Predicting Airbnb High Booking Rates

Project Group 5: Akshay Havalgi, Austin Hom, Bekzod Akramov, Shashank Rao, Vivek Ramanathan

Introduction

Over the past decade, Airbnb has become a widely popular company with many travelers using it for accommodation across many locations around the world. Recent statistics show that Airbnb has approximately over 150 million users in more than 65,000 cities. There are more than 5 million listings, and the platform has managed to tap into an inefficient market - rooms or entire homes that are unused and could earn the property-owner some extra income.

In this project, our aim was to predict the booking rates (high or not) of Airbnb listings in the test data set. Through exploring relationships within around 100,000 samples of Airbnb listing data, we tried to better understand the key variables that influence whether a property will be highly booked or not. The end goal is to then use that data and information to guide Airbnb hosts on how to maximize the profitability of their listings. Given that the aim of this project is to best predict whether a listing will have a high booking rate or not, the most important metric to assess the performance of our model is prediction accuracy.

The models we tried were Logistic Regression, Random Forest, and XGBoost. After running these models, we found that the XGBoost model gave us the best prediction accuracy at 84.59% using max_depth (depth of the trees) of 14, nrounds (number of iterations) of 3000 and learning rate of 3.9% on the variables. Although we can’t claim any direct interpretation of the data using XGBoost, it still provided us with a variable importance plot telling us the most influential variables in the model that contributed to the high prediction accuracy. These key variables included the listing’s longitude and latitude, price per guest, whether the property was available to book 30, 60, 90, and 365 days in advance, cleaning fee and minimum nights allowed to stay at the property, among others.

Exploratory Data Analysis

The dataset contained features for the training instances, labels for the training instances, and features for the test instances. Our goal was to make predictions for the instances in the test set. About 70 features were provided for each listing. After understanding all the features, those which appeared to be important were price per guest, cleaning fee of the room, longitude and latitude of the listing, number of bedrooms and bathrooms and the type of the room. Initial exploration of the dataset helped us learn about the data types and structures of the objects in the dataset, as well as missing and problematic data.

Data Cleaning and Feature Engineering

The data had a lot of missing values and a lot of errors & inconsistencies within the way the variables were coded. We systematically went through the data, examined the data type and structure of each variable, and cleaned it. We created two self-made helper functions for cleaning numeric columns and factor columns. NA’s were converted to be the median (numeric variables) or most common (factor variables) of the column manually. We removed redundant columns such as weekly price per guest and data with over 80% missing values such as square feet of the room, jurisdiction names. Lastly, we had one function in R to clean the entire dataset (see complete R code as attached).

We also created several new variables to include in our model:

- NumAmenities - The number of amenities at the listing. We counted the number of amenities present in the amenities column which was given in the form of a list.
- NumHostVerification - The number of host verifications at the listing. We counted the number of host verifications present in the host_verifications column which was given in the form of a list. We wanted to see how easily accessible the host was.
- TransitText - 1 if there is a transit summary, 0 otherwise. We wanted to see if the transit column which was filled had an impact on the high booking rate as compared to rows which did not have transit column populated.
- HouseRuleText - 1 if there are house rules, 0 otherwise. We wanted to see if the indication of rules in the house rules column had an impact on the high booking rate.
- Interaction sentiment- Positive, Negative, Neutral values on the interaction column. We wanted to see if the interaction with the host generated a Positive, Negative or a Neutral sentiment leading to an impact on high booking rate column.

Finally, all transformations done in the training were also done in the test.

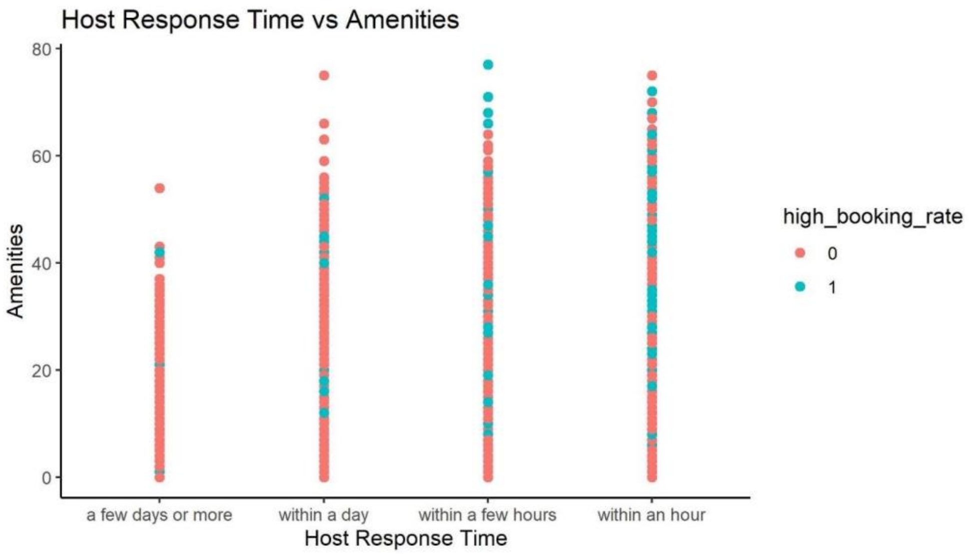
Examining relationships between variables

Based on domain knowledge and research, we explored relationships between:

1) High booking rate and several variables of interest (e.g., host’s response time, price, availability to book in 30, 60, 90, or 365 days, cleaning fee, room type, etc.)
2) Other predictors (e.g., price and host response rate, price and host response time, host response rate and availability, host response time and amenities, etc.)

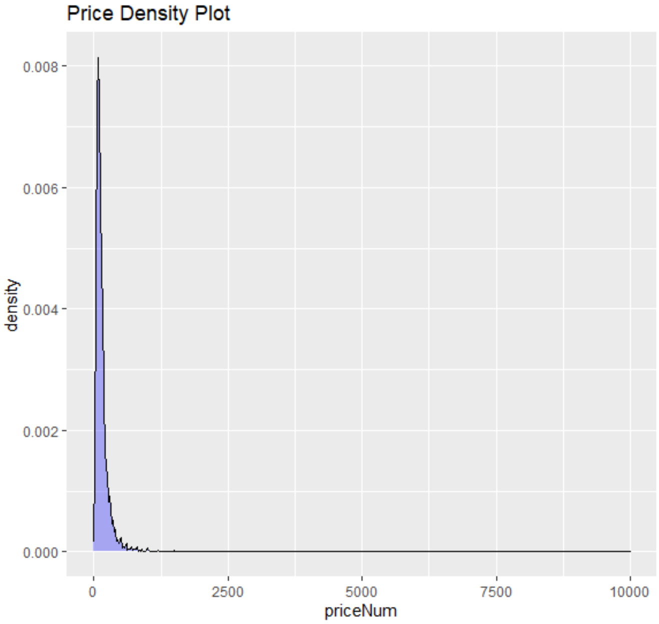
We got a good idea of which variables would be important to include in predictive models (including new variables to create**)**. We created many visualizations to examine the relationships between the variables, some of which are provided below and in the Appendix.

We found that high booking rate listings tended to have prices under $2,500 and cleaning fees under $500 (see Appendix Fig 1.1-1.2). In the scatterplot below, we also see shorter host response times and a larger number of amenities together tend to correlate with high booking rate listings.



Bekzod Akramov

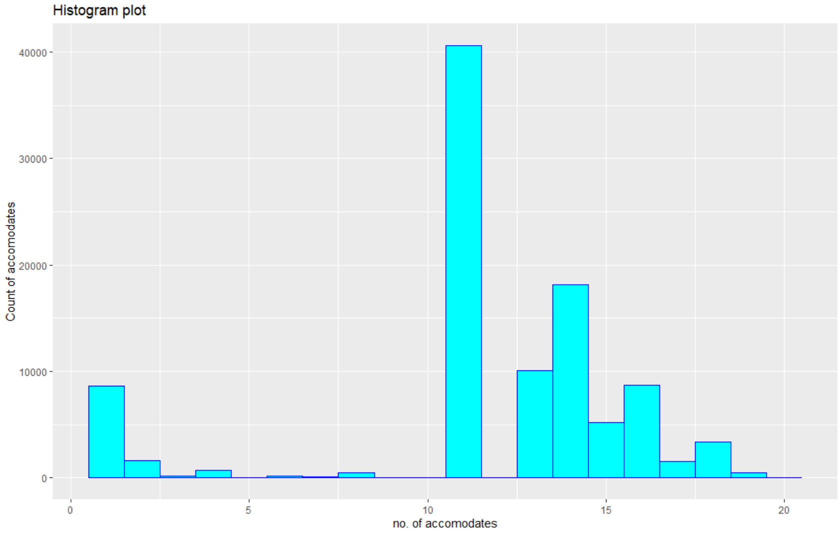
From the density plot below, we can see that the daily price of each listing is heavily right skewed. In a right skewed distribution, the mean is higher than the median. The min and max price don't affect our median. The majority of listings' room type belong to 'Entire home/apt' followed by 'Private room' and 'Shared room'. The variance of price for 'Entire home/apt' is higher compared to those of private room and shared room (see Appendix Fig 1.3).



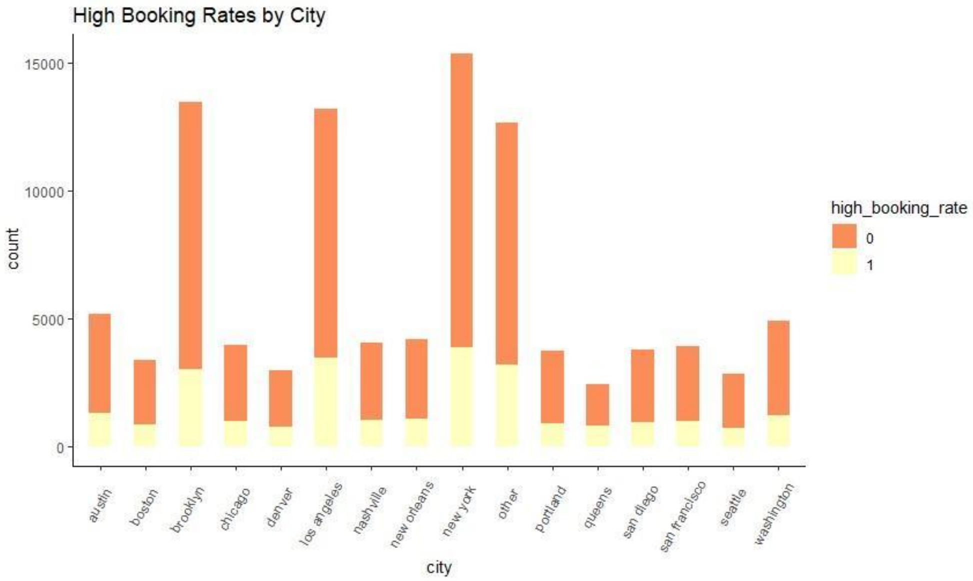
Austin Hom

As can be seen in the histogram below, a large majority of the listings can accommodate over 10 guests. We also found that high host response rate and availability in a smaller number of days (denoting a higher demand) is correlated to being a high booking rate listing (see Appendix Fig.

1.4). We also found that some cities (such as Brooklyn, Los Angeles, and New York) and some states (New York and California) (see Appendix Fig 1.5) have a relatively higher number of listings in the dataset and they also have a higher number of high booking rate listings.

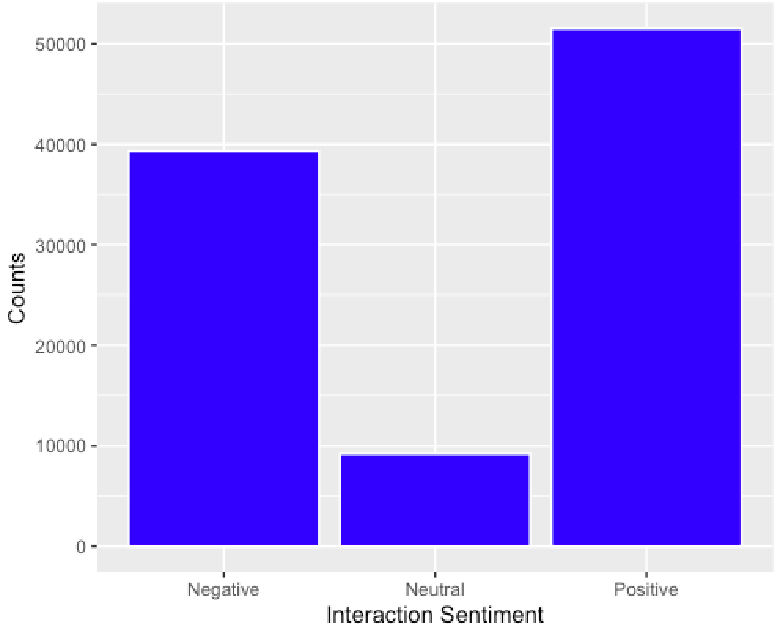


Vivek Ramanathan



Shashank Rao

We also found that most of the security deposits are concentrated in the range of $0-$500, with the average security deposit being around $180-$200. We also saw that most of the security deposits fall below $250 and rarely exceed $1000 (see Appendix Fig 1.6). The interaction column in the data represents the conversation between the host and the potential client. We decided to do a sentiment analysis on the interaction column to explore whether a potential client's first interaction with the host was positive, negative or neutral and whether the initial interaction has an effect on the high booking rate target variable. We have also classified null values as negative since we assumed that there was no interaction with the host. In the bar plot below, we can see that more than half of the interactions were positive.



Akshay Havalgi

Modeling, Results and Evaluation

Once the data was cleaned and imputed and the features were engineered, we moved on to modeling. We started off with a simple Logistic Regression model that included 43 predictors. We split the dataset into a 75 percent train and 25 percent validation. This Logistic Regression model produced a validation accuracy of 78.84%, which was a good baseline for us to improve upon.

Given the large number of instances in the training data, our next approach was building a classification based Decision Tree. These are flexible, robust to outliers and require very little data preparation. However, these are prone to overfitting. To reduce this, we converged to ensemble methods. We wanted to explore bagging methods. If a feature is considered very important, then this feature may be present in all the trees of our bagging model and result in high variance. To counter this, we chose to implement a Random Forest model. Here, each split is done with a randomly selected subset of features denoted with the hyperparameter ‘mtry’. This value was chosen as ‘7’ as this approximated to the square root of the total number of features we had in our cleaned dataset - ‘43’. Several numbers of trees were fit on the training data and were evaluated on the validation data. The optimal tree count was found to be ‘1000’. This high number of trees also ensured that we had a low bias, low variance model. This model resulted in an accuracy of 83.70%, which was a major improvement over the Logistic Regression model.

Looking at the resulting accuracy of the Random Forest model, we wanted to increase the accuracy further without overfitting. This was possible by using a boosting algorithm. However, as there were many tree boosting algorithms available, we wanted to select the right one for our dataset. XGBoost uses Newton tree boosting which has Hessian matrix, a higher-order approximation, which plays a crucial role to determine a better tree structure compared to other tree boosting algorithms. Hence, we used the XGBoost model on our dataset and obtained very good accuracy values. After performing grid search for determining the best hyperparameter values we used max_depth (depth of the trees) as 14, nrounds (number of iterations) as 3000 and learning rate as 3.9% which are the main hyperparameters for XGBoost to obtain the best accuracy from the model. In addition, we used a cutoff of .43 for this model as it increased the accuracy on the validation dataset. This XGBoost model resulted in an accuracy of 84.59%.

Given that our goal was to predict the listings with high booking rates, the evaluation metric we focused on was accuracy, which is the ratio of total correctly predicted to the total actual values. As we can see in the table below, XGBoost model produced the highest accuracy, followed by Random Forest.

Model	**Accuracy**
Logistic Regression	78.84%
Random Forest	83.70%
XGBoost	84.59%

We chose XGBoost to be our final model primarily because it gave us the best accuracy out of all the models we tested. Additionally, the XGBoost model provides feature importance of all the features which is very useful to answer feature importance questions. It also provides a better tree structure compared to all other tree based ensemble methods. XGBoost also includes an extra randomization parameter, i.e. column subsampling, which helps to reduce the correlation of each tree even further. The XGBoost model also performs automatic feature selection and captures high-order interactions without breaking down. After understanding all the advantages of using the XGBoost model, we finally chose it to be our final model.

As seen from the relative feature importance plot (see Appendix Fig. 1.7), longitude and latitude are the most important features which we too had assumed during the exploratory data analysis because the geographical location of the rental is very important to predict whether it will have a high booking rate or not. Surprisingly, the number of bedrooms, bathrooms and type of room do not appear to be very important contrary to what we had assumed earlier.

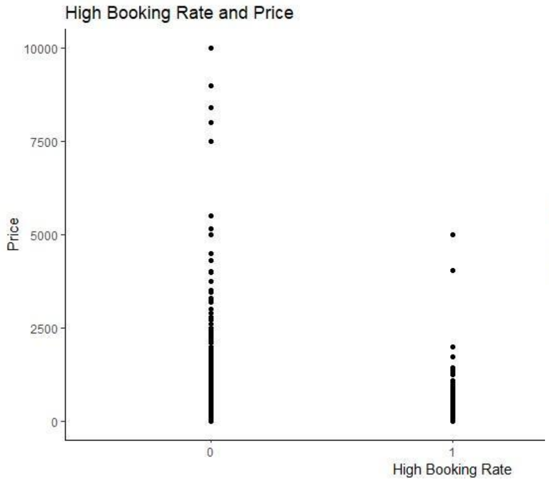
Conclusion

So, what is the business value of predicting high booking rate listings?

Our XGBoost model helps determine whether the host’s listings are predicted to have a high booking rate, as well as suggest which features the host can improve to reach a high booking rate. For example, we can advise the host that responding to potential clients in an hour instead of a few hours will boost the likelihood of having a high booking rate. Overall, we would also advise to increase the days prior to the booking date that the property is available to rent, to respond to as many potential customers as quickly as possible, and have positive interactions with potential clients. Based on the results and findings, the host can decide to sell the listing, rent out the listing to a potential tenant, remove the listing, switch to a competitor of Airbnb to list the same listing or reach out to Airbnb to help promote the listing for a small fee.

From the perspective of Airbnb, it helps them get an idea of how their listings are affected given the various features and what demand may look like for certain types of listings, which they can use to target the listings that are not popular by promotion.

APPENDIX (More visualizations)



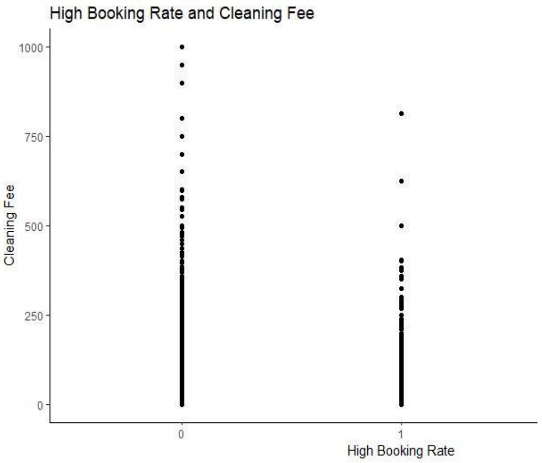


Figure 1.1

Figure 1.2

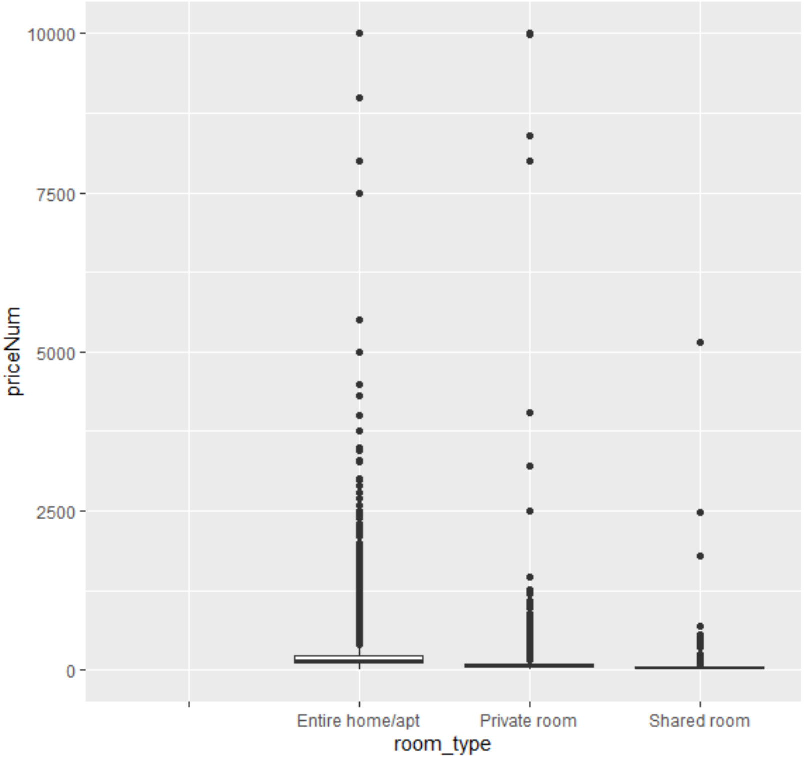


Figure 1.3

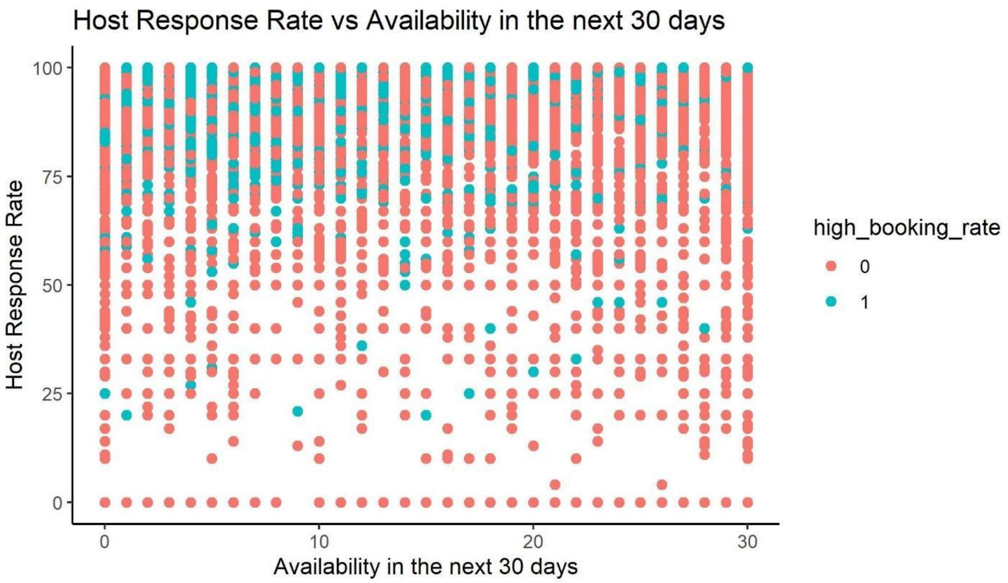


Figure 1.4

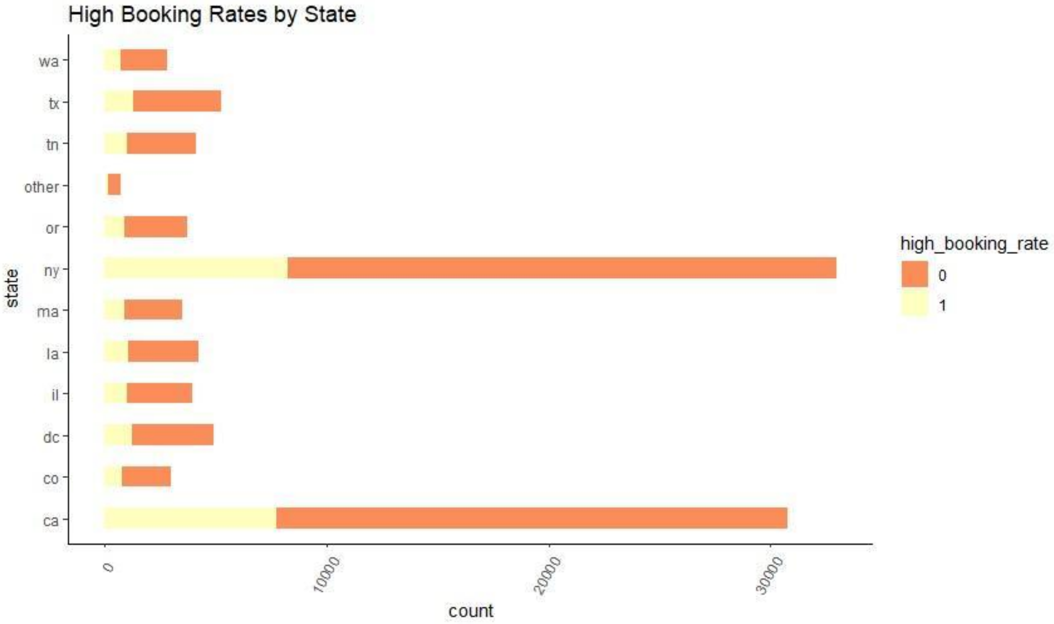


Figure 1.5

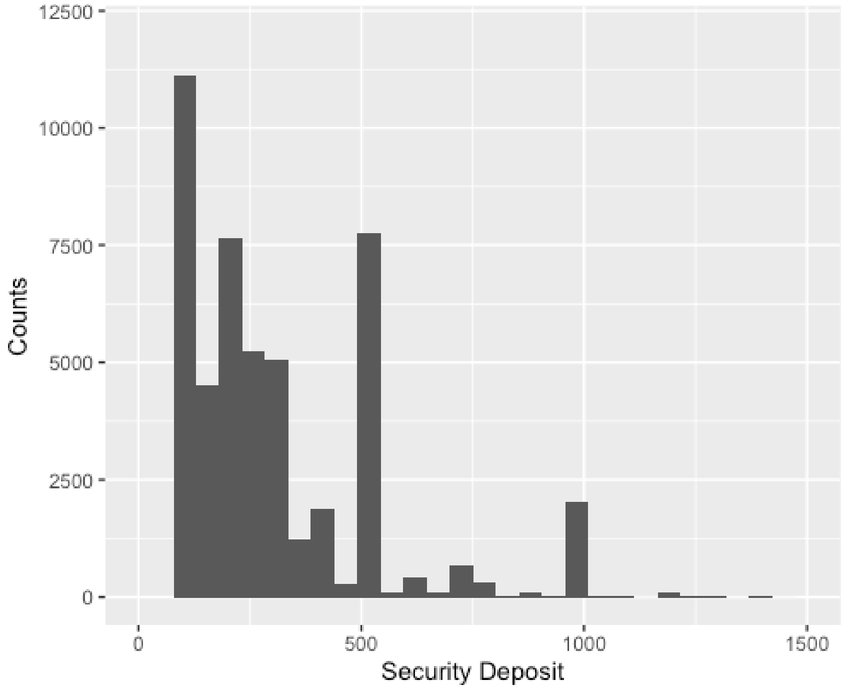


Figure 1.6

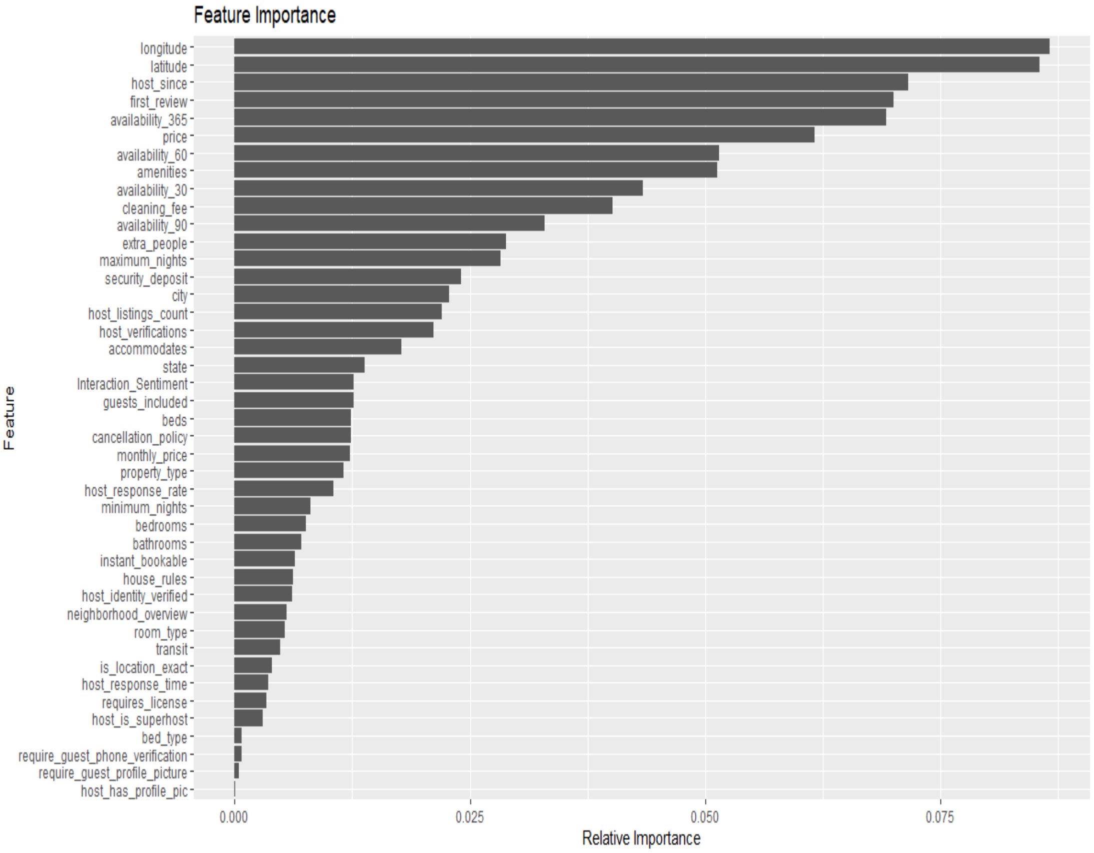


Figure 1.7