Airbnb High Booking Rate Predictions

1. Executive Summary

Airbnb is a global platform that gives everyday people the ability to list their properties online, ranging from full homes to spare rooms, on a marketplace. Even though Airbnb's 31 million users choose this as a less expensive alternative to staying in a hotel, the company is still widely successful, valued over $31 billion in 2017. With over 4 million listings worldwide, it is difficult to optimize how each one advertises themselves on the website. By finding relationships within 45,000 samples of Airbnb listing data, we hope to better understand the key factors that influence whether a property will be highly booked or not. The final intent is to then use that information to advise Airbnb hosts on how to maximize their listings' success. Since the goal of this project is to best predict whether a listing will have a high booking rate or not, the most important measurement to determine the success of our model is its prediction accuracy.

The models we tried were ensemble (bagging, boosting, random forest), KNN, LASSO and logistic regression, and neural network. After running these models, we found that a random forest gave us the best prediction accuracy at 85.53% using 3200 trees and 12 splits on the variables, which beat the baseline. Although we can't claim any direct interpretation of the data using a random forest, 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 were the host's response time, whether the property was available to book 30, 60, 90, and 365 days in advance, minimum nights allowed to stay at the property, the market, the longitude and latitude, price per guest, and the size of description.

Since we can't claim direct inference but we know the most important variables, we decided to run a logistic regression using the top 15 predictors identified in the variable importance plot from the random forest to see the directional influence between the variables and how each of them individually affects whether or not a property listening will have a high booking rate. From this logistic regression, we found that a host response time in a faster category, and the availability to book the property at least 60 days in advance has a positive influence on whether a property would have a high booking rate.

Our recommendation to Airbnb is to take our models and have a software engineering team add a new feature to the Airbnb host dashboard. This feature would use our random forest model to determine whether the host's listings are predicted to have a high booking rate, and then use our logistic regression model to suggest which features the host can improve to reach a high booking rate. For example, the feature can advise the host that answering user messages in an hour instead of a few hours will boost the likelihood of having high booking by x%.

The most general advice we can give to property owners is to increase the days prior to the booking date that the property is available to rent, to respond to potential customers as soon as possible, and have longer descriptions with bigger words. The Airbnb host relations team can also use this information in the host guideline packet.

2. Exploratory Data Analysis/Feature Engineering:

The dataset provided has several variables, many of which we intuitively believed to be very important to predicting high booking rate properties, such as ‘accommodates’, ‘host_is_superhost’, ‘market’, ‘price’, and ‘room_type’ before running any analysis. Before attempting to run any analysis or any model, we decided it was best to clean our data as much as possible. The better the data, the better the model.

Several numeric variables had NAs in them; we replaced them with numbers that logically made sense (for example, 0 for cleaning fees and 1 for minimum nights). R would automatically make a new category for categorical variables that had NAs (for example, all NAs in host_response_time would get their own category, signifying hosts that have either never been contacted or never responded to the contact).

There were 2 more types of variables - categorical lists and plain text. For the categorical list of amenities, we made 10 new binary variables, each representing whether or not a listing has an amenity or not. For example, ‘wifi’ indicates whether the property has WiFi or not. The R code for this and all other data cleaning/transformation is in Appendix C. The 10 amenities were ones we felt would be the most important of the lot. They didn’t end up being significant, which in a way is helpful for property owners to know because even if they have a lot of notable amenities, there are more significant ways to spend their time and money to increase their booking rate.

For plain text variables, we performed some very basic text analysis. We replaced ‘description’ with 2 variables - one for the length of the description (in number of words), and average length of each word (in number of letters). For summary, we only did average length of each word, since summaries have a length cap imposed by Airbnb. The rationale behind choosing average length of each word was to see if having longer words such as “exuberant” or “fantastic” affects high booking rate. Ultimately, our efforts paid off and the description variables made the top 20 in the variable importance plot.

Lastly, we added new variables to augment the data. These were variables such as price per guest (‘price’ divided by ‘guests_included’), number of beds per guest, number of bathrooms per guest and such. The rationale was that these were factors that logically affect the quality of a listing, and they were easy to infer from the given data. Again, our efforts paid off and most of these variables made the top 20 list.

Another decision we had to make was whether to include all variables in running our models, or just the ones we intuitively felt would be useful. We decided that even though we believed several variables in the original data set wouldn’t be significant, such as ‘availability_365’ and ‘longitude’ and ‘latitude’, we would still include them in the initial runs of each of the new models. If the variables truly aren’t significant, at least a majority of the models should remove them and we would have statistical evidence to support why they weren’t included in the final model.

3. Model Evaluation

When making the initial ensemble models, we chose to partition the data using a 75-25 training-testing split. For the neural network, we partitioned the data using a 75-25 training-testing split as well. When making the initial model for the lasso regressions, we partitioned the data to have 75-25 training-testing, and then took another 25-75 split of the training data to get the validation-training samples.

The goal of this project is to predict high booking rate for a property. The most important metric we used was accuracy. Looking for inference was a factor in model evaluation, but ultimately wasn't what we used to determine the most successful model.

The baseline we chose to compare our model against is that every property is classified as the most common case, which is 0, meaning it is a low booking rate property. We found that 74.8% of the training data is in the '0' case - giving us a baseline of 74.8% accuracy to compare to when classifying each property. We chose this baseline as with no model our training data has this split, and only a model with an improved accuracy over this baseline would be worth using to predict.

Most of our models did beat the baseline (KNN and lasso coming in around 78% to 80% and random forest at 83%), although a few initial iterations of the neural network did not (about 69% to 72%). When it came to choosing which model to fine-tune and use to predict on the unseen test data, we went with the random forest since its initial iteration which had a relatively fewer number of trees still beat all the other models. We briefly considered an ensemble of all of our models, but our computational capacity was restricted, and we decided to try and improve the random forest as much as possible.

4. Modeling

In order to get the best picture of this data, we chose to try multiple modeling types using R Studio, which were ensemble (bagging, boosting, random forest), KNN, LASSO and logistic regression. We also chose to use Radiant to create a neural network. We tried these models because each of them offers at least one slight advantage over the other, although they also have their individual disadvantages.

The ensemble methods allow us to run a huge number of models to get a representative average of the data. Boosting allowed us to reduce the errors of our predicted data points, which we thought would help increase accuracy. Bagging allowed us to see how different variations of the trees affected the predicted outcome. Although both these procedures are useful, the ensemble method that we predicted to have the best prediction power was random forest because it encompases many benefits specific to our data set.

The randomness of the random forest allowed us to de-correlate the trees to ensure there was no one dominating variable in each tree variation, which gave us a better glimpse of each variable's influence through the variable importance plot. Before beginning, we had a gut feeling that the random forest would perform the best, given that the data is gigantic with several unrelated variables, and the random forest makes fewer assumptions about the data than regression on KNN, and it implicitly accounts for "interactions" (in a way, since splitting on X2 after splitting on X1 accounts for their combined effect). The only downside of these three procedures (RF, bagging, boosting) is that there is no way to get a direct interpretation of the variables or inference among their relationships.

KNN is useful because it is so data-driven, which is helpful when you have large data sets. But since there are so many variables causing high dimensionality, the distance between the variables may be too large to give us an accurate or useful prediction. KNN is also very affected by mistyped entries, which could pose another problem considering we had to replace the data points with a N/A in the columns with a numeric alternative. It implicitly assumes that all the high booking rate properties will be "close-by", which may not necessarily be true.

LASSO regression gives us coefficients to provide inference between the variables and also may take away some of the variables with low coefficients, which could be helpful considering there were 50 variables in our modified data set. We decided to only use lasso regression instead of both lasso and ridge because of lasso's potential to reduce the number of variables and to decrease runtime. Logistic regression also allows us to see the variable coefficients and gives us a more simplistic view of what the data looks like without a lot of transformations and changes to the original data set.

We used Radiant to run the neural network with different variables and sizes to check the prediction accuracy in R. The neural network gives us the flexibility to fit the highly complex data which we can use for a supervised and an unsupervised problem. We initially ran the model with all variables but that proved computationally infeasible, so we ended up using several neural networks only a few variables each. Moreover, some variables had too many factor levels which Radiant was unable to handle. These may have been a cause of poorer performance from the neural network.

As we stated earlier, the modeling type that resulted in the highest prediction accuracy was the random forest, which we predicted would give us the best results. Since there are so many variables, the random forest procedure looks at the influence and importance between many variables by randomly choosing the variables used in each tree and then averaging the trees together to give us a holistic view of the data. Ensemble methods are generally known to have high predictive ability which is why we originally decided to try these modeling types. They are also less likely to over fit compared to other modeling types. After finding the most important variables from the random forest's variable importance plot, we ran a logistic regression using only those variables as predictors to get a better idea about how each of them affects the response variable.

When we ran the logistic regression with the most significant variables, we found that the categorical variable ‘market’ made the model a lot more complicated and all the individual markets other than New York were insignificant. We decided to not include this in this logistic model because the main point of running this model was to gain inference as to the direction of the effects of the predictors, and only one of the several markets seemed to have a significant relationship.

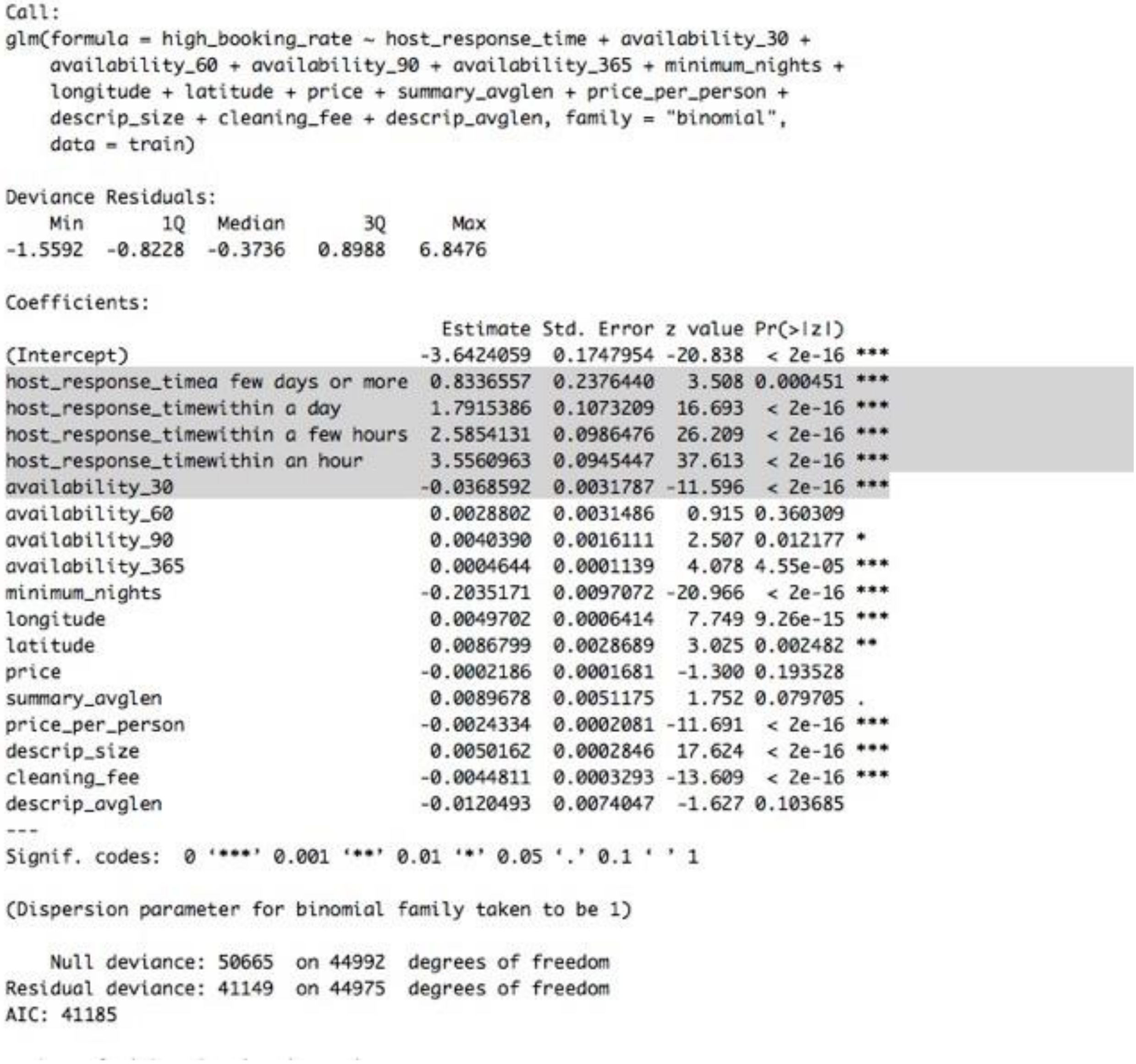
From the logistic regression, we found that if the host response time was within a few days, we expect odds of a property having a high booking rate to increase on average by a factor of 2.29 (e^0.83). If the host has an average response rate of within a day, the odds are expected to increase by an average of 5.99 (e^1.79). If the host’s average response time is within a few hours, we expect the odds to increase by a factor of 13.19 (e^2.58). And lastly, if the host’s average response time is within one hour, we expect the odds to increase by a factor of 34.81 (e^3.55).

This information is very valuable when it comes to telling property owners the most significant variables that can affect their booking rates. If they respond fast to potential customers, the quickest being within one hour, it increases the odds that the property will have a high booking rate by a factor of 34.81 (compared to the hosts which have never responded or never been contacted), which is a very significant increase in probability.

Appendices

Appendix A:

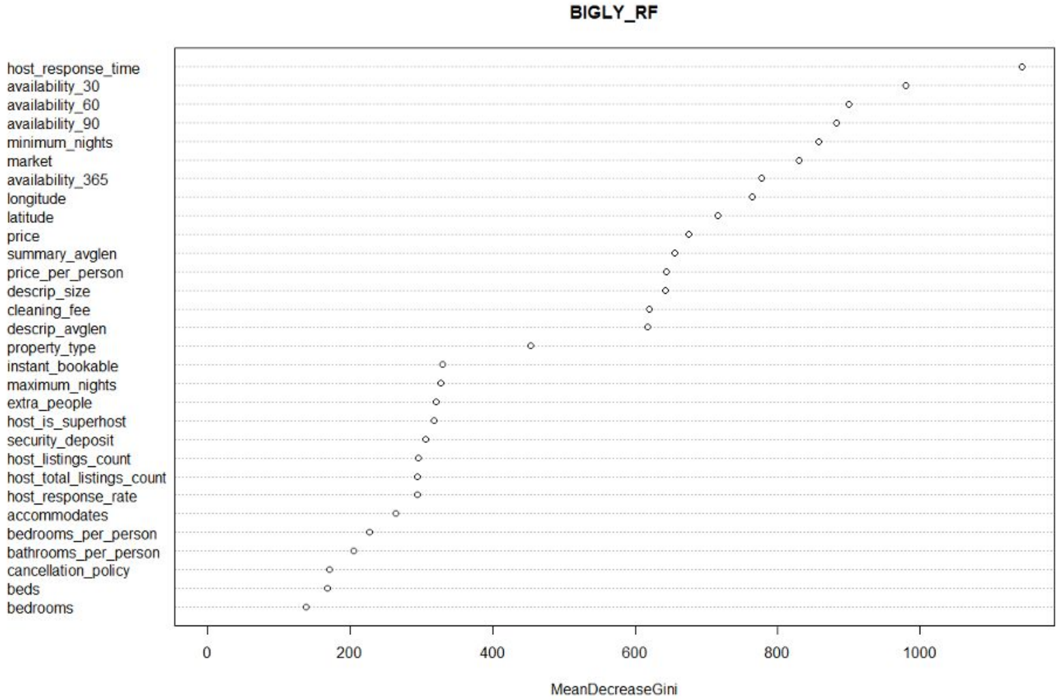
Logistic Regression using the most significant variables given by the variable importance plot in the random forest model.



```
Call:
glm(formula = high_booking_rate ~ host_response_time + availability_30 + 
    availability_60 + availability_90 + availability_365 + minimum_nights + 
    longitude + latitude + price + summary_avglen + price_per_person + 
    descrip_size + cleaning_fee + descrip_avglen, family = "binomial", 
    data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.5592  -0.8228  -0.3736   0.8988   6.8476  

Coefficients:
                                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)                         -3.6424059  0.1747954 -20.838  < 2e-16 ***
host_response_timea few days or more  0.8336557  0.2376440   3.508 0.000451 ***
host_response_timewithin a day       1.7915386  0.1073209  16.693  < 2e-16 ***
host_response_timewithin a few hours  2.5854131  0.0986476  26.209  < 2e-16 ***
host_response_timewithin an hour     3.5560963  0.0945447  37.613  < 2e-16 ***
availability_30                     -0.0368592  0.0031787 -11.596  < 2e-16 ***
availability_60                      0.0028802  0.0031486   0.915 0.360309    
availability_90                      0.0040390  0.0016111   2.507 0.012177 *  
availability_365                     0.0004644  0.0001139   4.078 4.55e-05 ***
minimum_nights                      -0.2035171  0.0097072 -20.966  < 2e-16 ***
longitude                            0.0049702  0.0006414   7.749 9.26e-15 ***
latitude                             0.0086799  0.0028689   3.025 0.002482 ** 
price                               -0.0002186  0.0001681  -1.300 0.193528    
summary_avglen                       0.0089678  0.0051175   1.752 0.079705 .  
price_per_person                    -0.0024334  0.0002081 -11.691  < 2e-16 ***
descrip_size                         0.0050162  0.0002846  17.624  < 2e-16 ***
cleaning_fee                        -0.0044811  0.0003293 -13.609  < 2e-16 ***
descrip_avglen                      -0.0120493  0.0074047  -1.627 0.103685    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 50665  on 44992  degrees of freedom
Residual deviance: 41149  on 44975  degrees of freedom
AIC: 41185
```

Appendix B:

Variable importance chart from the random forest model, used to determine the most significant variables to include in the final logistic regression.


BIGLY_RF
host_response_time
availability_30
availability_60
availability_90
minimum_nights
market
availability_365
longitude
latitude
price
summary_avglen
price_per_person
descrip_size
cleaning_fee
descrip_avglen
property_type
instant_bookable
maximum_nights
extra_people
host_is_superhost
security_deposit
host_listings_count
host_total_listings_count
host_response_rate
accommodates
bedrooms_per_person
bathrooms_per_person
cancellation_policy
beds
bedrooms
MeanDecreaseGini