The Price of Airbnb Listings

Pengjin Huang, Siyu Chen, Xinglong Liu, Zilin Lu, Zunke Ma

Columbia University

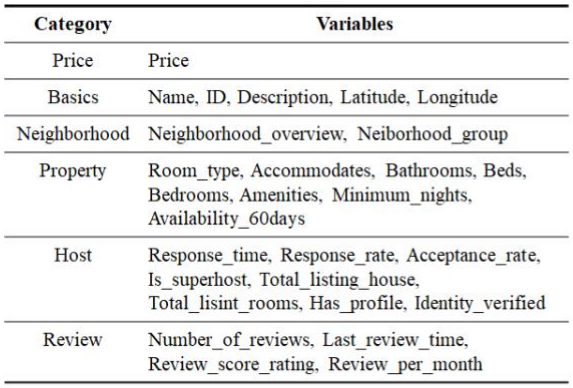
1. Introduction

Airbnb is a leading online marketplace for short-term rental accommodations. The platform allows users to search for and book a wide range of accommodations, including apartments, houses, and other unique properties. Our machine learning project aims to explore and analyze the factors that impact Airbnb pricing. By studying the characteristics of different properties in the Manhattan area, we aim to develop a model that can accurately predict the actual price of an Airbnb listing. This will hopefully provide valuable insights and information for travelers, giving them a better understanding of the factors that influence Airbnb pricing and helping them to find the best deals.

2. Exploratory Data Analysis

2.1. Basic Insights

Airbnb is a cross-sectional dataset containing detailed listings in Manhattan, New York. Exploratory data analysis is performed on the training data, which consists of 13477 entries and 29 columns, with 28 independent variables and 1 dependent variable (price). The variables were summarized into different categories to gain insights about the dataset (*Table 1*).



Category	Variables
Price	Price
Basics	Name, ID, Description, Latitude, Longitude
Neighborhood	Neighborhood_overview, Neiborhood_group
Property	Room_type, Accommodates, Bathrooms, Beds, Bedrooms, Amenities, Minimum_nights, Availability_60days
Host	Response_time, Response_rate, Acceptance_rate, Is_superhost, Total_listing_house, Total_lisint_rooms, Has_profile, Identity_verified
Review	Number_of_reviews, Last_review_time, Review_score_rating, Review_per_month

Table 1: Variables summarized into different categories

2.2. Univariate Data Analysis

The distribution of price in the Airbnb dataset is right skewed with a heavy tail (*Figure 1.1*), meaning that there exist extremely high prices. Specifically, there are 2.39% of listings with prices over 1000. After excluding these listings, the distribution of prices looks more reasonable (*Figure 1.2*).

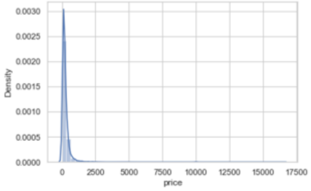


Figure 1.1: Price distribution before removing outliers

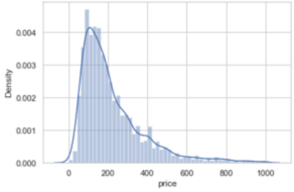


Figure 1.2: Price distribution after removing outliers

2.3. Bivariate Data Analysis for Categorical Variables

One important categorical variable in the dataset is the neighborhood. According to the box plot (*Figure 3*), the range of prices is positively correlated with the median.

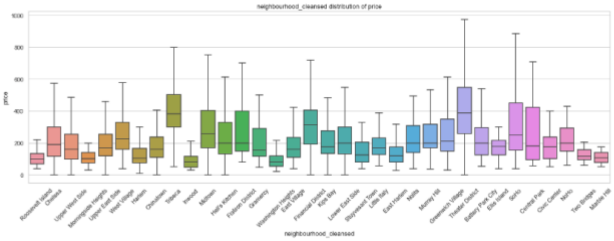


Figure 2: neighborhood distribution of price

Two heatmaps below correspond to price and number of listings respectively (*Figure 3*), where red indicates low and green indicates high. Tribeca and Theater are two price leaders. The area with higher median listing price tends to have a larger range of prices. The high-price listings centered at Midtown and Downtown which are tourist attractions. Going down along Manhattan Island, the price goes up. As for listing numbers, we've found that two centers exist, Midtown & Downtown (driven by tourism), and the lower part of Uptown (education). In the Financial district, the lowest part, there are only a few listings since lots of firms and hotels are around there.

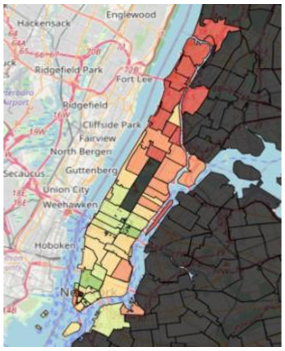


Figure 3.1: Mean prices

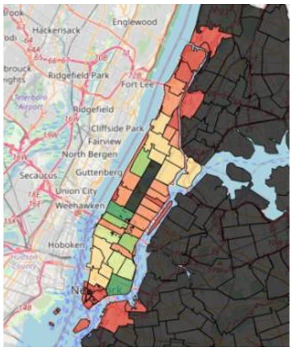


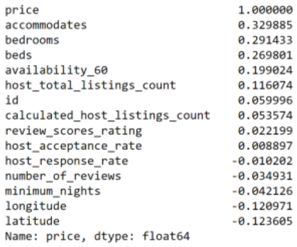
Figure 3.2: Number of listings

The relationship between other categorical variables and price is explored by box plots. After examining the data, it is found that there is no significant price difference between host-related variables (*Appendix: Figure 4*), suggesting that these factors may not be drivers of price in the Airbnb dataset.

2.3. Bivariate Data Analysis for Numerical Variables

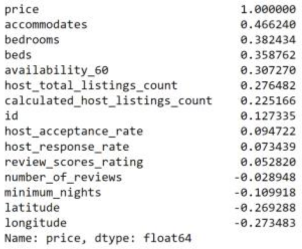
The scatter plot of all numerical variables against the price looks messy (*Appendix: Figure 5*). If extreme prices are excluded as suggested in the previous analysis, it is still difficult to observe any pattern or trend.

To better understand their relationships, the correlation coefficients are checked (*Table 2.1*). Overall speaking, the correlations are low since the dataset is noisy. After excluding all extreme prices, correlations are improved by a little bit (*Table 2.2*). Again, this verifies that it is better to remove these extreme prices.



```
price                             1.000000
accommodates                      0.329885
bedrooms                          0.291433
beds                              0.269801
availability_60                   0.199024
host_total_listings_count         0.116074
id                                0.059996
calculated_host_listings_count    0.053574
review_scores_rating              0.022199
host_acceptance_rate              0.008897
host_response_rate               -0.010202
number_of_reviews                -0.034931
minimum_nights                   -0.042126
longitude                        -0.120971
latitude                         -0.123605
Name: price, dtype: float64
```

Table 2.1: Correlations with price before removing outliers



```
price                             1.000000
accommodates                      0.466240
bedrooms                          0.382434
beds                              0.358762
availability_60                   0.307270
host_total_listings_count         0.276482
calculated_host_listings_count    0.225166
id                                0.127335
host_acceptance_rate              0.094722
host_response_rate                0.073439
review_scores_rating              0.052820
number_of_reviews                -0.028948
minimum_nights                   -0.109918
latitude                         -0.269288
longitude                        -0.273483
Name: price, dtype: float64
```

Table 2.2: Correlations with price after removing outliers

2.4. Multivariate Data Analysis

The correlations between independent variables are examined (*Appendix: Figure 6*). Some variable pairs have correlation coefficients above 0.7, but there is no extremely high correlation, so no need to worry about multicollinearity.

3. Data Preprocessing

3.1. Process Missing Value

As the *Appendix: Table 3* shows, the data set starts with a high ratio of missing values. And some variables are missing or present at the same time, like "host_response_time" and "host_response_rate".

Besides, combined with the situation in *Figure 7* and *Table 4*, we believe that the most of housing listings (rows) have less than 6 missing variables. In this case, we decide to eliminate the rows which are missing more than 6 variables. In addition to this, because host-response related and review-related variables are simultaneously and intuitively important variables, we also have a new criterion to delete it where host-response and review related variables are both missing. Based on this, 14,454 rows remained, accounting for 85.80% of the original data.

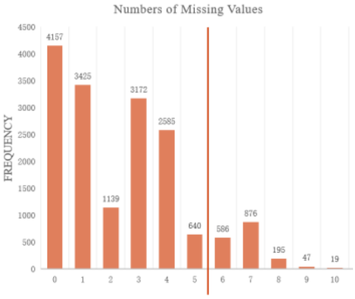
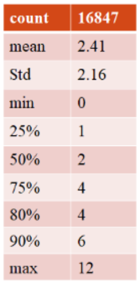


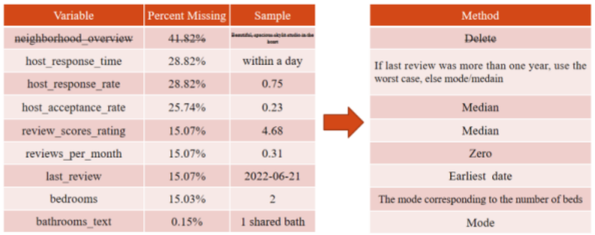
Figure 7: The distribution of missing variables per row



count	16847
mean	2.41
Std	2.16
min	0
25%	1
50%	2
75%	4
80%	4
90%	6
max	12

Table 4: The distribution of missing variables

After deleting rows with so many missing variables, we still need to fill in some missing values. Based on the properties of columns, different methods are adopted to fill in the missing value (*Figure 8*).



Variable	Percent Missing	Sample
~~neighborhood_overview~~	~~41.82%~~	~~Beautiful, spacious skylit studio in the heart~~
host_response_time	28.82%	within a day
host_response_rate	28.82%	0.75
host_acceptance_rate	25.74%	0.23
review_scores_rating	15.07%	4.68
reviews_per_month	15.07%	0.31
last_review	15.07%	2022-06-21
bedrooms	15.03%	2
bathrooms_text	0.15%	1 shared bath

Method
~~Delete~~
If last review was more than one year, use the worst case, else mode/medain
Median
Median
Zero
Earliest date
The mode corresponding to the number of beds
Mode

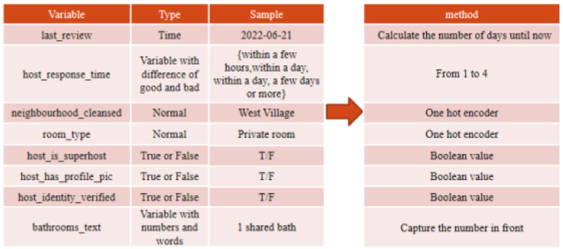
Figure 8: Methods for fill in the missing data

3.2. Process Categorical Variables

After dealing with the missing values, we still have some non-numerical variables to figure out. At first, the "last_review" variable is the timestamp type value, we transform it to a numerical value which is the number of days until now. Besides, for the ordinal variable "host response time", we use a sequence of numbers to represent them since the categories have an order. Furthermore, nominal variables "neighbourhood_cleansed" and "room_type" are dealt with by one hot encoder method. Moreover, we extract the number in front of the text variable "bathrooms_text" as its new value.

Lastly, one text variable named "amenities" is the most complicated variable to deal with. It is a list of all the amenities that the house listing has. *Appendix: Figure 9* has shown one sample. Based on this condition, we count the frequency of each amenity in the dataset and select the top 20 facilities that appear most frequently. And then calculate the percentage of each house listing containing the top 20 facilities we got before. The final result is a value between 0 and 1.

A summary of ways in preprocessing of non-numerical variables is shown as *Figure 10*.



Variable	Type	Sample	method
last_review	Time	2022-06-21	Calculate the number of days until now
host_response_time	Variable with difference of good and bad	{within a few hours,within a day, within a day, a few days or more}	From 1 to 4
neighbourhood_cleansed	Normal	West Village	One hot encoder
room_type	Normal	Private room	One hot encoder
host_is_superhost	True or False	T/F	Boolean value
host_has_profile_pic	True or False	T/F	Boolean value
host_identity_verified	True or False	T/F	Boolean value
bathrooms_text	Variable with numbers and words	1 shared bath	Capture the number in front

Figure 10: Preprocessing of non-numerical variables

3.3. Data Scaling

Z-scaling can better handle outliers and extreme values in comparison to min-max scaling. In this study, z-scaling is adopted because the data had outliers and noise, and z-scaling can indirectly avoid the impact of these issues by centralizing the data. The scaler is fit on the training data and then applied to transform both the training and test data using the mean and standard deviation of the training set. In reality, the algorithm should not have any information about the test set, thus the variance and mean of the test data should be based on the prior training set.

4. Model Performances

To predict the price of a renting apartment, we split the whole data set into a training set and a test set. Then, we fit several machine learning models on the training set, including linear regression, K-nearest neighbors, random forest, and neural networks. Use the test set to calculate the R-squared and Mean Square Error, the models' performances are summarized in the following table:

	Parameters	R-squared	MSE
Linear Regression	Regularization (Ridge / Lasso) applied	52.92%	13528.11
K-Nearest Neighbors	{'n_neighbors' = 9, 'p' = 1, 'weights' = 'distance'}	61.24%	10671.14
Random Forest	{'max_depth' = 20, 'min_samples_split' = 2, 'n_estimators' = 400}	68.65%	8630.32
Neural Network	{activation = 'relu', loss = 'mean_squared_error', optimizer = 'adam', epochs = '20'}	61.87%	10495.76

Table 5: The proportion of missing values for each variable before processing

Based on the test set's performance, we choose to use Random Forest as the final model because it has the highest R-squared value and lowest Mean Squared Error. The models' details are as follows.

4.1. Linear regression

The linear regression model is constructed with Ridge / Lasso regularizer. The performance of linear regression is not as expected. It suggests that the relationship between house properties and their Airbnb price is not linear. This is understandable because the pricing on Airbnb is associated with many subjective factors.

4.2. K-nearest neighbors

We first conducted normal standardization on independent variables excluding description and id. To have a rough idea about the KNN model's performance, a model of 10 neighbors is constructed, where the R-squared is around 53.56%. The predicted prices and actual prices are plotted as shown in *Appendix: Figure 11*. From the graph, there're three things we can conclude. First, our KNN model predicts the price in the correct direction. Second, as long as the actual price increases, the prediction errors' range increases. Another interesting finding is our model under-evaluates houses more than over-evaluates them. Since the major factor affecting the model performance is the number of neighbors. We looped from 1 to 50 for the parameter and checked for these models' performances shown as follows. Besides, the GridSearch method is applied in searching for the best set of parameters. The optimal parameters are {'n_neighbors': 9, 'p': 1, 'weights': 'distance'}, whose performances are shown. The R-squared of this optimal model is around 61.24%.

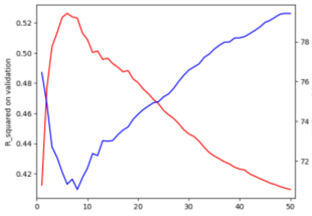


Figure 12: KNN model performances (k=1 to 30)

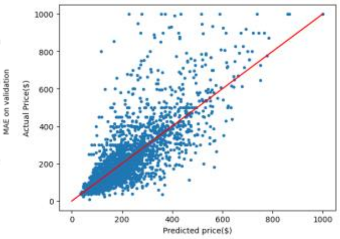


Figure 13:Predicted vs. Actual prices (optimal model)

4.3. Random forest

Since there are more than forty dummy variables for the "neighborhood_cleansed" variable, we use two different methods in the random forest model. The first method is just using these many dummy variables and the second one is using the average value of price in each neighborhood to instead the original string value. Considering its huge amount of dimensions, especially for the first one, we also used the PCA method to reduce the dimension (shown below). We found that its eigenvalue is close to zero after the 20th factor, in other words, the first 20th factors totally account for 95.31% of the variance. So, we used n=20 to do the PCA dimension reduction.

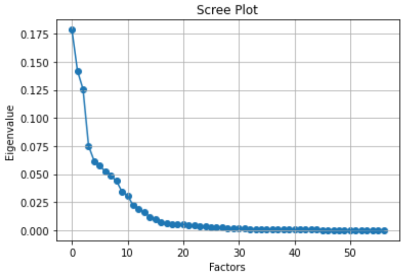


Figure 14: The scree plot of PCA

But his final result was not good, so we did not continue with the PCA. The R-square value of 0.57 was significantly lower than that of 0.66 without PCA.

So, in the case of not using PCA, we selected the best parameters for each method with the GridSearch, they are {'max_depth': 100, 'min_samples_split': 2, 'n_estimators': 400} and {'max_depth': 20, 'min_samples_split': 2, 'n_estimators': 400} for each method respectively.

After comparing metrics such as R-squared and MSE, we found the second approach performed better (Table 6). Besides, the model also helps us analyze the importance of each variable (*Appendix: Figure 15*)

	First method	Second method
R-square	66.78%	68.65%
MSE	9144.66	8630.32
RMSe	95.63	92.90

Table 6: Two method comparison

4.4. Neural network

When fitting a neural network model, we used the 'relu' as the activation function and mean squared error as the measure of loss. Epochs are selected to be 20 because a larger epoch faces great potential of overfitting.

5. Result Insights and Text Mining

From the model performances part, we can see that we only get R-squared values ranging in 0.5 to 0.7, which does not meet our expectation. For potential problems, we have two conjectures: the first one is that the mispricing phenomena indeed exists in Airbnb. Therefore, it is not possible to predict the price of a renting apartment accurately because the prices follow no rule. The second one is that our data set does not include other important features that are useful in predicting prices. For example, we find a specific apartment (*Appendix: Figure 18*), whose description can add great value to its price. This apartment lies in the Flatiron District. It is relatively small that only has one bedroom, one bed, one bathroom, and is suitable for accommodating three people.

Since 'accommodates' is the most important feature in our models, Random Forest predicts this apartment has a price of $205.98, which is even lower than the average price of the Flatiron District. However, the house owner mentioned that his house has a 360-degrees view of New York in the description part. This is a scarce and valuable characteristic of a house and is very attractive to travelers. The actual price of this house is $900, more than two times of our predicted value.

To try to handle these prediction errors, we conduct some text mining analysis on the review part and description part of the apartments. We first analyzed the reviews using LDA and sentiment analysis. What we found is that due to the nature of Airbnb and sometimes the request from the host, the reviews are mostly positive. Over 95% of the reviews are positive, and even for the negative reviews, the tone was close to neutral. For example, some of the negative reviews classified will be positive overall, but criticize certain features like the noise. Due to this, the word cloud for negative reviews has a big portion of positive words, making it similar to the word cloud for positive reviews, thus infeasible to draw many insights.

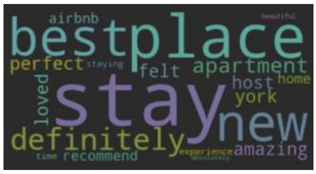


Figure 16.1: Positive Reviews Word Cloud

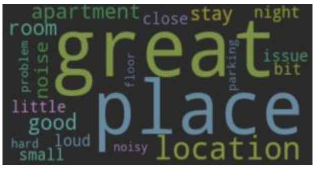


Figure 16.2: Negative Reviews Word Cloud

For analysis on descriptions, we adopted results from earlier models, and separated the data based on the difference between predicted prices and actual prices. We split the data that has a prediction difference greater than 100 and below 100, and visualized different words which might showcase the different focus on different groups.

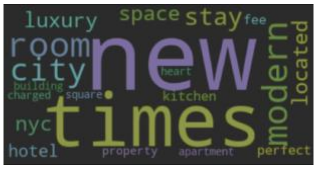


Figure 17.1: Word cloud for difference over 100

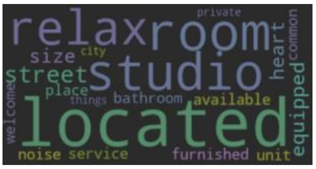


Figure 17.2: Word cloud for difference below 100

We see that even after the removal of "new york", the word "new" still appeared in the word cloud for the higher difference group, indicating that this is an important feature for prices (new units, new furniture, etc.). Also, the word "times" indicated the location near the Time Squares can also have a price fluctuation.

6. Further Improvements

Even from the results of our basic text mining, we can tell some valuable insights in predicting listing prices. Thus, there are mainly three directions for us to improve: (1) search for more comprehensive data of listings to include more variables in the model: such as surrounding landmarks and size, (2) perform more complicated text mining techniques, and (3) incorporate text mining results into machine learning models.

Appendix

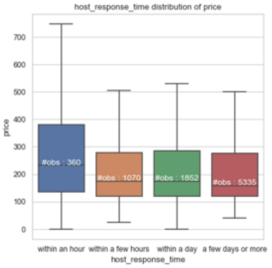


Figure 4.1: host_response_time distribution of price

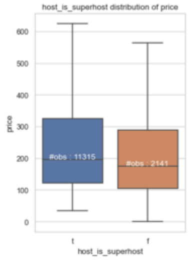


Figure 4.2: host_is_superhost distribution of price

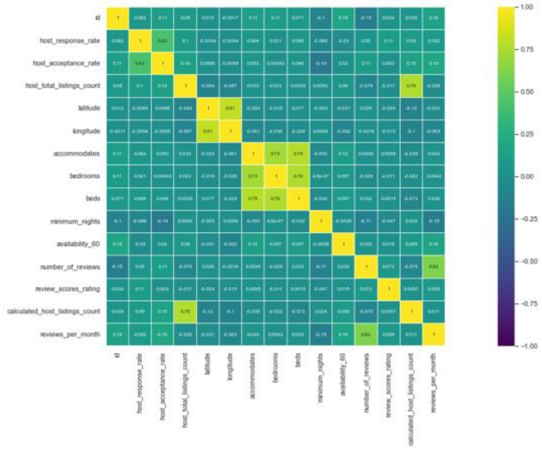


Figure 6: Correlation Matrix

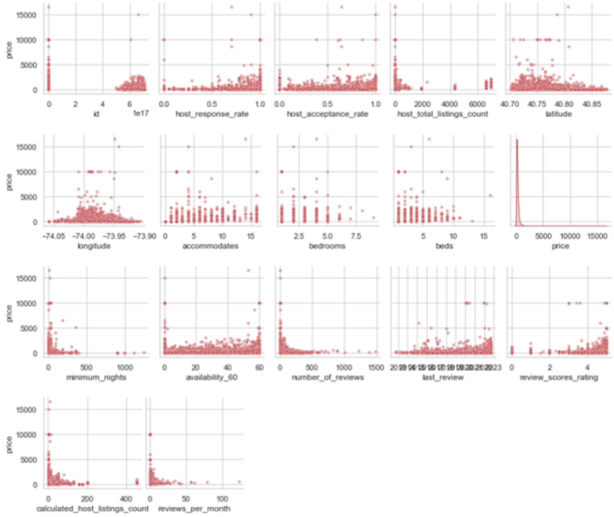
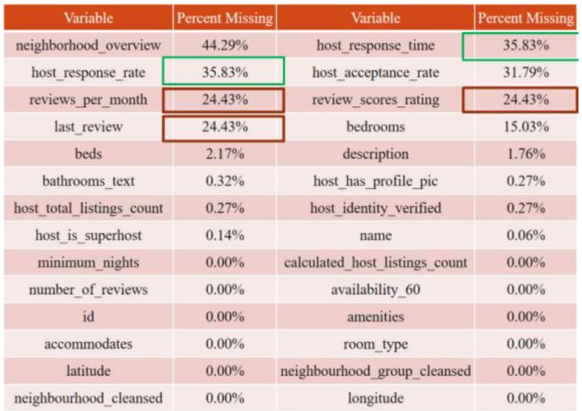


Figure 5.1: Scatter plot of numerical variables & price



Variable	Percent Missing	Variable	Percent Missing
neighborhood_overview	44.29%	host_response_time	35.83%
host_response_rate	35.83%	host_acceptance_rate	31.79%
reviews_per_month	24.43%	review_scores_rating	24.43%
last_review	24.43%	bedrooms	15.03%
beds	2.17%	description	1.76%
bathrooms_text	0.32%	host_has_profile_pic	0.27%
host_total_listings_count	0.27%	host_identity_verified	0.27%
host_is_superhost	0.14%	name	0.06%
minimum_nights	0.00%	calculated_host_listings_count	0.00%
number_of_reviews	0.00%	availability_60	0.00%
id	0.00%	amenities	0.00%
accommodates	0.00%	room_type	0.00%
latitude	0.00%	neighbourhood_group_cleansed	0.00%
neighbourhood_cleansed	0.00%	longitude	0.00%

Table 3: The proportion of missing values for each variable before processing

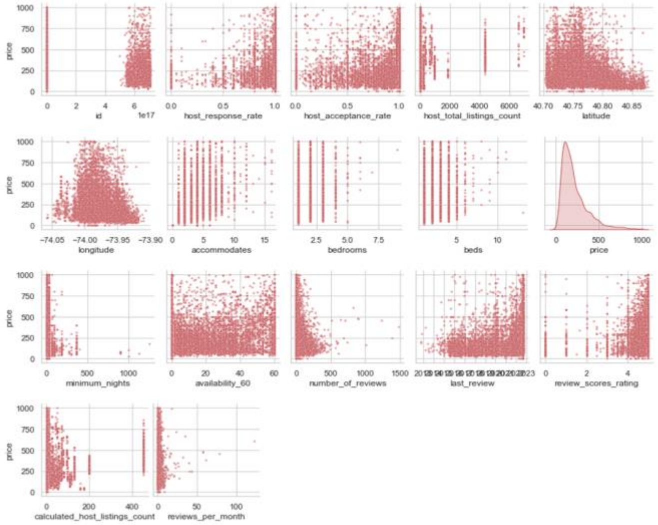
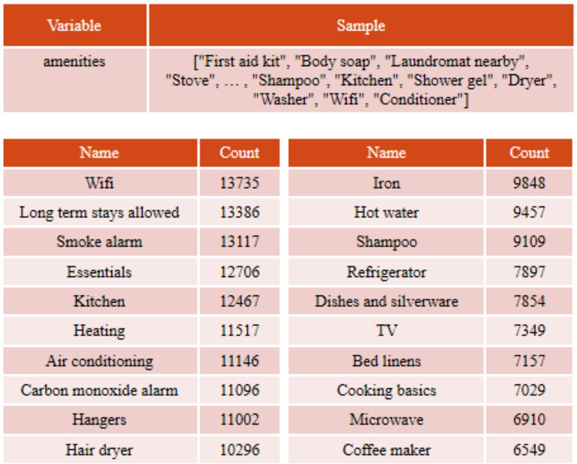


Figure 5.2: Scatter plot of numerical variables & price<=1000



Variable	Sample
amenities	["First aid kit", "Body soap", "Laundromat nearby", "Stove", … , "Shampoo", "Kitchen", "Shower gel", "Dryer", "Washer", "Wifi", "Conditioner"]

Name	Count	Name	Count
Wifi	13735	Iron	9848
Long term stays allowed	13386	Hot water	9457
Smoke alarm	13117	Shampoo	9109
Essentials	12706	Refrigerator	7897
Kitchen	12467	Dishes and silverware	7854
Heating	11517	TV	7349
Air conditioning	11146	Bed linens	7157
Carbon monoxide alarm	11096	Cooking basics	7029
Hangers	11002	Microwave	6910
Hair dryer	10296	Coffee maker	6549

Figure 9: Amenities

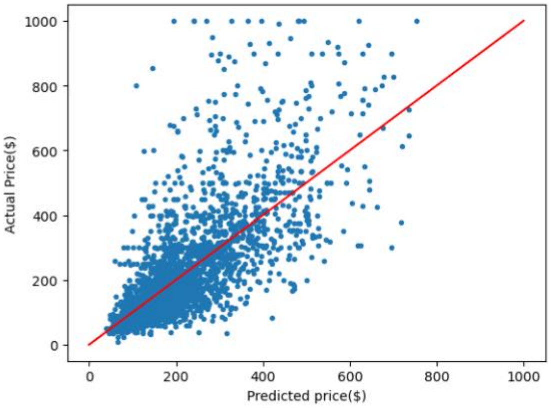


Figure 11: Predicted vs. Actual prices (k=10)

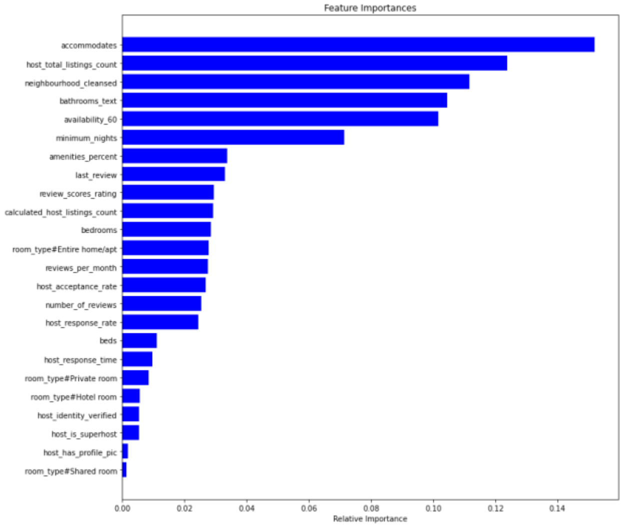
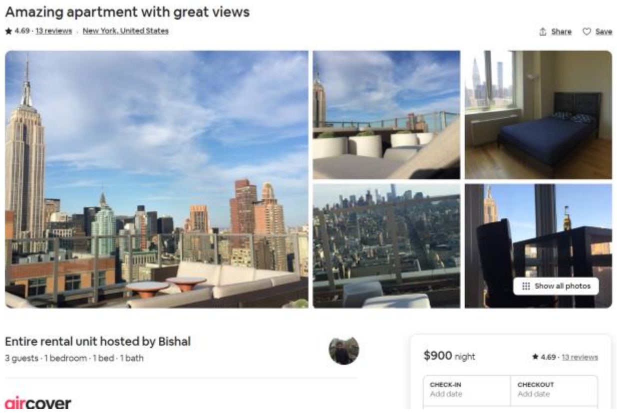


Figure 15: Feature importances in random forest model


Amazing apartment with great views

★ 4.69 · 13 reviews · New York, United States

Entire rental unit hosted by Bishal

3 guests · 1 bedroom · 1 bed · 1 bath

$900 night ★ 4.69 · 13 reviews

CHECK-IN Add date | CHECKOUT Add date

Figure 18: An example of the apartment with large prediction error