Application of AI in Business

Final Project Milestone

Airbnb Data Analysis and Machine Learning

Group 7

Vibha Hegde

Date: 12/7/2024

Table of Content

[Click Each Topic to Read More]

Executive Summary

In this project, we utilize Airbnb listing data to provide insights for hosts to increase their profit margins by leveraging the insights and patterns from the existing data.

Problem Statement

Based on the available historical listing data in the city of Boston for the year 2024, we aim to identify the features influencing the price of Airbnb listings. Our objectives for this project are as follows:

a) Identify the factors affecting the price of an Airbnb listing in Boston.
b) Identify the trends and the impact of these factors on the price of an Airbnb listing
c) Predict/forecast future prices of an Airbnb listing based on the information of the current factors using various machine learning models.
d) Stretch goal – based on textual information such as the amenities, determine the price of the Airbnb listing.

Methodology

To achieve the outlined objectives, we obtain the data from the Airbnb website. We clean the data to ensure that there are no outliers or missing variables. Later the data is formatted using feature engineering methods to ensure compliance for the various models we leverage to make predictions. Some of the models we used in this project are:

1. Linear Regression
2. RandForest Regressor
3. XGBoost Regressor
4. Neural Network

In terms of textual data, we leverage the property type and amenities to analyze the requirements that are of high importance to the customers. Again, we leverage Linear Regression and Decision tree regressors to analyze the impact of these textual features on the price of the Airbnb listing.

Findings

Some of the key findings from the various analyses are as follows:

1. Increasing availability to 365 days justified a higher listing price to the customers.
2. Security, safety, and self-check-in were the highest-rated amenities.
3. Customers preferred Entire units over shared units or private rooms within a unit.
4. Customers also preferred rental units over luxury stays.

Conclusion and Next Steps

From this project, we recommend that Airbnb hosts increase their availability to gather more interest and ensure that adequate security measures are in place. The hosts can increase their traffic

by advertising security features on their listings. Furthermore, they can obtain higher prices for listing entire units over listing private rooms within a unit.

1. Project Introduction

In this project, we analyze the Airbnb data to provide business recommendations utilizing machine learning concepts.

1.1 Problem Statement and Project Objective

In this project, our primary objective is to determine the leading factors that affect the price of Airbnb listings in Boston and identify the impact of these features on the price. We focus on the trends of the factors using various machine learning models to provide insights regarding the pricing strategy that can increase customer retention, increasing the value for hosts listing on Airbnb. Thus, our objectives can be summarized as:

e) Identify the factors affecting the price of an Airbnb listing in Boston.
f) Identify the trends and the impact of these factors on the price of an Airbnb listing
g) Predict/forecast future prices of an Airbnb listing based on the information of the current factors using various machine learning models.
h) Stretch goal – based on textual information such as the amenities, determine the price of the Airbnb listing.

2. Data Collection

For this project, we obtained the Airbnb listings data from the website: https://insideairbnb.com/get-the-data/

We specifically searched for the data regarding the listings available within and nearby Boston and loaded this data into our workbook. After inspecting the data, we had about 4325 listings with 75 different features. Of these, 45 features were numerical in nature and 30 were textual. The data contained information regarding the listing, whether the listing had reviews, features available with the listing, type of listing, details regarding the host, price of the listing and booking availability.

To view more regarding the data, please visit: https://data.insideairbnb.com/united-states/ma/boston/2024-06-22/data/listings.csv.gz

3. Data Cleaning

The raw data obtained was very large and had to be cleaned to be made available into machine readable formats. Some of the methods used for data cleaning are described below.

3.1 Identification of Various Data within the Dataset

Initially we started exploring the dataset by splitting the data into numerical and textual. We further split the data into numerical and categorical. We obtained around 45 numerical data and 30 textual data.

3.2 Removal of duplicate of irrelevant data

Some of the features within the dataset had duplicate information such as whether a listing was available for more than 30 days, 60 days, 90 days etc. Similarly, for the initial exploratory analysis, we did not need id information or URL information as these formats cannot be used to obtain correlations between features. Hence, we proceeded to remove these features from the dataset.

3.3 Conversion of percentage values to numerical data

Some of the features such as "host response rate" and "host conversion rate" were provided in percentage values as string. We need numerical values for these features and thus converted the percentages into numbers. Similarly, we had price data in currency format, which we converted into numerical value.

3.4 Binary Encoding

Features such as "host_is_superhost", "host_has_profile_pic", "host_identity_verified", "has_availability" and "instant_booking" had true or false values (Boolean values) in the string format. This is more useful to us as categorical values with 0 and 1 and hence we utilized binary encoding to format these values.

3.5 Handling Missing Data Values

The dataset does contain many missing values and "NaN" values in various fields. Our analysis will be flawed if we don't handle it. Hence, we have used some methods mentioned below for different fields to handle the missing data.

3.5.1 Omission:

Our analysis will not work if there are missing values in price data. Hence, we remove any of the observation that had no price value.

3.5.2 Filling Median values:

Specific features such as the number of bedrooms, accommodations, and number of bathrooms can have the median values filled in for those that have no missing values.

3.6 Removing Outliers

We identified and removed the top 1% of values in the price column as outliers. These extreme values could represent luxury listings or rare cases that would skew the analysis.

3.7 Removing Features with no values

In the dataset there were a couple of features which had no data. Since these data weren't adding any value, we removed these columns. Example: Neighbourhood cleaned

3.8 One hot encoding

Some of the categorical values had to be converted into multiple values. We created the dummies using One hot encoding. Examples of the variables are: Room Type and Host Response Time.

Post data cleaning methods we had **33 numerical variables and 4 textual features.**

4. Exploratory Data Analysis

Due to the large number of features, the feature analysis would be difficult. In addition to that it adds a dimensional overhead for any of the models. We wanted to reduce the dimensionality by finding the best features. Later, once we have a list of best features, we dive deeper into each of the feature statistics, by identifying the scale of the feature, minimum and maximum values etc. We find the correlation of these features with the price value using a correlation matrix and analyze the distributions of each of the variables.

4.1 Feature Selection

4.1.1 SelectkBest using Chi-squared

Initially, we did want to rate the features based on statistical measures such as the Chi-squared analysis. To rate the various features and to identify the top 10 features out of the 37 features, we ran the "select k Best" function in sklearn.

We obtained the following list of best features as compares to the price values: 'host_acceptance_rate', 'host_total_listings_count', 'accommodates', 'beds', 'minimum_nights', 'maximum_nights', 'availability_365', 'number_of_reviews', 'reviews_per_month', 'room_type_Hotel room'

4.1.2 RandomForest Classifier for decision tree based feature importance analysis

We were also curious about how different the results would be if we used a machine learning model to identify the top features. For this we utilized the RandonForest decision tree classifier to rate the importance of the various features against price values.

The list of features we obtained from RandomForest Classifer are:

Table 1: *Top 10 features based on the importance of the features.*

availability_365	0.162271
reviews_per_month	0.064704
number_of_reviews	0.058084
host_total_listings_count	0.055377

review_scores_value	0.050643
review_scores_location	0.048586
review_scores_rating	0.046334
review_scores_cleanliness	0.046211
review_scores_accuracy	0.044899
maximum_nights	0.043164

While some of the features were rated different, there was a significant overlap of many features. Based on the results we obtained we continue to use the following 10 features for the analysis:

Table 2: *List of selected features through feature analysis*

host_acceptance_rate
host_total_listings_count
accommodates
beds
instant_bookable
review_scores_cleanliness
maximum_nights
availability_365
number_of_reviews
room_type_Hotel room

Note: For the feature selection we have not used the textual features as we still need more time to understand how to analyze these. This is part of our stretch goal for this project.

4.2 Feature description

When we ran the feature description to obtain the maximum, minimum and mean values of each of the feature we obtained the following:

Table 3 – *Descriptive statistics of various features*

	price	host_acceptance_rate	host_total_listings_count	accommodates	beds	instant_bookable	review_scores_cleanliness	maximum_nights	availability_365	number_of_reviews	room_type_Hotel room
count	3507	3507	3507	3507	3507	3507	3507	3507	3507	3507	3507
mean	225.222127	91.250927	611.892501	3.12432	1.6424	0.38922	4.768252	563.545195	216.268035	51.967208	0.00656
std	167.559534	19.045995	1486.931035	2.09894	1.3647	0.48764	0.370699	421.900027	111.073954	98.265359	0.08073
min	25	0	1	1	0	0	0	3	0	0	0
25%	110	93	5	2	1	0	4.74	365	117	1	0
50%	189	98	25	2	1	0	4.86	365	233	11	0
75%	280	100	135	4	2	1	4.95	1125	318	57.5	0
max	1129	100	4975	16	20	1	5	1125	365	994	1

It is clear that some of the features are in a different scale and some features are also boolean in nature. During modelling and feature engineering we might have to modify it further to obtain accurate results.

4.2.1 Correlation Matrix

Accommodates and beds show high correlation (~0.8), indicating potential multi-collinearity. Both features have been retained based on the feature selection process. Their individual impact will be further evaluated during modeling, and if needed, we may prioritize accommodates due to its stronger relevance to pricing.

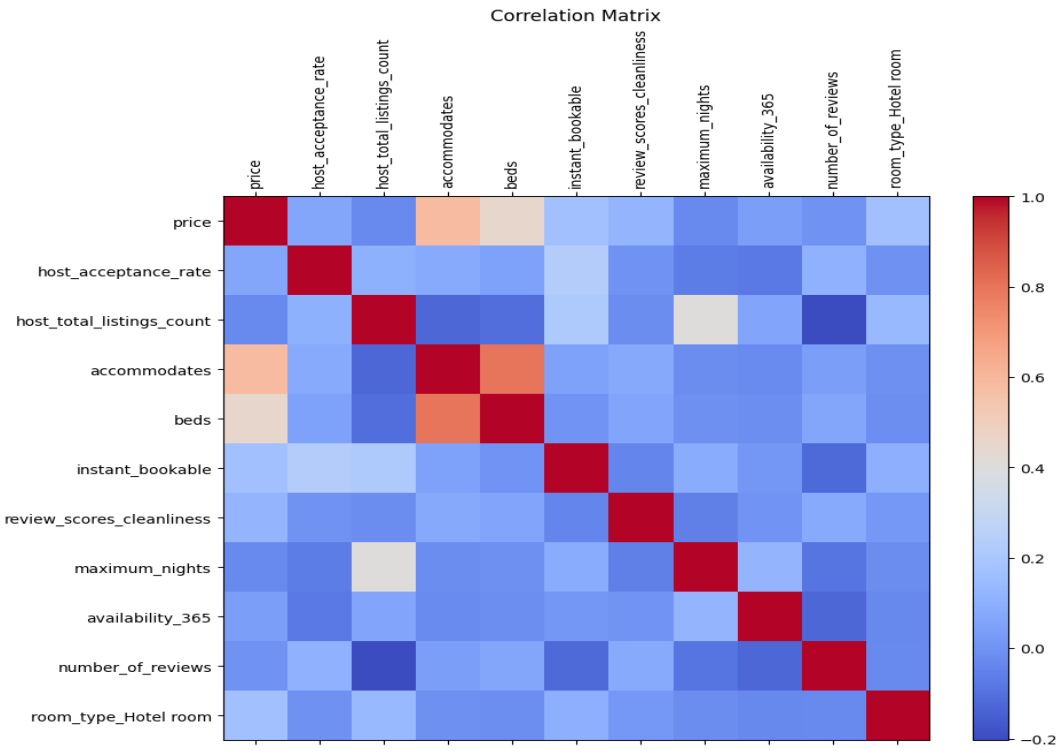


Figure 1 – *Correlation Matrix between price and other features.*

Furthermore, some of the features may be eliminated during modelling due to its lack of correlation to price and this helps reduce the dimensionality of the model.

4.2.2 Histograms of Features

Right-Skewed Features: Several features, including price, accommodates, and number_of_reviews, are positively skewed. This means that for models such as linear regression, this can skew the results. We might look into logistic transformations of some of these features during modelling to reduce the impact of the skew.

Accommodation distribution and the number of reviews is not a continuous distribution as these are quantified in whole numbers. Meanwhile, price distribution is continuous as prices can be floating values.

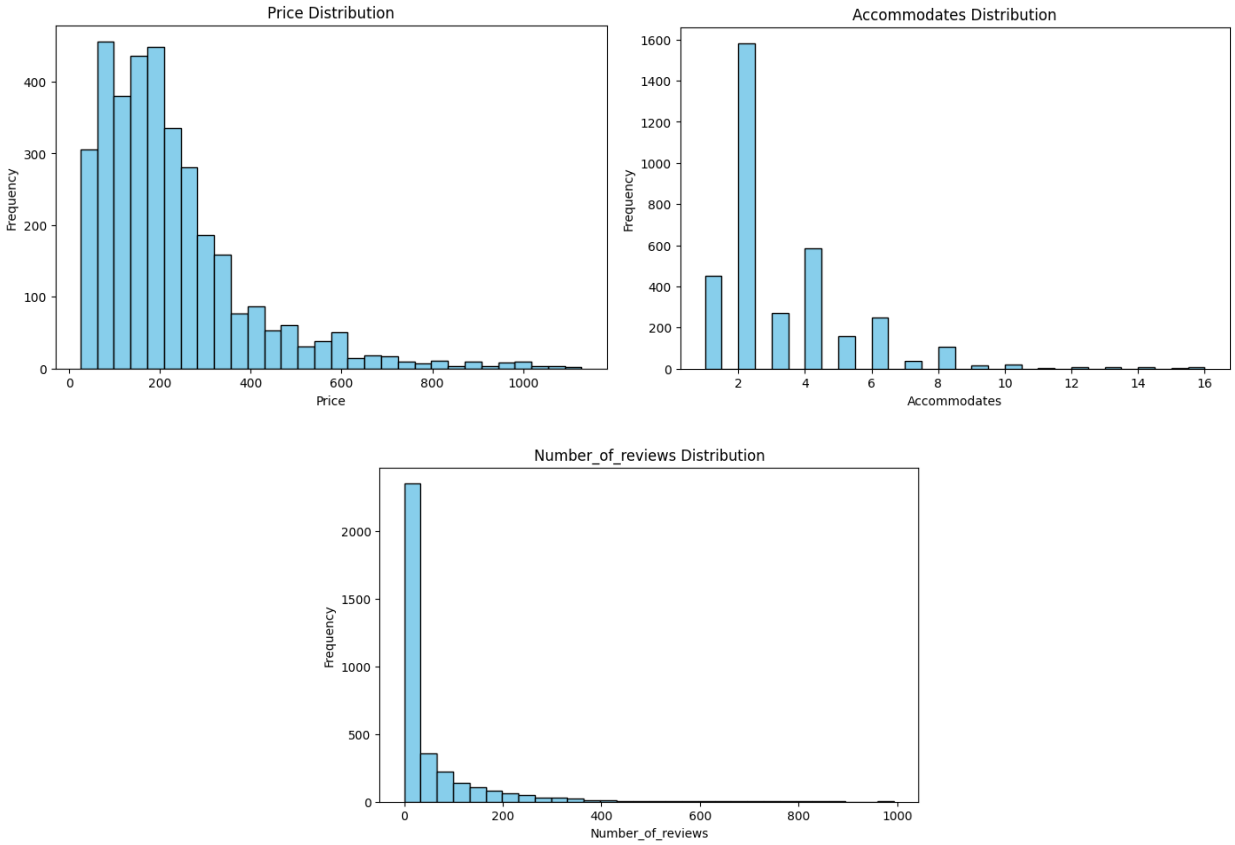


Figure 2 - *a) Histogram of Price Distribution, b) Histogram of Accommodation c) Histogram of Number of Reviews.*

Potential Outliers: High values in price, host_total_listings_count, and maximum_nights will be reviewed to ensure they don't distort model results.

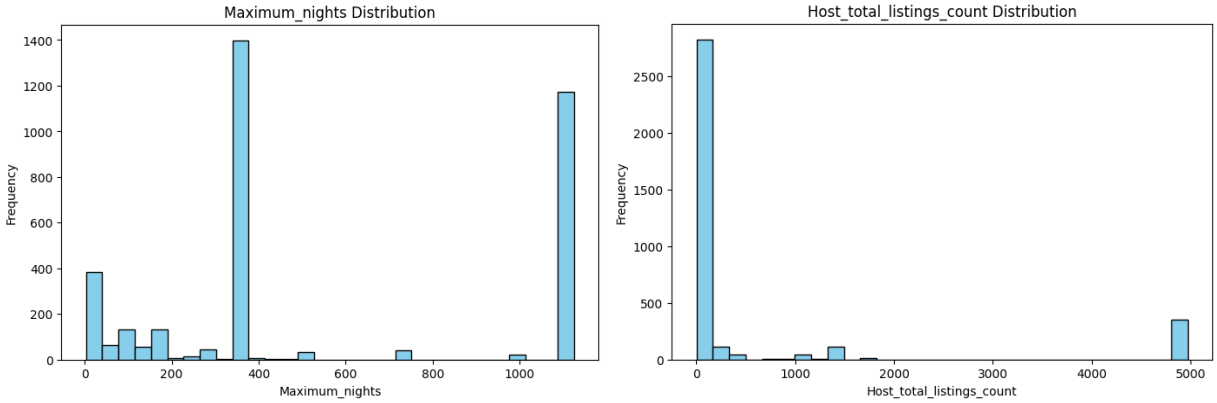


Figure 3 – *a) Histogram of Maximum Nights distribution and b) Host total listings count distribution*

In addition to the outliers, “Maximum nights” and “total host listings” do not have continuous distribution and thus, these features might have to be rescaled to reduce the impact of the distribution on the model results.

Imbalanced Features: room_type_Hotel room is heavily imbalanced, with most listings not being hotel rooms. While this is expected for a categorical variable, we cannot use this feature as it is without feature engineering. This may cause models to favor the majority class, but it will be addressed during modeling

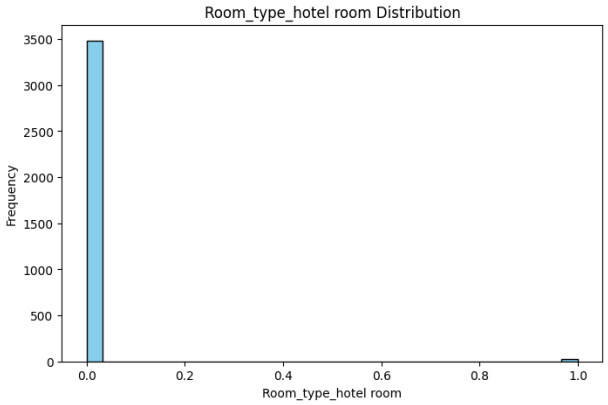


Figure 4 – *a) Hotel Room type frequency distribution*

4.2.3 Scatter plots of Features

Strong Positive Relationship: Accommodates and beds show strong positive trends with price, confirming their importance in price prediction.

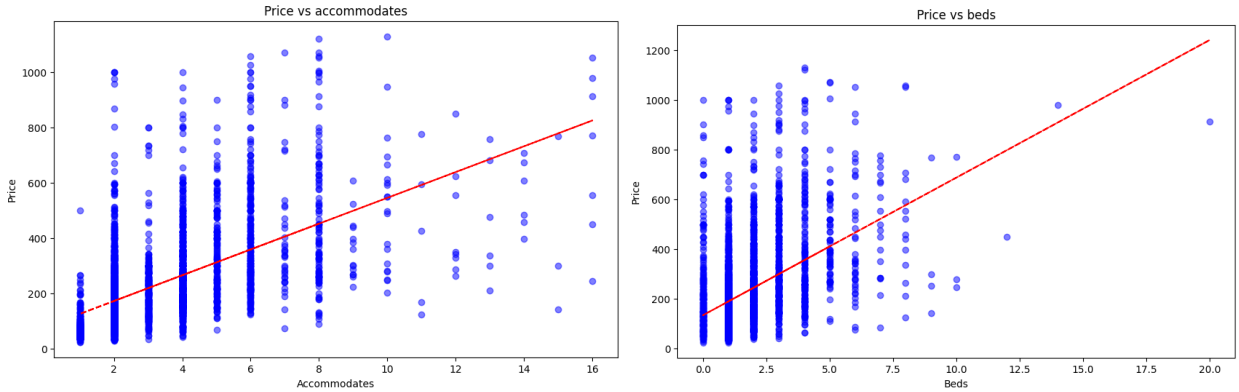


Figure 5 – *a) Price vs accommodation and b) Price vs beds scatter plots and trends.*

Weak or Insignificant Relationships: Host acceptance rate, instant bookable, and availability_365 show weak correlations with price. This might be due to how the categorical variable – instant bookable is distributed. So, we need to further analyze the impact using a different analysis method. As for the Host acceptance rate, the change in the rate might not have a significant impact on the price of the listing.

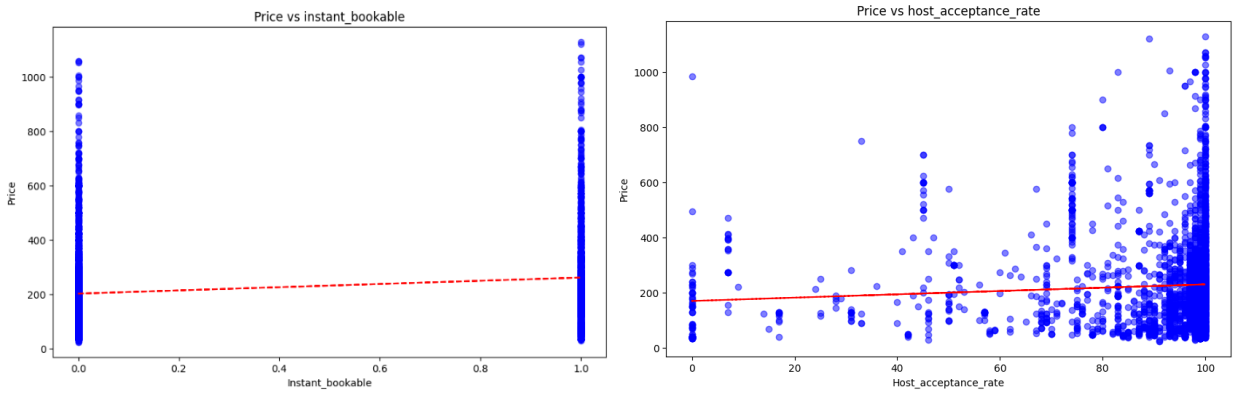


Figure 6 – *a) Price vs Instant Bookable and b) Price vs Host Acceptance Rate*

Flat Trends: Features like availability_365 do not show meaningful relationships with price, indicating that there is no change in the price of the listing even with a change in the availability of the listing.

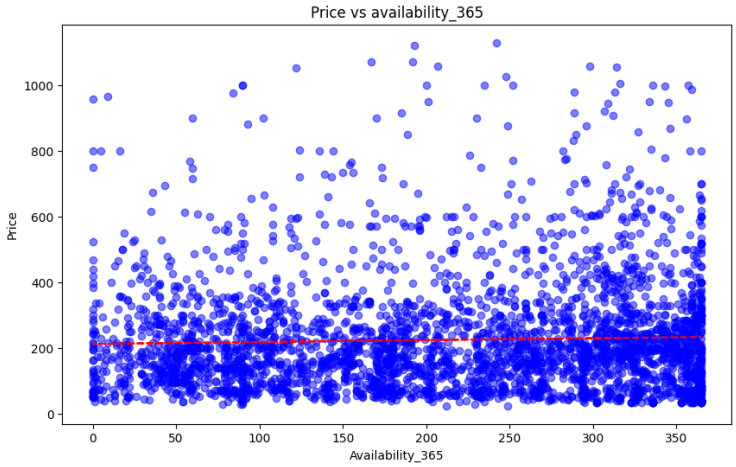


Figure 7 – *Price vs Availability*

5. Models for Price Prediction

We used four different models to analyze the impact of features on the price and model this relationship.

5.1 Linear Regression

We used linear regression as the base model to understand the relationship between various features. With no modifications and using only numerical values, the model yielded an absolute error of 89.122 and an RMSE value of 127.78, which is high for the price variable. The R-squared value is about 0.3532 (or 35.32%), which is a poor estimation of the model.

Here is the table with model coefficients matched to their respective feature names:

Table 4 – *Feature Names and Linear Regression Coefficients*

Feature Name	**Model Coefficient (Weight)**
host_acceptance_rate	-2.26841768
host_total_listings_count	0.11752013
accommodates	99.43430868
beds	-2.97205974
instant_bookable	23.82422153
review_scores_cleanliness	14.8401906
maximum_nights	-4.76535674
availability_365	9.15193357
number_of_reviews	-0.11900401
room_type_Hotel room	29.64066832

Based on this table, it is evident that larger accommodation leads to higher listing prices as well as the price would be higher if the room is a hotel room compared to luxury accommodation. Similarly, people value exclusivity, where the prices decrease if the host acceptance rate is higher. The prices are also reduced when there is a cap on the maximum number of nights the customer can stay.

While the model provides these insights, it is not perfect. From the model evaluation, the r_squared value is 0.35, which means the prediction variability is 35.32% demonstrated by the linear regression model presented. The RMSE is also quite high, indicating that there can be a difference of *127.78 between the predicted price and the actual price.*

5.2 RandomForest Regressor

Since the r_squared value of Linear regression was so low, it was evident that the model was not linear. Hence the next logical step was to use a Random Forest Regressor for determining the price prediction model. For the RandomForest Regressor, we used 3 different estimators, however, we found that the most optimal was when the estimator was set to 100. From the model evaluation, the r_squared value is 0.55.77, which means the prediction variability is 55.77% demonstrated by the random forest regressor model presented. The RMSE is also quite high indicating that there can be a difference of 105.66 between the predicted price and the actual price. The Mean Absolute error resulted in 64.74.

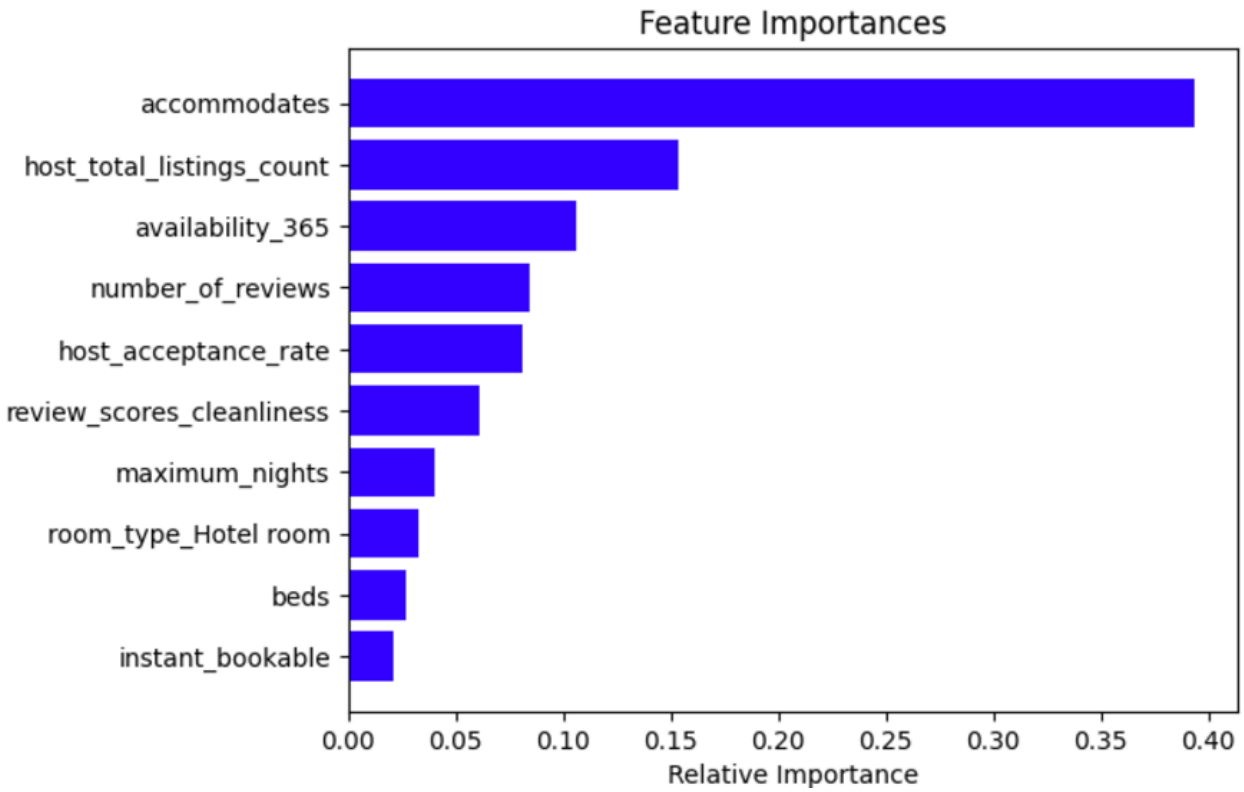


Figure 8 – *Relative Importance of various features in price prediction using RandomForest Regressor*

5.3 XGBoost Regressor

XG Boost is a gradient descent algorithm used for large datasets for higher efficiency and it uses a non-linear approach for estimating the relationship between various features. While we need to scale the features for Linear regression, XG Boost does not require data normalization and it still tries to capture statistical relationships between features unlike decision trees and random forest regressors.

To optimize the algorithm, we hyper-tune the parameters for XG Boost using GridSearchCV which provides us with the best feature set. For the Boston listings dataset, the best features were indicated as shown below:

```
Best parameters found:  {'alpha': 1, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
Best score found:  0.578536839973634
```

Furthermore, we used a cross-validation score to rate the performance of these estimators with different parts of the dataset to determine whether the model overfits or underfits. The overall scores for this estimator is comparable and better than the other models.

```
[0.69737895 0.6000573  0.67740777 0.42265867 0.62632692 0.54288559
 0.60403833 0.63928812 0.47442645 0.61566726]
```

From the model evaluation, the RMSE is lower than that of linear regression and the RandomForest Regressor. With an error of 103.58 and an absolute error of 63.87, this model has the best performance so far. However, with an r-squared value of 57.85, the model still fails to explain the volatility within the predictions.

Based on the predictions, we derived the importance scores of the top five features that impacted the price predictions. Based on the frequency of a feature, there were different features that made it to the top five, while the relevance ("gains") had a different set of features. However, on average, the features had a common pattern.

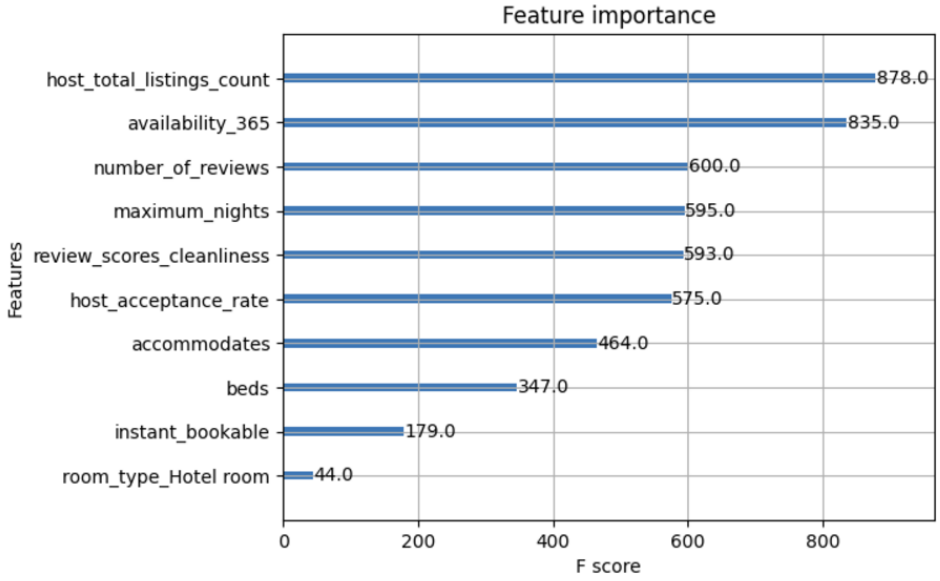


Figure 9 – *Feature importance score based on the frequency of the feature presence*

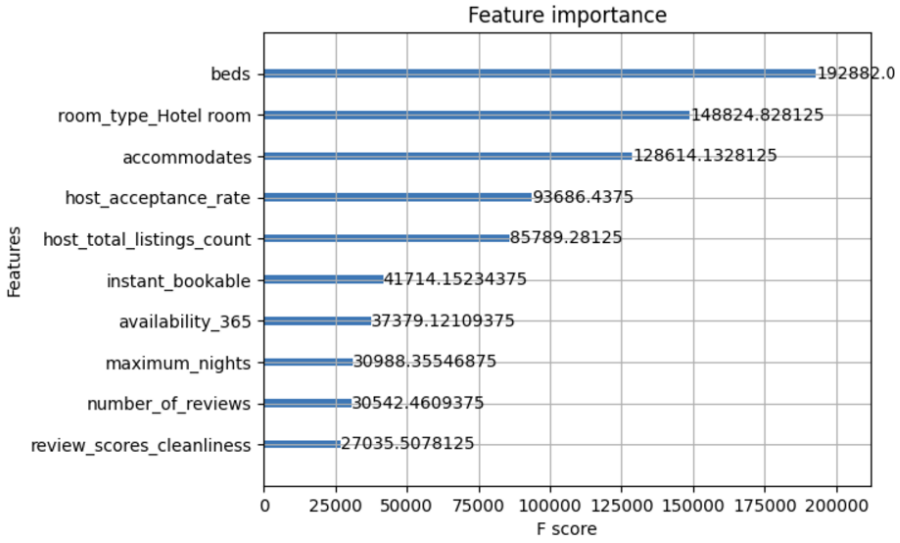


Figure 10 – *Feature Importance score based on the relevance or gains of the feature*

The feature importance with gains matches aligns with the initial insights we obtained in the feature selection.

5.4 Neural Network

Since we established that the features are related in a non-linear model structure, we continued to explore the price prediction using a Neural Network regressor model. In this model, we used 2 hidden layers to reduce the layer sizes with a learning rate of 0.001 and batch sizes [124,64]. We set a 50-epoch run for optimal results.

We did multiple runs with different layers and sizes and based on experimentation; we settled on the above parameters.

The training loss function was set to mean absolute error since that was the value used for comparison across the different models and the loss function looks as shown in the following figure for training and validation datasets.

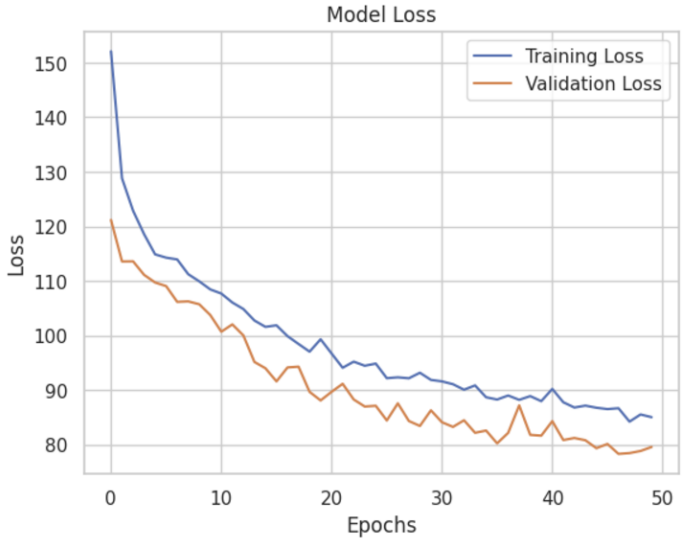


Figure 11 – *Training and Validation loss function for Neural Network*

The Training loss is much higher than the validation loss, this indicates that the model is not overfit, and they both have a gradual descent as expected of a model that is able to capture the patterns in the dataset.

Similar to other models, when we fetched the importance of different features from the price prediction neural network model, the results were as follows:

Table 5 – *Feature Importance mapping from Neural Network Prediction Model*

Feature	**Importance**
host_total_listings_count	39.81145713784416
maximum_nights	33.22829024486053
accommodates	22.378425073216103

Feature	Importance
availability_365	3.8832549157645304
number_of_reviews	2.7630236144758724
beds	2.108869373288928
instant_bookable	0.5769250190495768
host_acceptance_rate	0.29898300496941205
room_type_Hotel room	0.06905433893882958
review_scores_cleanliness	0.050063799996664216

The Predicted values from the Neural Network looked as follows:

Table 6 – *Actual vs Predicted prices using Neural Network Price prediction model*

Index	Actual	Predicted
0	212.0	169.599564
1	196.0	237.464478
2	196.0	218.871475
3	87.0	124.509048
4	186.0	169.008881

A pictorial representation of this model's actual and predicted values is shown below:

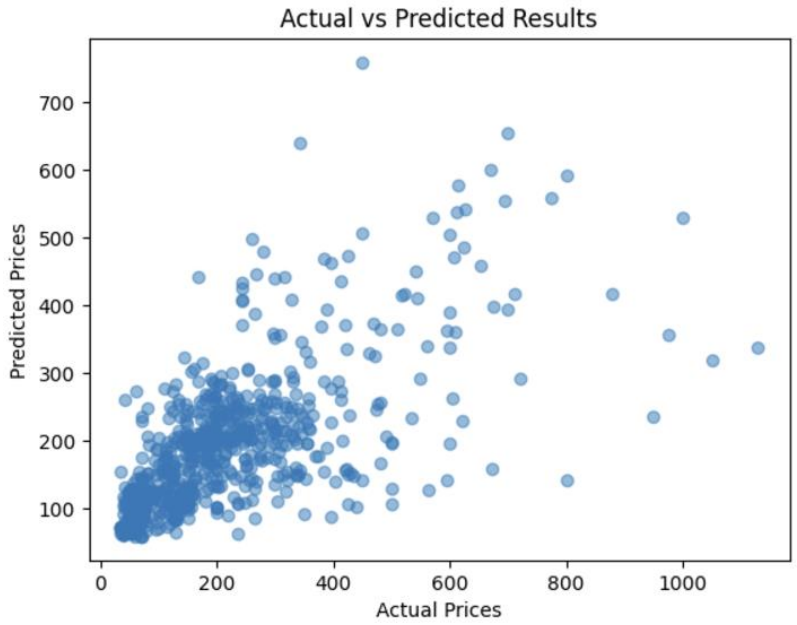


Figure 12 – *Actual vs Predicted Values from the Neural Network Model*

5.5 Comparison Between the Four Models

All four models increase in complexity starting from Linear regression to Neural Network. The performance also increases from linear regression to XGBoost Regressor and later decreases with the neural network.

The comparison of error rates (Mean Absolute error, MAE, and Root Mean Squared Error, RMSE) is used to determine the performance of each of the models. The lower the RMSE and MAE, the better the model. In addition to this, we also used R_squared to determine how much of the model volatility is explained by the model parameters.

From the following graph, it is clearly evident that the MAE and RMSE are lowest for the XGBoost Regressor and the highest for the simple Linear Regression model. The Neural Network and Random Forest Regressor came close to better performance; however, we did not want to stack the models as this would remove the explaining capacity of different feature influences for the model.

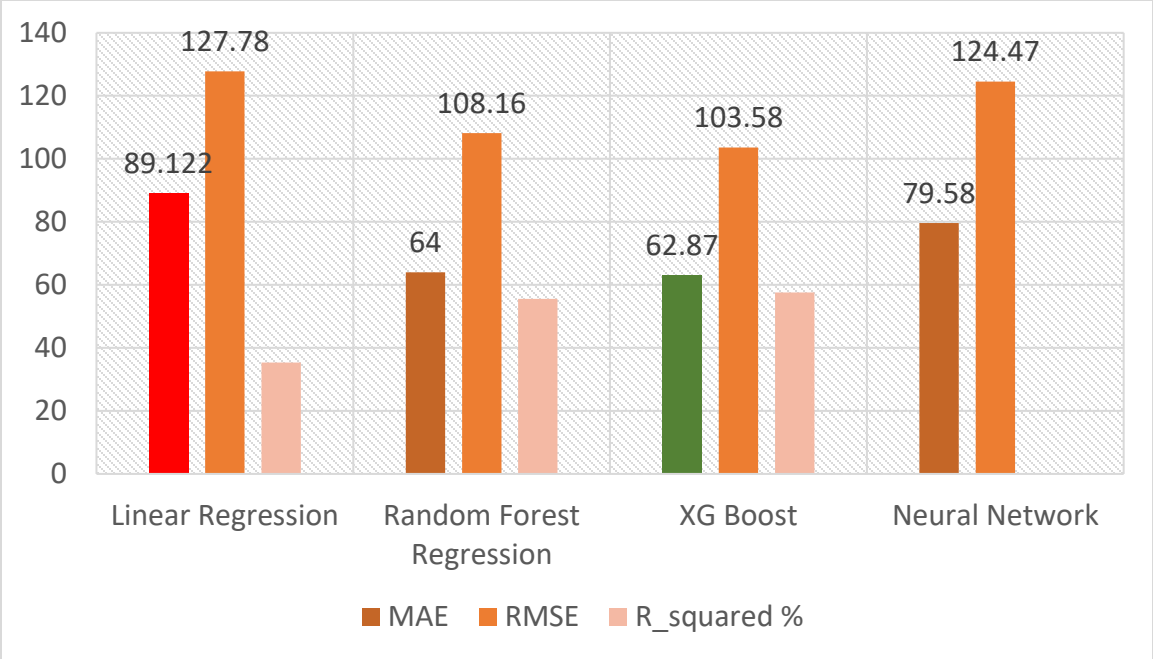


Figure 13 – *RMSE and MAE comparison between the four models*

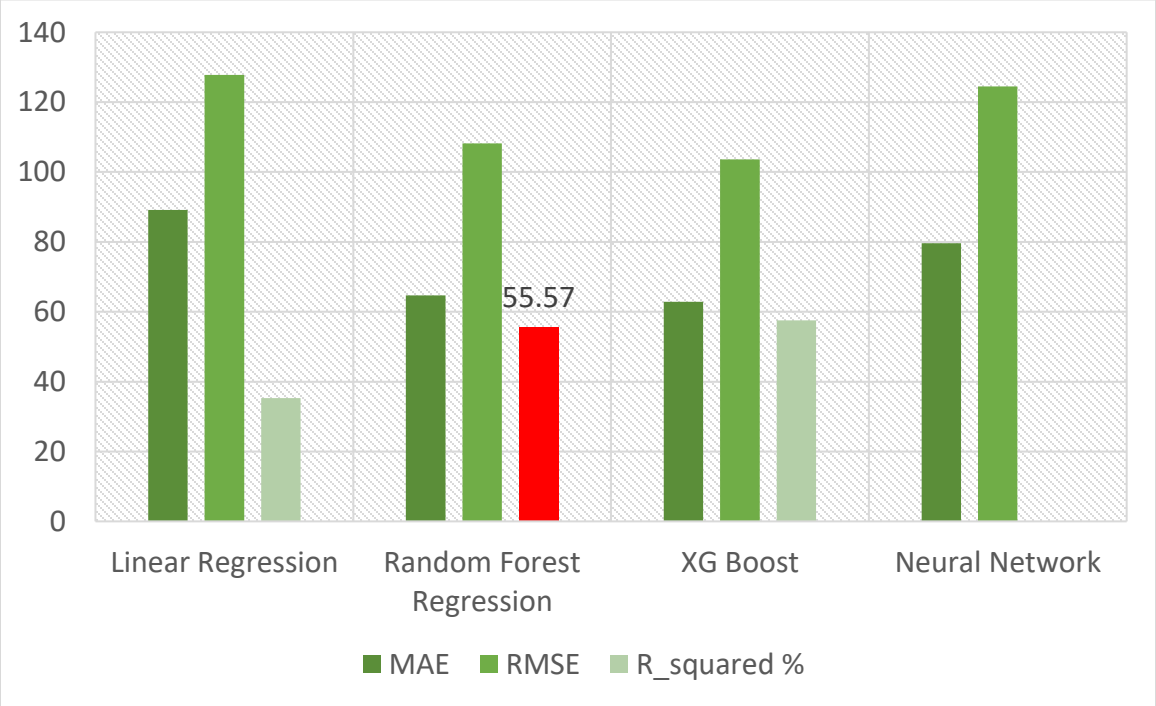


Figure 14 – *R_squared comparison between the four models with RandomForest Regressor having the highest RMSE*

Key Aspects of Note:

1. In the models, to improve the performance, we did not add or remove more features as we already had done a feature selection prior to running the model and chose only the most important features.
2. We specifically did not use the "neighbourhood_cleansed" feature as it is intuitive that downtown neighborhoods have higher listing prices due to the availability of amenities. We isolated the features which would have been overshadowed by the neighborhood data if included.
3. The performance is still comparatively poor as a model with an R-squared value as low as 57 is still very low predictability and only slightly better than completely random values.

6. Textual Analysis

In the refined dataset, we also had 4 textual data columns – property type, room type, host verifications, and amenities. In this section, we explore the data in property type, room type, and amenities and compare the price prediction based on these features.

6.1 Property Type Analysis

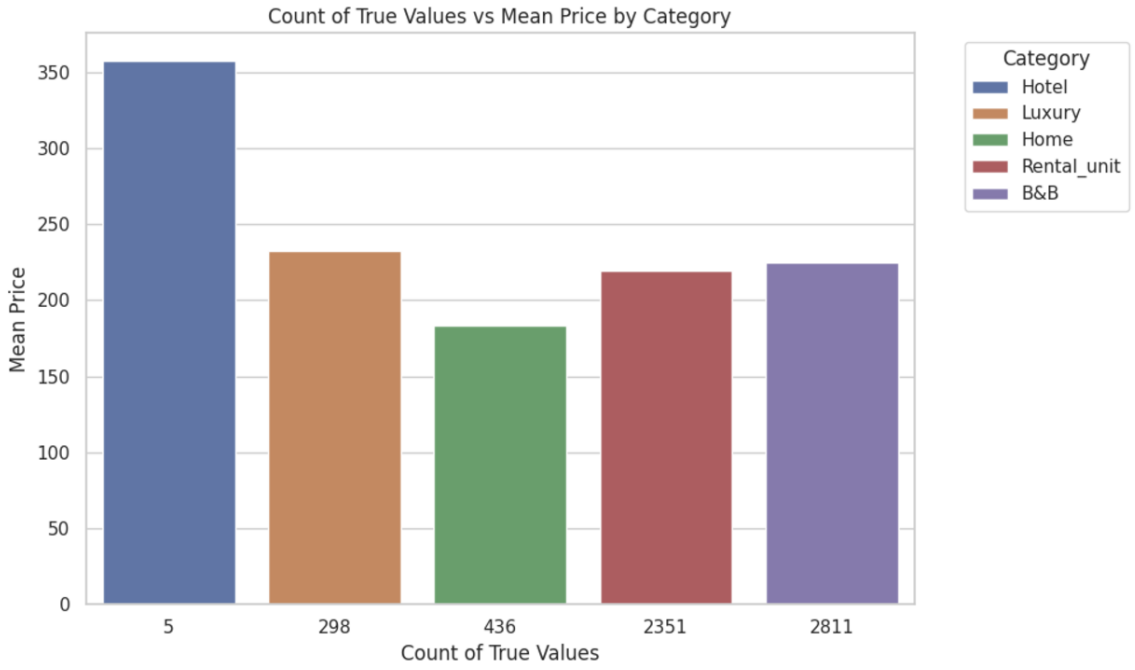


Figure 15 – *The distribution of mean prices of various property types*

The property type data had two types of information – the type of property such as whether it's a luxury room, hotel, motel, houseboat, bed and breakfast etc, and whether a unit was a shared unit, private unit, or entire unit. This data was split based on the string available within the property type column.

With the segregated data, a comparison between the mean price values and the frequency of a certain listing type brought about curious insights – the instinctive assumption that the Luxury listings might have higher mean prices was incorrect. Due to lack of variability in the pricing structure of Hotels, the mean price of hotel rooms is higher than that of luxury accommodations. Luxury or unique stays can have varied distribution in the price structure due to availability throughout the year (for example, a boathouse may not be available during the winter season).

The next step was to understand the impact of this feature on the price prediction of the listing. Hence, with the basic model – linear regression was used for the initial analysis. The linear regression model provided an R_squared value of **3.7%** and RMSE of **159.22** which is very high and doesn't indicate a strong linear relation between price and the property type.

However, to test whether the performance would improve with a non-linear model, DecisionTree Regressor was used. The following diagram shows the decision tree structure for price prediction with a single property type feature.

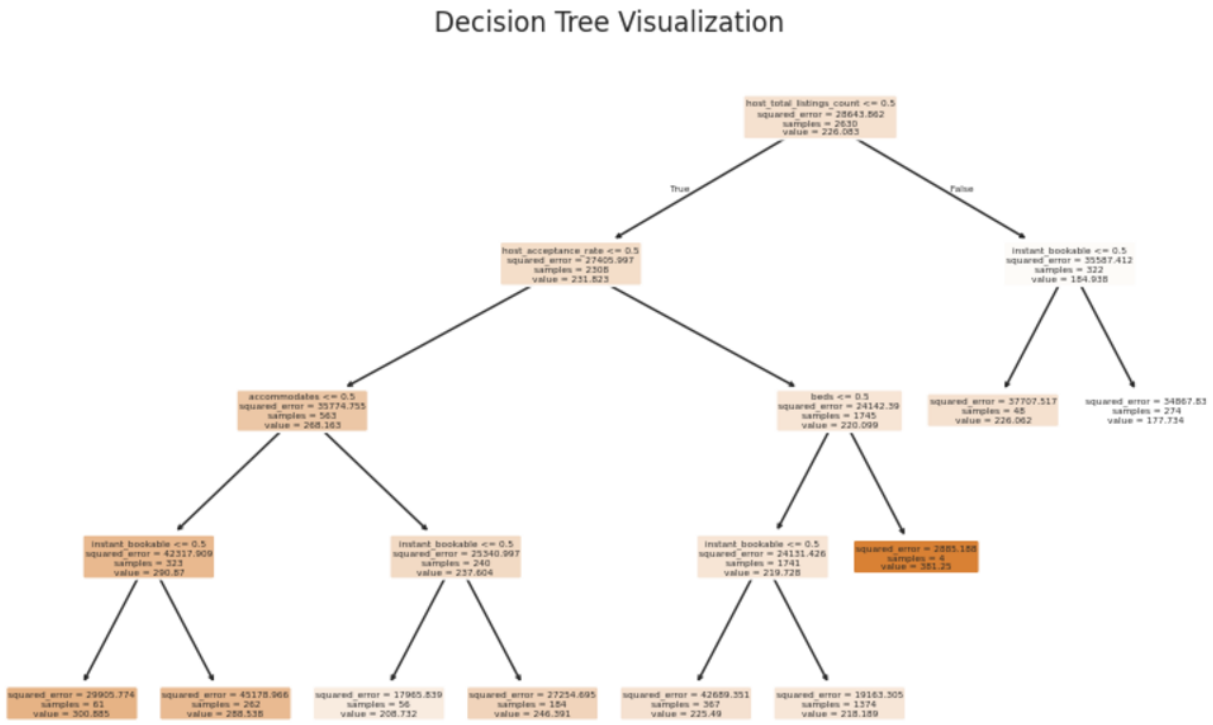


Figure 16 – *The decision tree visualization for the price prediction based on property type*

The decision tree had a similar performance as compared to the linear regression model and this hints that the relationship between price and property type might be complex and non-linear.

The decision tree yielded an R-squared value of **3.55%** and an RMSE value of **159.36.**

6.2 Room Type Analysis

Like the property type analysis, by segregating the keywords – "Entire", "Private" and "Shared", and plotting them against the mean price value, it was very evident that Entire units had higher mean prices as well as the frequency of entire units in the dataset is significantly higher than others.

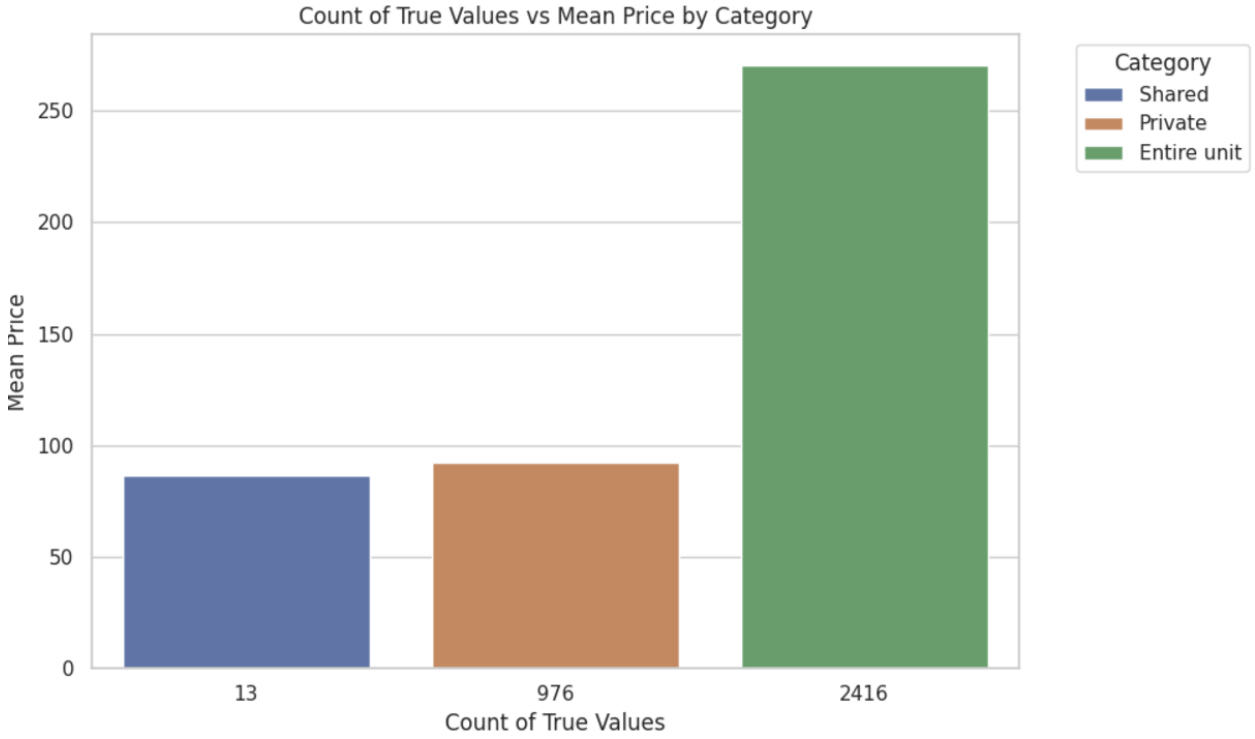


Figure 17 – *The distribution of mean prices of various room types*

Similar to the property type analysis, linear regression was used as the base model with a decision tree as a secondary model. The outcome was similar to that of property type analysis. The following table shows the performance metrics of linear regression and decision tree price predictions:

Table 7 – *Linear Regression and Decision Tree Results for Price Prediction*

Metric	**Linear Regression**	**Decision Tree**
Root Mean Squared Error (RMSE)	138.75099993177653	138.75099993177653
Mean Absolute Error (MAE)	91.39770534892799	91.39770534892794
R-squared	0.26890684897246564	0.26890684897246575

From the results, the performance of both models were almost the same. Thus, it is unclear whether the feature is linearly related to the model or not, however, the R_squared value is

higher than the compared to regular property type analysis. Hence it indicates that the room type has a higher influence on the price over the property type.

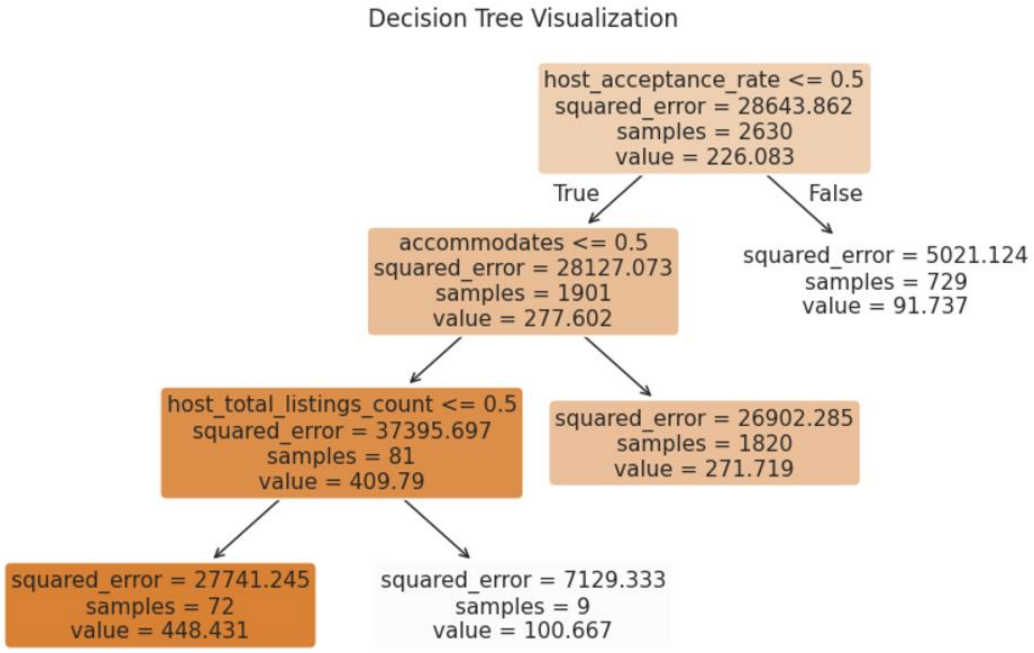


Figure 17 – *The decision tree visualization for the price prediction based on room type*

6.3 Amenities Analysis

The amenities are generally the most searched aspect of any listing and this definitely has a significant impact on the price prediction. However, in the dataset, the amenities were present as lists within a column. Hence for the analysis, the amenities column was split and segregated based on different unique parameters. Similar to the property type analysis, the Linear regression and decision tree models were executed to understand the overall impact of amenities on price prediction.

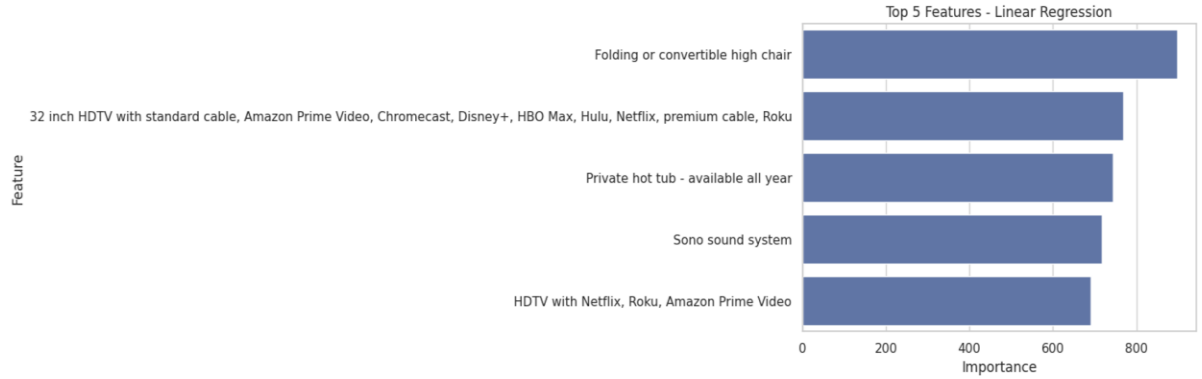
The Performance comparison is shown in the table below:

Table 8 – *Linear Regression and Decision Tree Results for Price Prediction*

Metric	**Linear Regression**	**Decision Tree**
Root Mean Squared Error (RMSE)	151.98757801343157	147.04164038319692
Mean Absolute Error (MAE)	99.19374283963538	80.17915135473896
R-squared	0.08500160802431034	0.14358390595965675

Here aswell, the Decision tree had better performance both in terms of R_squared values as well as the error rate.

Along with that, the top five features based on the feature importance were selected for both the models to identify if the models predict the same amenities as the importance for a price listing. However, it was interesting to note that, both the model weigh different amenities as important. This is indicative of how linear regression has selected features that linearly impact the price prediction while decision tree identifies the split points or cut off points between different categories and features.



	Feature	Coefficient	Importance
577	Folding or convertible high chair	897.385040	897.385040
710	32 inch HDTV with standard cable, Amazon Prime...	767.793993	767.793993
1138	Private hot tub - available all year	743.119077	743.119077
357	Sono sound system	717.032637	717.032637
590	HDTV with Netflix, Roku, Amazon Prime Video	689.622512	689.622512

Figure 18 – *Top five features from Linear Regression Co-efficient Analysis*

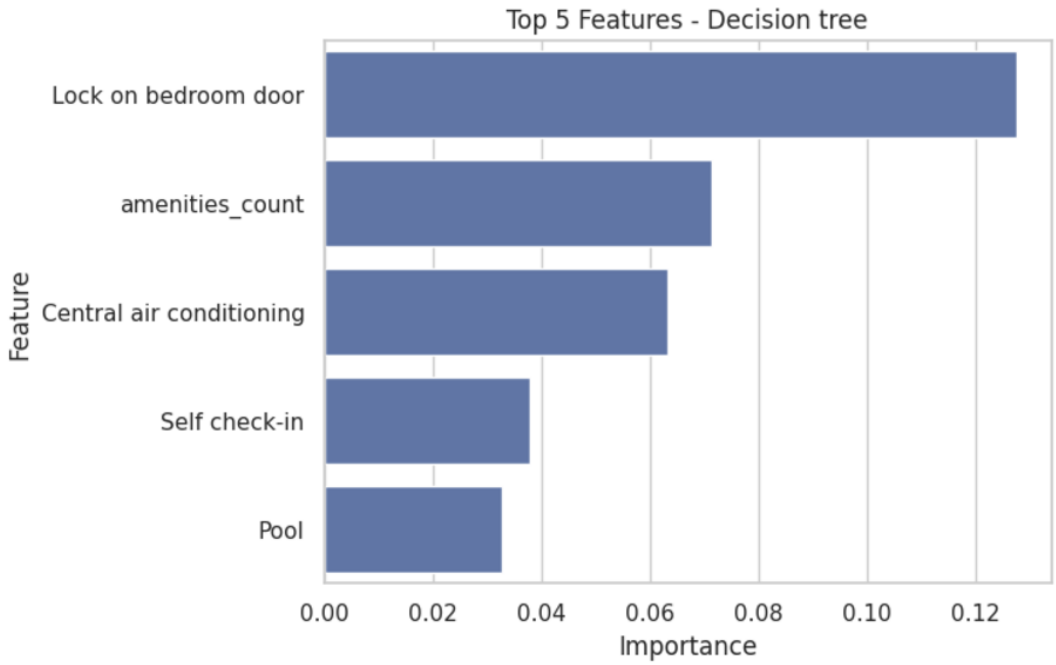


Figure 19 – *Top five features from Decision Tree Regressor Values*

7. Final Model

From the analysis of textual data, there is a clear indication that price is dependent on these features. So taking the best prediction model from the list of models tried, and testing the results when the dataset includes amenities and room type data is the next logical step. So going back to XGBoost model, rerunning the model with the modified textual data, where they are converted into categories based on their unique values.

Once again GridSearchCV is used to obtain the best parameters are they are follows:

```
Best parameters found:  {'alpha': 1, 'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
Best score found:  0.69818975405556
```

However, in the model, it is not a best practice to use alpha as 1. Hence retaining everything similar and keeping alpha as 0.1, the performance improvement was significant. The model R_squared is **0.6981 which is ~70%,** the model can explain a price variability of upto 70%. Meanwhile, the **Mean absolute error is 50.244 and RMSE is 82.63** which is the best out of all the models used in this project.

8. Conclusions and Key Recommendations

From the various analyses, the key insights that we derived were as follows:

1. Accommodation Capacity is one of the highest-ranked features
2. Entire Units are Preferred over private or shared rooms within a unit

3. Customers prefer security and room ambiance as top amenity needs
4. The factors affecting the price listing apart from neighborhood data are nonlinear
5. XGBoost was the best model to predict the price data with added textual data.

Key Recommendations for Hosts Listing on the Airbnb site:

1. Increase the availability and accommodation capacity to increase the listing price.
2. Lists entire units instead of listing shared rooms, or private rooms within a unit.
3. Highlight customer safety, safety measures implemented, and ambiance features in the listing description.

9. Next Steps

There is always room for improvement, and for this project, there are a few enhancements that can be made to give more time and resources. Augmenting this data from calendar and reviews dataset would increase more underlying feature relations and improve the performance of models listed in this project. Furthermore, other models like LightGBM or stacking of listed models could have been used to see if they have better results. However, these were out of scope for the current project.