Recommendation Systems - Final Project

Airbnb Price Recommendation

Linoy Azar - 208846741
Shahaf Yagoda - 305196586

Data Set: NY Listing
Code: Final Project Code

About Airbnb

Airbnb, Inc. is a global online marketplace that connects travelers with hosts offering unique short- and long-term stays, as well as local experiences. Founded in 2008, the company has transformed the hospitality industry by enabling individuals to rent out their private properties. The platform spans nearly every corner of the world, offering everything from urban apartments to countryside retreats and luxury villas.

Recommendation System's Objectives

The objective of our recommendation system is to assist property owners in New York in determining an appropriate nightly price range for their new properties. By analyzing the features and prices of existing listings, the system will identify pricing patterns related to factors such as location, property type, and size. The model will be trained using this data, with the price range as the target variable, enabling it to recommend a price range for new listings based on similar characteristics of existing properties.

Data Overview

This dataset contains information about the New York Airbnb listings. It contains 35 columns (Appendix A) and 75,749 records, representing a wide range of details about the various properties. This data includes useful features such as neighborhood, property type, room type, accommodates, amenities, prices, reviews and more.

Data Selection

The columns 'Street', 'City', 'Neighbourhood cleansed', 'Room type', 'Property type', 'Bedrooms', and 'Accommodates' were chosen as they directly influence property pricing.

1. Street: Contains the street name, important for exact location. The dataset includes 298 unique streets.
2. City: Indicates the city of the property. The dataset features 5 cities, with New York being the most common (34,375 listings).
3. Neighborhood cleansed: Specifies the neighborhood. There are 298 unique neighborhoods, with Williamsburg being the most frequent (6,017 listings).
4. Property type: Describes the type of property (e.g., apartment, house). The dataset includes 29 property types, with apartments being the most common (38,799 listings).
5. Room type: Describes the type of space offered (entire home, private room, etc.). There are 4 room types, with 'Entire home/apt' being the most frequent (38,306 listings).
6. Bedrooms: Indicates the number of bedrooms. Properties with up to 14 bedrooms are in the dataset, with 1 bedroom being the most common.
7. Accommodates: Specifies the number of people the property can host. Most properties accommodate up to 4 guests.

These columns provide essential details that impact pricing and demand, reflecting variations in price based on location, property type, room type, and size (Appendix B).
Note: We found a high correlation (0.72) between 'Accommodates' and 'Bedrooms', raising concerns about multicollinearity. After testing the correlation of each variable with 'Price', we found 'Accommodates' had a higher correlation (0.52), so we proceeded with this variable only. (Appendix C, Table 1).

Exploratory Data Analysis (EDA)

Missing Data

In the initial stage, we focused on handling missing values and applying appropriate data imputation to ensure the accuracy of the relevant columns using all the available data. We identified that the Street and Accommodates columns had missing values in approximately 41% of the records (Appendix C, Table 2).

- To address the missing values in the **Accommodates** column, we used the average capacity for each property type, ensuring the values were whole numbers, as the data refers to the number of people. After filling the missing values, we tested the impact on correlation and found that it reduced significantly to 0.12 (Appendix C, Table 3). As a result, we decided to remove the rows with missing values in the Accommodates column to maintain data integrity.
- For the **Street** column, we imputed missing values by using the most common street name within the corresponding neighborhood, which was determined from the Neighborhood column.

After rechecking for missing values, we chose to delete the remaining records with missing data, as their volume was minimal and imputing them was not feasible due to insufficient similarity with other records (Appendix C, Table 4).

Outliers (Appendix D)

- We identified outliers in the 'Price' column by removing properties priced at $0 and analyzing records above the upper bound ($335.5). We found that typical properties (e.g., apartments and houses) had more abnormal prices, while niche properties (e.g., caves, tents, castles) had fewer. Unique properties exhibited a higher percentage of outliers, while standard properties showed more price stability. We decided to remove extreme values to focus on average priced properties for a more accurate market analysis.
- In the **Accommodates** column, we confirmed that the minimum value is reasonable, with the lowest capacity being 1 person. We then checked for records where the capacity exceeded the upper bound of 7 people and found 647 records, making up only 0.91% of the total dataset. Given this small percentage, we chose to delete these records to maintain data consistency.

Models Creation

Data Preparation

1. **Selecting Columns**: we reduced the dataset to only the selected columns, after handling missing values and addressing outliers.
2. **Binning the 'Price' Column**: We used uniform width binning to divide the 'Price' column into 6 equal segments (Appendix E), ensuring that each price category spans a similar range. This approach provides users with clear, balanced price recommendations, avoiding large variations within any one category. This method maintains consistency across categories, making the price guidance more intuitive and relevant for users.
3. **Encoding Categorical Columns**: To convert textual variables to numeric values, we create a separate dummy column for each unique categorical value, allowing the model to interpret different values.

Defining Input and Output Matrices

- **X**: The input matrix includes all features (columns) except for 'Price_binned'.
- **y**: The output matrix, containing only the 'Price_binned' column, represents the categorical price values after binning.

Splitting into Training and Test Sets

We applied the holdout method with an 80/20 Train-Test Split to evaluate model performance, using 80% for training and 20% for testing to balance model training with generalization assessment on new data. Setting `random_state=42` ensures consistent, reproducible splits across experiments.

Success Metrics

1. Accuracy: The percentage of correct predictions out of all predictions. This is a general metric that provides an initial indication of the model's quality, but it can be sensitive in cases of class imbalance.
2. F1-Score: A weighted average of Precision and Recall, balancing both and providing a more accurate picture of the model's ability to distinguish between categories.
3. MCC: A metric that provides an overall picture of the model's quality while accounting for class imbalance. It gives a more accurate evaluation of the model's performance, especially when there are imbalanced categories.
4. Classification Report: Displays Precision, Recall, and F1-Score for each category separately. This is important when there is class imbalance and provides a more complete picture of the model's performance for each category.

Models Results (Appendix F)

	Accuracy	F1 Score	MCC
Random Forest	0.49	0.46	0.3
ID3	0.49	0.47	0.31
KNN	0.46	0.44	0.27
C4.5	0.49	0.47	0.31

1. **Comparing Model Performance:**
The decision-tree-based models (ID3 and C4.5) showed similar performance, slightly outperforming KNN and Random Forest in terms of F1 Score and MCC. However, these differences are minimal, and all models achieved relatively low accuracy overall. The KNN model had the lowest accuracy, F1 Score, and MCC, suggesting it may struggle with distinguishing between the price categories.

2. **Modeling and Data Issues:**
All models exhibit low accuracy, possibly indicating issues with the data itself, such as an imprecise price category distribution or outdated records. This could mean that the dataset includes properties that are no longer active in the market, leading to discrepancies in price relevance. Additionally, class imbalance within the price categories may be affecting model performance, as the models might focus on dominant categories, making it challenging to classify less frequent ones accurately.

3. **MCC as a Model Accuracy Indicator:**
The MCC (with a range from -1 to 1) gives a broader view of performance, especially with imbalanced data. The values obtained (around 0.27 to 0.31) reflect a low correlation, underscoring the difficulty the models have in accurately distinguishing between categories, potentially due to unremoved dependent variables.

Model Breakdown
Classification Report

Model	Category	0	1	2	3	4	5
ID3	**Precision**	0.54	0.56	0.39	0.37	0.27	0.32
	Recall	0.57	0.61	0.47	0.36	0.07	0.01
	F1-Score	0.55	0.59	0.43	0.37	0.11	0.02

KNN	**Precision**	0.51	0.53	0.37	0.35	0.19	0.14
	Recall	0.57	0.6	0.37	0.3	0.11	0.03
	F1-Score	0.54	0.56	0.37	0.32	0.14	0.05
C4.5	**Precision**	0.54	0.56	0.39	0.37	0.27	0.32
	Recall	0.57	0.61	0.47	0.36	0.07	0.01
	F1-Score	0.55	0.59	0.43	0.37	0.11	0.02

Overall Summary - All three models (ID3, KNN, and C4.5) demonstrate strong performance in the lower categories (0.0 and 1.0) but face challenges in accurately predicting the higher categories (4.0 and 5.0), largely due to data imbalance. Addressing this imbalance is essential to improve accuracy in these less represented categories.

Conclusions and Recommendations

- Price Binning: We first used uniform width binning to create equal price ranges, aiming for clear and balanced categories. However, this method might not have been ideal, as it could group unrelated data points, especially if the distribution is uneven. We then tried quantile-based binning, which creates groups with an equal number of data points, but the model results remained similar.
- Outdated Data and Active Listings: A possible reason for the low performance metrics is that the data may include outdated or inactive records. Some properties might have been active in the past but are no longer available, meaning their prices are not up-to-date or reflective of the current market. If it's unclear when each record was collected, this could lead to accuracy issues, as the prices may not reflect the current market conditions.
- Adding Features: The model could benefit from additional features that help improve predictive accuracy. For example, incorporating more detailed location data (latitude and longitude), specific property features, or the frequency of updates could enhance the model’s ability to differentiate between categories more effectively.

Appendix A

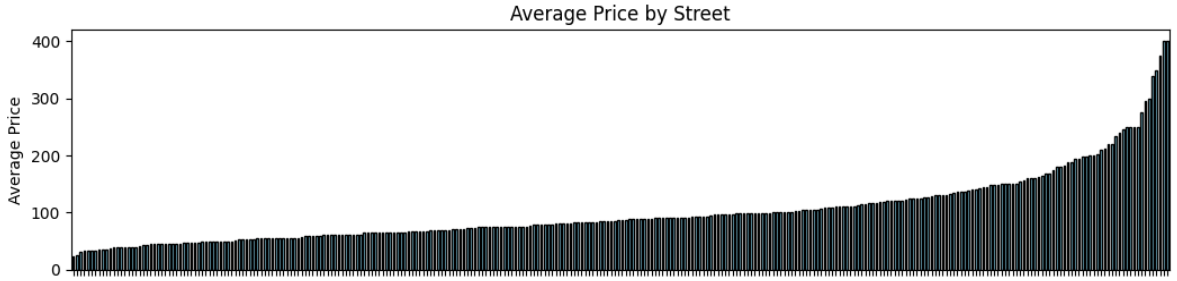
Column Descriptors

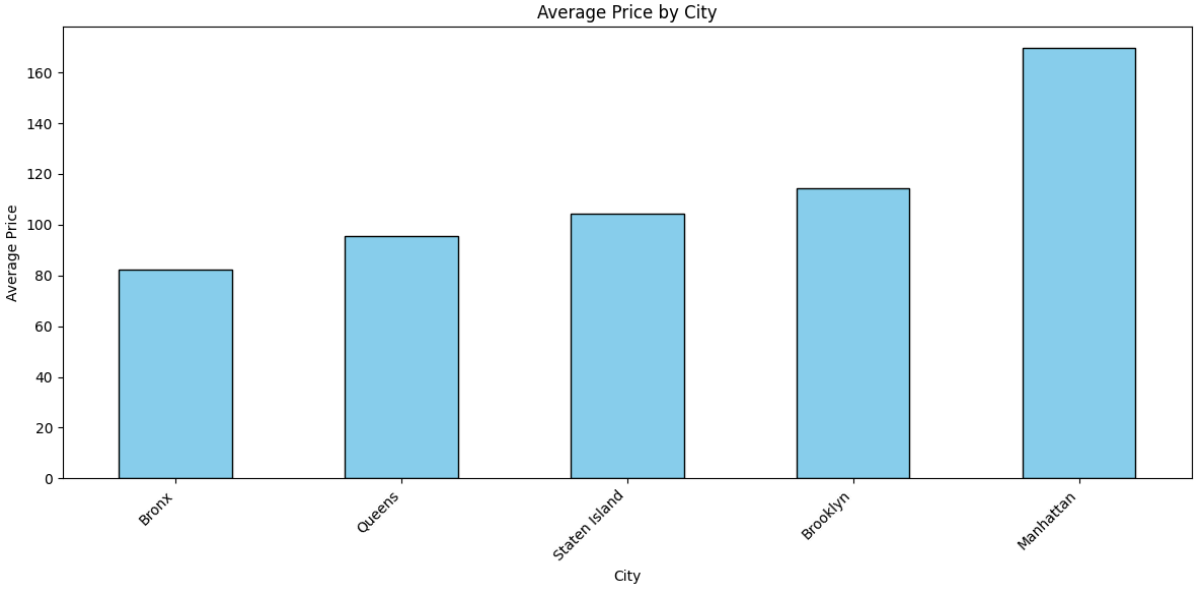
The dataset includes 4 data types: bool(1), float64(9), int64(13), object(12).

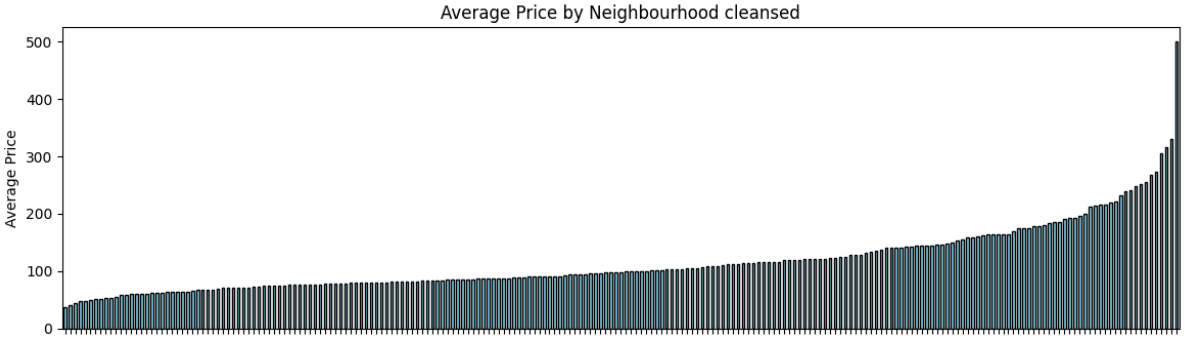
Column Name	Data Type	Description	Classification
Listing ID	float64	Unique identifier for each listing on the platform.	Numeric: Quantitative
Name	object	Name or title of the Airbnb listing.	Nominal
Host ID	int64	Unique identifier for each host on the platform.	Numeric: Quantitative
Host Name	float64	Name of the host associated with the listing.	Nominal
Host Response Rate	float64	Percentage rate at which the host responds to inquiries.	Numeric: Quantitative
Host Is Superhost	object	Indicates if the host is a 'superhost' (yes or no).	Binary
Host total listings count	object	Total count of listings managed by the host.	Numeric: Quantitative
Street	object	Name of street address of the listing.	Nominal
City	int64	City where the listing is located.	Nominal
Neighborhood cleansed	bool	Specific neighborhood the listing.	Nominal
State	object	State where the listing is located.	Nominal
Country	float64	Country where the listing is located.	Nominal
latitude	float64	Latitude coordinate of the listing.	Numeric: Quantitative
longitude	object	Longitude coordinate of the listing.	Numeric: Quantitative
Property type	int64	Type of property (e.g., apartment, house).	Nominal
Room type	float64	Type of room offered (e.g., entire home, private room).	Nominal
Accommodates	int64	Number of people the listing can accommodate.	Numeric: Quantitative

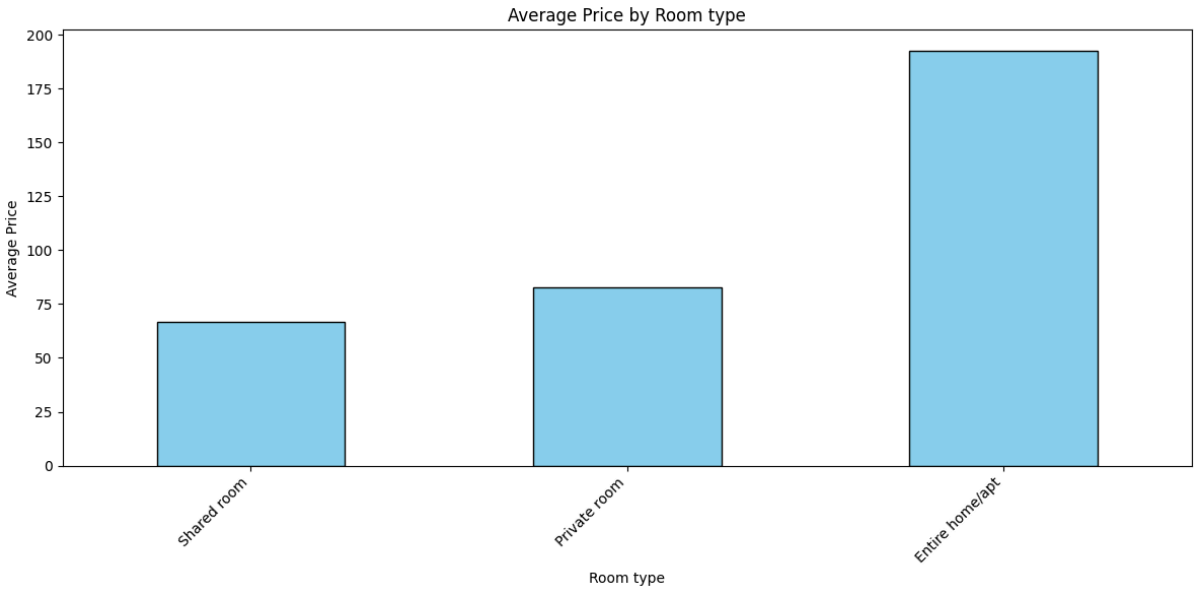
Bathrooms	object	Number of bathrooms available in the listing.	Numeric: Quantitative
Bedrooms	object	Number of bedrooms available in the listing.	Numeric: Quantitative
Amenities	int64	List of amenities available at the listing.	Nominal
Price	int64	Price per night for booking the listing, in dollars..	Numeric: Quantitative
Minimum nights	object	Minimum number of nights required for booking.	Numeric: Quantitative
Maximum nights	int64	Maximum number of nights allowed for booking.	Numeric: Quantitative
Availability 365	int64	Number of days the listing is available for booking over the next year.	Numeric: Quantitative
Calendar last scraped	int64	Date when the availability calendar was last updated.	Nominal
Number of reviews	int64	Total number of reviews left for the listing.	Numeric: Quantitative
Last Review Date	int64	Date of the last review received for the listing.	Nominal
Review Scores Rating	int64	Overall rating score based on guest reviews.	Numeric: Ordinal
Review Scores Accuracy	int64	Accuracy rating based on guest reviews.	Numeric: Ordinal
Review Scores Cleanliness	float64	Cleanliness rating based on guest reviews.	Numeric: Ordinal
Review Scores Checkin	object	Check-in rating based on guest reviews.	Numeric: Ordinal
Review Scores Communication	object	Communication rating based on guest reviews.	Numeric: Ordinal
Review Scores Location	object	Location rating based on guest reviews.	Numeric: Ordinal
Review Scores Value	float64	Value-for-money rating based on guest reviews.	Numeric: Ordinal
Reviews per month	float64	Average number of reviews per month.	Numeric: Quantitative

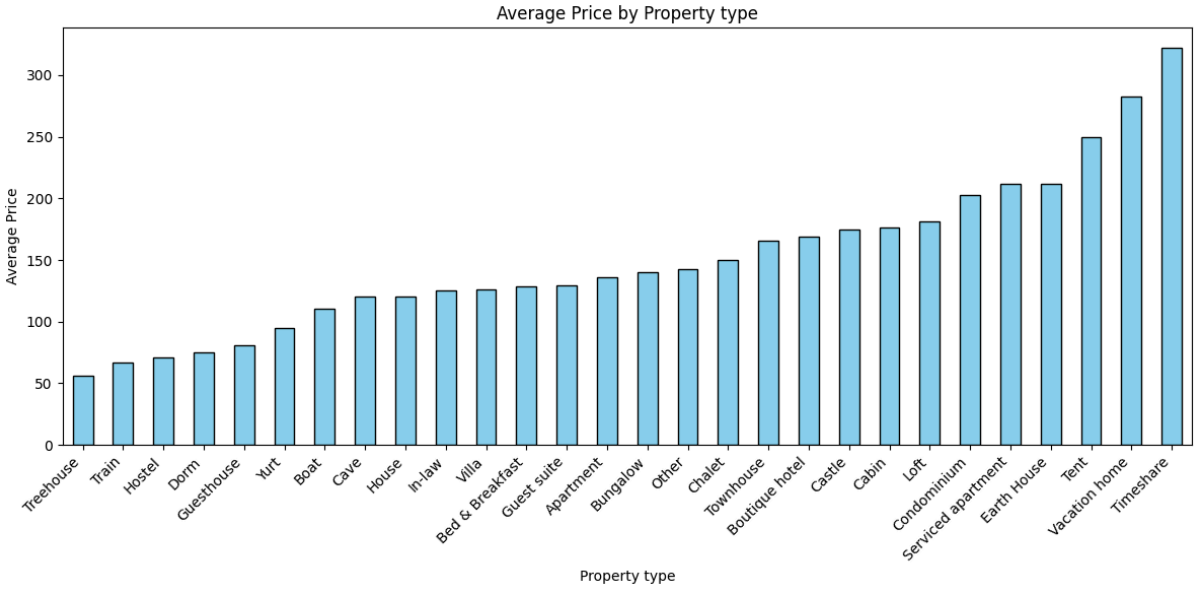
Appendix B


Average Price by Street
Average Price
0
100
200
300
400


Average Price by City
Average Price
0
20
40
60
80
100
120
140
160
Bronx
Queens
Staten Island
Brooklyn
Manhattan
City


Average Price by Neighbourhood cleansed
Average Price
0
100
200
300
400
500


Average Price by Room type
Average Price
0
25
50
75
100
125
150
175
200
Shared room
Private room
Entire home/apt
Room type


Average Price by Property type
Average Price
0
50
100
150
200
250
300
Treehouse
Train
Hostel
Dorm
Guesthouse
Yurt
Boat
Cave
House
In-law
Villa
Bed & Breakfast
Guest suite
Apartment
Bungalow
Other
Chalet
Townhouse
Boutique hotel
Castle
Cabin
Loft
Condominium
Serviced apartment
Earth House
Tent
Vacation home
Timeshare
Property type

Appendix C

Table 1:

	Correlation	P-value
Accommodates and Bedrooms	0.717449	0.0
Accommodates and Price	0.525359	0.0
Bedrooms and Price	0.474856	0.0

Table 2:

	Missing Values	Percentage (%)
Name	31	0.040925
Host Name	233	0.307595
Host Response Rate	45118	59.562502
Host total listings count	232	0.306275
Street	31439	41.504178
Accommodates	31439	41.504178
Bathrooms	31581	41.691639
Bedrooms	31439	41.504178
Amenities	31721	41.876460
Maximum nights	31439	41.504178
Calendar last scraped	31439	41.504178
Last Review Date	16683	22.024053

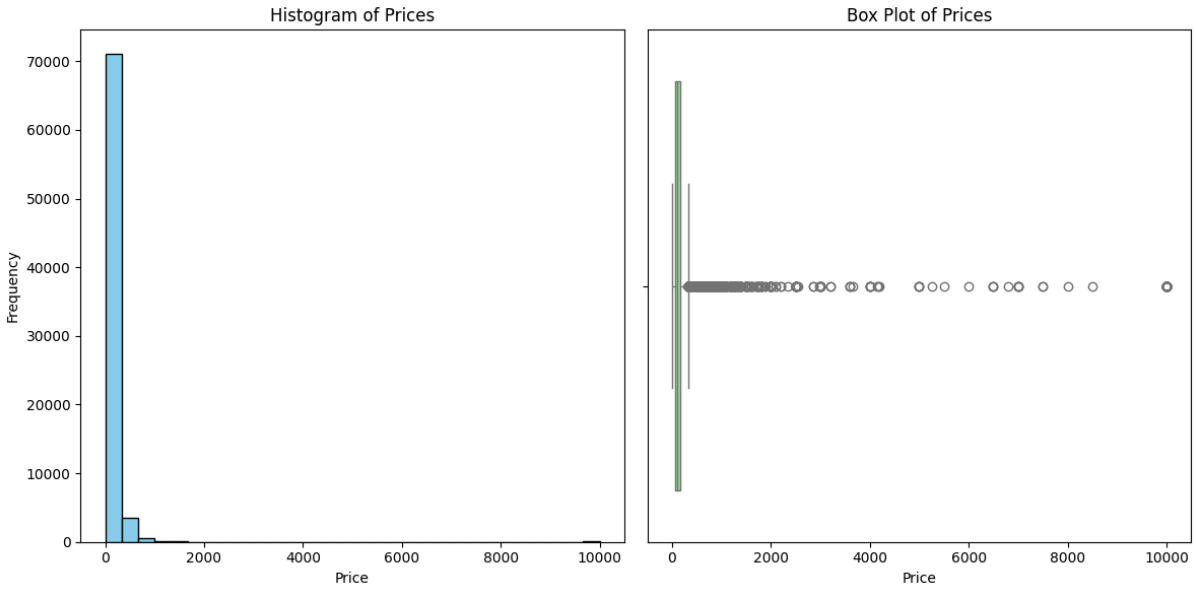
Table 3:

	Correlation	P-value
Accommodates and Price	0.124053	3.686184e-256

Table 4:

	Missing Values	Percentage (%)
Street	13	0.017249

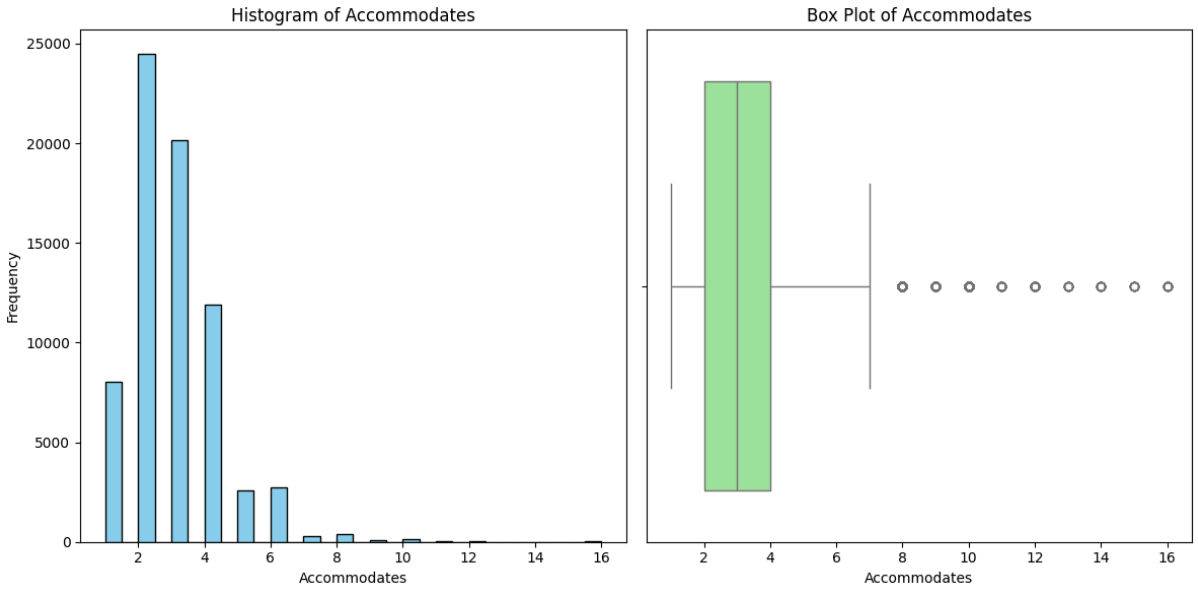
Appendix D



Upper Bound Outliers - Price [Count/ Proportion]

Property type	
Apartment	1733
House	308
Loft	215
Townhouse	190
Condominium	170
Other	119
Timeshare	114
Bungalow	98
Dorm	96
Guest suite	93
Cabin	91
Hostel	91
Bed & Breakfast	89
Villa	88
In-law	85
Chalet	83
Cave	81
Boutique hotel	79
Earth House	77
Guesthouse	76
Yurt	76
Boat	73
Serviced apartment	73
Vacation home	72
Treehouse	11
Castle	10
Tent	7
Train	2

Property type	
Cave	11.964549
Yurt	11.275964
Loft	9.944496
Condominium	9.037746
Townhouse	9.030418
Timeshare	8.662614
Other	7.517372
Bungalow	7.407407
Guest suite	7.237354
Dorm	7.207207
Villa	7.034373
Cabin	6.978528
In-law	6.865913
Hostel	6.780924
Chalet	6.726094
Boutique hotel	6.299841
Earth House	6.239870
House	6.185981
Bed & Breakfast	6.180556
Serviced apartment	5.887097
Boat	5.882353
Vacation home	5.815832
Guesthouse	5.584129
Apartment	4.487312
Train	2.816901
Treehouse	1.974865
Castle	1.801802
Tent	1.256732



Appendix E

Price Binned

Price binned	Count	Price Range in $
0	13350	5-60
1	26320	60-115
2	15011	115-170
3	8786	170-225
4	4314	225-280
5	2325	280-335

Appendix F

Random Forest: A machine learning model based on decision trees that uses multiple trees (forest) to improve performance and reduce the risk of overfitting. Each tree in the Random Forest is trained on a random subset of the data, and the final output is determined by the majority vote of all the trees. It performs well in identifying non-linear relationships and is especially effective when there is class imbalance in the data.

ID3 (Iterative Dichotomiser 3): A decision tree algorithm that splits the data based on the maximum information gain from each feature (using a metric called "information gain" or "Entropy"). ID3 is well-suited for classification problems but tends to generate deep trees if the stopping criteria for splits are not properly controlled.

KNN (K-Nearest Neighbors): A classification model that assigns a class to a new data point by finding the K nearest neighbors (K is predefined) and taking the majority vote of their class labels. It's simple to implement but can be slow with large datasets and does not perform internal optimization like other models.

C4.5: An enhanced version of ID3, this decision tree model uses new splitting techniques (like handling missing data) and performs splits based on maximizing information gain. C4.5 is widely used due to its simplicity and ability to handle imperfect data.