Real State Price Prediction

Machine Learning: Regression

Ricardo Schmid

Data Scientist

2024

Table of Contents

1. Business Understanding

1.1. Problem Statement

Real estate pricing can vary significantly based on numerous home features, leading to challenges in accurately estimating a fair market value. Manual estimation is prone to error, potentially resulting in homes being listed at unrealistic prices. This inconsistency highlights the need for a robust machine learning model capable of predicting home prices with greater accuracy. Such a model could significantly benefit real estate businesses by providing reliable price estimates based on comprehensive home features.

1.2. Objectives

Develop a Machine Learning Model: Create a machine learning model capable of predicting house prices using a diverse set of features. This model should effectively capture the complex relationships between home attributes and market values.

Feature Importance Analysis: Identify and prioritize key features that significantly influence home prices. Conduct thorough feature selection to decrease model complexity and enhance interpretability.

2. Data Understanding

2.1. Data Overview

The dataset, sourced from Kaggle, consists of detailed real estate listings scraped from Realtor.com.

It contains 2,000 entries providing information on various property characteristics for properties located in Chicago, as shown below:

- type: The type of property (e.g., single-family home, condo).
- text: A textual description of the property.
- year_built: The year in which the property was constructed.
- beds: The number of bedrooms.
- baths: Total number of bathrooms (including full and half).
- baths_full: Number of full bathrooms.
- baths_half: Number of half bathrooms.
- garage: Garage capacity (number of cars).
- lot_sqft: Size of the lot in square feet.
- sqft: Living area size in square feet.
- stories: Number of stories/floors in the property.
- lastSoldPrice: The price at which the property was last sold.
- soldOn: The date on which the property was last sold.
- listPrice: The listing price of the property at the time of data collection.
- status: The current status of the listing (e.g., for sale, sold).

3. Methodology: Supervised Learning - Regression

3.1. Exploratory Data Analysis

The performance of a model can be impacted when comparing different types of properties with diverse features. To enhance model effectiveness, focusing on a specific property type is advisable. Figure 1 illustrates that single-family homes dominate the dataset in terms of observations, making them the ideal candidate for modeling purposes. Therefore, the decision was made to filter the dataset to include only single-family homes. Subsequently, the property type column was removed as it no longer varied.

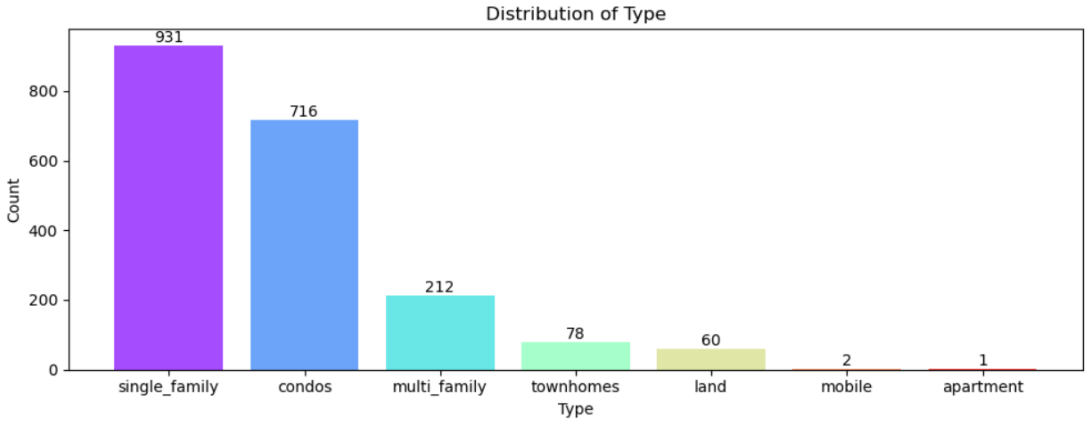

Figure 1 – Quantity of observations for each type of home

After selecting single-family homes, the status column was examined. Figure 2 shows two unique values, with "ready_to_build" appearing only once. We decided to remove this observation and retain only properties marked as "for_sale". With only one status remaining, the status column was also removed from the dataset since it no longer added useful information for our analysis or modeling.

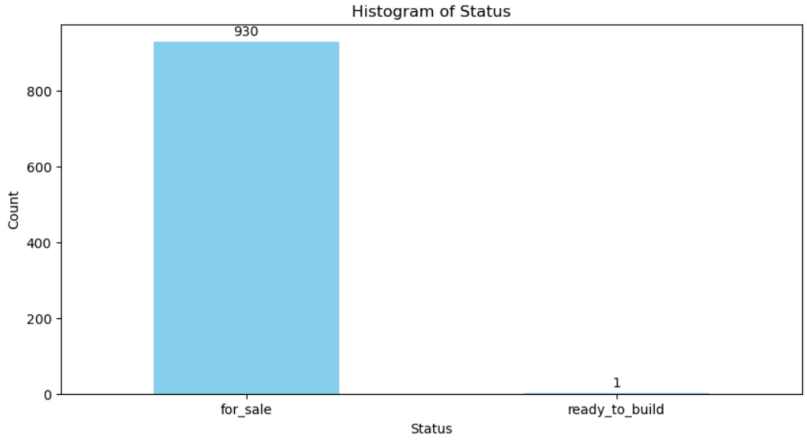

Figure 2 – Status column observations

Another column that was also removed from the dataset was the "text" column. It contained free-form property descriptions, which are unstructured data. Extracting any relevant information from

this column would require complex natural language processing and might not significantly improve the model.

Figure 3 shows that the "year_built" column contains values extending over 160 years. These values may not be very trustworthy and could lead to misleading information in the model. Therefore, the dataset was filtered to include only data from 1900 onwards to avoid removing too much data while ensuring accuracy.

```
count     877.000000
mean     1936.541619
std        33.252592
min      1856.000000
25%      1915.000000
50%      1930.000000
75%      1955.000000
max      2024.000000
```

Figure 3 – Year Built column

The "soldOn" column contains data in date format, which is not directly suitable for the model. Therefore, the year of the last sale was extracted from this column to be used alongside the last sale price in the model.

At this point, as shown in Figure 4 below, the dataset has some missing values that need to be addressed. The first one is the "listPrice" column, which is the target variable. There are two rows with missing values in this column, and these rows were removed from the dataset.

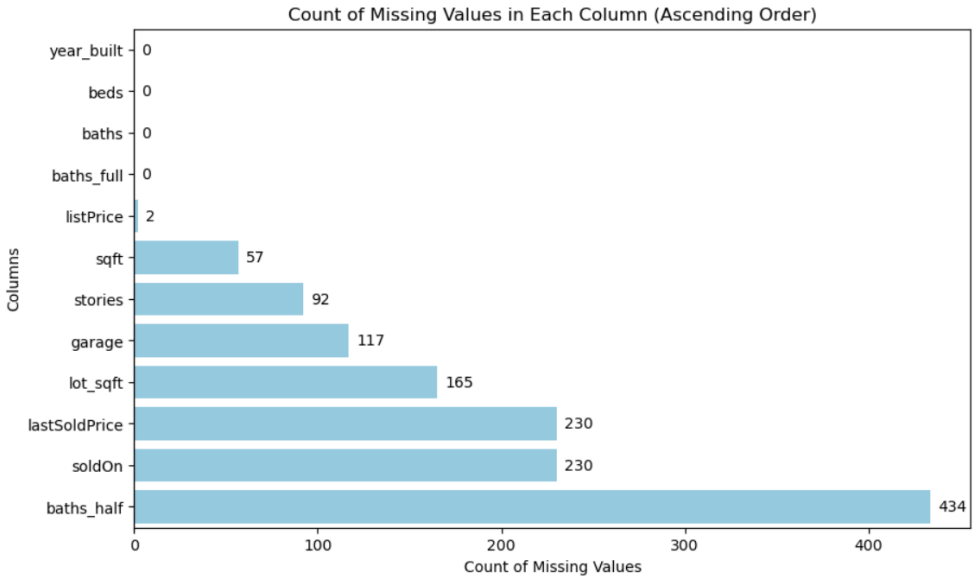

Figure 4 – Count of missing values

For the columns "square feet (sqft)," "lot square feet (lot_sqft)," and "last sold price," the missing values were filled with the mean value of the respective columns. For the columns "stories," "garage," and "soldOn (year)," the missing values were filled with the median value of the respective columns.

For the "baths_half" column, since the "baths" column is the sum of "baths_full" and "baths_half," the missing values in the "baths_half" column were considered as zero. This is because the total number of bathrooms in the house was already accounted for by the "baths_full" column.

Regarding the outliers in the dataset, the columns "year_built" and "soldOn" were not considered for outlier identification. As shown in Figure 5, outliers were identified and handled as follows: for the columns "bedrooms," "baths," "baths_full," and "baths_half," all identified outliers were removed. For the "garage" column, values above 3 were considered outliers, so only garage values equal to or less than 3 were retained in the dataset.

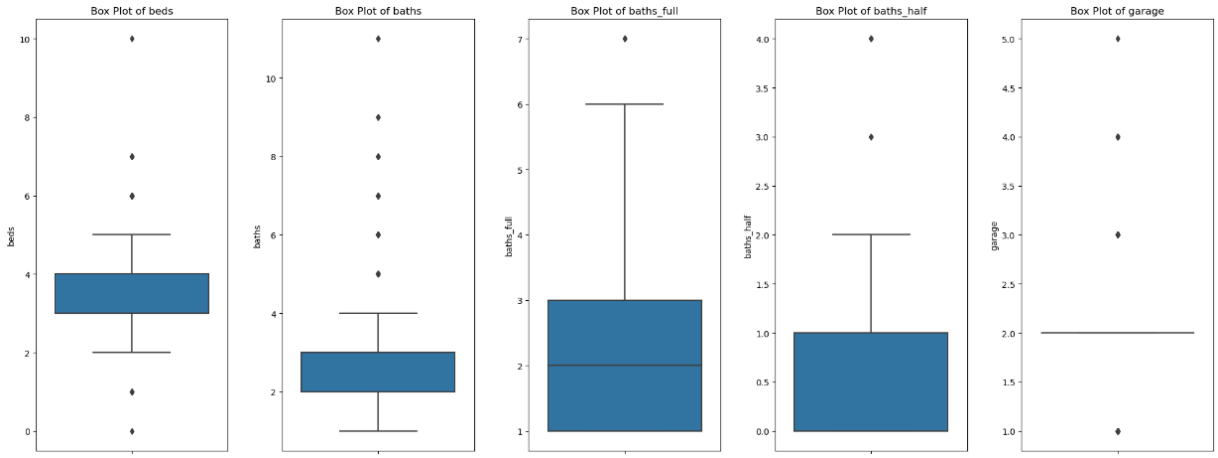

Figure 5 – Boxplot for outlier detection

As per the outliers identified in Figure 6, the maximum value for the number of stories (floors) was 4. Although it is uncommon for single-family houses to have four stories, there were only two such values, so they were not removed from the dataset. However, the outliers for lot square feet, square feet, last sold price, and list price were all removed from the dataset.

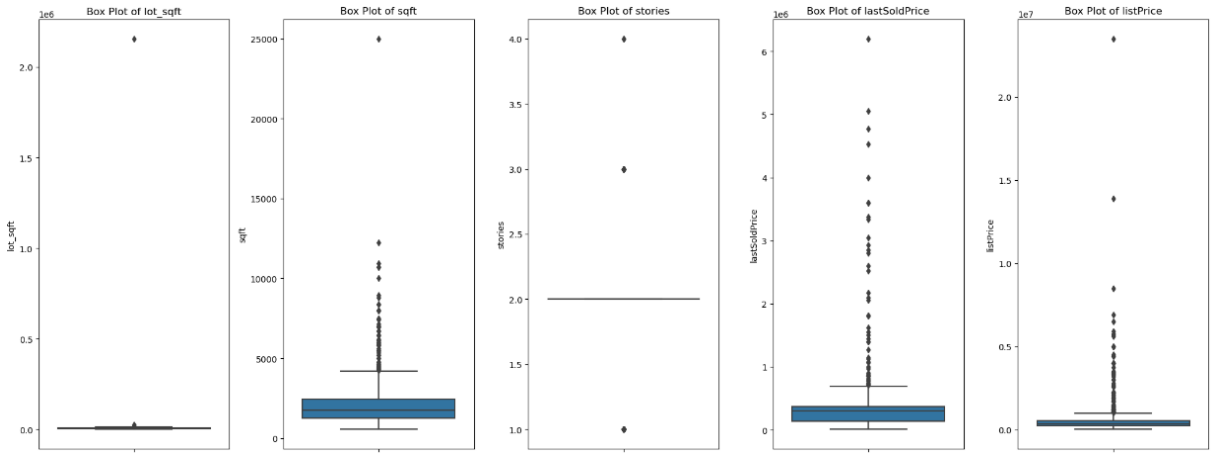

Figure 6 – Boxplot for outlier detection

After completing the data cleansing steps, 586 rows and 12 columns remain in the dataset. Figure 7 illustrates the correlations between all features. It reveals that square feet exhibit the highest

correlation with the target variable, followed by the number of bathrooms. Conversely, garage, stories, and lot square footage show the least correlation with the target variable, suggesting potential errors in the lot square footage column. Additionally, there is a notable strong correlation observed between square feet and the number of bathrooms.

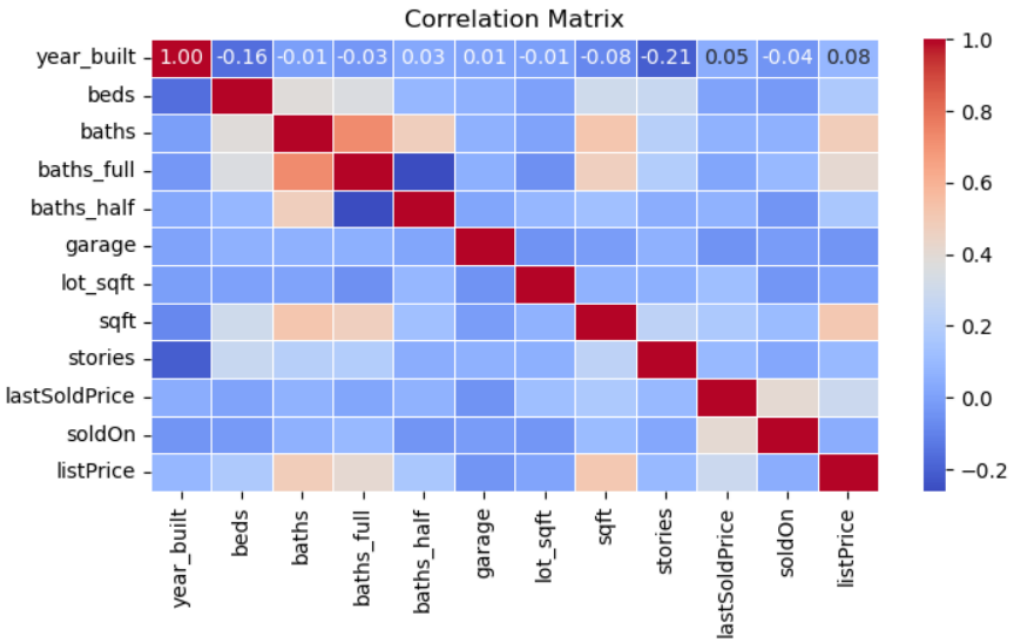

Figure 7 – Correlation Matrix

3.2. Feature Engineering

From this point onward, all steps were executed within mlOS, a low-code platform provided by Braintoy, a startup based in Calgary, Alberta. mlOS offers a user-friendly environment for executing complex machine-learning tasks with minimal coding requirements.

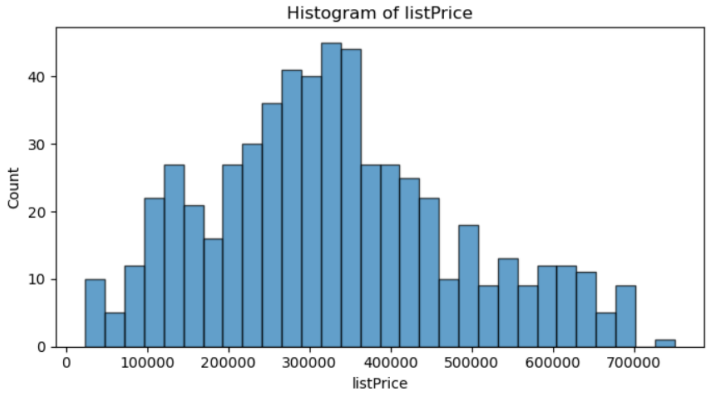

Figure 8 – List Price distribution (Slightly Skewed)

Three features — garage, stories, and lot square footage — were identified as having very low correlation with the target variable during feature importance analysis conducted in mlOS and also

in the correlation matrix on Figure 8. These features were considered less influential in predicting the target variable and were excluded from the model. This removal improved the overall performance of the model by focusing only on the most relevant predictors.

3.3. Model Building

The dataset was split into training and validation sets, with 80% of the data allocated for training the model and the remaining 20% reserved for testing. Within the mlOS environment, the Auto Pilot function was employed to automatically identify the optimal model for the dataset. Both the Random Forest Regressor and Linear Regression emerged with the lowest Mean Squared Error among all models tested, indicating their superior performance.

4. Results

4.1. Model Comparison

According to the results presented in Figure 9, the Random Forest Regressor emerged as the top-performing model, achieving the lowest mean squared error of 0.13, followed closely by linear regression with 0.14.

Version-Tag	Dataset	Algorithm	Rank	Error	Doc.	Publish	Delete
v.3-v.ada	cleaned_realstate3-Target_Norm3	RandomForestRegressor	1	0.13			
v.1-v.8c3	cleaned_realstate3-Target_Norm3	LinearRegression	2	0.14			

Figure 9 – Comparison between models

Upon reviewing all the error parameters in Figure 10, it is evident that the errors were consistently slightly higher for Linear Regression compared to the Random Forest Regressor. Specifically, the Random Forest model achieved a Root Mean Squared Error (RMSE) of 0.17, while Linear Regression had 0.18. For Mean Absolute Error (MAE), the values were 0.13 for Random Forest and 0.14 for Linear Regression. Similarly, Mean Squared Logarithmic Error (MSLE) was 0.01 for Random Forest and 0.02 for Linear Regression, and for Median Absolute Error (MEDAE), the values were 0.1 and 0.12, respectively.

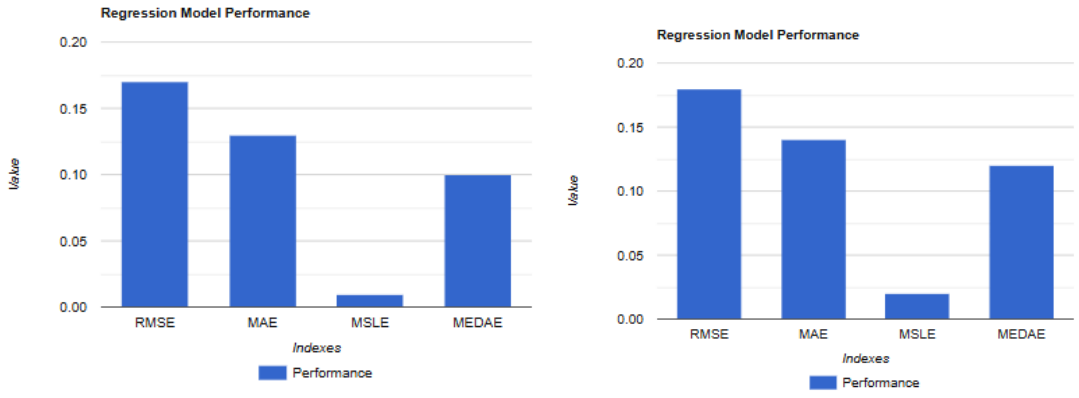

Figure 10 - Random Forest Regressor (left) vs Linear Regression (right)

Figure 11 below depicts the Residual vs Predicted Target Plot, illustrating the distribution of prediction errors (residuals) relative to the predicted target values. In these plots, the residuals are predominantly centered around zero, with some dispersion. Most residuals fall within the range of -0.5 to 0.5, suggesting that the model generally predicts the list price accurately. This indicates that while there is variability, the model does not consistently overestimate or underestimate, reflecting a positive performance indicator.

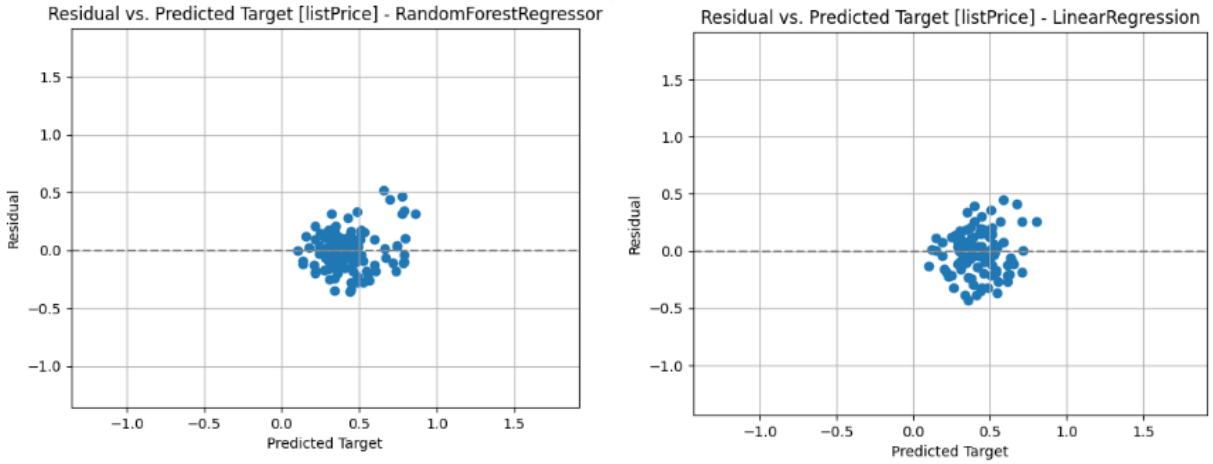

Figure 11 – Residual vs Predicted Target Comparison

Figure 12 illustrates the Predicted vs True Target Plot, comparing the model's predicted values against the actual target values. In these plots, points are scattered around the diagonal line, indicating that the models generally predict values close to the true values. While there is some deviation, the plot shows a strong correlation between predicted and true values.

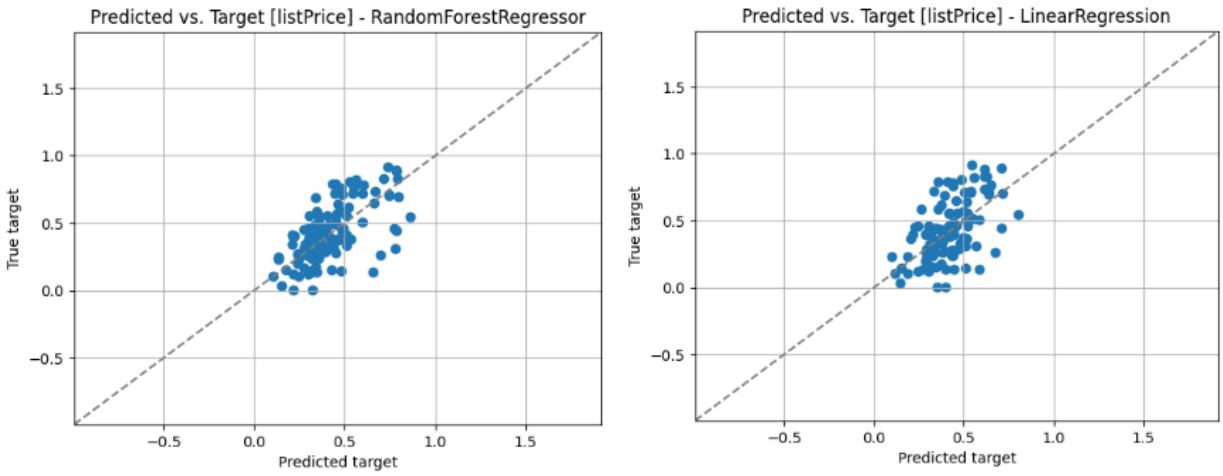

Figure 13 – Predicted vs True Target Plot Comparison

Given the visual similarities in the plots, the choice between the two models can be informed by their MSE values. The RandomForestRegressor, having a lower MSE, demonstrates slightly better

predictive accuracy. This choice ensures better predictive performance and more accurate house price predictions.

4. Conclusion

4.1. Real-World Applications

The development of a robust machine learning model for real estate price prediction presents significant real-world applications. By leveraging a diverse set of home features such as square footage, number of bathrooms, and others identified through rigorous feature importance analysis, the model can provide accurate and reliable price estimates. This capability is valuable for real estate businesses, enabling them to optimize pricing strategies, reduce listing errors, and enhance overall market competitiveness.

4.2. Future Improvements

Incorporating additional features such as neighborhood or region, proximity to public transportation, and availability of amenities like supermarkets could enrich the model's predictive power. These factors can play crucial roles in determining property values and could provide deeper insights into market dynamics.

4.3. Lessons Learned

Throughout this project, several key lessons have emerged. Feature selection has proven to be critical in building a model that effectively captures the relationships between home attributes and prices. The use of normalization techniques has significantly improved model performance by addressing data skewness and enhancing predictive accuracy. Furthermore, strategies for handling missing data, such as imputing with mean or median values, can diminish the variability in the data. This is because it does not capture the true distribution and relationships inherent in the missing data points. Consequently, the feature values become more homogeneous, potentially weakening their statistical association with the target variable. These insights underscore the importance of meticulous data preprocessing and thoughtful feature engineering in achieving optimal machine learning outcomes.