Journal of Economics, Finance and Accounting Studies
ISSN: 2709-0809
DOI: 10.32996/jefas
Journal Homepage: www.al-kindipublisher.com/index.php/jefas

JEFAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Machine Learning Empowered Insights into Rental Market Behavior

Florina Livia Covaci
Business Information Systems Department, Babeș-Bolyai University, Cluj-Napoca, Romania
Corresponding Author: Florina Livia Covaci, **E-mail**: florina.covaci@ubbcluj.ro

| ABSTRACT
The aim of the current study is to determine which models are most suited for forecasting a property's rental price given a variety of provided characteristics and to develop a predictive model using machine learning techniques to estimate the rental prices of apartments in Cluj-Napoca, Romania, in relation to market dynamics. Given the absence of a comprehensive dataset tailored for this specific purpose, a primary focus was placed on data acquisition, cleaning, and transformation processes. By leveraging this dataset, the model aims to provide accurate predictions of fair rental prices within the Cluj-Napoca real estate market. Additionally, the research explores the factors influencing rental prices and evaluates the model's performance against real-world data to assess its practical utility and effectiveness in aiding rental market stakeholders.

| KEYWORDS
Predictive modeling, Rental market, Machine learning.

| ARTICLE INFORMATION
ACCEPTED: 02 April 2024 **PUBLISHED:** 23 April 2024 **DOI:** 10.32996/jefas.2024.6.2.11

1. Introduction

In the contemporary landscape, the decision to rent a property holds financial significance for individuals and families alike. The capacity to make an informed choice regarding housing arrangements relies heavily on our ability to forecast and comprehend rental costs. Rental pricing is a complex process that takes into account a number of elements, such as location, size, amenities, and general condition of the property. This complex equation is also greatly influenced by the number of available properties, the demand for rental accommodation, and the state of the economy.

Using technology to its full potential—in particular, machine learning—offers a fascinating way around this complexity. Machine learning algorithms have the capability to assimilate numerous variables and generate predictions for house rental prices. In this study, we explore the realm of machine learning to anticipate the cost of renting a house within a specific city (Cluj-Napoca, Romania). The rental market in Romania is influenced by several key factors, economic trends and demographic shifts being the most significant. Specific cities or regions in Romania are experiencing demographic changes that impact their rental markets. For example, university cities like Cluj-Napoca have a high demand for rentals due to the large student population.

Employing a range of machine learning techniques, we systematically evaluate their performance and draw comparisons to identify the most effective approach. Our overarching goal is to develop a finely tuned machine learning model capable of furnishing estimates of rental costs in any desired city.

The beneficiaries of this cutting-edge model span across the housing market spectrum. Property owners can utilize it to determine optimal rent prices for their listings, while renters can leverage it to secure affordable accommodations. Real estate professionals can wield it as a powerful tool to provide clients with valuable insights into rental trends. Moreover, our study aims to tackle important questions, such as the key determinants influencing rental prices and the most accurate machine learning method for

prediction. Ultimately, we aim to enhance the precision of our prediction model, empowering individuals to make informed decisions regarding the critical choice of renting or buying a home. Our model, born from this study, serves as a point of guidance in this pivotal decision-making process.

2. Literature Review

In the exploration of the housing rental market, numerous studies have explored the factors that impact rental prices. However, research specifically focused on predicting rental prices within this market remains relatively scarce, with more attention directed towards forecasting property prices in the broader real estate market. In addition to conventional valuation techniques, researchers have continuously sought to innovate by introducing new methods for real estate assessment, including the integration of machine learning approaches, particularly artificial neural networks.

In (Raga et al., 2019), housing prices are forecasted for individuals currently not owning homes, taking into account their financial capabilities and preferences. By analyzing past property trends, fare ranges, and anticipated developments, the paper seeks to make well-informed estimations of future prices. To achieve this, the research employs a variety of regression techniques, including Multiple Linear Regression, Ridge Regression, LASSO Regression, Elastic Net Regression, Gradient Boosting Regression, and Ada Boost Regression. These methods are utilized to predict housing prices using a dataset, with the primary objective of determining the most effective technique among them.

The paper (Varma et al., 2018) proposes an advanced housing price prediction system that emphasizes accuracy. This system utilizes Linear Regression, Forest Regression, and Boosted Regression techniques, further enhancing its efficiency by integrating Neural Networks. The aim is to provide customers with precise predictions, reducing the risk of investing in unsuitable properties. Furthermore, the system is designed to incorporate additional customer-oriented features without compromising its core functionality. A significant future update involves expanding the database to include larger cities, enabling users to explore a wider range of properties, improving prediction accuracy, and ultimately making more informed decisions.

The authors introduce (Shi et al., 2022) a novel approach: treating past sale records as an evolving data stream. The study's findings suggest that the data stream approach outperforms traditional regression methods, underscoring the potential of data stream techniques in enhancing prediction models for residential property prices.

The research of (Thamarai et al. 2020; Madhuri et al., 2019; Kumar et al., 2021) employs machine learning models to forecast house prices, likely exploring various regression and ensemble methods. The study is expected to utilize multiple datasets and assess model performance metrics to develop accurate predictive models for the dynamic housing market.

Machine learning algorithms are employed (Park, 2015) to forecast housing prices in Fairfax County, Virginia, exploring the complexities of this regional housing market, potentially uncovering factors that influence housing prices in this area while (Phan, 2022) the authors investigate house price prediction in Melbourne City, Australia, considering unique factors such as location, demographics, and urban development in predicting housing prices.

The work of (Truong et al., 2020; Vineeth et al., 2018; Wang et al., 2021; Winky et al., 2021) explores advanced machine learning techniques aiming to improve housing price prediction models. It involves tasks such as feature engineering, optimization, or innovative algorithmic approaches to enhance model accuracy.

The paper (Satish et al., 2019) explores the utilization of machine learning for house price prediction, likely addressing the selection of suitable features and algorithms. It may also evaluate model performance using metrics such as RMSE and R-squared.

The research (Zulkifley et al., 2020; Shahirah et al., 2021) presents a comprehensive literature review, summarizing significant findings from existing studies on house price prediction using machine learning.

The research of (Adetunji et al., 2022) focuses on the Random Forest machine learning technique for house price prediction, while the study of (Wang et al., 2019) combines deep learning and ARIMA models to predict housing prices, considering both temporal and structural factors. The work (Soltani et al., 2022) explores the incorporation of spatio-temporal dependencies into machine learning algorithms for housing price prediction, potentially benefiting urban planning applications.

In summary, several key challenges persist in rental market research, including a lack of comprehensive understanding regarding the factors influencing rent and a need for detailed comparative studies on different rent prediction algorithms. Therefore, this paper aims to address these limitations by comparing the effectiveness of various methods, such as Linear Regression, Ridge

Regression, Decision trees, KNN, Random Forest, and SVR, in predicting rent prices. Our aim is to identify the most suitable method and conduct extensive research on various factors influencing rent, providing valuable insights into the real estate market.

3. Methodology

This section provides a detailed exploration of the data collection, research, and results acquisition process in the context of predicting property rental prices in Cluj-Napoca. The methods, techniques, and tools employed in this research process are presented. Readers will uncover how data was collected preprocessed, and relevant attributes were selected. Additionally, explanations regarding choices made regarding the methods and models used and how the validity of the obtained results was ensured will be provided. Specifically, readers will explore the details concerning data collection, preprocessing steps applied to the data, criteria used for selecting relevant attributes, models tested and chosen, and the evaluation of their performances. Thus, this section offers a comprehensive perspective on the approach and methodology employed in the research process.

3.1 Data set description and methods used to collect and process data

It is well recognized that data has become a significant resource in today's world and is foundational in the realm of research and analysis. Within online resources, numerous datasets containing diverse information can be found, with the majority focusing on properties available for sale. However, none of these datasets specifically address the real estate market in Romania or Cluj-Napoca. Hence, we employed web scraping to retrieve only the factors of interest for the current analysis. Web scraping involves the practice of collecting data without a dedicated API. This is commonly achieved by writing source code that queries a web server, requests data (typically in the form of HTML and other files comprising web pages), and then parses that data to extract the necessary information. It encompasses a wide variety of programming techniques and technologies, such as data analysis, natural language processing, and information security (Mitchell, 2018).

Using the web scraping technique, we managed to extract the necessary data; the data source used was the real estate company's website, Storia (www.storia.ro), a complex real estate online platform. The entire code for this work was written using the Python programming language, which was fully executed in Google Colab. Also known as Google Colaboratory, it is a cloud computing platform provided by Google that offers a browser-based integrated development environment (IDE), allowing users to create and run Python code. Among its advantages are the following: it does not require any installation or configuration process, users can write code immediately, access to processing power for free, easy distribution of written files and collaboration with other participants. For collecting data, we chose Selenium, which is a popular framework used for web browser automation. It provides a set of tools and libraries for browser automation, allowing interaction with web pages, performing actions, and programmatically extracting data. After installing the libraries (Selenium, ChromiumDriver, etc.) and making the necessary configurations, we navigated to the main announcements page on www.storia.ro, opened a CSV file, and began to iterate through each individual announcement, selecting the factors that we deemed relevant after consulting other specific datasets. Once the script finished going through all available announcements, our dataset consisted of 1680 rows and 20 attributes, which we will present in the following:

1. URL - was extracted as a unique identifier for the advertisement and also allowed us to open the advertisement in the browser to verify if all the information was correctly extracted. It is a text attribute and was not used for model construction.
2. Title - represents the title of the advertisement on the source website. Although this attribute was not used for model construction, extracting it was helpful, as some information was derived from the title text. It is also a text attribute.
3. Price - represents the rental price of the property, typically expressed in EUR. Those expressed in Romanian currency were converted to EUR during the preprocessing step to standardize all the data. It represents our target variable, based on which this study was initiated. It is a numeric attribute.
4. County - represents the county in which the advertisement was posted. Although our study is clearly focused only on properties in Cluj-Napoca, a city located in Cluj County, we chose to extract this attribute to ensure that the data was correctly extracted, as on the Storia website, we can obtain data from all over Romania. Additionally, this attribute was not used in model construction. It is a text attribute.
5. City - the same considerations are for the County attribute. This variable was not used in model construction either. It is a text attribute.
6. Neighborhood - represents the neighborhood where the posted property for rent is physically located in the advertisement.
7. Street - represents the street where the property is located. The street attribute had a lot more missing data, which was a consideration in not using this attribute in model construction. It is a text attribute.
8. Area - represents the surface area of the rental property in square meters. It is a numeric variable, and it plays an important role in model construction.
9. Number of rooms - represents the number of rooms the apartment has. It is a numeric attribute.

10. Floor - represents the floor level at which the property is located. It is a numeric attribute.
11. Partitioning - represents the division of space into distinct and functional rooms and compartments. It is a text attribute.
12. Number of bathrooms - refers to the total number of rooms intended for personal hygiene and various bathroom-related activities. It is a numeric attribute.
13. Orientation - represents the direction in which a property is oriented in relation to the cardinal points (north, south, east, west). This can influence many aspects of a home, including sun exposure, ventilation, natural lighting, and energy efficiency. This variable was not used for model construction and is a text variable.
14. Central heating - a system used for heating a property. In our case, we refer to whether there is a central heating system. It is a text attribute that takes values yes/no.
15. Air conditioning - a system used for cooling and controlling the temperature, humidity, and air quality in a home. It is a text attribute that takes values yes/no.
16. Parking - the space intended for parking associated with the property in question. We could not determine the form of this parking (underground, covered, garage, etc.), so we only looked to see if the property comes with this facility. It is a text attribute that takes values yes/no.
17. Elevator - It is a text attribute that takes values yes/no.
18. Balcony - It is a text attribute that takes values yes/no.
19. Status - refers to the condition of the apartment at the time of rental (renovated, new, used, etc.). It is a text attribute.
20. Year of construction - refers to the year in which the building of the property was completed and put into operation. This information is important as it can influence both the physical condition of the apartment and any renovations or modernizations that may be required. It is a numerical attribute.

We applied a series of transformations and operations during the data preprocessing stage to ensure a clean dataset. A preliminary step in obtaining a cleaner dataset involved removing duplicates, a task performed within Google Sheets, as we opted to keep all the data in the cloud. Therefore, Google Sheets' available functions and Python scripts were utilized for data cleaning and preprocessing. Subsequently, to handle missing or incomplete values and manage errors, we attempted to fill in missing values where possible and rectify errors. If these actions were not feasible, we chose to remove the data. Additionally, extreme values or real estate listings that were not relevant to our study were also eliminated.

3.2 Exploratory data analysis

Once the data preprocessing stage was completed, a clean dataset was obtained, structured according to our study's requirements, and irrelevant variables were removed, and we proceeded to generate various graphs and statistics. These aimed to understand the data and the relationships established between them.

To observe the distribution of apartment rental prices, I used the matplotlib. pyplot library and created a histogram that can be seen in Figure 1.

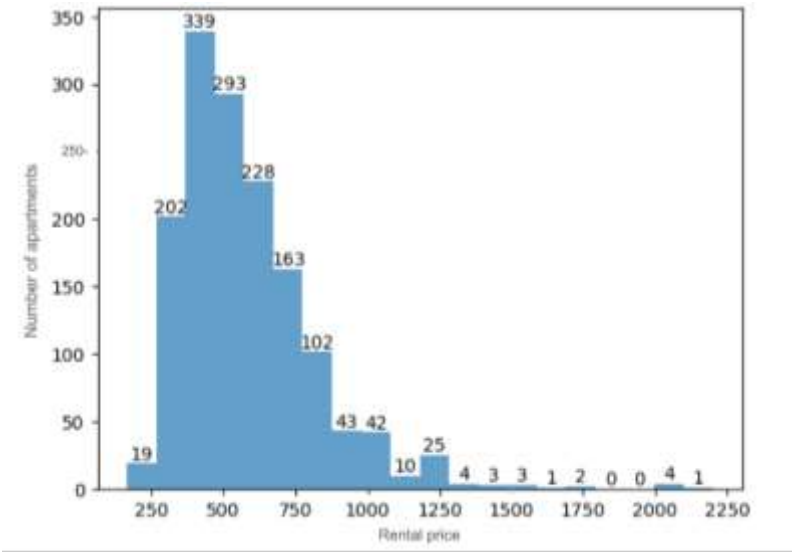

Figure 1 Distribution of properties rental prices

As can be seen from the graph, most (over 600) apartments are located in a price range of 400-600 EUR. Looking at the graph, we can say that our data follows an asymmetric distribution to the right, with some extreme values on the right side, representing the higher priced apartments, most likely those in the central neighborhoods.

Another important aspect to watch was the relationship between the rental price and the size of the apartments. As can be seen in Figure 2, most apartments are between 30 and 100 square meters in size, with prices between 250 and 800 EUR. Of course, we also have exceptions that do not fall within these ranges, and these apartments are those in the central areas.

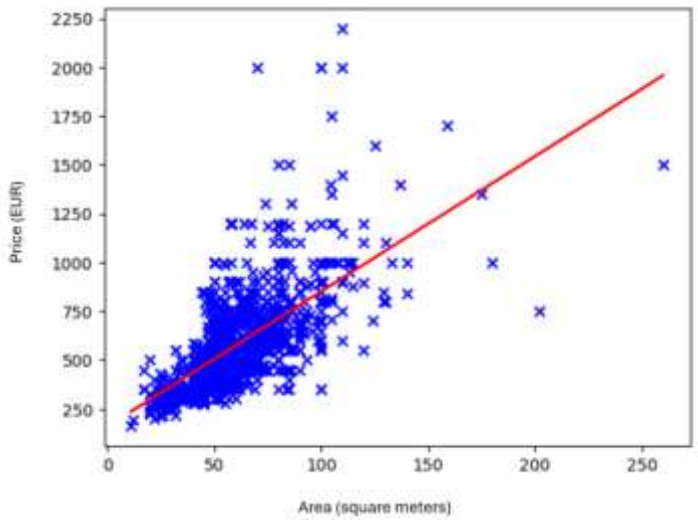

Figure 2 Relation between price and area

To find out the most representative variables that influence the rental price, we created the correlation matrix in Figure 3. Following the correlation matrix, we can observe that the most important factors influencing the price are size, number of rooms and number of bathrooms.

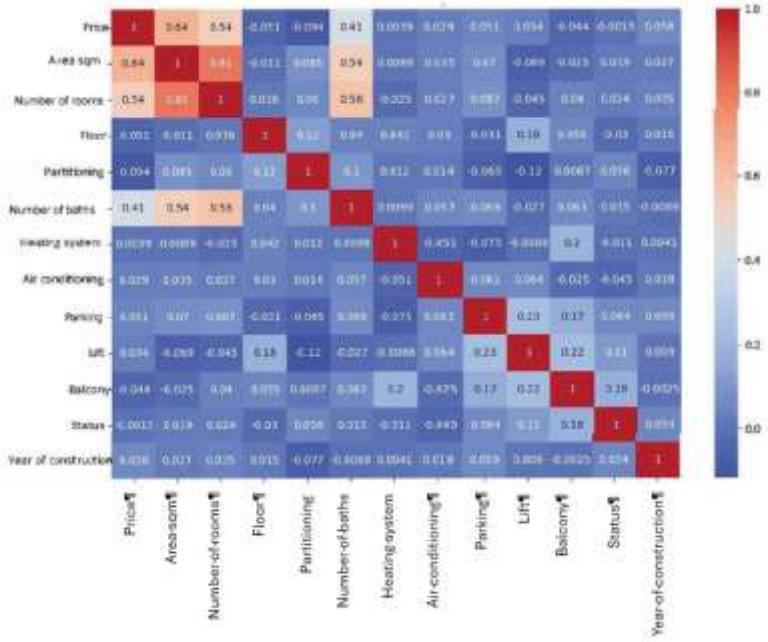

Figure 3 Correlation Matrix

Observing the variables, we can make the following interpretations:

● Area (m^2) - property price has a moderate positive correlation with its area. In other words, properties with a larger area tend to have a higher price.

● Number of rooms - and in this case, there was a significant positive correlation between the number of rooms in the properties and their price. Properties with more rooms tend to have a higher price.

● Number of bathrooms - the correlation, in this case, is also positive but weaker than that between area and price or number of rooms and price. However, properties that have more bathrooms may be priced higher compared to properties that have fewer bathrooms.

●Year of construction - the existence of a very weak positive correlation between the year of construction and price indicates that newer properties may tend to be priced higher, but the influence of this factor on the price is considerably reduced.

● Parking, Elevator - the correlations between the existence of parking and elevator facilities and the price properties are very weak. From these values, we can consider that their influence on the price is relatively negligible.

3.3 Prediction models

To build the prediction models, the first step was to separate the dependent variable Price from the rest of the independent variables. Afterwards, we divided the data set into training data and test data, respecting the proportion of 80% training and 20% test. Thus, the split data set has 1187 values and 32 columns for training and 297 values with 32 columns for test.

In the implementation process, we defined a function (Predictive_Model) that is called for each implemented prediction model. The function also computes R-squared and Square Root of Mean Squared Error. The function will also display a histogram of both values' current price as well as predicted price values.

R-squared measures how well the model performs relative to a simple average of the target values. R squared = 1 indicates a perfect fit, R squared = 0 indicates that the model does no better than simply averaging the data.

The root mean square error (RMSE) measures the average difference between the values predicted by a model and the actual values. It provides an estimate of how well the model is able to predict target value (accuracy). The smaller the Root Mean Squared Error value, the better the model is. A perfect model (a hypothetical model that would always predict the exact expected value) would have a Root Mean Squared Error value of 0.

3.3.1 Linear Regression

Linear regression is a learning algorithm that represents a linear combination of features of the input sample. These features of input are real values, from minus to plus infinity. We will use this model to achieve predictions of the dependent variable according to the optimal independent variables provided so that we can make the most accurate predictions. In the case of our linear regression implementation, the model did not achieve very high performance, having an R-squared value of 0.55, meaning that only 55% of cases can be correctly predicted and an RMSE of 141 (EUR).

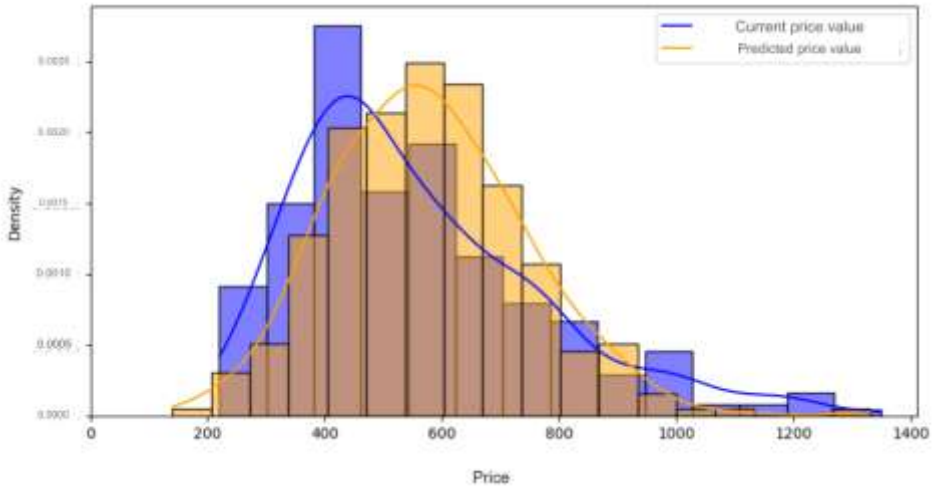

Figure 4 Predictions made using linear regression

Figure 4 depicts a histogram where both the actual values (those in blue) and the predicted values (those in orange) are displayed. We can see that where the two graphs overlap, the actual value of the price and the predicted one match, and in the other case, with significant differences, the two values do not match.

3.3.2 Ridge Regression

The linear regression finds coefficients that best fit the data, but it doesn't find the unbiased coefficients; it does not take into account which independent variable is more important than others.

The Ridge Regression adjusts the lambda parameter so that the model coefficients change.

For the current dataset, Ridge regression performed slightly better than linear regression, obtaining an R-squared of 0.57, meaning that only 57% of cases can be correctly predicted and an RMSE of 138 (EUR). We got the best score with a lambda parameter of 50.

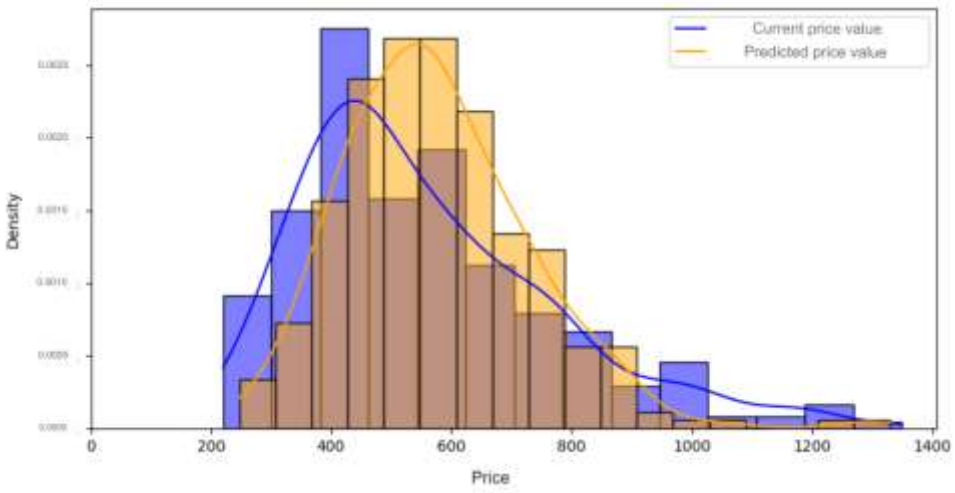

Figure 5 Predictions made using Ridge regression

Figure 5 depicts a histogram where an improvement over linear regression can be observed, but predicted values in a good proportion of cases are still far from the current values.

3.3.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric learning algorithm. In contrast to the other learning algorithms that allow the training data to be removed after the model is built, KNN keeps all training examples in memory. Once a new example, x is previously unseen, the KNN algorithm finds the k most training examples close to x and returns the mean of the label values in the case of regression. The distance between two examples is given by a distance function.

In the case of our implementation, we selected the value of K equal to 5. In order to select the right value for our data, we ran the KNN algorithm several times with different values of K, and we chose the value 5 because this value reduced the number of errors the most, maintaining the algorithm's ability to make predictions.

The advantages of this algorithm would be the following:

- The algorithm is simple and easy to implement
- There is no need to build a model, tune more parameters or make additional assumptions
- The algorithm can be used for both regressions and classifications

Conversely, a major drawback of this algorithm is that it becomes significantly slower as the number of examples and/or predictors increases. However, in our situation, this algorithm obtained an R-squared value of 0.56, meaning that only 56% of cases can be predicted accurately and an RMSE of 140 (EUR).

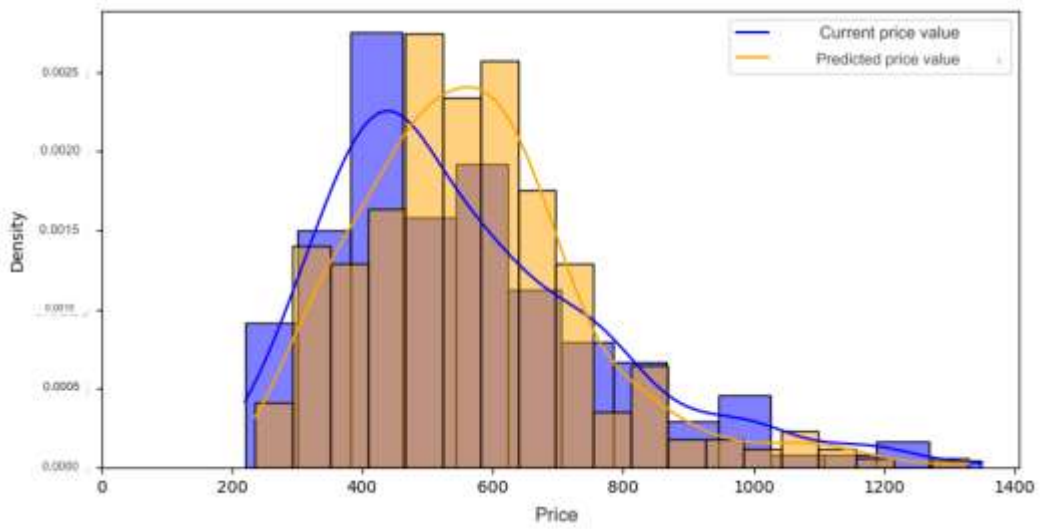

Figure 6 Predictions made using KNN

In Figure 6, a very small improvement can be observed over linear regression but a poorer result than Ridge regression.

3.3.4 Decision tree

A decision tree is an acyclic graph that can be used to make decisions. In every branch node of the graph, a specific feature *j* of the characteristics vector is examined. If the feature value is below a specific threshold, the left branch is followed; otherwise, the right branch is followed. Once the leaf node is reached, the decision is made about the class the example belongs to. The decision tree is used both in classification and regression problems.

The process of building a decision tree involves selecting the characteristics which provide the best split of the data and the construction of the corresponding subtrees. This is done based on metrics such as the Gini index or entropy, which measures the purity of the results of each division in order to minimize uncertainty.

The decision tree can be used to make predictions on new data sets. For regression, the average value or the predominant value of the samples is calculated from a given node to predict the value of a new instance. The decision tree has the advantage that it is easy to understand and interpret, and its decision-making process can be visualized in an intuitive way. However, there is a risk of creating overly complex trees that fit the training data too well and perform poorly on the new data (overfitting). To solve this problem, you can adjust the parameters or use the pruning technique.

Pruning consists of going back through the tree once it has been created and removing branches that do not contribute significantly to reducing errors by replacing them with leaf nodes (Burkov, 2019). In our situation, this algorithm obtained an R-squared value of 0.53, meaning that only 53% of cases can be predicted correctly, and an RMSE of 145 (EUR) represents the least performing model so far. After several runs of the algorithm, the best value for the max_depth parameter (which represents the maximum number of nodes that a decision tree can have) is 4.

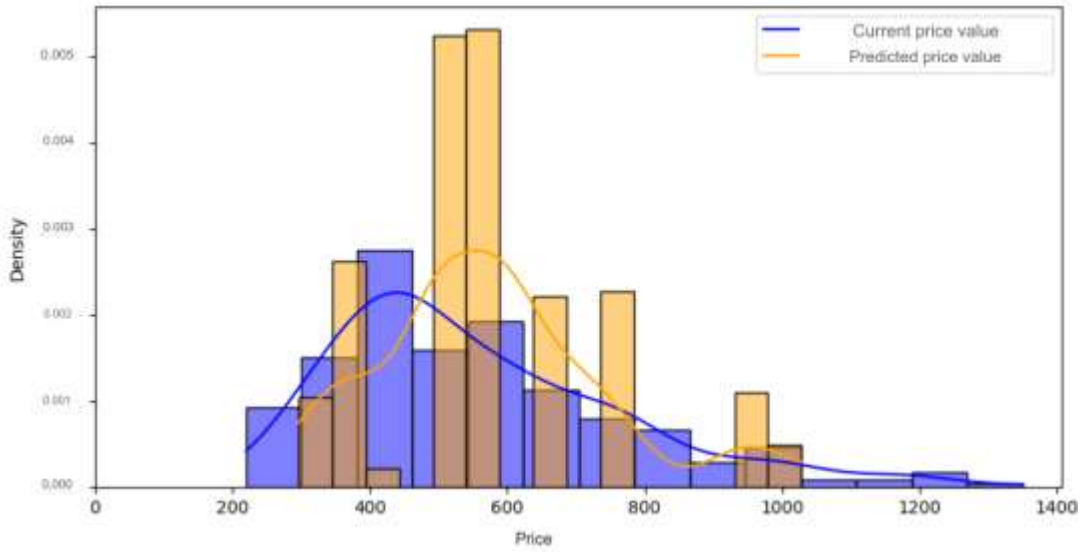

Figure 7 Predictions made using Decision Tree

3.3.5 Random Forest

Random Forest is part of the ensemble learning algorithms, a learning paradigm that, instead of trying to learn a super accurate model, focuses on training a large number of models with low accuracy and then combines the predictions given by those weak models to obtain a high-precision meta-model.

Low-accuracy models are typically produced by learning algorithms that are quick during training and prediction because they are unable to recognize complicated patterns. With decision trees, the training set splitting procedure is frequently terminated after a small number of rounds. The theory behind ensemble learning is that if the trees are not identical and each tree is at least marginally better than random guessing, we can combine a large number of such trees to attain high accuracy, even though the resulting trees are shallow and not particularly accurate.

To obtain the prediction for an input x, the predictions of each weak model are combined using a form of weighted voting. The specific form of the weighting of votes depends on the algorithm used.

Bagging consists of creating multiple "children" of the training set (each copy being slightly different from the others) and applying the weak learner to each copy to obtain several weak models, which we combine later. A widely used machine learning algorithm based on the idea of bagging is a random forest. The random forest algorithm uses a modified tree learning method that inspects, at each split in the learning process, a random subset of the features. The reason for this is to avoid correlation between trees: if some features are strong predictors of the variable target, these features will be selected to split the examples into many trees. This would lead to the appearance of correlated trees in our "forest".

Correlated predictors cannot help improve prediction accuracy. The main reason behind the better performance of blended learning is that good models will be likely to agree on the same predictions, while weak models will be likely to agree on some different predictions. Correlation will make weak models more likely to agree, which will make majority voting or averaging difficult.

The most important hyperparameters to adjust are the number of trees and the size of the random subset of features to consider at each split. Random forest is one of the most widely used ensemble learning algorithms. The reason is that by using more samples of the original data set, we reduce the variation of the final model. Reduced variance means reduced overfitting. Overfitting occurs when our model tries to explain small variations in the data set because our data set is only a small sample of the population of all possible examples of the phenomenon we are trying to model. If we are unlucky about how our training set was sampled, it may contain some unwanted artifacts (but unavoidable): extreme values and over- or under-represented examples. By creating more random samples with the replacement of our training set, we reduce the effect of these artifacts (Burkov, 2019).

In our case, given that we are dealing with predictions, we used the RandomForestRegressor algorithm, where we used the parameters: criterion, n_estimators and random_state.
For the parameter "criterion='absolute_error'", we specify the criterion used to measure the quality of a split during tree construction. In this case, we used absolute error as a metric for evaluating divisions. The number of estimators (n_estimators)

specifies the number of trees to be built within the model. In our case, we will be building 40 trees. The parameter random_state=20 specifies a value for the generation of pseudo-random numbers, ensuring the reproducibility of the results because the use of the same values for random_state will generate the same results in each run.

This algorithm achieved an R-squared value of 0.69, meaning that 69% of cases can be correctly predicted, and an RMSE of 117 (EUR) represents the best performing model so far. After several runs of the algorithm, the best value for the n_estimators parameter is 20.

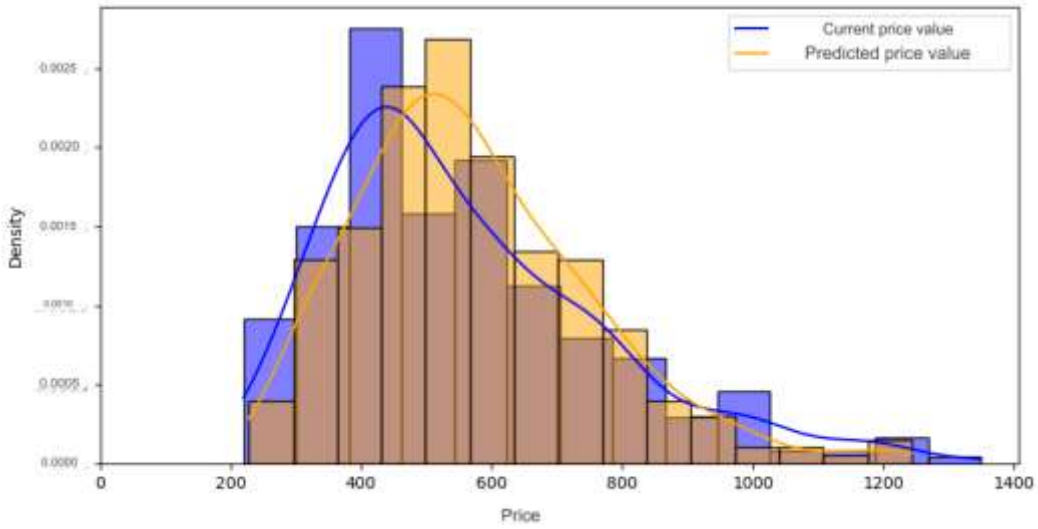

Figure 8 Predictions made using Random Forest

3.3.6 Support Vector Regression

Support Vector Regression (SVR) is a regression technique based on Support Vector Machine (SVM). SVM is mainly used for classification problems, while SVR is adapted for solving regression problems. In simple regression, we try to minimize the error rate. Meanwhile, in SVR, we try to frame the error using a specific threshold.

To use SVR, it is necessary to select the right kernel (the function used to map data from a lower dimension to a higher dimension) and to adjust model parameters such as C and epsilon. SVR is based on the idea of finding a hyperplane in a higher dimensional space that maximizes the edges between the points of data and the hyperplane. Thus, an attempt is made to find a regression function that fits within these limits. C, or the penalty coefficient, controls the trade-off between the correct adjustment of training data and preventing overfitting. The value of C must be a positive number, and the higher it is, the higher the error penalty will be. Its epsilon error tolerance specifies the maximum size of the error that the SVR algorithm can tolerate in the edge area.

In our case, this algorithm obtained an R-squared value of 0.56, meaning that only 56% of cases can be predicted correctly, and an RMSE of 140 (EUR), which is similar to the performance of other models described so far. After several runs of the algorithm, the values chosen for the parameters were kernel = linear, a coefficient of penalty with the value of 10 and epsilon equal to 0.1.

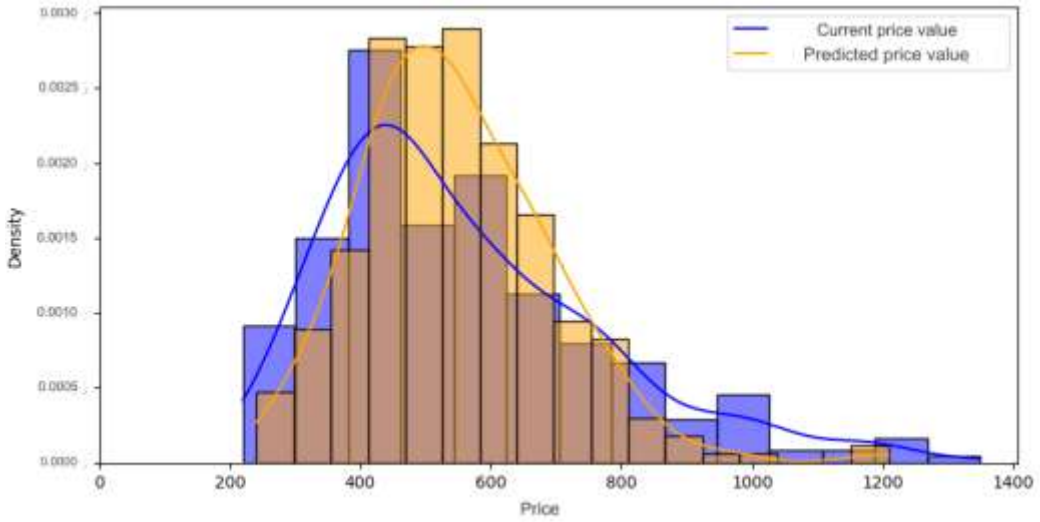

Figure 9 Predictions made using SVR

4. Results and Discussion

In this section, we explore and explain the results obtained in the analysis carried out so as to observe to what extent the results obtained responded to the aims established at the beginning of the work and to what extent they represent a benefit to the field studied. At the same time, we will present the results in a simplified form so that they will be easy to compare and track. We will analyze the relevance of these results and to what extent they can serve as a basis for further studies and developments.

4.1 Synthesized visualization of the results and their interpretation

To be able to view and compare the results in an easier way, we have chosen to display them in Table 1, where we will present the score obtained by each model.

	R squared	*RMSE*
Linear Regression	0.553	140.943
Ridge Regression	0.574	137.522
KNN	0.558	140.026
Decision Tree	0.509	147.706
Random Forest	0.690	117.312
SVR	0.541	142.765

Table 1. Synthesized view of the results

The table shows the scores obtained by the models, showing the name of each model, how well the model fits the observed data (R squared), and the estimate of the mean prediction error (RMSE).

The R-squared values obtained are moderate and indicate a significant relationship between the independent variables and the dependent variable (price), but there was still a significant part of variation that cannot be explained by the models.

Regarding RMSE (Root Mean Squared Error), we also have values that may lead to less accurate predictions but represent a good base for further improvements.

The random forest model shows a higher level of accuracy compared to the other models evaluated, stating that this model performed best and was able to correctly predict 69% of cases. Among the assumptions for which this model performed the better compared to the rest of the compared models are the following:

● Random Forest is a more advanced algorithm based on ensemble learning, which combines multiple decision trees to get more accurate results.
● This model reduces variance (how much the model results change when applied to different data sets or different training sets) compared to a single decision tree, so each model has randomly selected features and the training set, achieving more diversity in final predictions. This helps reduce errors caused by variation and can lead to better results.
● The model has the ability to identify features that are important and relevant to prediction because when building those trees that take the data and characteristics randomly, several sub conjuncts are analyzed; thus, the model is able to assess the importance of each feature and assign appropriate weights in the variable selection and decision making process. Thus, complex relationships can be established between the independent variables and the dependent variable.
● By aggregating the results from multiple decision trees, Random Forest has a good ability to manage the phenomenon of overfitting (overfitting), which then occurs when the model fits the training set too well and does not perform as well on the training data. Thus, it reduces the risk of overfitting, leading to predictions that can be used to analyze various data.

5. Conclusion

In this paper, our primary aim was to identify optimal models for predicting rental prices in the real estate market of Cluj-Napoca, Romania. We sought to assist both property owners seeking fair rental prices for their properties and potential renters aiming to secure equitable rental rates, thus mitigating overestimations and underestimations.

We selected algorithms commonly utilized in the literature to fulfill this objective. Despite rigorous efforts in data cleaning and preprocessing, our results were inevitably influenced by limitations within the dataset. Specifically, the dataset's relatively small size constrained the generalizability of our findings. Moreover, the concentration of price values within a narrow range of 400-600 EUR further restricted predictive capabilities, leaving other price ranges inadequately represented.

When interpreting our findings, it is important to take into account the constraints of our research, specifically the size of the dataset and the skewed distribution of price values. Future research should focus on expanding the dataset to encompass a broader spectrum of price ranges and incorporate additional variables to enhance predictive accuracy.

In conclusion, while our study provides valuable insights into rental price prediction models for the Cluj-Napoca real estate market, it also highlights the importance of addressing dataset limitations and encourages further research to refine predictive capabilities and broaden applicability.

Aknowledgements This study benefited greatly from the support of Miodrag-Miroslav Bilici, whose contributions substantially improved the quality and depth of our data analysis and preprocessing efforts.
Conflicts of Interest: Declare conflicts of interest or state, "The authors declare no conflict of interest."
ORCID iD https://orcid.org/0000-0003-1184-5992

References

[1] Abigail B A, Oluwatobi N A, Funmilola A A, Ololade O, Yetunde F A and Gbenle O. (2022) House Price Prediction using Random Forest Machine Learning Technique. *Procedia Computer Science. 199,*, 806-813.
[2] Ali S, Mohammad H, Fatemeh A and Christopher J P. (2022) Housing price prediction incorporating spatio-temporal dependency into machine learning algorithms. *Cities 131, December*, 103941.
[3] Ayush V, Abhijit S, Sagar D and Rohini N. (2018) House Price Prediction Using Machine Learning And Neural Networks. Second International Conference on Inventive Communication and Computational Technologies (ICICCT)
[4] Byeonghwa P. and Byeonghwa P. (2015) Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications. 42, 6, 15 April 2015, 2928-2934.
[5] Andriy B, (2019) The Hundred-Page Machine Learning Book, ISBN 199957950X, 2019
[6] Donghui S, Jian G, Jozef Z and Alan S. L. (2022) Predicting home sale prices: A review of existing methods and illustration of data stream methods for improved performance. In Proceedings of the Wires Data Mining And Knowledge Discovery 2022. Electronic copy available at: https://ssrn.com/abstract=4587725
[7] Feng W, Yang Z, Haoyu Z and Haodong S (2019) House Price Prediction Approach based on Deep Learning and ARIMA Model. 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT).
[8] Kiran K G, Malathi R D, Neeraja K and Syed A (2021) Prediction of House Price Using Machine Learning Algorithms. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI).
[9] Mitchell R. (2018) Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media, 2018
[10] Naalla V, Maturi A & Bharathi B. (2018) House Price Prediction Using Machine Learning Algorithms. ICSCS 2018: Soft Computing Systems pp 425–433.
[11] Naga S G., Raghavendran C V. and Sugnana R, C (2019) House Price Prediction Using Machine Learning. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-8 Issue-9, July 2019.

[12] Nor H Z, Shuzlina A R, Nor H U and Ismail I (2020) House Price Prediction using a Machine Learning Model: A Survey of Literature. I.J. Modern Education and Computer Science, 2020, 6, 46-54.
[13] Nur S J. Junainah M and Suriatini I. (2021) Machine Learning for Property Price Prediction and Price Valuation: A Systematic Literature Review. 19 (2021): *Planning Malaysia Journal*: Volume 19, Issue 3, 2021.
[14] Pei-Ying W, Chiao-Ting C, Jain-Wun S, Ting-Yun W and Szu-Hao H (2021) Deep Learning Model for House Price Prediction Using Heterogeneous Data Analysis Along With Joint Self-Attention Mechanism. IEEE Access: 9:2021.
[15] Quang T, Minh N, Hy D and Bo M. Housing Price Prediction via Improved Machine Learning Techniques. Procedia Computer Science. 174, 2020, 433-442.
[16] Raga M CH., Anuradha G. and Vani P M. (2019) House Price Prediction Using Regression Techniques: A Comparative Study. 2019 International Conference on Smart Structures and Systems (ICSSS).
[17] Thamarai M. and Malarvizhi S P. (2020) House Price Prediction Modeling Using Machine Learning. I.J. Information Engineering and Electronic Business, 2020, 2, 15-20.
[18] The Danh Phan. (n.d) Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. International Conference on Machine Learning and Data Engineering (iCMLDE).
[19] Winky K.O. H, Bo-Sin T and Siu W W. (2021) Predicting property prices with machine learning algorithms. *Journal of Property Research.* 38, 2021- 1.