New York City Airbnb Price Prediction & Analysis - Methods

Team: Akshi Saxena, Rakshit Soni, Rajat Manish Bhagat

Data Source: http://insideairbnb.com/get-the-data/
Dataset:http://data.insideairbnb.com/united-states/ny/new-york-city/2023-03-06/data/listings.csv.gz
Project Repository:
https://github.com/akshi-saxena/airbnb-price-prediction/blob/main/Improvement_Airbnb_Price_Prediction.ipynb

Data Preprocessing and Exploratory Data Analysis

This step involves performing type conversions, exploring data distribution, detecting outliers, imputing missing values and finally performing transformations on data to make it useful in the construction of machine learning models.

Data Type Conversions

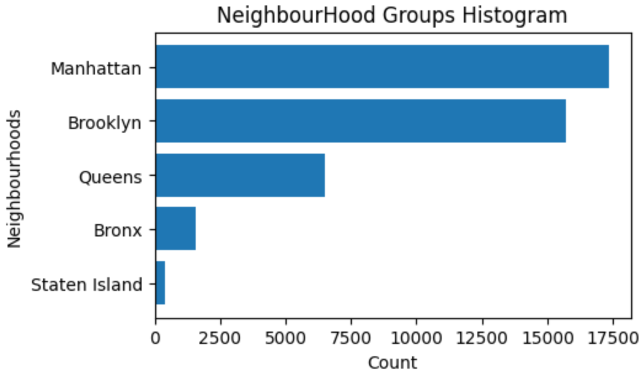
We used the full *listings* (.csv) dataset consisting of approximately 42,000 accommodation listings in New York City from the AirBnB website. The data consists of 75 features, not all of which are relevant. Features describing scraping information *(scrape_id, last_scraped, etc.)* are dropped as they do not provide any information about the price. Some features contained sensitive information about the host *(host_name, host_location, host_about, etc.)*. They were also dropped so as to not include any biases. It is not ethical for a model to be based on any individual.

Binary Features

Some features *(instant_bookable, host_identity_verified, etc.)* were in a textual format with values like *'t'* and *'f'* in the text format of 't' and 'f'. We converted them to binary values. The rate features *(host_acceptance_rate, host_response_rate)* are textual and have a *'%'* sign in them. The price feature also has dollar signs and commas in the text. We converted those values to numeric. The feature *'bathroom_text'* consists of the number of bathrooms in text format (e.g. 1 bath, half bath, 2 shared baths, etc). This feature is converted to a numeric feature.

Features conveying similar information

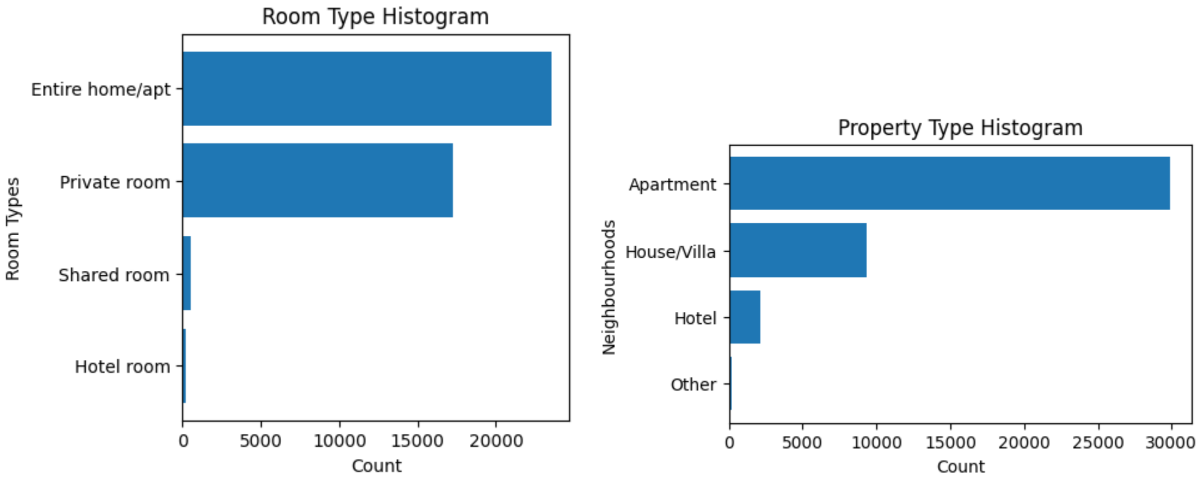
There are columns with the same information - *neighborhood, neighborhood_cleansed, neighborhood_group_cleansed*. The *'neighborhood'* feature consists of text data with area, city, and country (irrelevant as we are only seeing NYC), the *'neighborhood_cleansed'* consists of an area in textual format, and *'neighborhood_group_cleansed'* consists of a larger area grouped together. We decided to use only the *'neighborhood_group_cleansed'* column which has 5 major areas in New York and can be treated as a categorical variable.



The plot shows the number of data points in each neighborhood group in New York City. Most of the accommodations are available in Manhattan and Brooklyn.

Categorical Features

The *'property_type'* feature has information about the AirBnB accommodation type (e.g. shared room in a villa, guest suite, etc.). It had 80 unique accommodation types. To simplify them, we classified them into 4 categories: *Apartment, House/Villa, Hotel, and Other*. We do not consider information about the accommodation being private or shared as it is described by another feature, *room_type* (See Appendix item 1 for classification information).



The above plots show the number of data points in each type of room and each type of property. Most of the accommodations are private apartments or homes. There are very few shared accommodations in the dataset.

Outlier Detection

To detect any outliers, we used boxplots to see the distribution of each numeric column. The plots show we have highly skewed data. (See Appendix item 2 for plots) Our target variable 'price' is right skewed. So we will later transform it into a logarithmic scale.

We then remove outliers using the Inter Quartile Range (IQR) method, by finding upper and lower boundaries. Usually Q1 is 25th percentile and Q3 is 75th but looking at the distribution of our data, we chose Q1 as the minimum feature and Q3 as the 90 percentile. Outliers are 1.5 below Q1 and above Q3.

Textual Features

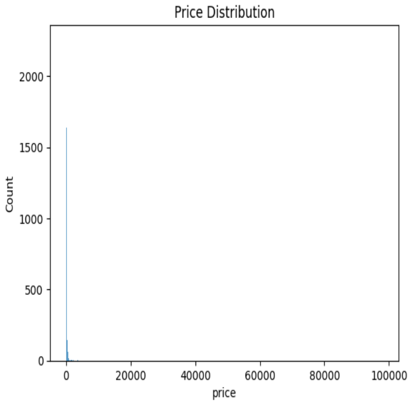
The feature *'amenities'* consists of a string with names of amenities in an accommodation separated by a comma. Most amenities could be grouped under one category or another. So, we mapped the amenities into the following major amenities determined by a word cloud - security (*smoke alarm, carbon monoxide detector*), long-term stay, air conditioning, TV, parking, and kitchen. (See Appendix item 3 for word cloud)

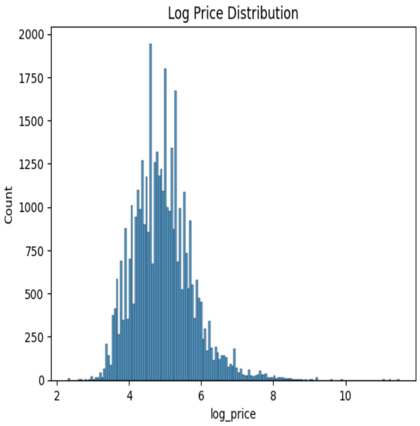
Missing Values

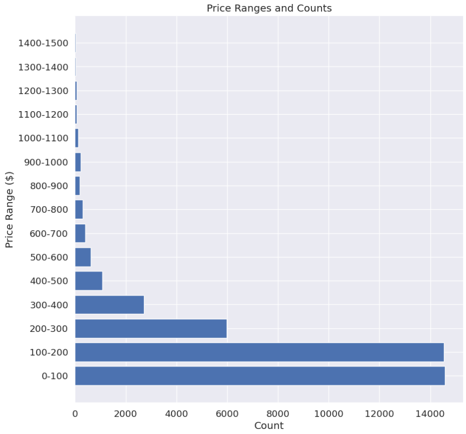
We calculate the percent of missing values for each feature. For numeric features, we use mean imputation, and for categorical variables, we use the most frequent imputation method. (See Appendix item 4 for details table)

Transformations

Upon analyzing the price distribution for the different listings that were present, we found that the prices were heavily skewed and the majority of the prices were in the range of $0 - $300. To counter this, we performed a log transformation on the prices to get a normalized distribution.







Furthermore, we found that the majority of the features that were present as part of the data were categorical in nature. We decided to perform *One-Hot Encoding* on these variables. This gave us more meaningful values which we could use to help make our model perform better.

Features like '*property_type, room_type, neighborhood_group_cleansed, host_response_time, amenities*' were the ones where we felt the need to perform this transformation. For *property_type* we manually decided to break them into 4 separate categories to reduce dimensions (as mentioned in data type conversions) and ensure that we maintain the trend that we have observed in *room_type*.

Feature Importance and Modeling

To see the relationship between price and other variables, we use a correlation matrix. (See Appendix item 4 for the correlation matrix). Refer to the price row for readability we are not so interested in the other variables.

The figure shows the correlation of numeric features. Since we are doing an analysis on which features impact the prices of the Airbnbs, we find that only *beds, bathrooms, and accommodations* have a slightly stronger positive correlation. Others have a very weak relationship with price. We might consider removing the host response and acceptance as well as some review features during modeling.

We then observe the relationship between categorical variables and the price using a correlation matrix. (See appendix item 5) We can see the *host_response_time* (within a day, few hours) has little to no relation with the price. We see that *Entire Home/Apartment, Manhattan, and TV* have the most positive correlation with price.

We are comparing 3 different models and looking at the feature importances to see which predictors are most important for predicting price.

Models

1. Linear Regression

Linear regression is a statistical modeling technique used to predict a continuous target variable, in our case- price, based on one or more input features. The linear model assumes a linear relationship between the input features and the target variable. We first used passed all features to the model. We observed residual plots for each feature as a predictor to see if the variables show a model violation (See Appendix item 8). From these plots, we can see that all have some relation with the target variable. We looked at the coefficients to determine which features were important.

2. Lasso Regression

We use Lasso Regression to address the problem of having a large number of features in the data. We pass all features as Lasso uses L1 regularization and adds a penalty term making some coefficients zero, so it helps with selecting features. Lasso gave a very low R-squared so we tuned it, finding the best alpha value as 0.0001. We observe the features with coefficients more than zero to see which predictors affect this model (See Appendix item 9).

3. Decision Tree

Since our linear models did not perform too well, we used a Decision Tree Regressor. The default parameters of the model did not do well. So we used GridSearchCV to systematically search for the best hyperparameters. We experimented with different depths of the tree, maximum features, and the number of leaf nodes. This improved the model's performance. The best estimator has hyperparameters:

max_depth=100, max_features=20, max_leaf_nodes=100. We then found feature importances which are calculated based on the Gini, ie., how often that feature is used to split the data (See Appendix item 10).

4. Random Forest

Random forest regression is a type of ensemble learning method that uses a collection of decision trees to make a prediction. In the random forest model, multiple decision trees are trained on different subsets of the data, and the final prediction is made by averaging the predictions of all the trees. We fit the model with all features and take a look at the feature importances. The feature importances are calculated based on the decrease in the impurity of nodes that make up the random forest for a particular feature that is used in the split. (See appendix item 11)

Evaluation

We used the following evaluation metrics.

- R squared - The R-squared score is used to evaluate the performance of the model on the test set, which measures how much of the variation in the target variable is explained by the model. A higher R-squared score indicates a better fit.
- RMSE - We then calculate the root mean squared error (RMSE) of the model's predictions on the test set. The RMSE is a measure of the average distance between the predicted values and the true values. Lower RMSE values indicate better model performance.

We did k-fold cross-validation to get an accurate evaluation of the models.

Model	R-squared	RMSE	Cross Validated R-squared	Cross Validated RMSE
Linear Regression	0.629	0.427	0.629	0.426
Lasso Regression	0.629	0.427	0.629	0.427
Decision Tree Regression	0.631	0.426	0.634	0.424
Random Forest Regression	0.741	0.357	0.744	0.354

From the table, we can say that the Random Forest Regression outperformed the other models. Linear Regression and Lasso Regression have similar performance.