$= E[(Y_x - \mu_\phi)^2] + E[(\mu_\phi - \phi(x))^2]$$

$$= E[(Y_x - \mu_\phi)^2] + Var(\phi(x))$$

$$\geq E[(Y_x - \mu_\phi)^2]$$

Since in nontrivial situations, the variance of the $\phi(x)$ predictor is positive so that the inequality above is strict, this result gives if $\mu_\phi = E(\phi(x))$ could be used as a predictor [8]. It would have a smaller Mean Squared Error in the prediction than does $\phi(x)$, however, the information needed to obtain the value of $E(\phi(x))$ is not known. In principle, it is possible to obtain this value, but in practice, it is typically too difficult to obtain sensibly, and so the bagged prediction of Y_x is taken to be:

$$\frac{1}{B}\sum_{b=1}^{B} \phi_b^*(x)$$

Where $\phi_b^*(x)$ is the prediction obtained when the base regression method is applied to the b-th bootstrap sample drawn with replacement from the original learning sampling. For the bagging method to obtain a prediction of Y_x in a regression setting, one chooses a regression method (which is referred to as the base method) and applies the method to B bootstrap samples drawn from the learning sample. The B predicted values obtained are then averaged to produce the final prediction. The bagging method works best when the base regression is not very stable, that is, small in the learning sample that can often result appreciably different in the average predictions error [8].

Extra Trees

This is a method similar to the Random Forests algorithm in the sense that it is based on selecting, at each node, a random subset of K features to decide on the split [9]. Extra-trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each from the *max_features* randomly selected features and the best split among those is chosen. When *max_features* is set 1, this amounts to building a totally random decision tree.

Consequently, when K is fixed to one, the resulting tree structure is selected independently from the output labels of the training set. In practice, the algorithm only depends on a single main parameter, K. Good default values of K have been found empirically in general $K = p$ for regression problems, where p is the number of input features.

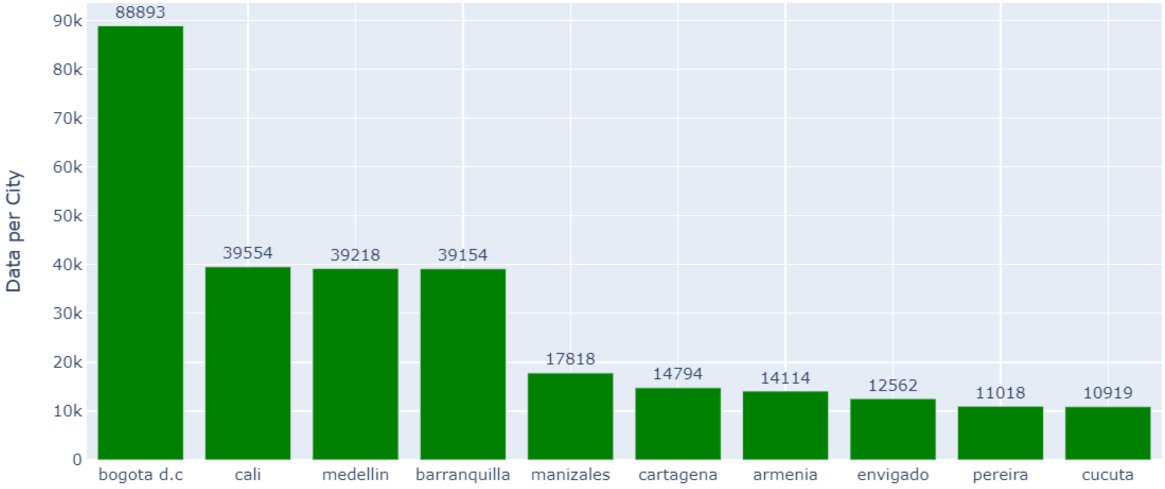
Data Preprocessing

The preprocessing of the data focused on the following: eliminating data that was not within the scope of the study (as well as outliers), performing undersampling, rescaling certain features, enriching the dataset with imputation techniques, and restructuring the dataset.

Firstly, the dataset was withdrawn from the data that does not correspond to the scope of the study. The cleaning was performed according to the following criteria: the advertisement must correspond to a selling action of a house or apartment located in Colombia, and consequently, be priced in COP. This means that any item that did not meet the requirements in features like "operation_type", "property_type", "l2" and "currency" was dismissed from the dataset. Afterward, outliers and numerical values equal to cero were eliminated. In "price", data located above the 97th and below the 0.5th percentiles were dismissed. In "surface_total", data above the 98th and below the 0.5th percentiles, and in "bathrooms" and "bedrooms", data above the 98th percentile were removed. Performing outlier elimination through non-parametric approaches such as elimination according to Mahalanobis distances was attempted. However, the large presence of missing values in the dataset restrained the range of this approach, which makes it ineffective when it is performed uniquely, and unnecessary when it is implemented along with classical marginal elimination.

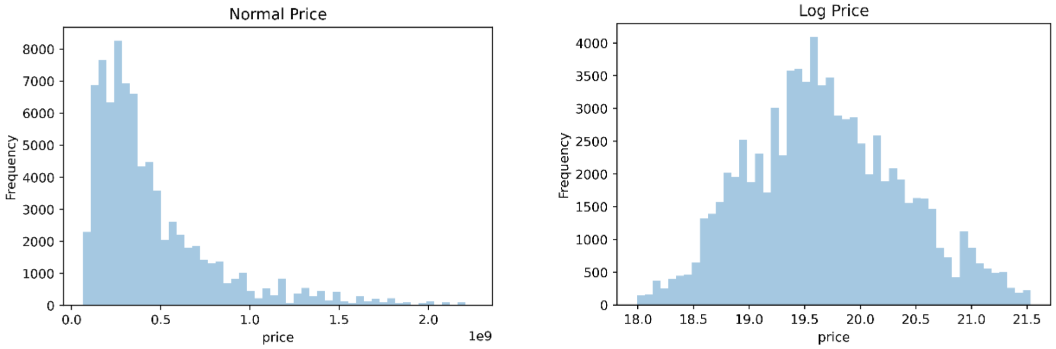
Secondly, undersampling was performed. Since the dataset is heavily imbalanced, it was decided to work with equal samples of the n most frequent cities contained in the dataset. This affected dramatically the number of workable items. Figure 2 displays the data amount of the ten most frequent cities of the dataset. The model was tested with three different datasets: the first containing data of the ten most frequent cities, the second containing the seven most frequent, and third containing the four most frequent cities. All datasets are employed in order to show the model's capacity of working with several cities of the country and to evidence how accurate it could be if it is fed with the largest amount of data possible.

Figure 2. Data per city



Thirdly, numerical variables were rescaled according to two different criteria. The objective of transforming numerical features is to end up with Gaussian-like distributions, which highly impacts the performance of the model. The Yeo-Johnson transformation was applied to “bathrooms”, “bedrooms” and “surface_total”. The natural logarithm was applied to “price”. Despite the fact that other transformation and rescaling techniques were tested, these generated the best overall results. Transforming “price” was an essential part of the study. The range and the amplitude of the distribution were severely impacted, which improved the accuracy of the model. Further justifications of the transformation are mentioned in the evaluation.

Figure 3. “price” distribution before and after of applying logarithm



Fourth the dataset was enriched by using imputation techniques. Missing values in “lat”, “lon”, “bedrooms” and “bathrooms” categories were filled according to the mean criteria. Despite the fact that imputation is commonly performed according to the median criteria due to its robustness, the mean criteria was implemented since it generated slightly better results than the median. The model was tested with and without imputation of values located on “surface_total” in order to compare the obtained results. Further analysis is included in the following section. Finally, categorical features were encoded into binary vectors, which indicate where the property is located and the type of property.

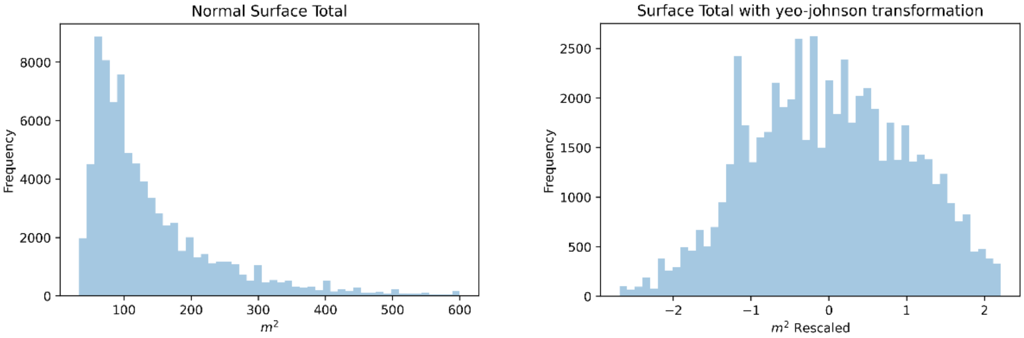
EXPERIMENTATION AND RESULTS

Several experiments were performed in the study in order to provide a complete analysis of the different preprocessing tools utilized, the algorithms implemented to realize the prediction, and their performance on several datasets with different characteristics.

Firstly, three datasets were chosen in order to evidence the models’ performances under different conditions. The criteria used to select the datasets were their length, the number of different cities that they held, and the features that would not be withdrawn from their missing values. As a result of the first and the second criteria, each dataset contained data corresponding to the four, seven, and ten most frequent cities, respectively. The dataset containing four cities is the largest one and the one containing seven cities is the smallest one. The models and the preprocessing tools were tested over the dataset containing four cities with no imputed data since the models’ accuracies initially showed the best results when they worked with it, and consequently, the differences between the results will be more pronounced. After testing several regression algorithms, the three algorithms that presented higher scores were the Bagging Regressor, the Bagging Regressor with ExtraTree Regressor, and the LightGBM, the last one presenting the best accuracy of the three.

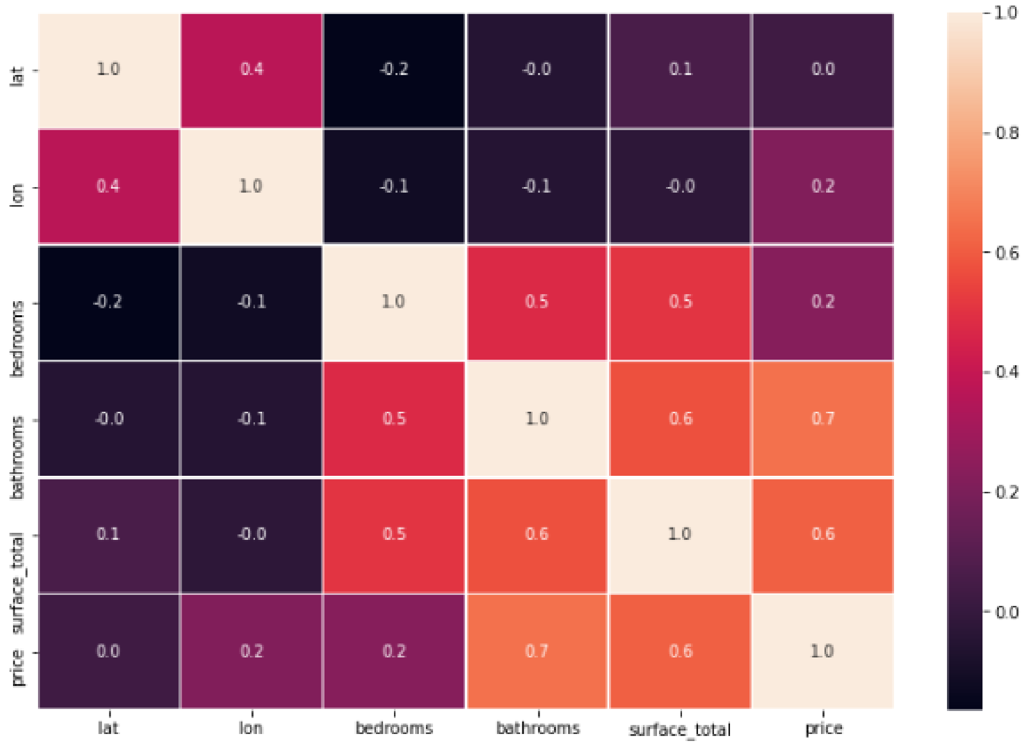
Several transformations and rescaling techniques were tested in order to determine which method is the most suitable for the models. Different standardization methods and non-linear transformations were evaluated. Finally, it was found that applying the Yeo-Johnson transformation method to numerical features such as “bedrooms”, “bathrooms” and “surface_total” generates the best results. This is attributed to its ability to transform heavily skewed distributions into Gaussian-like, mesokurtic, 0-centered distributions. This can be evidenced in Figure 4 that presents how "surface_total” distribution is transformed. Variables such as “lat” and “lon” are not transformed since its normalization will affect the models’ ability of clustering data corresponding to different cities. Further explanations are provided in the following paragraphs.

Figure 4. “surface_total” before and after applying Yeo-Johnson transform.



Moreover, enriching different numerical features was put into discussion. Whilst evaluating the models’ performances, it was observed that imputing values in certain numerical features severely affected the scores of the model. This means that the technique implemented in the study does not represent sufficiently accurate certain variables. Figure 5 presents that certain variables such as “surface_total” and “bathrooms” have a high linear correlation with “price”, which means that its imputation must be done carefully. Even though “lat” and “lon” do not possess high values, it was proved empirically that they were important variables.

Figure 5. Correlation Matrix.



In order to find which combination of enriched features improves the models' accuracies the most, the three datasets were tested over four different conditions: not imputing any value, filling every missing value, enriching "bedrooms", "lat" and "lon", and enriching "bathrooms" and "bedrooms". It was decided to always impute the values of "bedrooms" due to its low linear correlation and the high missing rate that it presents on the dataset. Imputing values will mean gaining data while not sacrificing the quality of the information provided. The overall performance of the models under each of the four conditions was measured calculating the average of the performances of the models in each of the three datasets and then calculating the mean of them.

Initially, the models were tested over the first two conditions and, despite the amount of data that they had, the models who were fed with enriched datasets performed markedly worse than the models who were fed with non-enriched datasets. Enriching every feature of the dataset led to an increase of 15.4% of the RMSLE, increasing its value in almost five units, which, considering that the metric works on a logarithmic scale, is an excessively large difference. The mean of the average performances of the models was 0.272188 and 0.321767. Afterwards, the two enriching combinations were evaluated. Former evaluations do not include the imputation of "surface_total" to demonstrate that the imputation of such variable must be done carefully, hence, enriching the feature with a simple method as the mean criteria will impact the performance of the models.

After evaluating the models with both datasets, it was proven that the "surface_total" do decrease the models' accuracies since the overall performances of the models were similar to the ones achieved with no data imputed. Preserving the directly proportional relation between “surface_total” and “price” is essential in order to ensure the competence of the model, thus, imputing constant values will undoubtedly worsen the results. Moreover, the importance of the geographical coordinates was evidenced as well, since the performance of the models was worse than the ones obtained with a non-enriched dataset. This is attributed to the creation of an “imaginary” city “at the eyes” of the model. This means that the models find a relation between the coordinates and the prices of the properties and imputing such features clusters a set of data linked to certain characteristics. The mean of the average performance of the models was 0.274837.

Finally, the dataset with “bedrooms” and “bathrooms” enriched showed the best overall performance, with the mean of the average performances being 0.269944. Despite the high linear correlation between “bathrooms” and “price”, the quality of the information provided is not corrupted by imputing values due to the low range that “bathrooms” have and the small percentage of missing values in the dataset. In addition, it generated the best overall results for the dataset corresponding to four and seven cities, being the second-best regarding the ten cities dataset. This demonstrates that this imputation combination favors the dataset regardless of the number of cities considered, thus, it does not affect the models’ ability to generalize into general Colombian data.

After doing this, it was decided to test the three algorithms under the most suitable conditions in order to evaluate their general performance and determine if there exists a statistical difference between the accuracy of the methods. To do so it was decided to run a 7-fold cross validation for the three datasets (four, seven and ten cities) and calculate the difference of means and difference of variances confidence intervals. The cross-validation results are listed below in Table 2, 3, and 4. It was found that there is no statistical difference between means and comparison of standard deviations of neither of the three algorithms under any of the three datasets.

Table 2. 7-Fold Cross Validation Results of RMSLE for four cities.

Algorithm	Mean	Standard Deviation
LightGBM	0.25354	0.00699
Bagging Regressor	0.25296	0.00511
Bagging Regressor with ExtraTree Regressor	0.25312	0.00559

Table 3. 7-Fold Cross Validation Results of RMSLE for seven cities.

Algorithm	Mean	Standard Deviation
LightGBM	0.27557	0.00478
Bagging Regressor	0.27959	0.00527

Bagging Regressor with ExtraTree Regressor	0.27912	0.00511

Table 4. 7-Fold Cross Validation Results of RMSLE for ten cities.

Algorithm	Mean	Standard Deviation
LightGBM	0.28288	0.00946
Bagging Regressor	0.28374	0.00806
Bagging Regressor with ExtraTree Regressor	0.28382	0.00758

In addition, it the best results achieved during the cross validation of the four cities dataset where collected, and they are listed in Table 5. The dataset has 81,021 items and has data corresponding to Bogotá D.C., Medellín, Cali and Barranquilla.

Table 5. Best RMSLE results of each algorithm.

Algorithm	RMSLE
LightGBM	0.24077
Bagging Regressor	0.24533
Bagging Regressor with ExtraTree Regressor	0.24475

CONCLUSIONS AND FUTURE WORK

The present study evidenced that the correct practice of preprocessing techniques is essential to generate the best possible results. Performing proper standardization of heavily skewed data is key in order to guarantee an optimal functioning of the models. Implementing imputation techniques correctly is crucial as well. Differentiating the most suitable imputation technique for each variable is a determinant factor since enriching the dataset appropriately allows to feed the model with larger amounts of data without compromising the quality of the information contained in it. Executing such tasks improperly can severely worsen the accuracy of the model. Additionally, the LightGBM, Bagging Regressor, and Bagging Regressor with ExtraTree provide a competitive approach to the problem.

Imputing values according to the mean criteria offered an effective approach to preserve the largest amount of data by enriching “bedrooms” and “bathrooms” features without compromising the quality of the information. However, such approach failed to appropriately reflect the nature of features such as “lat”, “lon”, and “surface_total”. In order to impute “surface_total” values accurately, an approach that preserves the relation of such features and “price” must be considered. Furthermore, finding a proper approach represents a much harder task, since the technique needs to find a way for identifying the relationships among the data and successfully clustering them. It is key to find a way to do so, and it would be taken into account in later studies.

In future work, studying and covering more web pages is essential to improve predictive quality, as it would broaden coverage of properties for sale and would introduce a longer period of data collection. Consequently, the fluctuation of house prices over time can be assessed to analyze whether it is atypical or not. An analysis of the economic variables in this sector is also expected. It should comprise the behavior of monetary flows of real estate by using housing prices datasets so that data-related insights can be displayed into the economic situation. Approaching the problem within the perspective of first assigning a range of price values that the property will possess and later estimating the price in order to end with a more accurate prediction will be considered as well. Finally, providing an effective approach to impute values in “surface_total”, “lat”, and “lon” should be introduced as well for the future models to work more accurately.

ACKNOWLEDGEMENTS

We wouldn't be able to go without thanking David Barco and Juan Esteban Rivera for their continued assistance, which proved vital for streamlining and clarifying all of our ideas during the de writing process. This paper would have never come into being were it not for their support. We also need to thank Prof. Mauricio Toro for his guidance, support, and expertise during the research, writing, and submitting processes.

REFERENCES

[1] Peter, Nkolika & Okagbue, Hilary & Emmanuela C.M, Obasi & Akinola, Adedotun. “Review on the Application of Artificial Neural Networks in Real Estate Valuation”. International Journal of Advanced Trends in Computer Science and Engineering. (2020).

[2] T. D. Phan. “Housing price prediction using machine learning algorithms: The case of Melbourne city”. International Conference on Machine Learning and Data Engineering Sydney, Australia (iCMLDE). 2018.

[3] Hong, Jengei & Choi, Heeyoul Henry & Kim, Woo-sung. “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. (2020).

[4] Manjula, R & Jain, Shubham & Srivastava, Sharad & Kher, Pranav. “Real estate value prediction using multivariate regression models”. IOP Conference Series: Materials Science and Engineering. (2017).

[5] J. I. Perez, F. Gonzalez, and J. C. Correa. “Modeling of apartment prices in a Colombian context from a machine learning approach with stable-important attributes”. DYNA. Vol.87 N° 212, pp.63-72. 2020.

[6] B. Klaus , L. Carvalho Melo , W. D. Gomes de Oliveira , S. B. da Silva Sousa and L. Berton “Housing Prices Prediction with a Deep Learning and Random Forest Ensemble”, pp. 1-12. 2019. Accessed on: July 7 of 2020. URL:

https://www.researchgate.net/publication/335527230_Housing_Prices_Prediction_with_a_Deep_Learning_and_Random_Forest_Ensemble

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. NIPS. 2017.

[8] C. D. Sutton. “Classification and Regression Trees, Bagging, and Boosting” Handbook of Statistics, Vol. 24 ,pp. 303-329. 2005.

[9] P. Geurts, D. Ernst., and L. Wehenkel, “Extremely randomized trees”, Mach Learn, Vol 63 N° 1, pp. 3-42. 2006.

[10] S. Li, X. Ye, J. Lee, J. Gong, and C. Qin. “Spatiotemporal Analysis of Housing Prices in China: A Big Data Perspective”. Applied Spatial Analysis and Policy, pp 421-433. 2017.

[11] R. Zhu, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett “Gradient-based sampling: An adaptive importance sampling for least-squares in Advances in Neural Information Processing Systems”. Eds. Curran Associates, Inc., pp. 406–414. 2016.

[12] K.V Rashmi, and R. Gilad-Bachrach. “DART: Dropouts meet Multiple Additive Regression Trees”. ArXiv 2015.