AirBnb Kaggle Report

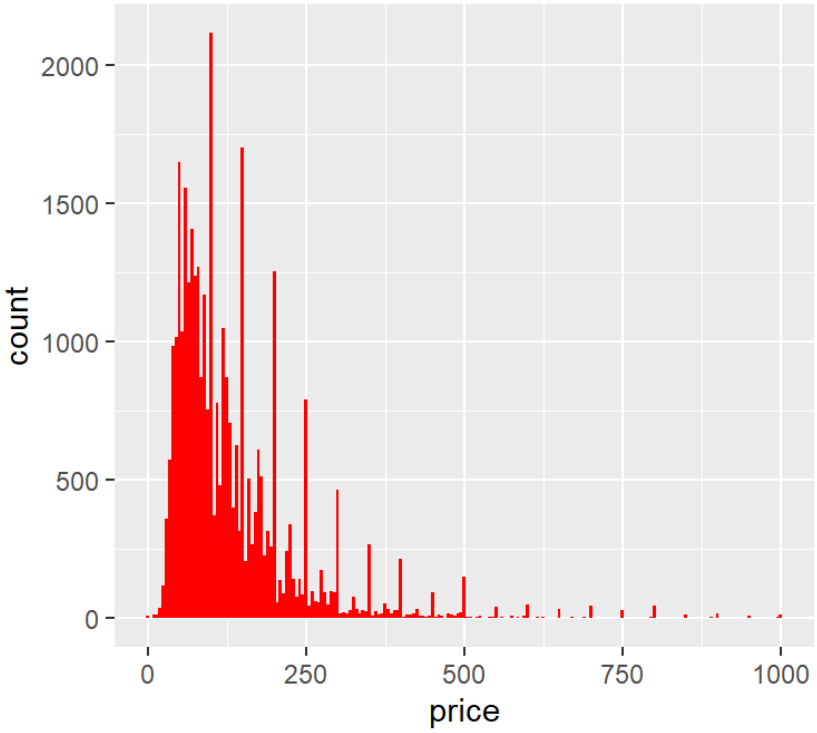
Data Exploration

The first step in data exploration is to examine and clean the data. I started by looking at how to account for the data's null values. I replaced the N/A values with median. To improve the model's performance, the feature levels were reduced. Columns with too much text, such as summary, description, and so on, are then removed. Columns with an excessive number of NA values, such as square feet, weekly price, and monthly price, were also removed. I used correlation as a starting point for discovery and exploration.

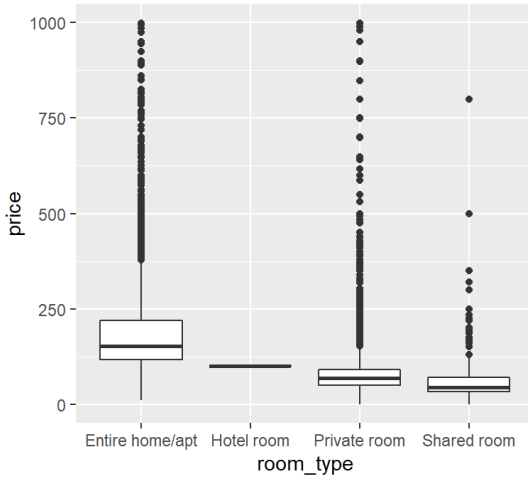
To examine the values in the categorical variable and examining data functions like unique, head, glimpse etc. were used for both test and train data.

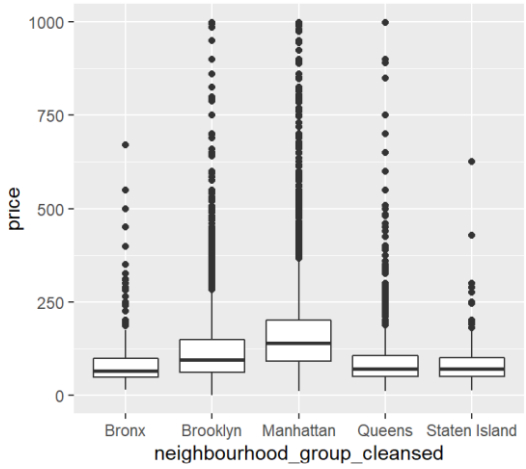
Data Exploration through visualization

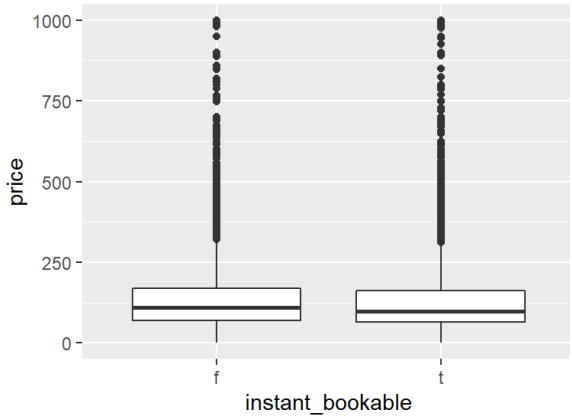
To understand the distribution of price we create a histogram. We can see from the graph that the majority of the properties range between 50 to 250 dollars.

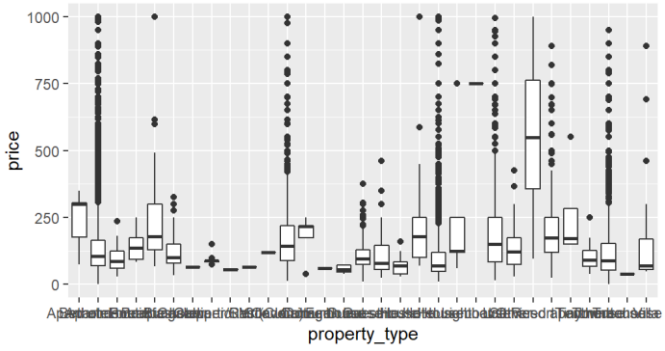


We can use box plots to see if there are any outliers in the data. We can see that there are a lot of outliers in roomtype, instant bookables, property type, and neighbourhood group cleansed.









Feature selection

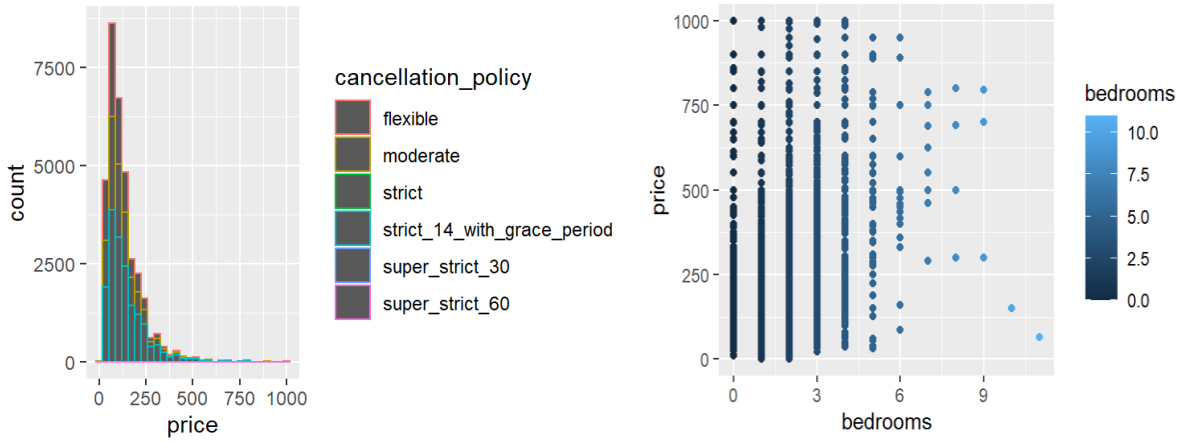
The first step in determining which features will work best for our model was to create a simple linear regression model, which helped me understand the statistical significance of the variables in relation to the price. Another thing I did was look at the correlation. We've already removed the columns with excessive text and NA values. After observing the correlation between the variables. My strategy was to divide the variables into two categories, with a few variables in the must-have category having a high correlation value and performing well when tested on the test data. The optional variables were another category that could be used to mix and match and train the model.

It’s always good to train the model if we have many numeric values. I made transformation on few of the columns such as mentioned below.

Some transformations such as changing percentage to numeric, changing dates to number of days, uniformity across zip codes was done

Host_response_rate was character type initially, I transformed it into numeric format by removing the percentage sign and dividing it by 100.

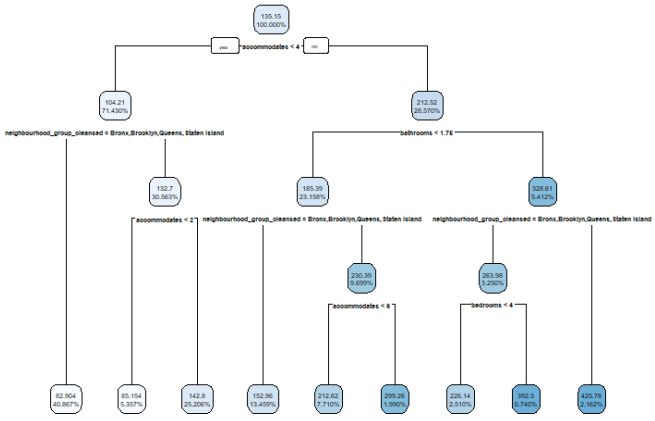
The zip code was a mix of digits and character values, transformation was done to make it uniform and all digits removed the N/A values. Plotted the following visualization to decipher some pattern between the variables and price. We can see that frequency of flexible, moderate and strict_14_with_grace_period is high in the specified distribution. From the other graph we can see the pattern more the number of bedrooms the higher the price



Data Modeling

Simple train and test sampling; initially, all of these variables were trained using linear regression, which gave me an idea of which variables were performing well. Even after using what appeared to be the most accurate variables, the RMSE was only around 80. I concluded that linear regression is not the best model and that something more advanced must be used to train the model in order to achieve lower RMSE.

Then I tried random forest. The following is an example of a tree that was created.



Ultimately the best performing model was XGboost. I tried various multiple modifications on the parameters for the eta, rounds, depth, repeats, subsample to obtain the well performing model

After using XGBoost I was able to get RMSE near 60, on further investigation and adding and removing variables to this model I was able to reach the RMSE of about 59 which was my best score.

Learning

I recognized the significance of data cleansing and preparation. More than just fitting a model, it is critical to prepare the dataset so that the model can be trained properly. I spent a significant amount of time switching back and forth between constructing the model and cleaning the data.

Dropping the columns with lot of text was initially a good practice but there may be useful insights derivable from it which I could use next time.