Airbnb price prediction

Using by Machine Learning

Avishay Aknin

Bar illan University

Data Science Course

Anatomy of an ML Project

1 .Domain Knowledge-

Defining the Questions:

- What factors most influence Airbnb property prices in Berlin?
- How accurately can we predict these prices given various features?

Understanding the Problem:

Airbnb property prices are influenced by factors such as location, property type, and amenities. Historical price data and property details are available for analysis, which will be used to understand these influences and build a predictive model.

Outcome Definition:

The goal is to create a predictive model with a maximum 10% error rate, enabling accurate price predictions for hosts and helping renters make informed choices. The model should also identify key factors impacting property prices**.**

Identifying Influences:

Key factors include response time, host status, location attributes (such as neighborhood and postal code), property features (including type, size, and room configuration), booking conditions, and review ratings.

Exploring New Knowledge:

Potential new information could include the experience level of hosts, such as their years of activity on the platform. Additionally, analyzing price distributions by categorizing properties into very high-end and budget-friendly segments could provide further insights into pricing dynamics.

Project Design

Optimal Design Selection:

To achieve our goals, we selected a combination of regression models, Random Forest, Support Vector Machines (SVM), and other advanced techniques. This mixed approach leverages the strengths of different models to enhance the accuracy of price predictions.

Defining Research Subjects:

We define and select research subjects by focusing on Airbnb listings in Berlin, analyzing various property features such as location, type, and amenities to identify what influences pricing.

Data Handling Strategy:

Excluding Irrelevant Columns:

- Removed columns that were deemed irrelevant for prediction due to their unique or unsuitable data types for the model like hostID, ReviewID columns .
- Excluded columns with excessive cardinality or redundancy that could potentially introduce noise like review_date , ListingID , Listing URL columns.

Filtering Location:

- Eliminated all listings not located in Berlin to ensure the dataset is focused and relevant to the target area.
- Extracted latitude and longitude columns into a separate dataframe to ensure they do not influence the prediction model but remain available for any future geographic analysis.

Cleaning Data:

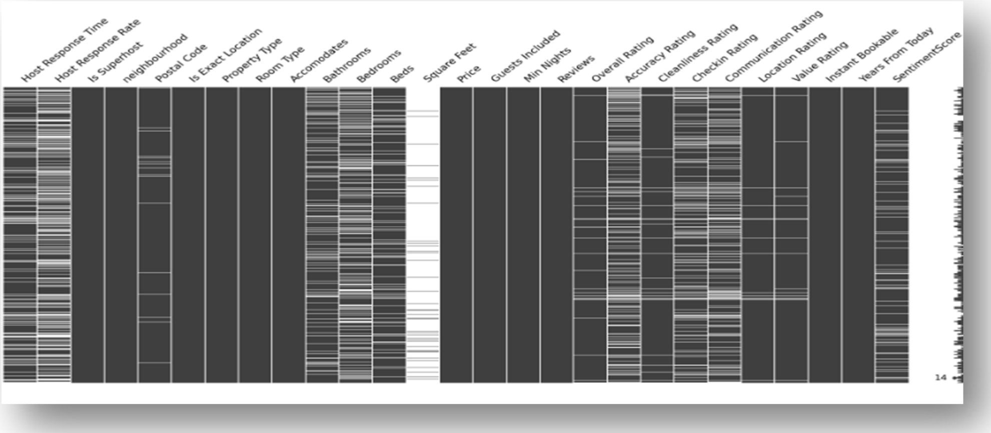
- Addressed discrepancies in fields with anomalous values, including properties listed with zero or negative prices, by either correcting or removing these entries.

Handling Missing Values:

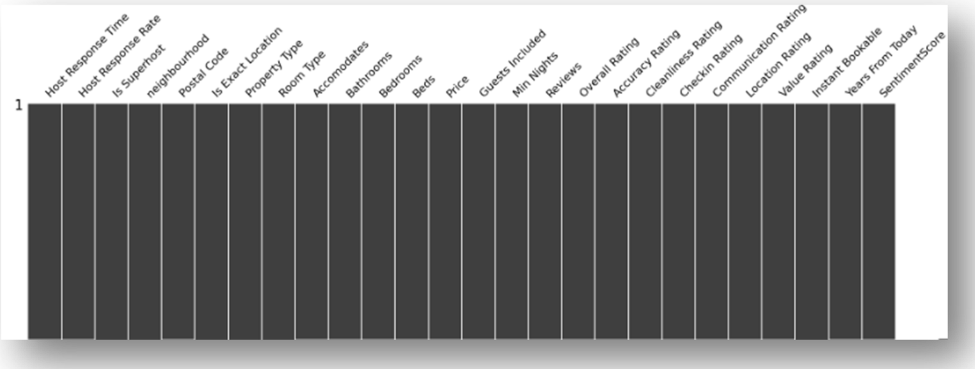
- Initially filled missing values using K-Nearest Neighbors (KNN) imputation. because it assumes that apartments with similar characteristics have similar prices. This approach helps estimate missing values based on the values of similar listings.
- Removed over 10,000 records with incomplete or unusable data because K-Nearest Neighbors (KNN) imputation was unable to fill them due to insufficient similar data points.
- Completed remaining missing values using Label Encoding to ensure data integrity and usability and filling numerical columns with their mean values to ensure data integrity and usability.

Msno module plot – display nulls values on plot

Before knn imputation and mean filling



After

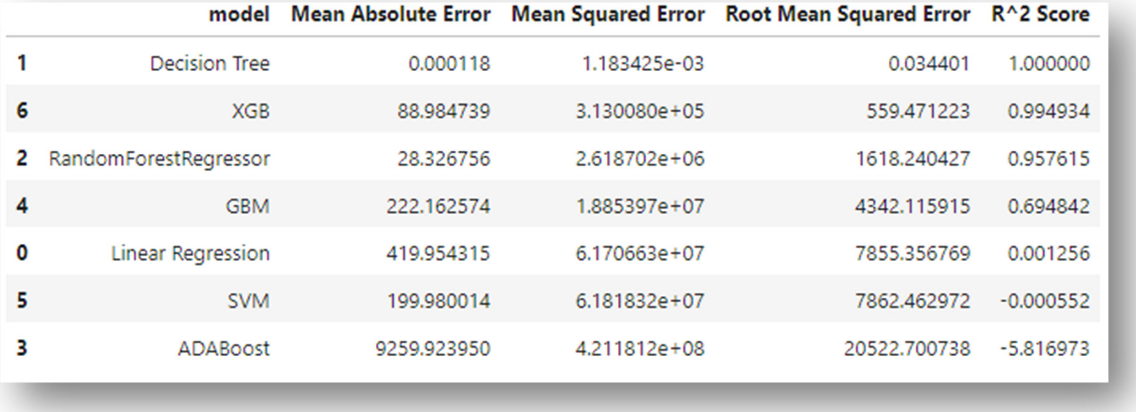


Feature Selection:

Utilized multiple feature importance assessment models, including Lasso Regression, Support Vector Machine (SVM), Gradient Boosting, and Random Forest, to evaluate and select features for the final dataset. Features were included based on their consistent selection across at least three of the models.

Algorithm Validation:

Validate algorithms through cross-validation and assess performance using metrics like RMSE or MAE for continuous price predictions to ensure accuracy and reliability.



	model	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R^2 Score
1	Decision Tree	0.000118	1.183425e-03	0.034401	1.000000
6	XGB	88.984739	3.130080e+05	559.471223	0.994934
2	RandomForestRegressor	28.326756	2.618702e+06	1618.240427	0.957615
4	GBM	222.162574	1.885397e+07	4342.115915	0.694842
0	Linear Regression	419.954315	6.170663e+07	7855.356769	0.001256
5	SVM	199.980014	6.181832e+07	7862.462972	-0.000552
3	ADABoost	9259.923950	4.211812e+08	20522.700738	-5.816973

2.Data Preparation

Data Extraction

Collect data from the Kaggle Airbnb Berlin dataset.
https://www.kaggle.com/datasets/thedevastator/berlin-airbnb-ratings-how-hosts-measure-up

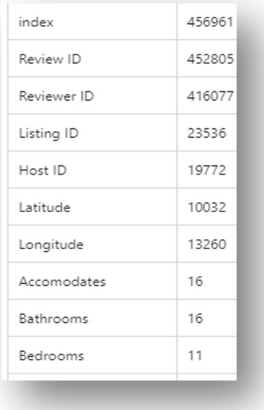
File Preparation

Organize the data into structured flat files for consistency and ease of use.

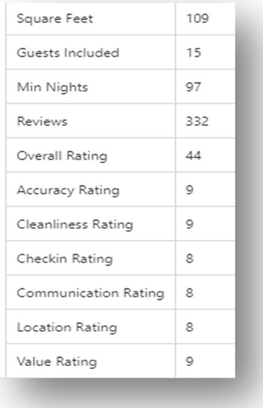
Data Preparation

1. **Clean Text Data:** Remove non-alphanumeric or numeric characters - (e.g., ;@#$%) from text fields to standardize and normalize the text.
2. **Reduce Large Categories:** • Consolidated categorical variables with excessive unique values, such as index, reviewID, ReviewerID, ListingID, and HostID, into broader categories. This simplification aims to streamline the dataset and improve model performance.

Uniqueness of columns dataframe-



index	456961
Review ID	452805
Reviewer ID	416077
Listing ID	23536
Host ID	19772
Latitude	10032
Longitude	13260
Accomodates	16
Bathrooms	16
Bedrooms	11



Square Feet	109
Guests Included	15
Min Nights	97
Reviews	332
Overall Rating	44
Accuracy Rating	9
Cleanliness Rating	9
Checkin Rating	8
Communication Rating	8
Location Rating	8
Value Rating	9

Exploratory Data Analysis (EDA) Steps:

1. **Basic Statistics:**

- Calculate minimum, maximum, mean, and unique values for each column to get an initial understanding of the dataset.

2. **Data Overview:**

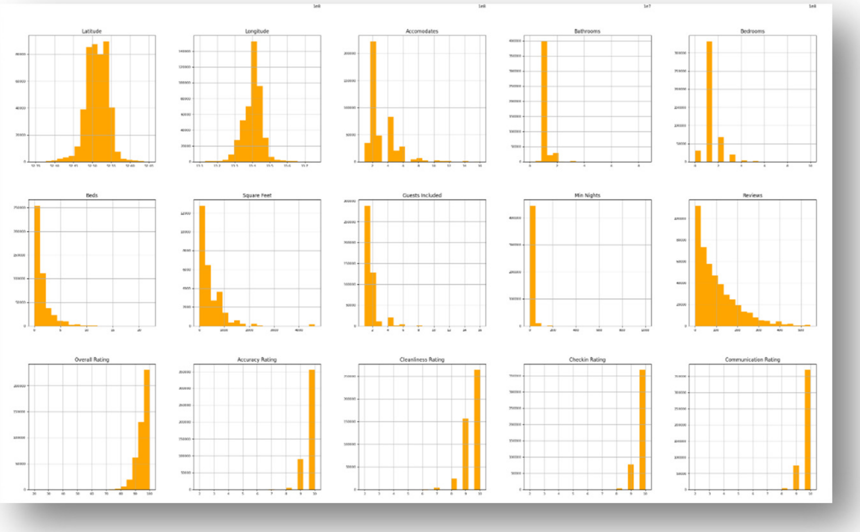
- Use df.info() and df.describe() to understand data types, missing values, and summary statistics.

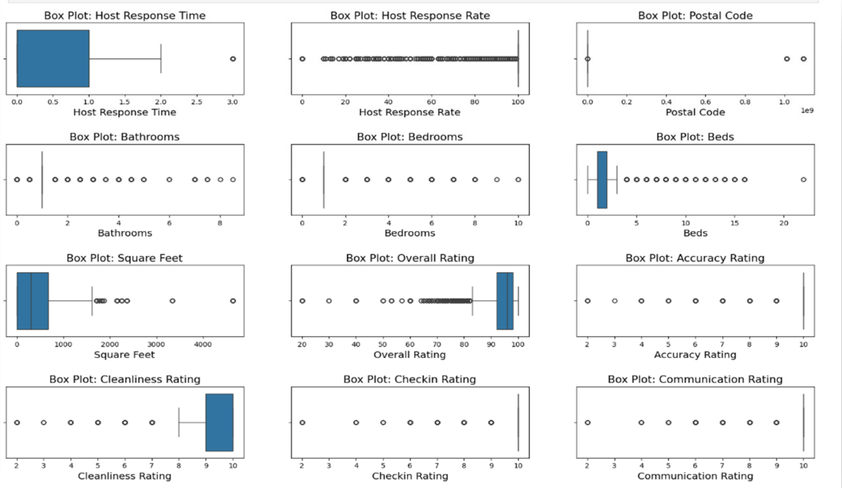
3. **Visual Analysis:**

- Utilize the AUTOVIZ library to automatically generate visualizations and uncover patterns and trends in the data.

4. **Distribution Analysis:**

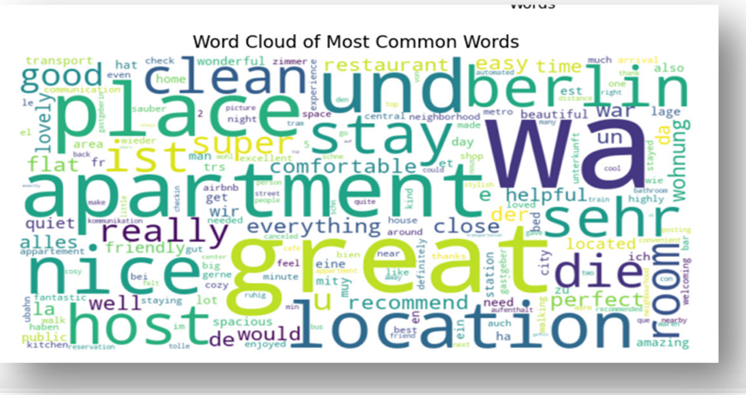
- Examine the distributions of key columns, such as numeric columns such as price, to understand their ranges and variances.

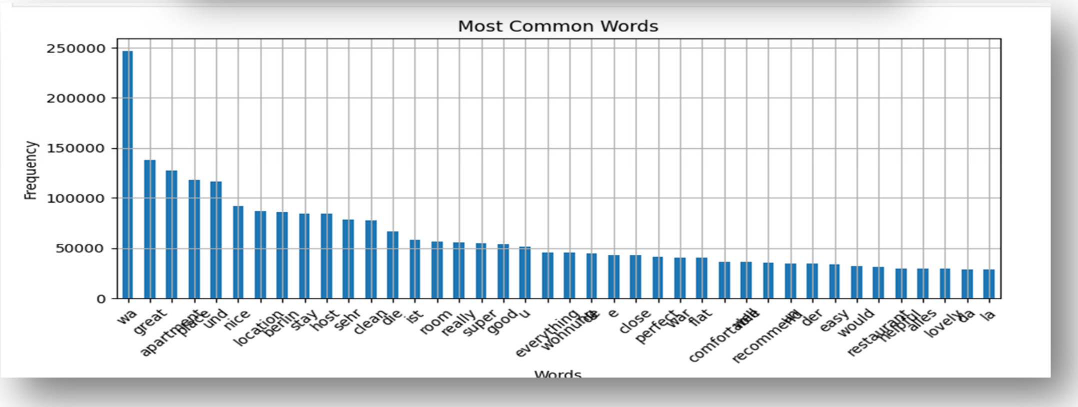




5. **Sentiment Analysis:**

- Analyze the sentiment of customer reviews to gauge overall customer satisfaction and its potential impact on the target variable.





Sentiment Analysis Workflow

Preprocess and Tokenize:

- Convert text to lowercase.
- Remove punctuation.
- Tokenize and lemmatize words.

Define Sentiment Words:

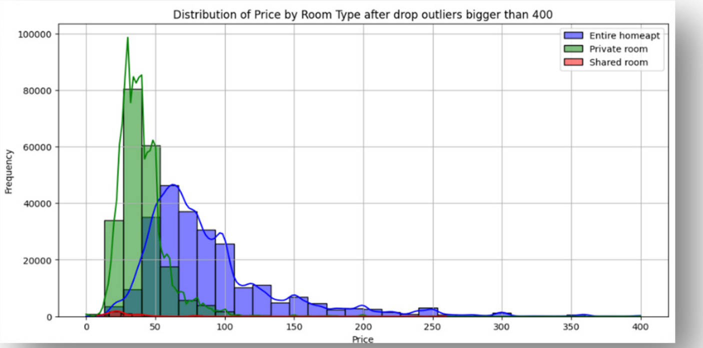
- Choose and lemmatize positive words (e.g., "great," "good") and negative words (e.g., "bad," "horrible").
- Positive Words: Assign a positive score (+1) to each positive word.
- Negative Words: Assign a negative score (-1) to each negative word.

Calculate Sentiment Scores:

- Count occurrences of positive and negative words in each comment.
- Compute sentiment score as positive word count minus weighted negative word count. Normalize to a 0-1 scale.

6. **Price Target columns Distribution:**

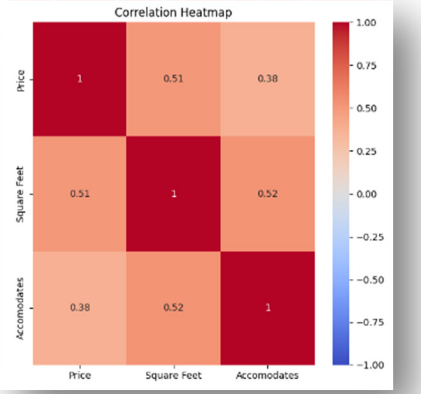
- Investigate the distribution of the price column, which is the target variable, to identify trends and potential outliers.

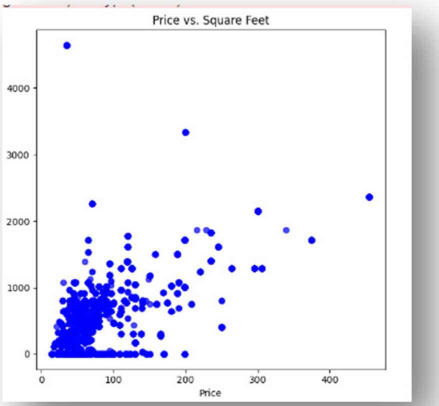


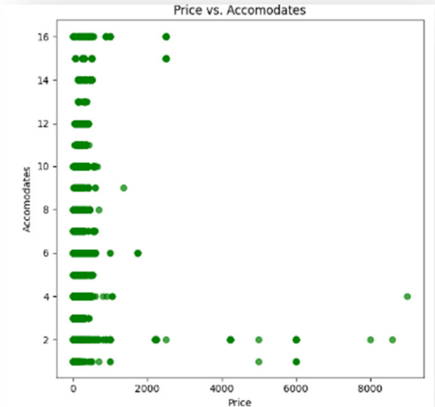
7. **Correlation Analysis:**

- Evaluate the correlation between variables to understand relationships and dependencies, using correlation matrices or heat maps.

Price vs square feet & accommodate

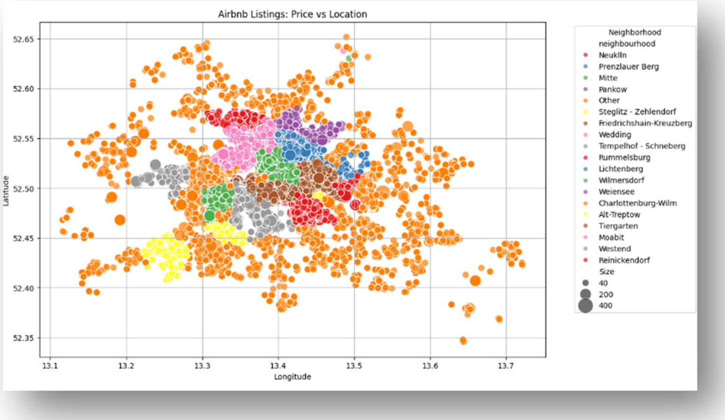






8. **Geographic Analysis:**

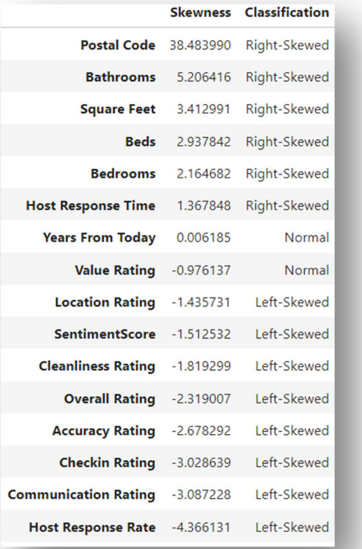
- Analyze geographic data by mapping property locations to identify patterns and trends based on location, using tools like Folium.



Data Cleansing – Outliers and Missing Values

Outlier Detection and Handling:

1. **Check Distribution:**
   - Assess the distribution of data for skewness. Determine if data is left-skewed, right-skewed, or normally distributed.
2. **Normalize Using Z-Score:**
   - For data that is normally distributed, use Z-score normalization to identify and handle outliers. Values with Z-scores beyond ±2.5 are considered outliers.
3. **Apply IQR Method:**
   - For non-normally distributed data, use the Interquartile Range (IQR) method. Calculate IQR (Q3 - Q1) and define outliers as values outside 1.5 * IQR below Q1 or above Q3.



	Skewness	Classification
Postal Code	38.483990	Right-Skewed
Bathrooms	5.206416	Right-Skewed
Square Feet	3.412991	Right-Skewed
Beds	2.937842	Right-Skewed
Bedrooms	2.164682	Right-Skewed
Host Response Time	1.367848	Right-Skewed
Years From Today	0.006185	Normal
Value Rating	-0.976137	Normal
Location Rating	-1.435731	Left-Skewed
SentimentScore	-1.512532	Left-Skewed
Cleanliness Rating	-1.819299	Left-Skewed
Overall Rating	-2.319007	Left-Skewed
Accuracy Rating	-2.678292	Left-Skewed
Checkin Rating	-3.028639	Left-Skewed
Communication Rating	-3.087228	Left-Skewed
Host Response Rate	-4.366131	Left-Skewed

One-Hot Encoding and Label Encoding

Label Encoding:

- Applied label encoding to categorical columns where variables have a natural order or where the number of unique categories is high.

One-Hot Encoding:

- Used for the Room Type column, which has only two categories: "Private room" and "Shared room." This column was one-hot encoded to capture its impact on the price, representing each category with a separate binary column to avoid introducing ordinal relationships that do not exist.

Feature Selection

1. **Apply Models:**
    - Used Lasso, SVM, Gradient Boosting, and Random Forest to identify important features.
2. **Aggregate Results:**
    - Created a Data Frame to combine feature importance from each model.
3. **Select Features:**
    - Chose features selected by at least three models.
4. **Finalize Dataset:**
    - Constructed a final dataset with selected features and the target variable Price.

Project Summary: Deployment and Beneficiaries of Machine Learning

How will we deploy the Machine Learning?

The machine-learning model will be deployed through an API that integrates with existing property management or booking platforms. This API will provide real-time price predictions and insights based on the model. The deployment process will include setting up cloud-based infrastructure for scalability, ensuring the model can handle high volumes of requests, and establishing data pipelines to continuously update the model with new data.

Who will use and benefit from the Machine Learning?

- **Property Hosts**: They will use the model to set competitive prices for their listings, optimizing their earnings while remaining attractive to potential guests.
- **Renters**: They will benefit from accurate price predictions, helping them make informed decisions and find better deals.
- **Property Management Platforms**: These platforms can integrate the model to offer value-added services, enhancing user experience and providing actionable insights for market trends and pricing strategies.

Algorithms

Method Selection:

For this Airbnb price prediction project, chosen supervised learning methods. Supervised learning is appropriate because we have labeled data (i.e., historical prices) and we aim to predict a continuous target variable (price). Algorithms like regression models, Random Forest, and Gradient Boosting are used to leverage this labeled data effectively.

Model Validation Techniques:

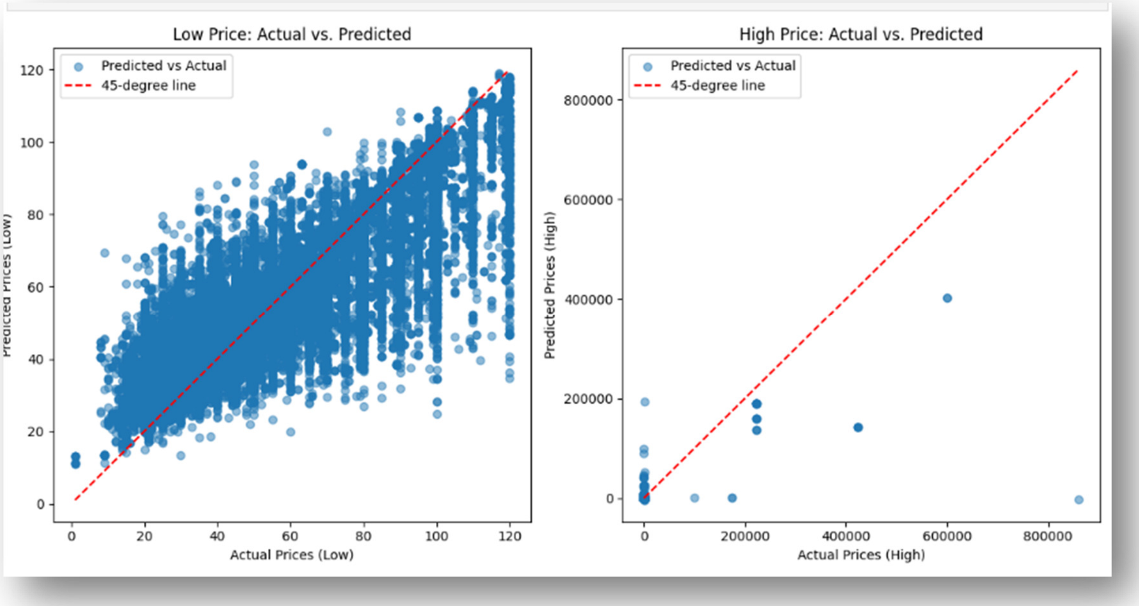
To validate model accuracy, we employ statistical tests such as Mean Squared Error (MSE) and R-squared. These metrics help in assessing how well the model predicts the price compared to the actual values.

Dataset Partitioning

We partition the dataset into training and testing sets, using a 70% training and 30% testing split. Additionally, we further divide the test set into two groups based on price categories: low-priced and high-priced apartments. This segmentation allows us to evaluate the model's performance separately for different price ranges, providing insights into how well the model predicts both ends of the price spectrum and ensuring that predictions are accurate across various price points.

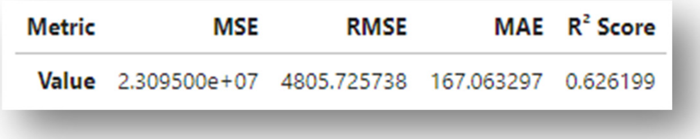
To enhance prediction quality, we decided to separate the data into two data frames due to the significant imbalance in price distribution. Approximately 85% of the data is concentrated in the lower price range (below $120), while the remaining 15% includes extreme values reaching up to $500,000 or more. By creating separate data frames for low-priced and high-priced apartments, we aim to improve model performance and accuracy for each distinct price range.

Example Segmentation of Low and High Price Categories



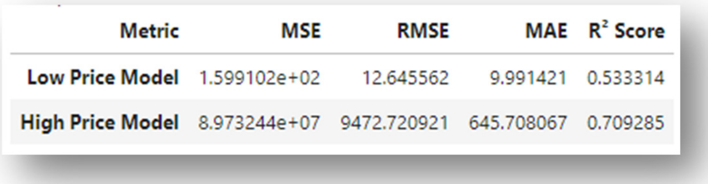
Example of Random forest model

Validation parameter before segmentation



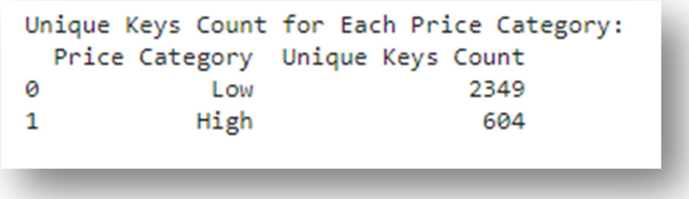
Metric	MSE	RMSE	MAE	R^2 Score
Value	2.309500e+07	4805.725738	167.063297	0.626199

Validation parameter after segmention



Metric	MSE	RMSE	MAE	R^2 Score
Low Price Model	1.599102e+02	12.645562	9.991421	0.533314
High Price Model	8.973244e+07	9472.720921	645.708067	0.709285

Without segmenting the data by price range, the model exhibited a high RMSE of 4805. However, when divided into low and high price categories, the RMSE improved significantly for low-priced apartments to 12.6, while it remained exceptionally high at 9472 for high-priced apartments. This improvement in accuracy for low-priced apartments, which constitute 85% of the dataset, is due to the higher density of unique data points in this range, allowing the model to achieve better performance and higher quality prediction.



```
Unique Keys Count for Each Price Category:
  Price Category  Unique Keys Count
0            Low               2349
1           High                604
```

Boosting:

We use Gradient Boosting to enhance model accuracy. It builds multiple decision trees sequentially, where each tree corrects the errors of the previous ones, improving prediction precision and handling complex patterns effectively.

Hyperparameter Tuning:

We used Grid Search to fine-tune hyperparameters. This method involves systematically testing various combinations of hyperparameters to identify the optimal settings that maximize model accuracy and minimize error.

Best Model Analysis:

Model Comparison: XGBoost vs. Random Forest

When comparing the performance of XGBoost and Random Forest models for low and high-priced apartments, XGBoost demonstrates superior accuracy in both categories.

XGBoost Performance:

- Low Price: MSE = 168.21, RMSE = 12.97, MAE = 9.40, R^2 Score = 0.73
- High Price: MSE = 302,070,500, RMSE = 17,380.18, MAE = 1,235.54, R^2 Score = 0.58

Random Forest Performance:

- Low Price: MSE = 159.91, RMSE = 12.65, MAE = 9.99, R^2 Score = 0.53
- High Price: MSE = 89,732,440, RMSE = 9,472.72, MAE = 645.71, R^2 Score = 0.71

Evaluation:

- Low Prices: XGBoost has a marginally better R^2 score, indicating better overall accuracy, while RMSE and MAE values are very similar between XGBoost and Random Forest.
- High Prices: Random Forest outperforms XGBoost with lower RMSE, MAE, and a higher R^2 score, demonstrating better handling of high variability in high-priced listings.

Implementation

Model Selection:

We initially selected the best models based on their evaluation metrics. From these, we chose the top three performers for further refinement. We then optimized their hyperparameters and segmented the data into low and high price categories for more targeted predictions.

Model Ensembling:

We evaluated the top three models separately, examining each one's performance individually to determine the best one for accurate predictions.

Testing on New Data:

In this project, we did not use completely new, unseen data for testing. Instead, we relied on cross-validation and split the existing data into training and testing sets to evaluate the models' performance.