Exploratory Data Analysis and Supervised Price Prediction of Residential Real Estate Listings

Esra Yapıcı

1 Introduction

This project was conducted within the scope of the Exploratory Data Analysis (EDA) course and focuses on the analysis and prediction of residential real estate prices using real-world data. The main objective is to examine how structural attributes and location-related information influence housing prices and to evaluate different supervised regression models under realistic data constraints.

Unlike standard EDA assignments based on pre-curated datasets, the dataset used in this study was collected manually through a custom web scraping pipeline. This allowed direct observation of data quality issues that naturally occur in real online listings.

2 Data Collection and Scraping Process

The dataset was collected from publicly available residential listings on the Storia platform. Multiple scraping approaches were tested during the data collection phase.

Initial attempts using Selenium were unsuccessful due to frequent detection of automated browsing behavior by the website. A BeautifulSoup-based approach was also evaluated; however, it proved insufficient because many listing elements were dynamically loaded via JavaScript, leading to incomplete extraction.

The final and successful scraping pipeline was implemented using Playwright. This approach enabled more realistic browser interaction and allowed stable extraction of listing attributes such as price, total area, number of rooms, and textual location descriptions. The scraped data was stored in CSV format and used as the basis for further analysis.

3 Data Preprocessing and Missing Values

The raw dataset contained missing values, inconsistent formatting, and extreme observations. All numeric variables were converted to numerical form.

Observations with missing values in critical variables (price, total area, or number of rooms) were removed, as these attributes are essential for meaningful exploratory analysis and supervised regression modeling. Missing values in location-related categorical variables were retained and encoded as `Unknown`.

Mean imputation was deliberately avoided for key numerical variables to prevent distortion of the underlying data distribution. Extreme values were filtered by removing unrealistic area values and prices above the 95th percentile.

3.1 Dataset Size

The reduction in dataset size is primarily due to the removal of listings with missing or invalid values in critical numerical attributes such as price, total area, and number of rooms. Before filtering, the dataset contains 635 observations and 11 feature columns. After applying all filtering rules, including area and price-based outlier removal, the final dataset used for exploratory analysis and supervised modeling consists of 600 observations and 11 feature columns.

All subsequent visualizations and machine learning models are based exclusively on this filtered dataset.

4 Exploratory Data Analysis

4.1 Feature Engineering

In addition to the raw attributes provided in the listings, several derived features were created to enhance the analytical and predictive capabilities of the dataset.

An area-per-room feature was computed to capture spatial efficiency and layout characteristics of the properties. This feature provides a normalized representation of space usage that is less sensitive to extreme area values.

Sector information was extracted from unstructured location text using regular expressions. Listings without explicit sector information were retained and labeled as `Unknown` to avoid unnecessary data loss.

Feature engineering was intentionally limited to transformations supported by the available data. Artificial or speculative features were avoided to prevent introducing bias into the analysis. The supervised regression models were trained using the following features:

- Total area (m^2)
- Number of rooms
- Area per room (engineered)
- Sector (categorical, extracted from location text)

These variables represent the complete and consistently available feature set across the 600 filtered observations. The following engineered feature was created:

- Area per room, computed as the ratio between total area and number of rooms

No additional synthetic or inferred features were introduced.

4.2 Price Distribution

Figure 1 shows the distribution of property prices after preprocessing and filtering. The distribution is strongly right-skewed, indicating the presence of a small number of high-priced properties.

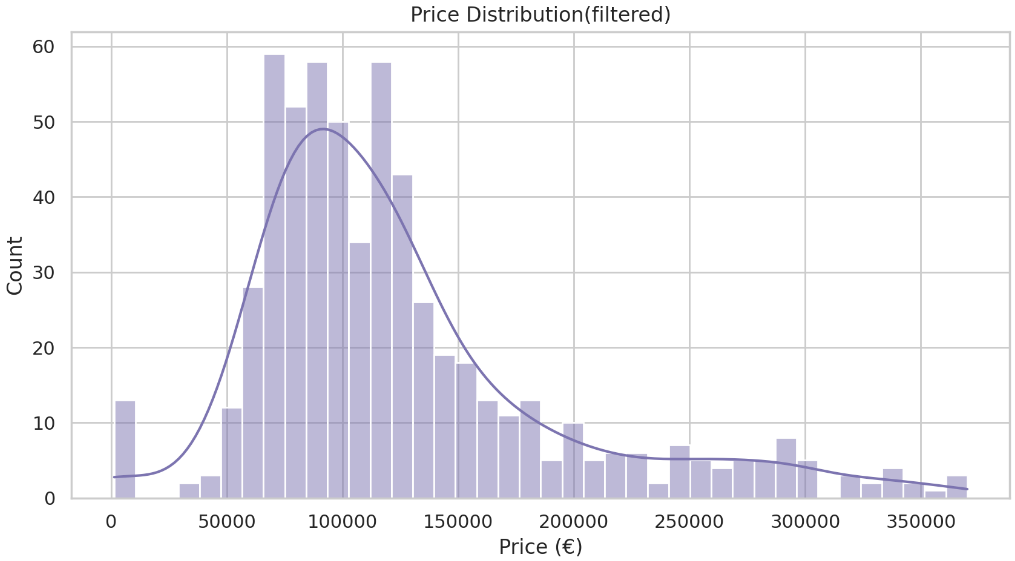


Figure 1: Distribution of property prices after preprocessing and filtering.

To reduce skewness and stabilize variance, a logarithmic transformation was applied. The resulting distribution is shown in Figure 2.

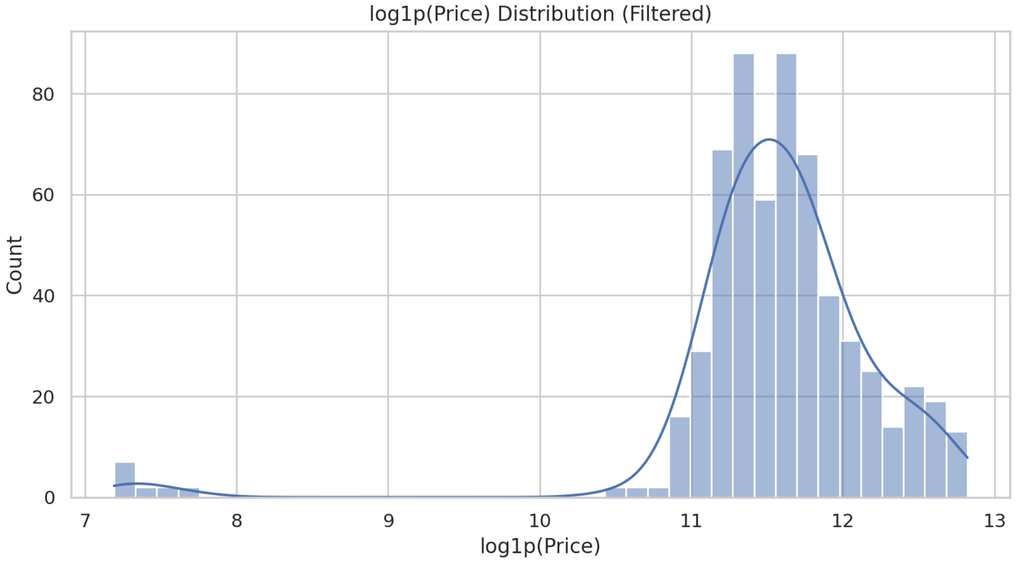


Figure 2: Distribution of $\log(1 + \text{Price})$.

4.3 Outliers and Data Filtering

Real estate price data is known to contain extreme values that can distort both visual analysis and model training. To mitigate this effect, domain-informed filtering rules were applied.

Total area values were constrained to a realistic range between 20 and 350 square meters. Additionally, listings with prices above the 95th percentile were removed to reduce the influence of extreme outliers.

These filtering steps were applied prior to visualization and modeling to ensure more stable patterns and interpretable results, while still preserving the overall complexity of real-world data.

4.4 Price and Area by Sector

The relationship between price and total area segmented by sector is illustrated in Figure 3.

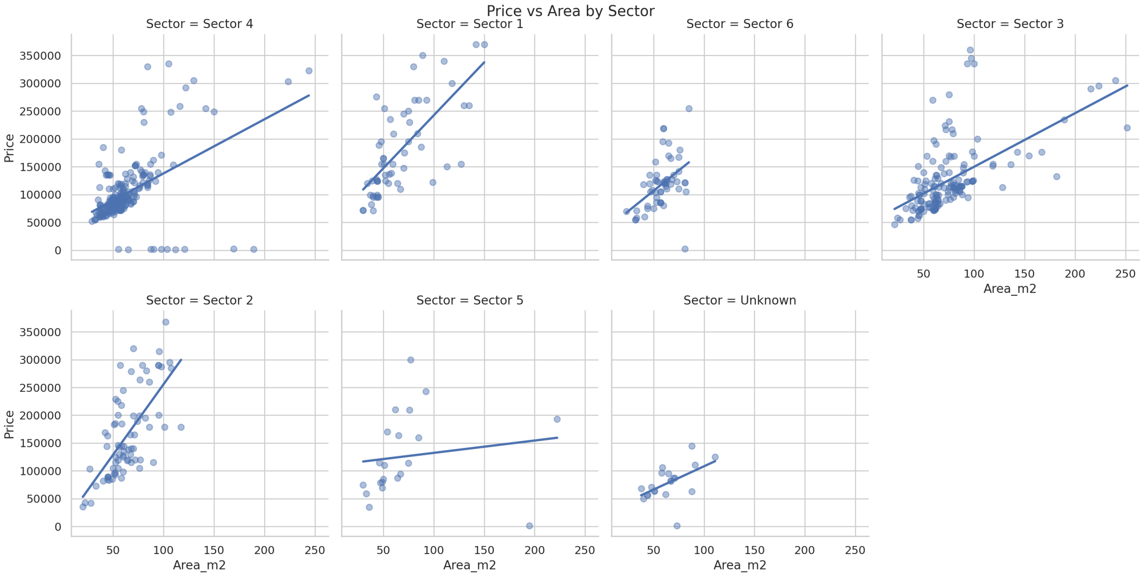


Figure 3: Price vs. area by sector. Some sectors contain fewer observations due to missing area information in the original listings.

The absence of area information in certain sectors is a direct consequence of incomplete metadata provided by sellers, rather than an error introduced during preprocessing.

4.5 Area per Room

To capture spatial efficiency, an engineered feature representing area per room was introduced. Figures 4 and 5 show its relationship with price.

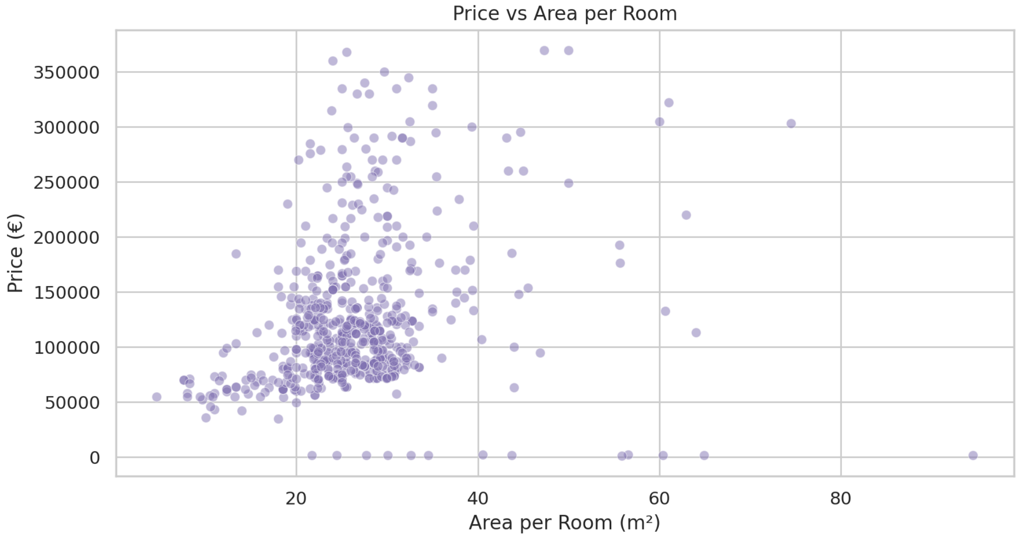


Figure 4: Price vs. area per room.

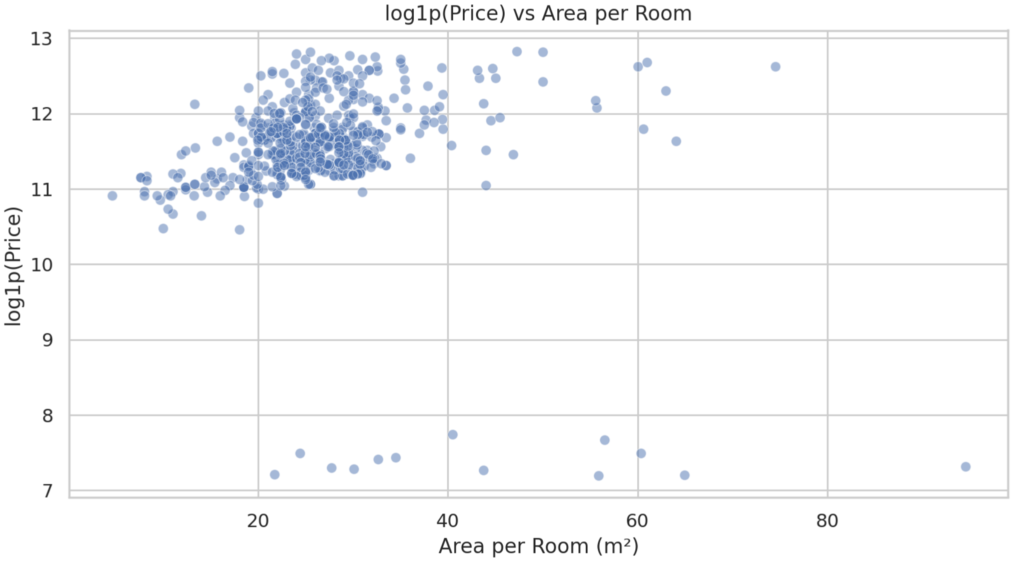


Figure 5: Log-transformed price vs. area per room.

4.6 Price Distribution by Sector

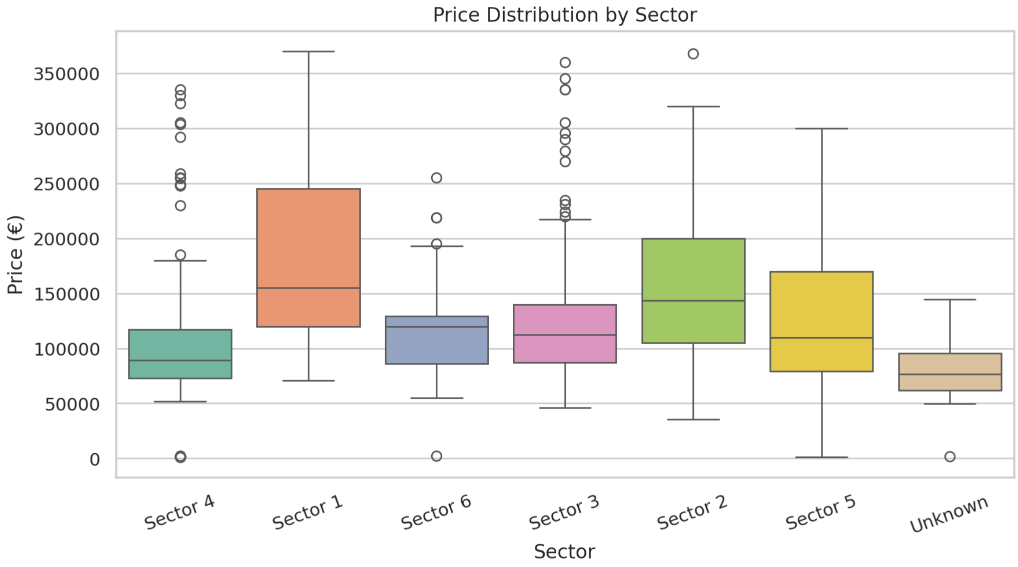


Figure 6: Price distribution across sectors. Central sectors exhibit higher median prices and greater variance.

5 Modeling and Evaluation

The problem was formulated as a supervised regression task. Several models were evaluated, including Linear Regression, Ridge, Lasso, Decision Tree, Random Forest, and Gradient Boosting.

5.1 Holdout Results (Raw Price)

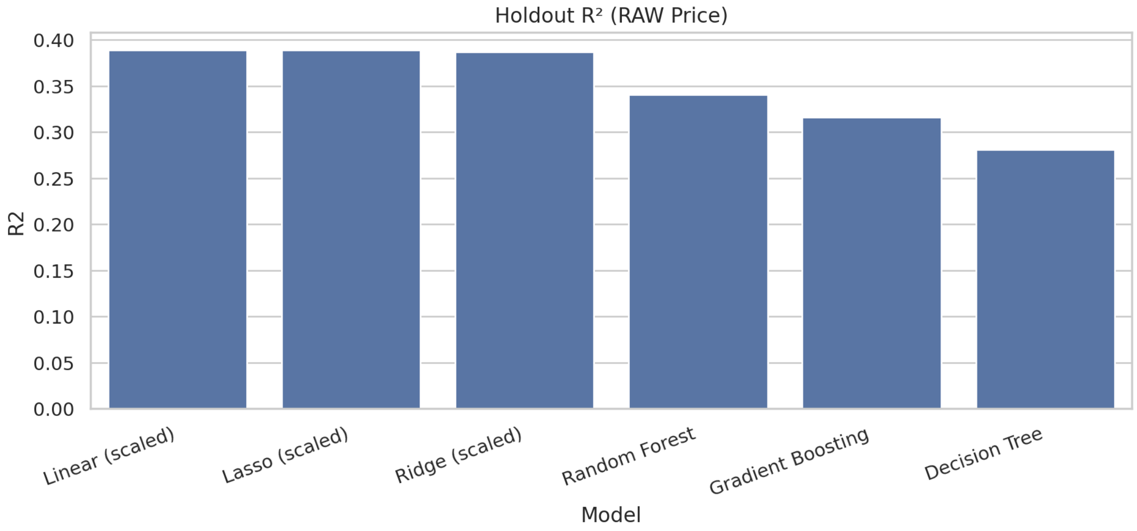
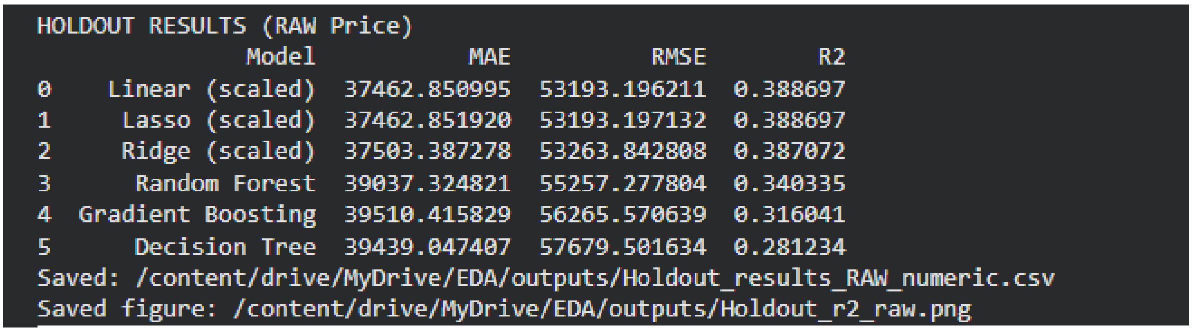


Figure 7: Holdout R^2 results for models trained on raw prices.



```
HOLDOUT RESULTS (RAW Price)
               Model           MAE          RMSE        R2
0    Linear (scaled)  37462.850995  53193.196211  0.388697
1     Lasso (scaled)  37462.851920  53193.197132  0.388697
2     Ridge (scaled)  37503.387278  53263.842808  0.387072
3      Random Forest  39037.324821  55257.277804  0.340335
4  Gradient Boosting  39510.415829  56265.570639  0.316041
5      Decision Tree  39439.047407  57679.501634  0.281234
Saved: /content/drive/MyDrive/EDA/outputs/Holdout_results_RAW_numeric.csv
Saved figure: /content/drive/MyDrive/EDA/outputs/Holdout_r2_raw.png
```

Figure 8: Holdout evaluation metrics (MAE, RMSE, R^2) for raw price models.

5.2 Holdout Results (Log-Transformed Target)

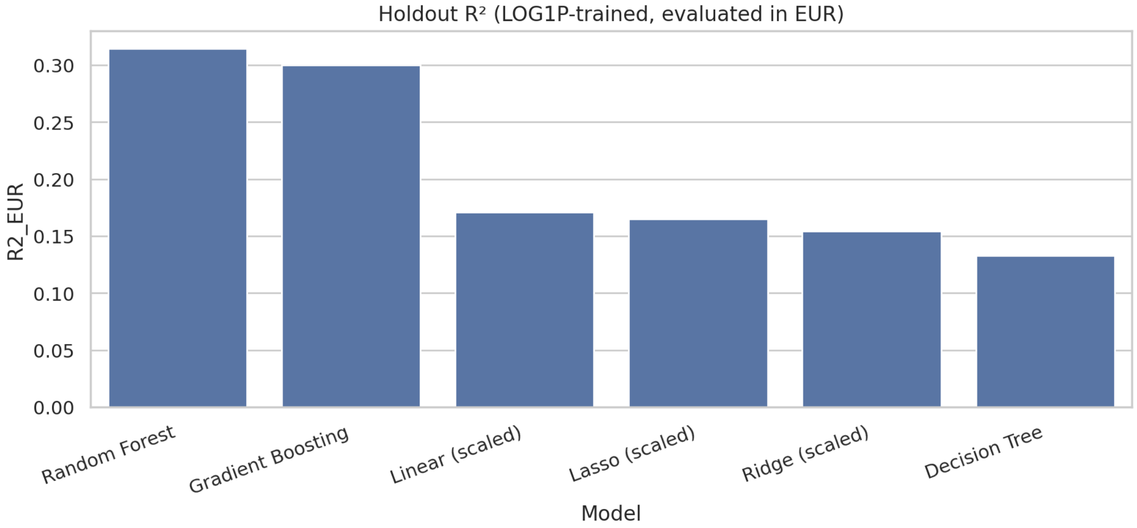


Figure 9: Holdout R^2 results for models trained on log-transformed prices and evaluated back in EUR.

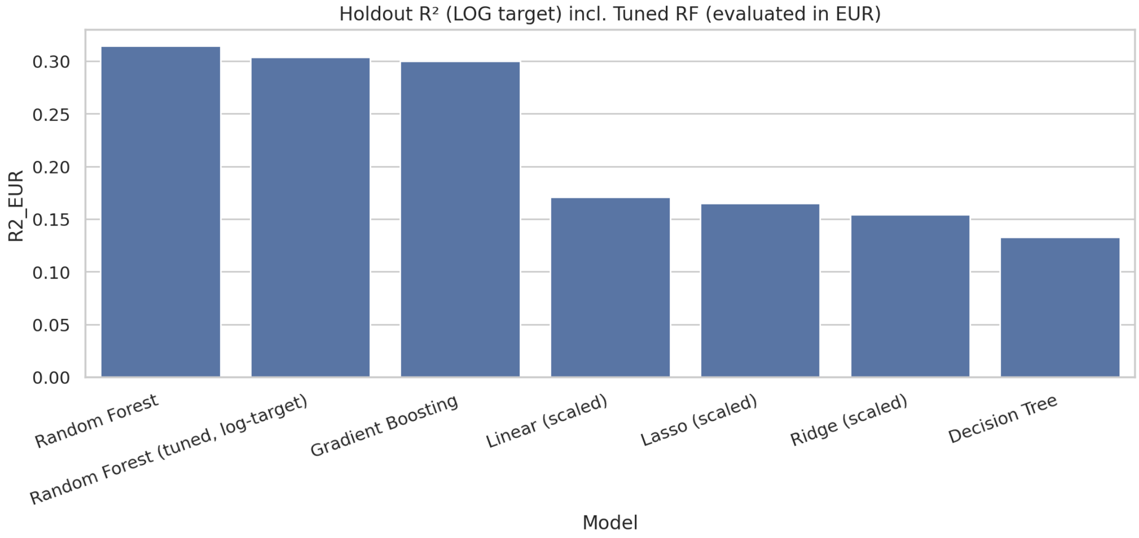
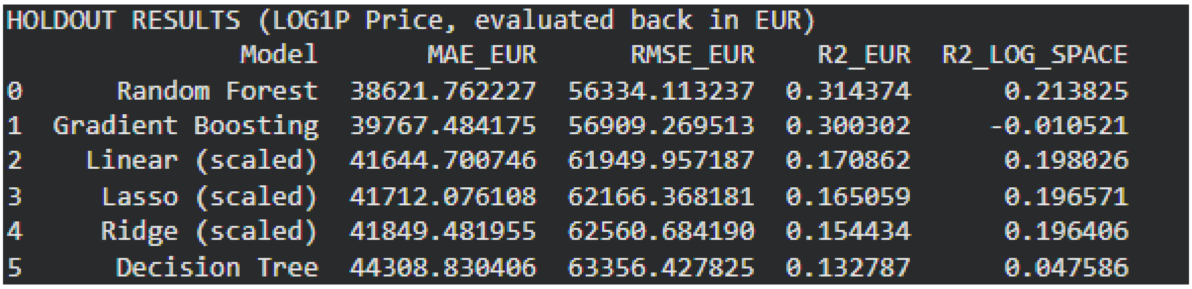


Figure 10: Comparison including tuned Random Forest model.



```
HOLDOUT RESULTS (LOG1P Price, evaluated back in EUR)
               Model       MAE_EUR      RMSE_EUR    R2_EUR  R2_LOG_SPACE
0      Random Forest  38621.762227  56334.113237  0.314374      0.213825
1  Gradient Boosting  39767.484175  56909.269513  0.300302     -0.010521
2    Linear (scaled)  41644.700746  61949.957187  0.170862      0.198026
3     Lasso (scaled)  41712.076108  62166.368181  0.165059      0.196571
4     Ridge (scaled)  41849.481955  62560.684190  0.154434      0.196406
5      Decision Tree  44308.830406  63356.427825  0.132787      0.047586
```

Figure 11: Holdout evaluation metrics for log-trained models evaluated back in EUR.

Hyperparameter optimization was performed exclusively for the Random Forest model using grid search with cross-validation. Random Forest was selected for tuning due to its sensitivity to parameter choices such as tree depth and number of estimators.

The grid search resulted in only marginal performance improvements, indicating that model performance is primarily constrained by data quality and feature availability rather than suboptimal hyperparameter settings.

Given the limited number of explanatory variables and the absence of richer structural or neighborhood-level attributes, additional feature engineering or more complex predictive models are unlikely to yield meaningful performance improvements.

Therefore, the selected set of supervised regression models is considered sufficient and appropriate for the given dataset.

6 Residual Analysis

Residual analysis was conducted for the best-performing log-trained Random Forest model.

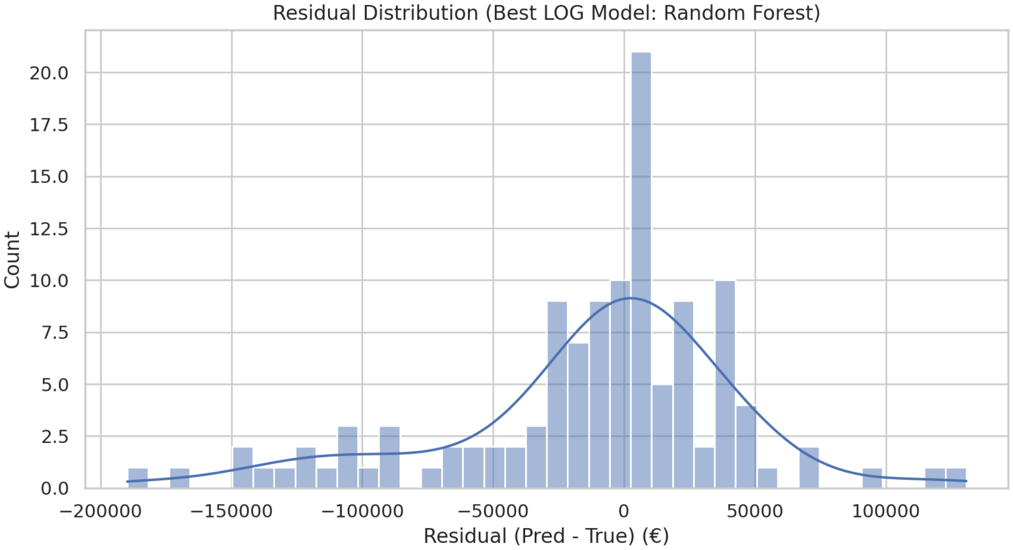


Figure 12: Residual distribution of the best-performing log-trained model.

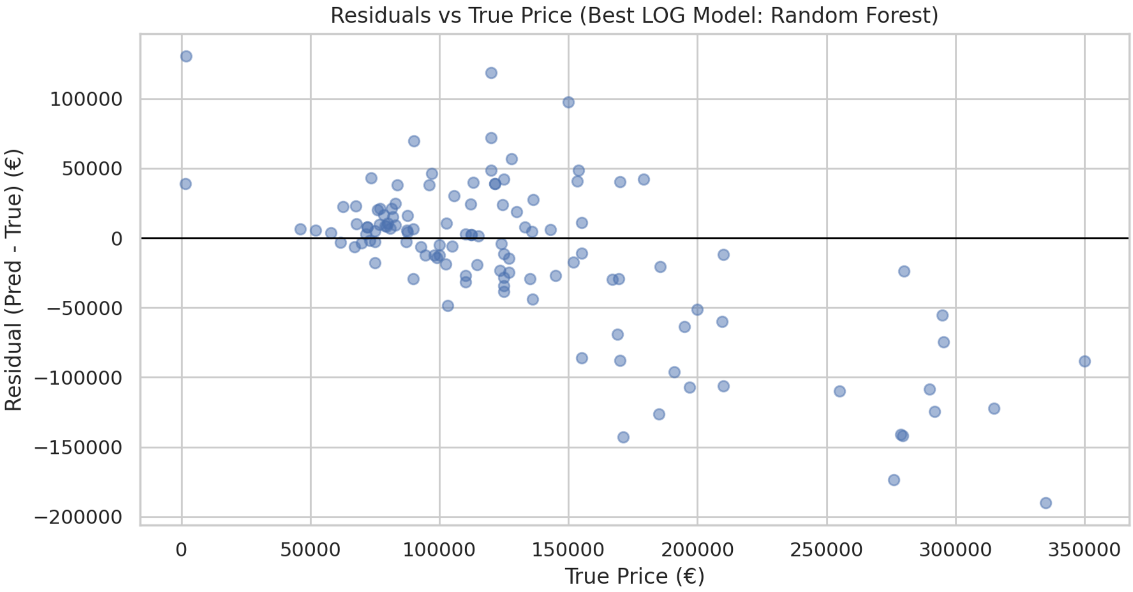


Figure 13: Residuals vs. true prices. Larger errors occur in the high-price range due to data sparsity.

7 Conclusion

This study demonstrates the challenges of applying exploratory data analysis and supervised learning techniques to real-world scraped housing data. While model performance is limited by missing metadata and unobserved factors, careful preprocessing and feature engineering enable meaningful insights into price dynamics. The results highlight that data quality and representation play a more critical role than model complexity alone. Based on the final dataset of 600 observations, the results indicate that model performance is primarily constrained by data availability rather than modeling choices.

References

[1] Storia.ro (2024). Residential real estate listings platform. Available at: `https://www.storia.ro`

[2] JetBrains. (2024). PyCharm: The Python IDE for Professional Developers. Available at: `https://www.jetbrains.com/pycharm/`

[3] Google. (2024). Google Colaboratory: Hosted Jupyter Notebook service. Available at: `https://colab.research.gov`

[4] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer.

[5] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[7] Osborne, J. W. (2002). Notes on the use of data transformations. *Practical Assessment, Research & Evaluation*, 8(6).

[8] Microsoft. (2023). *Playwright Documentation*. Available at: `https://playwright.dev`

[9] OpenAI. (2024). ChatGPT: Large language model for text generation and assistance. Available at: `https://chat.openai.com`

[10] Google. (2024). Gemini: Large language model for information retrieval and assistance. Available at: `https://gemini.google.com`