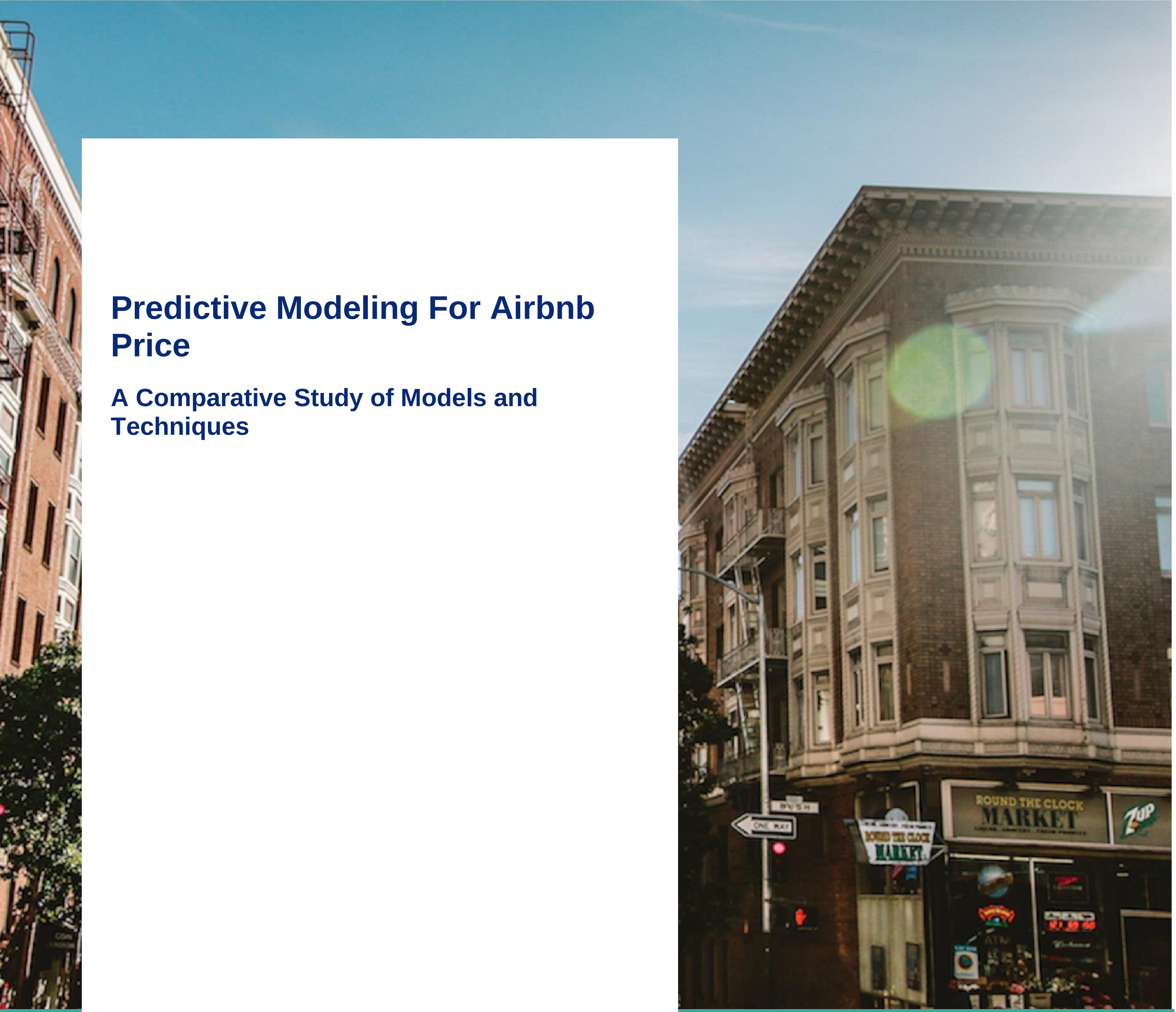


Predictive Modeling For Airbnb Price

A Comparative Study of Models and Techniques

12. 1. 2024

Springboard Data Science Career Track Capstone Project
Suk Won Choi

PREDICTIVE MODELING FOR AIRBNB LOG PRICE

ENHANCING ACCURACY THROUGH ENSEMBLE TECHNIQUE

The predictive model developed for Airbnb pricing offers valuable insights into optimizing pricing strategies and improving property values. The following key findings and strategies emerged from the analysis:

Performance Comparison and Final Model Selection

- The **Ensemble model (weighted average of XGBoost, Random Forest, Gradient Boosting, and MLP)** is the top performer, showing the lowest error rates (Test MAE: 0.2699, RMSE: 0.3734) and explaining 72.5% of variance (R^2: 0.7252).
- **XGBoost** follows closely, with strong predictive capabilities (R^2: 0.7228, MAE: 0.2708).

Feature Importance

- The most important features influencing pricing are **Room Type**, **Property Size** (bedrooms, accommodates), **Location** (distance to downtown, neighborhood), and **Amenities**.
- **Guest Experience** features like **Number of Reviews** and **Host Response Rate** also affect pricing, though less significantly.

MODEL UTILIZATION STRATEGIES

Practical Application of Findings

- **Optimal Pricing**: The model helps hosts set competitive prices by comparing their listings to similar properties, avoiding underpricing or overpricing.
- **Dynamic Pricing**: Separate modeling for peak and off-peak seasons allows for adjusting prices based on demand, maximizing revenue during high-demand periods and minimizing vacancies during low-demand periods.
- **Property Value Enhancement**: Hosts can increase property value by adding in-demand amenities, improving property descriptions, and emphasizing location and property condition, which can justify higher pricing and increase bookings.

I. Introduction

1. Problem Statement

For Airbnb, optimizing price for listings is essential to maximize revenue while ensuring high occupancy rates and customer satisfaction. Pricing strategies often rely on qualitative or overly simplistic models that fail to consider the diverse factors influencing daily rental prices, such as property attributes, amenities, location, and real-time market conditions.

This inefficiency can lead to suboptimal pricing recommendations, resulting in missed revenue opportunities, reduced competitiveness, and dissatisfied hosts who may seek alternative platforms or pricing solutions.

Airbnb faces the challenge of developing a scalable, data-driven pricing model that integrates a wide range of features and dynamically adjusts to market-specific and seasonal influences. Such a solution must not only deliver accurate price predictions but also provide clear, interpretable insights to empower hosts and build their trust in the platform's recommendations.

By addressing this challenge, Airbnb can solidify its position as a leading platform in the short-term rental market, enhance host retention, and improve overall profitability.

2. Steps To Solution

The primary objective of this project is to develop a sophisticated, data-driven pricing model for Airbnb listings that maximizes revenue potential while maintaining competitiveness and customer satisfaction. To achieve this, the project focuses on the following goals:

- **Comprehensive Feature Engineering**: Identify, engineer, and refine critical features, including property attributes, location, amenities, host characteristics, and seasonal or market-specific influences, to enhance the model's predictive capabilities.
- **Model Development**: Build a scalable and adaptive pricing model using advanced machine learning techniques, such as ensemble methods, to accurately predict log-transformed daily rental prices.
- **Performance Evaluation**: Evaluate the model using robust metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared, ensuring it provides reliable and accurate predictions for diverse listings.
- **Transparency and Interpretability**: Incorporate feature importance analysis and interpretable model outputs to offer actionable insights for hosts, empowering them to understand key pricing drivers and adjust their strategies accordingly.

By achieving these goals, the project aims to address the inefficiencies of static pricing models, providing Airbnb with a practical pricing solution. This solution will help strengthen host trust, maintain competitiveness in the short-term rental market, and drive overall profitability for the platform.

II. Dataset

The dataset for this project is sourced from the Kaggle Airbnb Price Dataset, which provides comprehensive details on Airbnb listings. It contains 74,111 entries with 29 columns, representing various features of Airbnb listings, which provide information about rental property listings, focusing on various attributes that may impact the log-transformed daily rental price (log_price). The features are categorized as follows:

- Target: log_price (log-transformed daily rent price).
- Numerical Features:
  Includes quantitative attributes such as the number of people the property can accommodate, geographic coordinates, and review scores.
- Categorical Features:
  Includes qualitative attributes like property type, room type, and host-related information.
- Textual Features:
  Includes descriptive data such as property descriptions and lists of amenities.
- Additional Features:
  Includes metadata like review dates, host starting date, property ID, and thumbnail URL.

Category	Name	Description
Target (Numerical)	log_price	Log-transformed daily rent price.
Numerical Features	accommodates	Number of people the property can accommodate.
	bathrooms	Number of bathrooms in the property.
	bedrooms	Number of bedrooms in the property.
	host_response_rate	Percentage of messages the host responds to.
	latitude	Latitude coordinates of the property.
	longitude	Longitude coordinates of the property.
	number_of_reviews	Total number of reviews received by the property.
	review_scores_rating	Average review rating of the property.
	beds	Number of beds in the property.
Categorical Features	property_type	Type of property (e.g., apartment, house, etc.).
	room_type	Type of room offered (e.g., entire home/apt, private room, shared room).
	bed_type	Type of bed provided (e.g., real bed, sofa bed, etc.).
	cancellation_policy	Cancellation policy of the property (e.g., flexible, strict).
	cleaning_fee	Indicates whether a cleaning fee is charged (True/False).
	city	City where the property is located.
	host_has_profile_pic	Indicates whether the host has a profile picture (True/False).
	host_identity_verified	Indicates whether the host’s identity has been verified (True/False).
	instant_bookable	Indicates whether the property can be booked instantly (True/False).
	neighbourhood	Neighborhood where the property is located.
	zipcode	ZIP code where the property is located.
Textual Features	description	Text description of the property.
	amenities	List of amenities offered by the property.
Additional Features	first_review	Date of the first review for the property.
	host_since	Date when the host first started hosting.
	last_review	Date of the last review for the property.
	name	Name of the property listing.
	id	Unique identifier for the property listing.
	thumbnail_url	URL of the property’s thumbnail image.

<Table 1. Dataset description>

III. Data Cleaning

The data cleaning process for this project involved several steps to ensure the dataset was clean, consistent, and ready for analysis. This included handling categorical features, managing missing values, and engineering new features to enhance predictive power.

1. **Conversion of Data Types**

- The date-related columns (first_review, last_review, host_since) were converted to a proper datetime format to enable calculations and comparisons.
- Categorical variables, such as property_type, room_type, and city, were explicitly converted into categorical data types to prepare them for encoding.

2. **Handling Missing Values**

Several columns had missing values that needed to be addressed:

- Numerical Features: Features such as bathrooms, bedrooms, and beds were imputed using K-Nearest Neighbors (KNN), leveraging related features (beds, bathrooms, bedrooms).
- Categorical Features: Columns like neighbourhood and zipcode had missing values imputed using classification models (Random Forest), utilizing geographic features (latitude and longitude).
- Other Features: The review_scores_rating and host_response_rate columns were filled with their respective median values.

3. **Removal of Irrelevant Columns**

Columns deemed non-informational, such as thumbnail_url, id, and name, were removed to streamline the analysis and avoid unnecessary complexity.

4. **Creation of New Features**

- A binary feature, is_new_listing, was added to identify listings without reviews or ratings, allowing the model to account for the absence of customer feedback.
- This feature was derived by checking if the host_response_rate, review_scores_rating, first_review, and last_review columns were all missing.

5. **Validation and Final Clean-up**

- The dataset was validated to ensure that all missing values had been addressed. Remaining missing values were filled using appropriate methods, and no duplicated rows were found.
- By the end of the wrangling process, all 74,111 rows and 26 features were ready for further analysis, with all missing data addressed and all features properly imputed.

The data wrangling phase ensured that the dataset was comprehensive and prepared for feature engineering and predictive modeling, allowing for robust and reliable analysis.

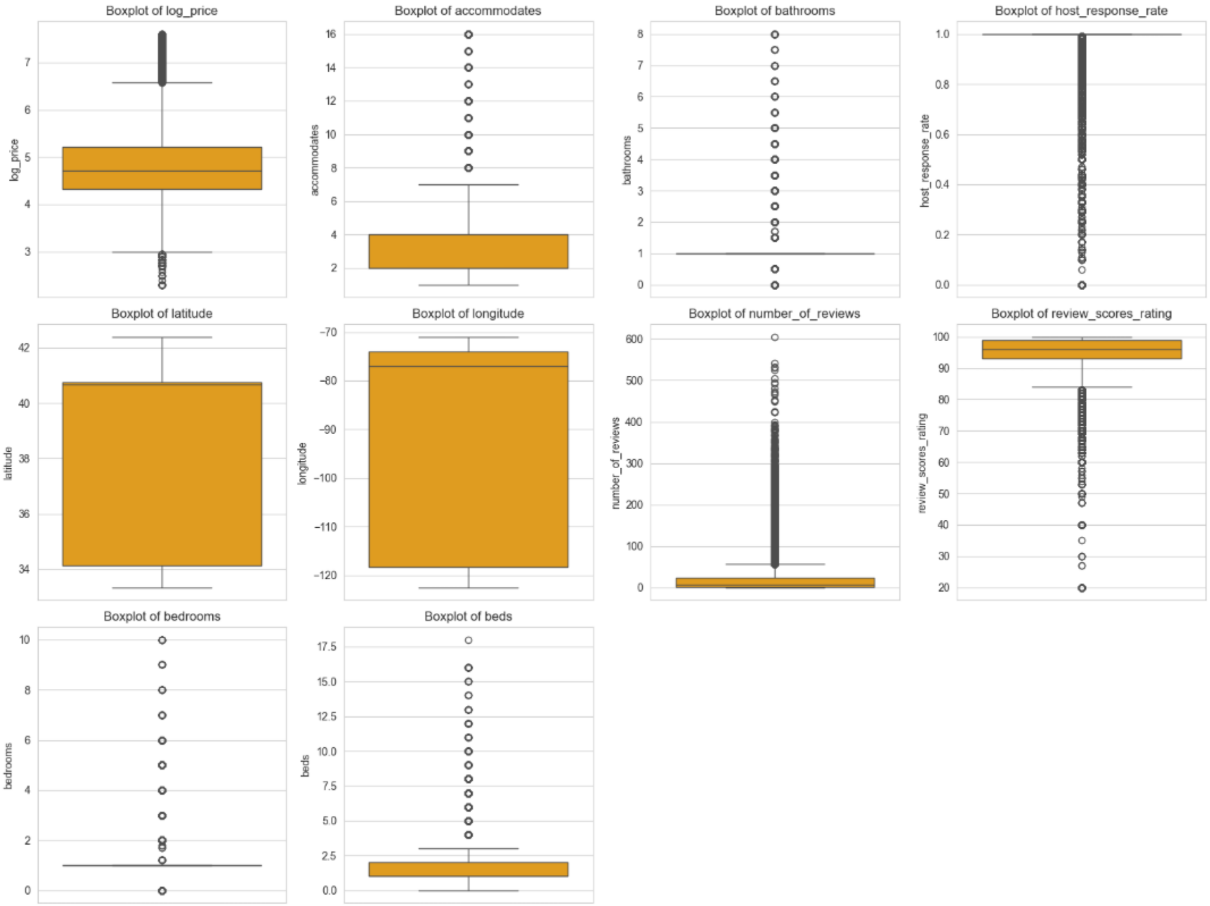
IV. Exploratory Data Analysis

1. Univariate Analysis

1.1 Distribution of Numerical Variables

The characteristics of the distribution of numerical variables, including the target variable, are as follows (Figure 1):

- The target variable, log_price, exhibits a distribution that is close to symmetric. Some unusually low prices were identified as potential data entry errors and were subsequently removed.
- Features expected to be closely related to rental prices, such as accommodates, bedrooms, and bathrooms, exhibit right-skewed distributions with observable outliers. However, as the outliers do not represent unrealistic values, they will remain in the dataset.
- Both latitude and longitude show left-skewed distributions without outliers. While they do not seem to be highly predictive of rental prices on their own, they will be retained for potential use in feature engineering.
- Other numerical variables also show left-skewed distributions, but since none of the outliers appear to be abnormal, they will be used as they are for now.

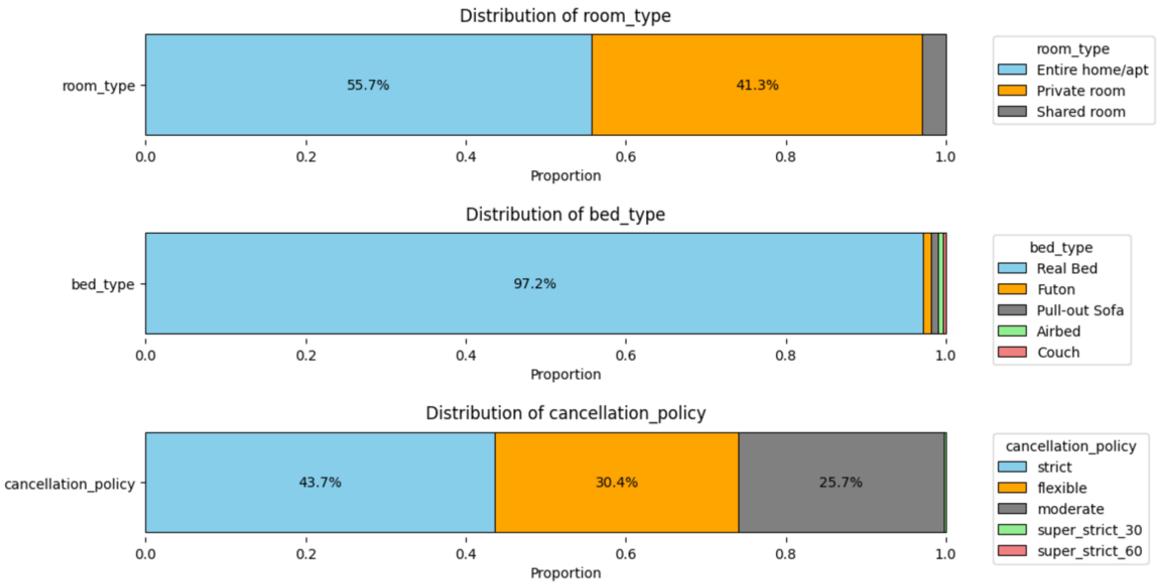


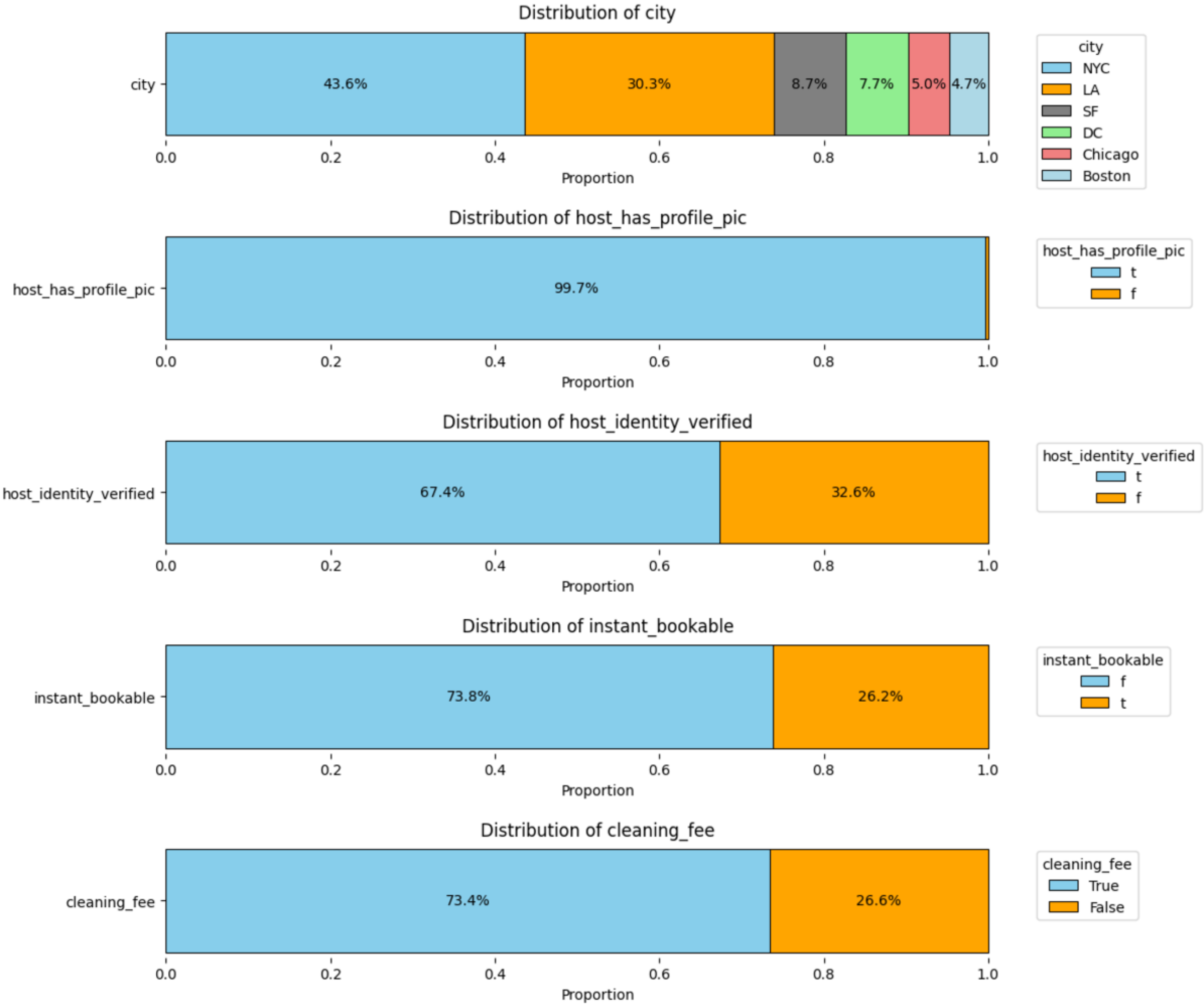
<Figure 1. Boxplots of Numerical Variables>

1.2 Distribution of Categorical Features

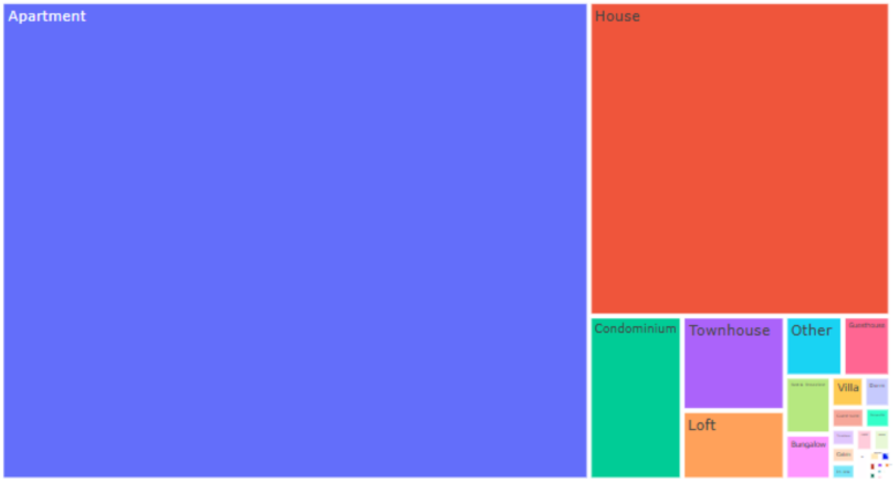
The characteristics of the categorical variables are as follows. Certain variables with an excessively high proportion in specific categories are expected to have low discriminative power for predicting the target variable, so these variables will be removed (Figure 2):

- Most room_type entries in the dataset are either "Entire home/apt" or "Private room."
- The bed_type feature shows that "Real Bed" is overwhelmingly the most common, accounting for nearly all listings. The remaining bed types, such as "Futon," "Pull-out Sofa," "Airbed," and "Couch," make up only 2.8% of the listings. Given this highly skewed distribution, we will remove the bed_type feature to simplify the dataset and focus on more balanced features.
- The cancellation_policy distribution shows that "Strict" is the most common policy, accounting for 43.7% of listings, followed by "Flexible" at 30.4% and "Moderate" at 25.7%.
- Regarding location, New York has the highest representation with 43.6%, followed by Los Angeles at 30.3%, San Francisco at 8.7%, Washington D.C. at 7.7%, Chicago at 5.0%, and Boston at 4.7%.
- Nearly all hosts (99.7%) have a profile picture, indicating very little variance in this feature. Since most hosts meet this condition, it can be considered saturated and unlikely to provide additional value for further analysis, so we will remove this feature.
- The ratio of identity-verified hosts to non-verified hosts is approximately 6.5 to 3.5.
- Additionally, 73.8% of listings are instant-bookable, while the remaining 26.2% require host approval. This feature could offer valuable insights into host preferences for booking flexibility.
- The property_type feature has more than 30 categories but is heavily skewed. For simplification, we will group the categories into "Apartment," "House," "Condo_Townhouse," and "Other," with 66.1% of listings being "Apartment" and 22.3% being "House" (Figure 3).




Distribution of city
city
43.6%
30.3%
8.7%
7.7%
5.0%
4.7%
0.0
0.2
0.4
0.6
0.8
1.0
Proportion
city
NYC
LA
SF
DC
Chicago
Boston
Distribution of host_has_profile_pic
host_has_profile_pic
99.7%
0.0
0.2
0.4
0.6
0.8
1.0
Proportion
host_has_profile_pic
t
f
Distribution of host_identity_verified
host_identity_verified
67.4%
32.6%
0.0
0.2
0.4
0.6
0.8
1.0
Proportion
host_identity_verified
t
f
Distribution of instant_bookable
instant_bookable
73.8%
26.2%
0.0
0.2
0.4
0.6
0.8
1.0
Proportion
instant_bookable
f
t
Distribution of cleaning_fee
cleaning_fee
73.4%
26.6%
0.0
0.2
0.4
0.6
0.8
1.0
Proportion
cleaning_fee
True
False

<Figure 2. Distribution of Categorical Features>

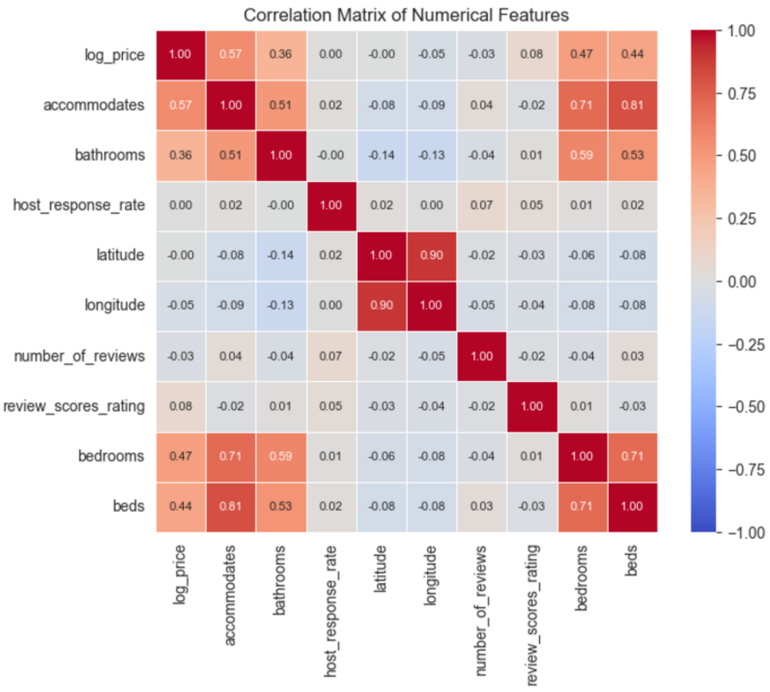

Apartment
House
Condominium
Townhouse
Other
Guesthouse
Villa
Loft
Bungalow
Cabin

<Figure 3. Distribution of Property Type>

2. Bivariate Analysis

Examining the correlation coefficients among the numerical variables reveals that features like accommodates, bedrooms, bathrooms, and beds exhibit a high correlation with the target variable, log_price. In contrast, other features do not show significant correlations. When analyzing correlations among the features, the coefficients between accommodates, bedrooms, and beds exceed 0.7, indicating potential multicollinearity.

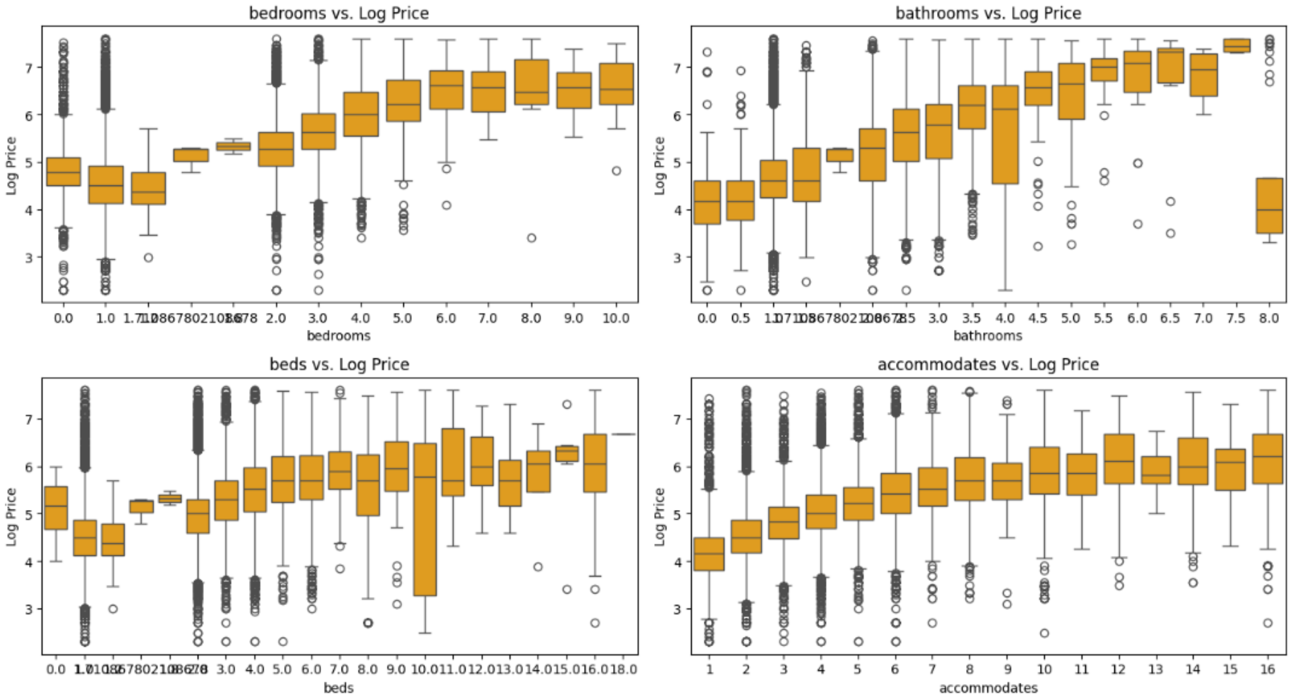
To address this, Lasso Regression will be employed as a baseline model in the modeling and prediction section, as it can mitigate multicollinearity by performing feature selection. Additionally, the performance of Lasso Regression will be compared with tree-based models, such as Random Forest and XGBoost, as well as MLP, which are less affected by multicollinearity.



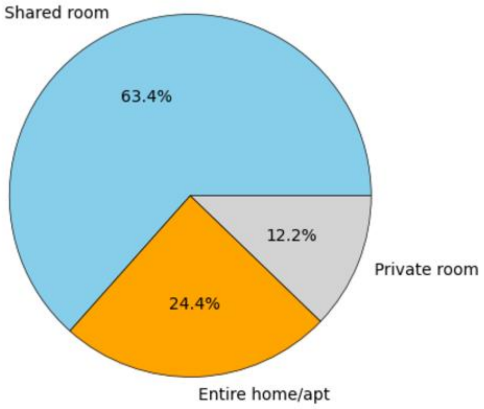
<Figure 4. Correlation Matrix of Numerical Variables>

2.1 Target vs Non-Continuous Numerical Features

Examining the relationship between discrete numerical features and log_price reveals a generally positive correlation (Figure 5). This trend is particularly evident in features such as bedrooms, bathrooms, and accommodates.
An interesting observation is that for bathrooms, the log_price drops sharply at the maximum value of 8. This sharp drop appears to be due to the high proportion of shared rooms, which tend to have lower prices, in these cases.
As will be discussed later, shared rooms are the least expensive room type, and for properties with 8 bathrooms, shared rooms account for 63.4% of the listings. This high proportion likely drives the average price down (Figure 6).



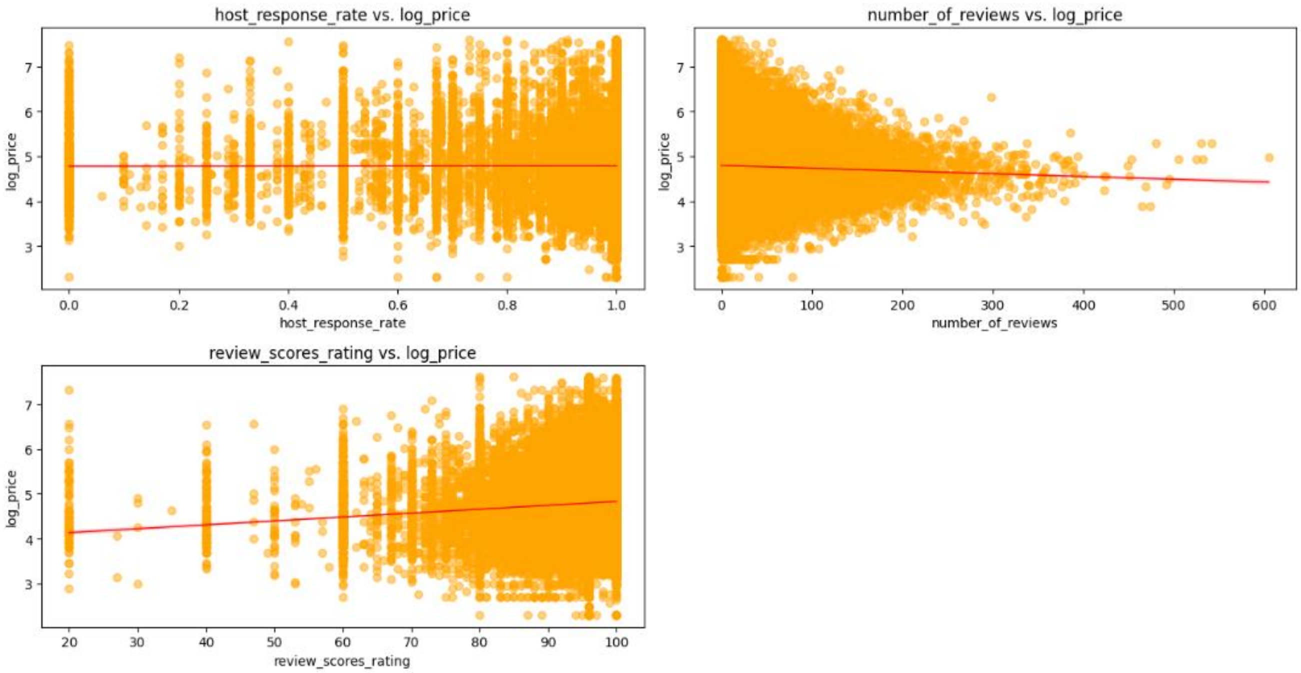
<Figure 5. Log Price vs Non-Continuous Numerical Features >



< Figure 6. Room Type Distribution for Properties with 8 Bathrooms >

2.2 Target vs Continuous Numerical Features

Among the continuous numerical features, review_scores_rating shows a positive correlation with the target variable, log_price. However, the relationship between other variables and the target appears less clear (Figure 7).



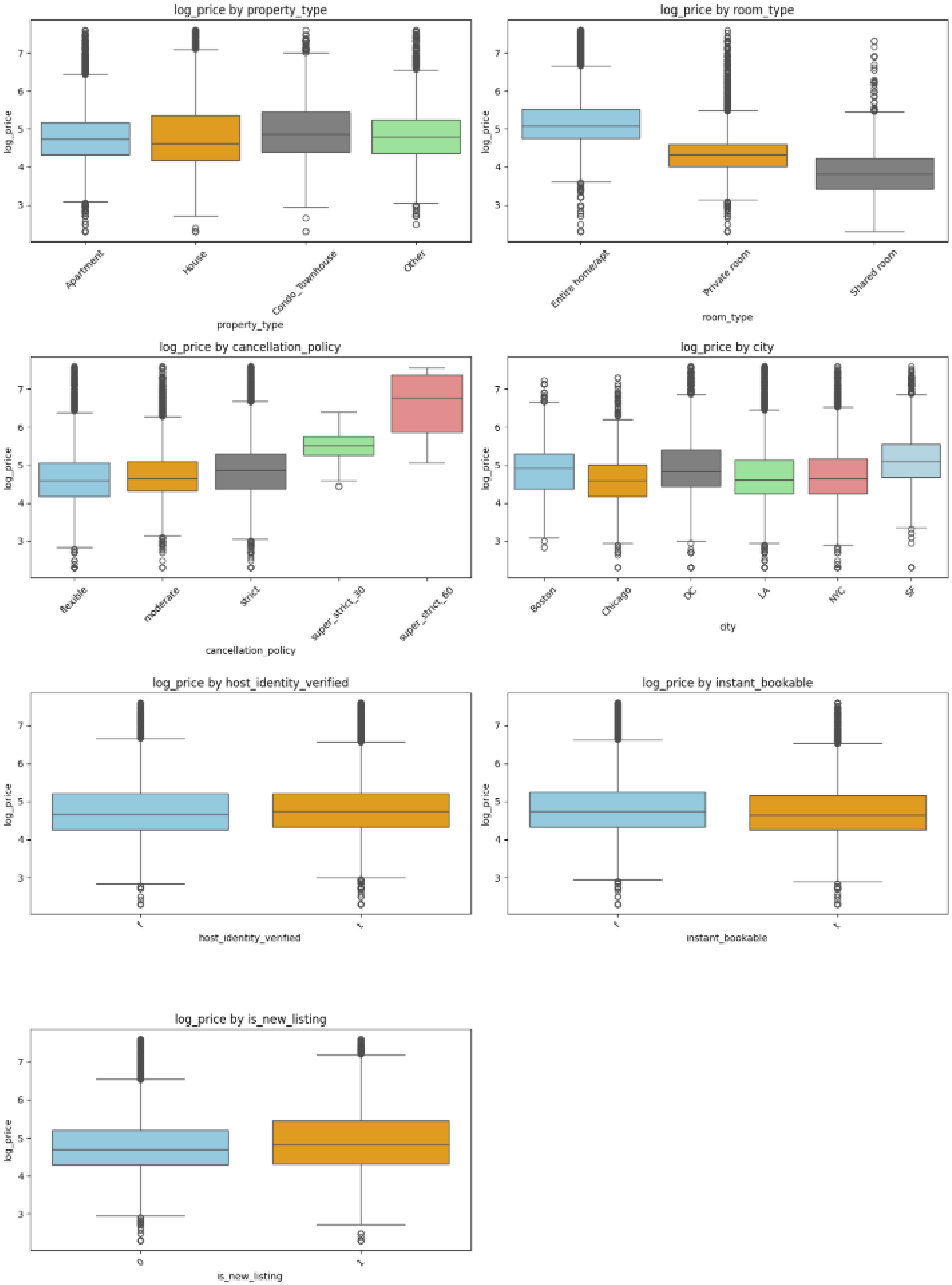
<Figure 7. Log Price vs Continuous Numerical Features >

2.3 Categorical features vs target variable

The relationship between categorical features and the target variable reveals interesting patterns in the distribution of log-transformed prices (Figure 8). Notably, certain features, such as Room Type, Cancellation Policy, City, and New Listing Status, exhibit clear distinctions in log_price distributions across their respective categories:

- Room Type demonstrates significant differentiation in pricing. Listings categorized as Entire home/apt command the highest prices, followed by Private room, with Shared room being the least expensive. The differences between these groups are striking and well-defined, reflecting the inherent value of privacy and exclusivity.
- Cancellation Policy also shows a notable trend. As the policy becomes more restrictive, prices tend to increase. Listings with the two most stringent cancellation policies are significantly more expensive than those with more lenient policies. This suggests that stricter cancellation policies may signal premium offerings or attract a different market segment willing to pay a premium for perceived quality or reliability.
- City is another critical determinant of pricing. Cities like Boston, Washington, D.C., and San Francisco stand out with substantially higher prices compared to other locations. These price differences likely reflect variations in demand, local market dynamics, and the general cost of living in these metropolitan areas.
- New Listing Status also has a discernible impact. Listings that are categorized as new tend to have higher prices compared to non-new listings, possibly indicating a premium placed on fresh and potentially less-utilized properties.

In contrast, other categorical features do not exhibit pronounced differences in the distribution of log_price across their categories. This suggests that while certain features significantly influence pricing, others may have a more muted or negligible effect on price differentiation.

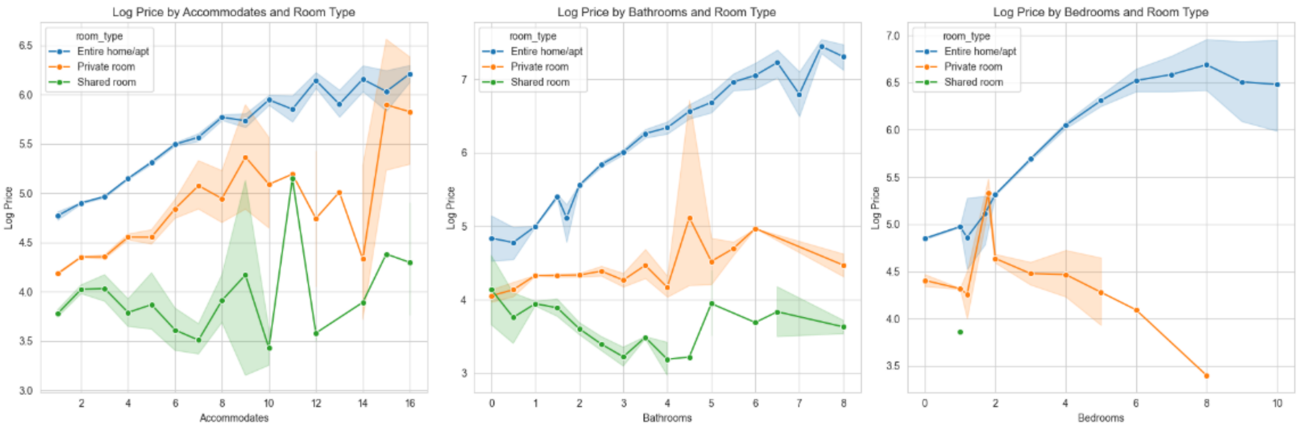


< Figure 8. Log Price vs Categorical Features >

2.4 Interaction Effects

The analysis of interaction effects highlights the complexities between different features and their relationships with the target variable, log price. As shown in Figure 9, certain variables like room_type, bedrooms, bathrooms, and accommodates interact differently with the target depending on the room type. These interactions underscore the importance of capturing how one feature's influence on price can be conditional on another feature's value. For example, the effect of bedrooms on price may vary depending on whether the listing is an entire home or a shared room.

While traditional linear regression models require the manual inclusion of interaction terms, which can be computationally expensive and prone to overfitting, tree-based models such as Random Forest, XGBoost, and Gradient Boosting naturally account for these interactions. These models excel at handling nonlinear relationships and interactions without the need for explicit specification of interaction terms, making them more suitable for this dataset. Their flexibility in modeling high-dimensional, complex relationships ensures that they can capture subtle patterns more effectively than linear models, likely resulting in superior predictive performance.



< Figure 9. Interaction between Room Type and Numerical Features >

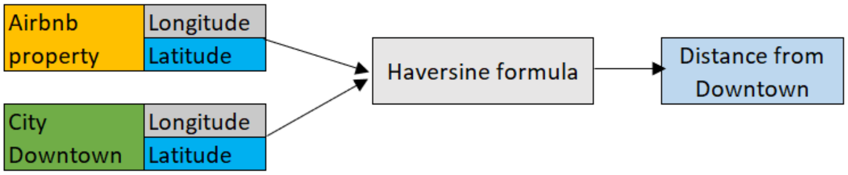
V. Feature Engineering

1. New Feature Generation

As noted, several features in the dataset are not in a format that can directly inform the prediction of log price. By engineering these features into more usable forms, we aim to improve model performance and better explain the variability in price.

1.1 Distance from downtown

Using the latitude and longitude coordinates of each property, we computed the distance to the downtown area of its respective city using the Haversine formula. This distance is expected to influence pricing, as properties closer to city centers often enjoy higher demand and revenue potential. The Haversine formula[1], which calculates the great-circle distance between two points on the Earth's surface, is ideal for this task as it accounts for the Earth's spherical shape, ensuring accurate distance measurements.



< Figure 10. New Feature: Distance from Downtown using Longitude and Latitude >

1.2 Hosting Duration

The host_since feature, which indicates when the host began offering their property, was converted into a numerical feature reflecting the number of years a host has been active. This "Hosting Duration" feature is expected to be a useful predictor, as more experienced hosts may have established reputations that influence their pricing strategies.

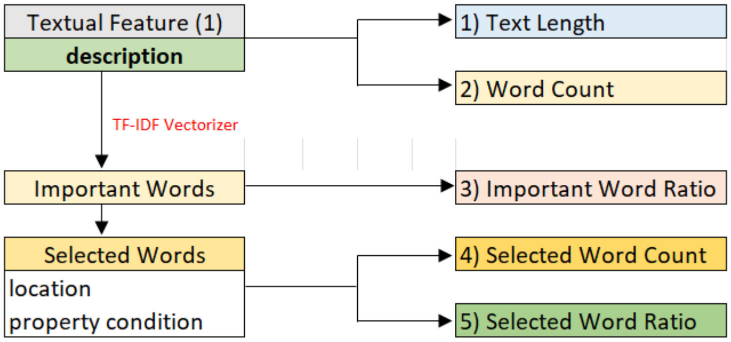
1.3 New Features from 'description' column: Features Generation From Text

The unstructured text data in the description column was transformed into useful features by quantifying its contents:

- Text Length and Word Count: We calculated the total length of the description and the word count. Longer descriptions may suggest higher-quality listings, which could be associated with higher prices.
- TF-IDF and Important Word Ratio: We applied a TF-IDF vectorizer to extract the 300 most important words in the descriptions. The ratio of these words to the total word count was then calculated, providing a measure of how much attention is given to impactful language that could potentially correlate with higher-priced listings.
- Selected Word Features: A set of curated words (e.g., "luxury," "downtown," "beach") was defined based on their potential relevance to pricing and demand. Two additional features were created:
    - Selected Word Count: The number of these key words in each description.
    - Selected Word Ratio: The proportion of selected words to the total word count in each description.

These features aim to capture the descriptive qualities of the listings, which may be indicative of their attractiveness and pricing.

[1] https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/



< Figure 11. New Features: Generating Numerical Features From Textual Feature>

1.4 New Features from 'amenities' column: Features Generation From String

The amenities column, which contains a string of different property amenities, was processed into a list format. We then generated two new features:

- Total Number of Amenities: The total count of amenities available in each property.
- K-means Clustered Amenities: We applied K-means clustering to group the amenities into five distinct clusters[2], identifying combinations of amenities that frequently appear together. This feature simplifies the dataset and helps the model better understand the relationship between amenities and pricing.

2. One-hot Encoding

Several categorical features, including property_type, room_type, cancellation_policy, cleaning_fee, city, host_identity_verified, instant_bookable, and amenities_cluster, were transformed using one-hot encoding.

This technique converts categorical variables into a series of binary columns, enabling the model to interpret these categorical variables in a numerical format, improving the model's ability to detect relationships in the data.

3. Binary Encoding[3]

The features zipcode and neighbourhood have 669 and 619 unique categories, respectively. One-hot encoding these features would drastically increase the dimensionality of the dataset, leading to computational inefficiency and a higher risk of overfitting due to the large number of additional features.

[2] **The elbow method analysis determined that the optimal number of clusters (k) is 5.**
[3] **https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/**

To address this, binary encoding was applied as a more efficient alternative. Binary encoding[4] transforms each category into a series of binary digits, significantly reducing the number of resulting columns compared to one-hot encoding. For the zipcode feature with 669 unique categories, binary encoding generated 10 columns ($\mathbf{log_2(669) = 9.39}$), while the neighbourhood feature with 619 unique categories generated 10 columns ($\mathbf{log_2(619) = 9.28}$).

This approach preserves the uniqueness of each category while maintaining a more compact and manageable feature set, making it a practical solution for high-cardinality categorical variables.

VI. Modeling and Performance Evaluation

After completing the feature engineering process, 57 features were selected to predict the target variable. The next step involves modeling, where various algorithms will be used to train the model on the training set, followed by hyperparameter tuning to optimize performance.

1. Modeling and Prediction Process

The algorithms selected for this task include Lasso Regression, Random Forest, XGBoost, Gradient Boosting (GBM), and MLP (Multilayer Perceptron). Here's why these algorithms were chosen:

- Lasso Regression: Lasso (Least Absolute Shrinkage and Selection Operator) is beneficial when dealing with multicollinearity or when some features might not contribute significantly to the prediction. It performs feature selection by shrinking some coefficients to zero. While Lasso is less capable of capturing interactions between features, it will serve as a baseline model for comparison against more sophisticated models.

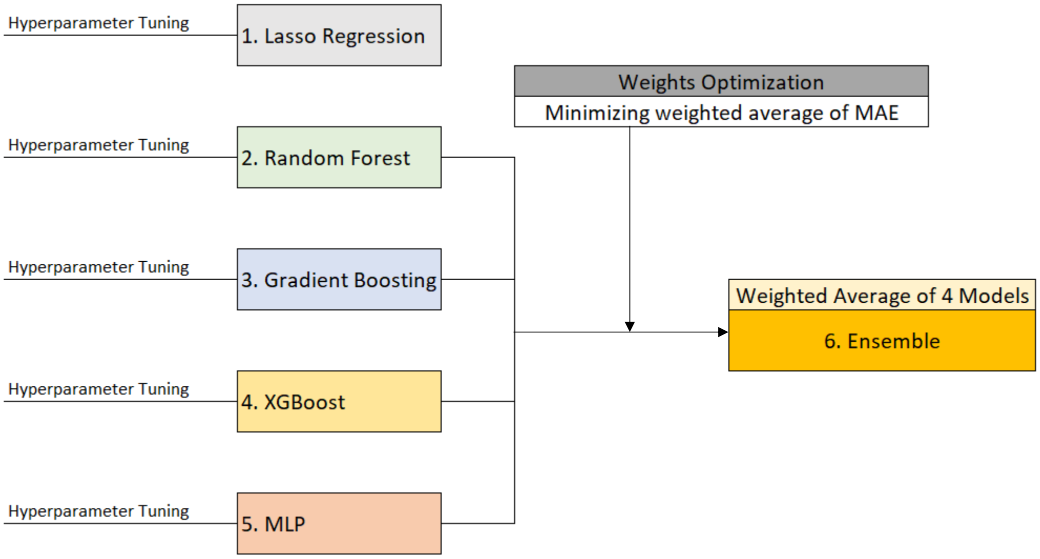
- Random Forest: Random Forest is a robust ensemble method based on decision trees. It excels at handling non-linear relationships and is resistant to overfitting, particularly when the dataset is large or complex. Random Forest can also estimate feature importance, making it valuable for understanding which features are driving predictions.

- XGBoost (Extreme Gradient Boosting): XGBoost is an efficient and powerful gradient boosting algorithm. It often delivers high performance in machine learning tasks, particularly with complex or large datasets. XGBoost is effective in capturing both linear and non-linear patterns in the data, making it suitable for this problem.

- Gradient Boosting (GBM): Gradient Boosting builds an ensemble of decision trees sequentially, with each tree learning from the errors of its predecessor. This approach has been shown to achieve high predictive accuracy in regression tasks, and it is particularly useful for problems where relationships between features are complex.

[4] https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

- MLP (Multilayer Perceptron): MLP is a type of artificial neural network (ANN) that can model highly non-linear relationships through its layers of neurons. It is capable of capturing intricate patterns in the data, making it useful for problems where traditional models may fall short.



< Figure 15. Modeling Process>

2. Performance Comparison and Final Model Selection

The performance of each model was evaluated using three metrics: MAE, RMSE, and R-squared.

From the <Figure 14>, we can observe the following:

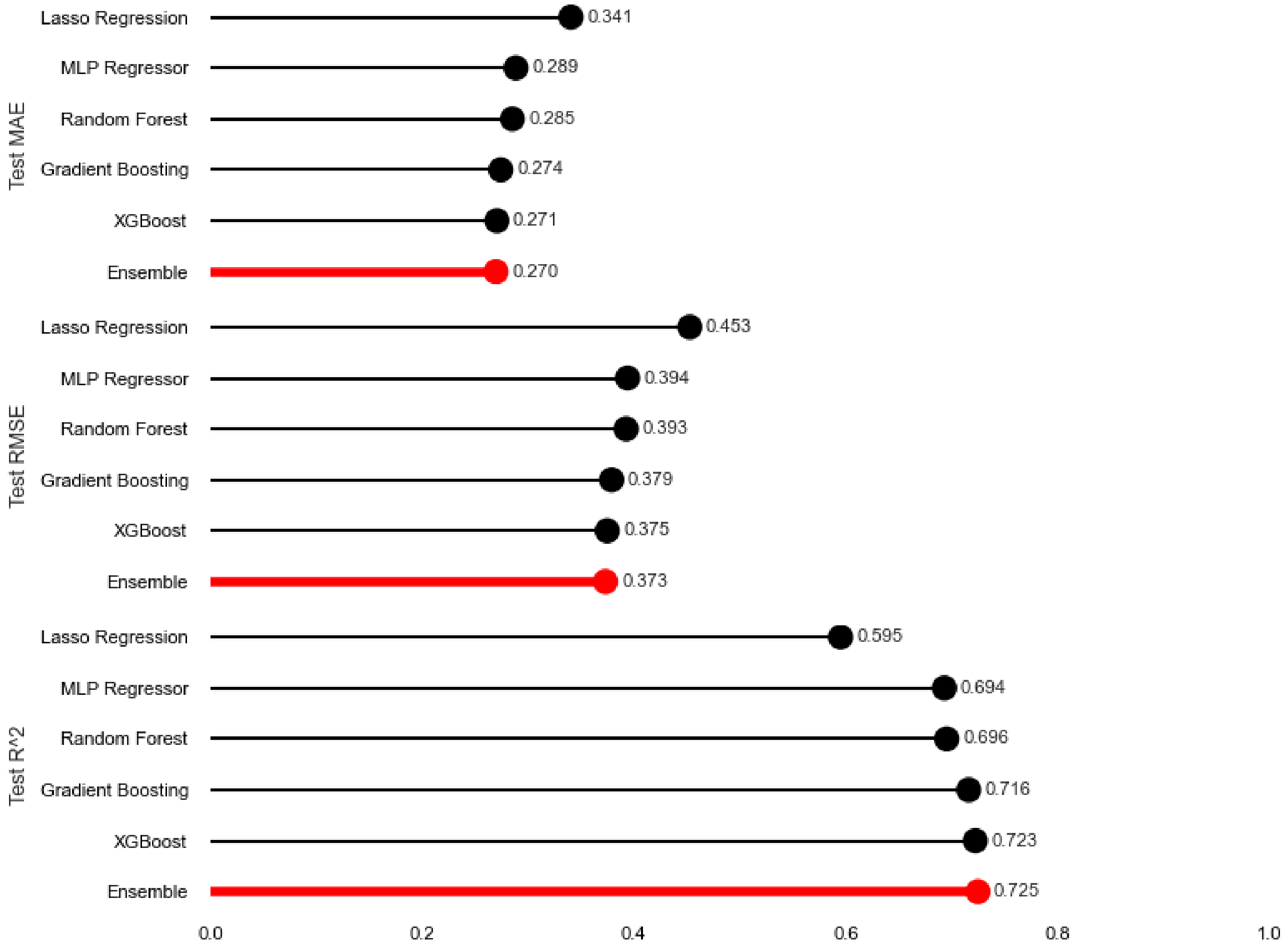
- Overall, the Ensemble model emerges as the top performer, with the lowest Test MAE (0.2699) and Test RMSE (0.3734), indicating the smallest error in predictions. Its Test R^2 of 0.7252 also suggests it explains a significant portion (72.5%) of the variance in the target variable. This makes the Ensemble method the most reliable and effective model for this dataset, combining the strengths of multiple models to provide superior predictive power.

- XGBoost follows closely, with a Test R^2 of 0.7228 and strong performance in both error metrics (MAE: 0.2708, RMSE: 0.3750). XGBoost has consistently shown excellent performance across various machine learning tasks, and its results here confirm its capability in capturing complex patterns in the data.

- Random Forest and Gradient Boosting are competitive, with Random Forest achieving an R^2 of 0.6957, and Gradient Boosting slightly outperforming with an R^2 of 0.7165. The MAE and RMSE values are similar for both models, with Random Forest slightly leading in error metrics (MAE: 0.2854, RMSE: 0.3929) and Gradient Boosting having slightly better explanatory power with its R^2 but slightly higher error metrics (MAE: 0.2745, RMSE: 0.3793). These models perform similarly, with no major difference in overall effectiveness.

- The MLP Regressor, while providing reasonable performance, falls behind the other models, with an R² of 0.6935, and higher error values (MAE: 0.2888, RMSE: 0.3943). However, compared to Random Forest and Gradient Boosting, the MLP's performance is not significantly worse, showing that its ability to model complex relationships is somewhat limited compared to the tree-based methods.
- Finally, Lasso Regression is the weakest performer, with a Test R² of only 0.5955 and relatively high error metrics (MAE: 0.3407, RMSE: 0.4530). This model's inability to capture non-linear relationships and interactions between features likely contributes to its poor performance. Since Lasso Regression is a linear model, it does not account for the potential interaction effects between features or the non-linear relationships that may exist between the features and the target variable. This limitation is a significant reason why Lasso does not perform well on this dataset, as the target variable likely involves more complex patterns that Lasso cannot capture.

In conclusion, the Ensemble model is the best overall performer, closely followed by XGBoost. Both models show the best combination of low error rates and high R² values, indicating they are the most reliable for this dataset. Random Forest and Gradient Boosting are also strong contenders with similar performance, while the MLP Regressor performs adequately but is outperformed by the tree-based models. Lasso Regression, on the other hand, struggles due to its inability to model non-linear relationships and feature interactions, which makes it the least effective model in this comparison.



< Figure 14. Performance Metrics Comparison>

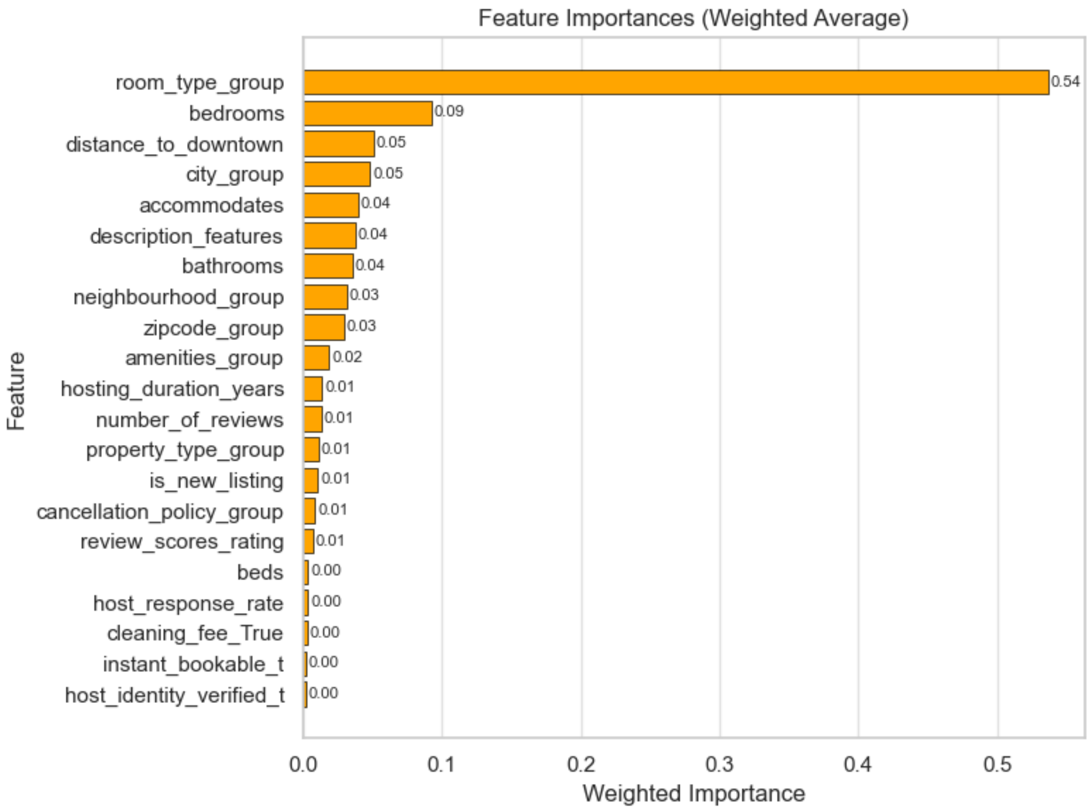
3. Feature Importance

The best model is an ensemble model that averages the predictions of XGBoost, Random Forest, Gradient Boosting (GBM), and MLP using weighted averages. However, deriving feature importance from this ensemble model is challenging because MLP (a neural network) does not inherently provide feature importance scores, and combining different models' importance metrics is not straightforward due to differences in calculation methods and scales.

Therefore, feature importances were calculated by normalizing the importances from the remaining three models (XGBoost, Random Forest, and GBM) and combining them using a weighted average based on each model's contribution. In addition, to facilitate interpretation, the importances of encoded features were aggregated to calculate the importances of the original features.

The results, shown in Figure 14, highlight the following characteristics:

- Room Type is the most important feature, playing a dominant role in determining the value or attractiveness of a listing.
- Features related to the size of the property and basic facilities, such as bedrooms and accommodates, are also crucial. These features describe the capacity and scale of the property, which strongly influences the decision-making process for guests.
- Location-related features, including distance to downtown, city, zipcode, and neighborhood, have a notable influence. These factors indicate how desirable or accessible the property is based on its geographical location, affecting its popularity and appeal.
- Features derived from the property description and those grouped under Amenities, such as facilities offered, also have meaningful explanatory power. These attributes describe the quality and attractiveness of the listing in terms of the amenities and features guests are looking for.
- Guest experience-related features, such as Number of Reviews and Host Response Rate, though having a smaller weight, still have a significant impact. Positive reviews and prompt responses contribute to a property’s perceived reliability and trustworthiness.
- Other features, such as Beds, Host Response Rate, Cleaning Fee, and Instant Booking, have less explanatory power and appear to have a minimal impact on the model’s prediction. These features do not contribute as strongly to determining the value or desirability of the listing.



< Figure 15. Ensemble Model’s Feature Importance: Weighted Average of XGBoost, GBM, and Random Forest>

VII. Practical Application of Findings

The findings from this predictive modeling approach can enhance decision-making processes in the hospitality and property rental industries. By utilizing the model's results, businesses can make data-driven decisions that optimize various aspects of their operations. Below are the key practical applications of the findings:

1. Enhancing Booking Rates through Optimal Pricing

The pricing prediction model evaluates the competitiveness of a specific listing in the market and suggests an appropriate price range based on similar listings' characteristics. This can be highly valuable for property managers and hosts in maximizing their booking rates while maintaining profitability.

- **Accurate Price Setting**: The model helps avoid underpricing or overpricing by comparing the listing with similar properties. By using historical data and similar listings’ performance, it ensures that the price is competitive and attractive to potential customers.

2. Dynamic Pricing Implementation

While the current model is based on data from a specific time point, separately modeling the data for peak and off-peak seasons can enable dynamic pricing that reflects changes over time, optimizing revenue or reducing vacancy rates to cover fixed costs.

By modeling peak and off-peak data separately:

- **For peak seasons**, where demand is high, the model can increase prices **to maximize revenue**. For example, during specific seasons or events, prices can be automatically adjusted upwards to match high demand, capturing bookings at higher rates.
- **For off-peak seasons**, where demand is lower, the model can decrease prices to reduce vacancy rates and **minimize losses from fixed costs.** By lowering prices, the model can still attract bookings during low-demand periods, ensuring occupancy and covering essential costs.

This approach allows for dynamic pricing strategies tailored to the characteristics of each season, optimizing revenue and occupancy while responding to demand fluctuations effectively.

3. Guide to enhance the property’s value

The company can provide hosts with a guide to improve the value of their properties, helping them optimize features under their control to positively impact pricing.

- **Amenities**: Hosts can add in-demand amenities to increase the property’s perceived value and justify higher pricing.
- **Property Description**: By offering more detailed descriptions that highlight unique features or nearby attractions, hosts can make their listings more appealing.
- **Location and Property Condition Emphasis**: Highlighting the property's proximity to popular areas, such as the city center or tourist attractions, as well as its condition (e.g., recently remodeled), can enhance its appeal and justify higher pricing.

These enhancements allow hosts to strategically improve their listings, leading to better pricing and increased bookings.