# Predictive Modeling on Airbnb High Booking Rate

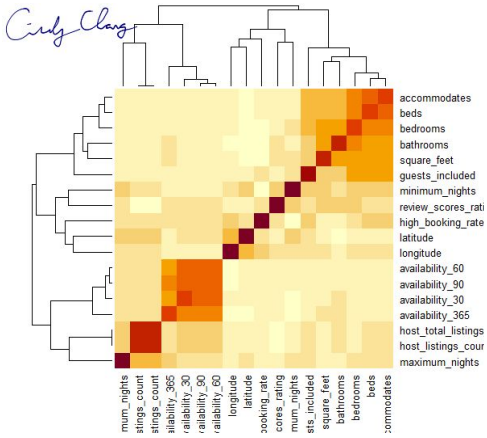
Group 7: Aishwarya Bhangale, Cindy Chang, Huyen Nguyen, Jiakun Luo, Wenjing Cui

## 1.INTRODUCTION

With more than 7 million listings around the globe and 660000 listings in the U.S, Airbnb's is built on unbeatable match-making service, namely, providing quality listing recommendations for guests and also helping hosts promote their property. However, recommending listings to guests becomes especially difficult for a brand new listing with very few reviews, ratings, or booking history to refer to. If a host lists a new house, it is also hard for Airbnb to suggest "smart pricing" for hosts due to lack of historical data. Therefore, Airbnb needs to identify which new listing will likely be popular using the basic information of the listing so that they can make listing and pricing recommendations accordingly.

This study aims to help Airbnb build a machine learning model to predict whether a listing will have a high booking rate(popular) or not by using features such as number of bathrooms, property type, room type, and etc. Three types of machine learning algorithms: logistic regression, random forest, and XGBoost are compared. The study uses accuracy as the metric to determine the best model, which is the XGBoost model.

## 2.EDA




Before cleaning and transforming the data, we briefly examine the dataset. The correlation matrix of numerical variables suggests that *accommodates, beds, guests_included* might be highly correlated to the label variable, *high_booking_rate*. *Host_listings_count* and *host_total_listings* are highly correlated, so we drop *host_total_listings* from our model. *Availability* variables are highly correlated with each other, thus in the feature engineering section, we extract their individual effects by subtracting overlapping availability time windows.

## 3.FEATURE ENGINEERING

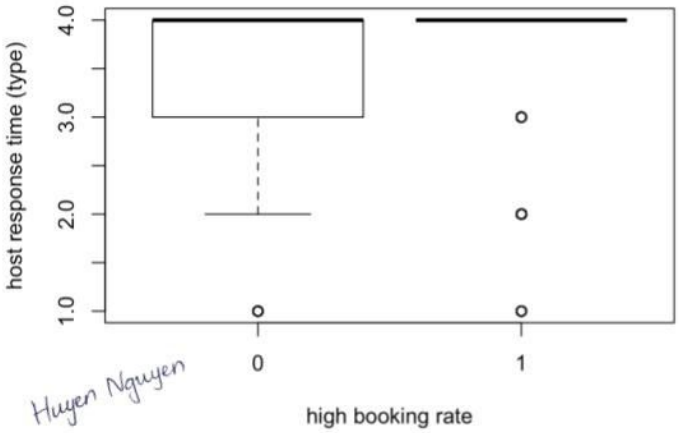
### I) Data Preprocessing Principles:

- ❖ Missing values are imputed with the mean for the numerical variables, while they are replaced with the most frequent classes for the categorical variables.
- ❖ For numeric variables, the minimum and maximum values are adjusted for the purpose of maintaining high quality of data.
- ❖ All categorical variables are one-hot encoded.
- ❖ Information of all variables is reserved as much as possible.

**II) Notable Features:**

A total of 100 features were used, and the full list of features used in our final model can be found in Appendix1. The below are notable mentions:

**a. Host_response_time(top 10 important features)**




The "host_response_time" variable states how long it takes a host to reply, which has four levels. The level will be higher if a host responds within a shorter time period.

The box plot shows a clear relationship between host response time and whether the listing has a high booking rate. Note that the quickest response (within an hour) is denoted as number 4. Almost all of the hosts who have high booking rates respond within an hour. This variable ended up as one of the top 10 important features in the XGBoost model.

**b. Density bins**

Location is expected to have an impact on the booking rate, since listings in big cities usually get high occupancy rates. We augment pre-existing features "zipcode" with geographical information "population density" obtained from external databases to create a new feature, "density". After that, we create a new column named "density_bins" by cutting column "density" into 10 bins by the magnitude of its values. With this new variable, we take into consideration the effect of location and population density to the booking rate.

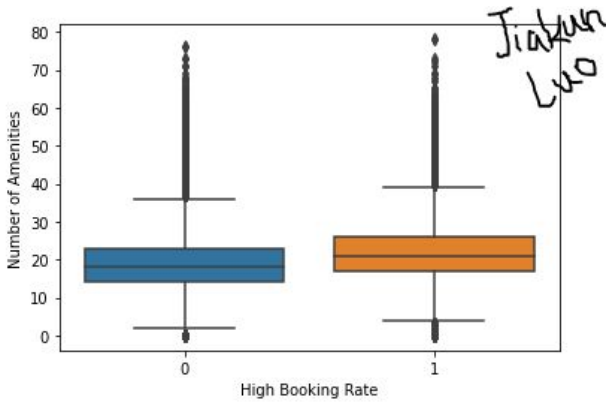
**c. Price*PropertyApartment (top 15 important features)**

We believe that property type is an important feature because different types of listings, for example, an apartment and a tent, will have different booking rates. The category distribution of *property_type* is very skewed, so we group 46 types into 5 major property types based on business sense (Appendix2). Then, we create an interaction variable by multiplying *propertyApartment* with *Price* to capture the effect of high-end and cheap apartments on a high booking rate. The *Price*PropertyApartment* variable is among the top 15 important features in our final model.

**d. Amenities**

The "amenities" column entails amenities that the hosts provide for guests in the listings, and we expect that guests prefer listings with more amenities rather than those with less facilities as well as services.

From the boxplot, the mean of the total number of listings with lower booking rate is lower than those with higher booking rate, indicating that the more amenities a listing has, the more attractive it becomes.




With the total amenity count, different kinds of amenities are taken equally, but we should also consider the distinct influence each amenity has on booking rate. For example, basic amenities are expected by almost every guest during their stay such as the internet and air conditioning, while extra amenities enhance the guest experience and add the "wow" effect of listings, depending on the type of traveler. New amenities' features and further analysis are attached at the Appendix 3, and the further analysis some of which are important features in the final model.

**e. Description_wcount(top 10 important features)**

How detailed the listing descriptions written by the hosts are and how much information they reveal help guests decide to book or not. We use the number of word count of listing description to measure its detailedness.

**f. Transit--->Flexible (top 10 important features)**

The transit column is the text variable in which hosts integrate public transit options around the living place. In order to quantify the traffic accessibility, the number of keywords related to public transport, including "bike",

"bus", "subway", "train", "transportation", "metro" and "line" is recorded in a new variable named "flexible". The new feature serves for people out there who travel without their own vehicle and whether the living place is reasonably accessible by public transport turns into an assignable factor in their booking decisions.

### g. Neighbourhood

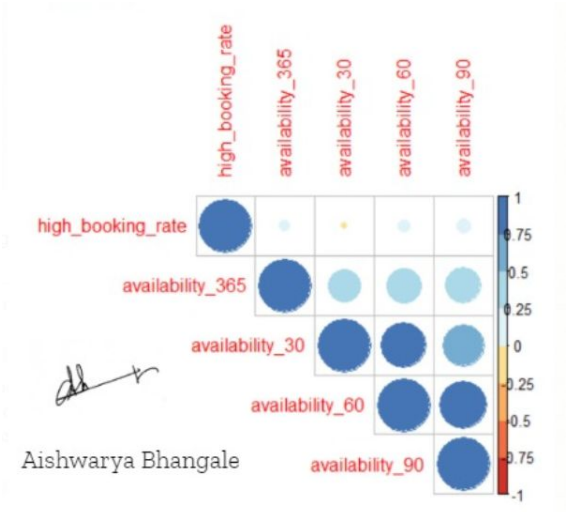
The "neighborhoods_overview" column consists of detailed descriptions of the listings' surroundings. Information such as location, access to transportation and nearby buildings are included in it. We separate the effect of whether a listing is nearby restaurants and whether it is close to the city center by creating two binary variables: The neighbourhood_restaurant variable and the city_centrality variable.

```
> chi_square(train$neighbourhood_restaurant)

        Pearson's Chi-squared test with Yates' continuity correction

data:  sd
X-squared = 603.72, df = 1, p-value < 2.2e-16

> chi_square(train$city_centrality)

        Pearson's Chi-squared test with Yates' continuity correction

data:  sd
X-squared = 356.73, df = 1, p-value < 2.2e-16
```

The x-squared of the neighbourhood_restaurant is 603.72 and the x-squared of the city_centrality is 356.73, so two variables associated with neighbourhood are useful in determining whether a listing is highly booked.

### h. Availability(top 10 important features)

There are 4 variables associated with availability of an Airbnb listing in the dataset- availability_30, availability_60, availability_90, availability_365. We feature engineered these four variables to include the days between, for example the availability between 30th and 60th day, as a value for availability_60. On the right is a correlation graph between these four variables after their collinearity was removed.




We can clearly see from the correlation plot, that these variables are still highly correlated. If the total days the Airbnb is available for the next 30 days is high, then we can expect the total days the airbnb is available between the 30th and 60th day to also be high.

Aishwarya Bhangale

## 3. MODELLING & MODEL EVALUATION

The Airbnb data is partitioned into 70% training data and 30% validation data. Due to computing power limitations(each run takes too long), we do not hold out a test data to test our models on. We use the majority class, high_booking_rate=0, as our baseline model prediction and it resulted in a 0.744 accuracy.

There are three models that we experimented with, and the reasons of choosing them are as follows:

1. Logistic Regression Model: Basic and quick to run for binary classification problems.
2. Random Forest Model: A standard ensemble classification approach that is known to be suitable for data with complex features. Less likely to overfit.
3. XGBoost Model: A model that can reduce bias and is known for its good performance.

The first model that we experimented with is the logistic regression model. With a validation set accuracy of 0.7929503 at an optimum cutoff of 0.4648672, the logistic regression model beated the baseline model, but we believe there could be even better models. So, we decided to pursue models better suited for more complexed data.

When comparing the accuracy of the logistic regression model to that of random forest and XGBoost, we find that logistic regression is outperformed by the other two. We believe it may be that logistic regression is better suited for smaller datasets and that its linear boundary does not classify well in our dataset.

The second model we thus tried is the random forest model. Following are the reasons we selected this model to experiment with next:

- It is not affected by multicollinearity between the feature variables
- It creates decorrelated decision trees with respect to the randomness in its algorithm
- It is far easier to tune than any other ensemble model
- It is not affected by too many binary feature variables in the training set
- It is unaffected by over-fitting if the parameters are tuned properly.

With a number of features randomly sampled at each split at an optimal value of 18, our random forest model yielded a validation accuracy of 0.8380622. We used “tuneRF” function in “randomForest” library to tune the “mtry” variable. For the rest of the variables, we did not have enough computation power at hand or even online(Kaggle, Databricks platforms) that could allow grid search over variables “sampsize”, ”ntree”, ”nodesize”, and “maxnodes”.

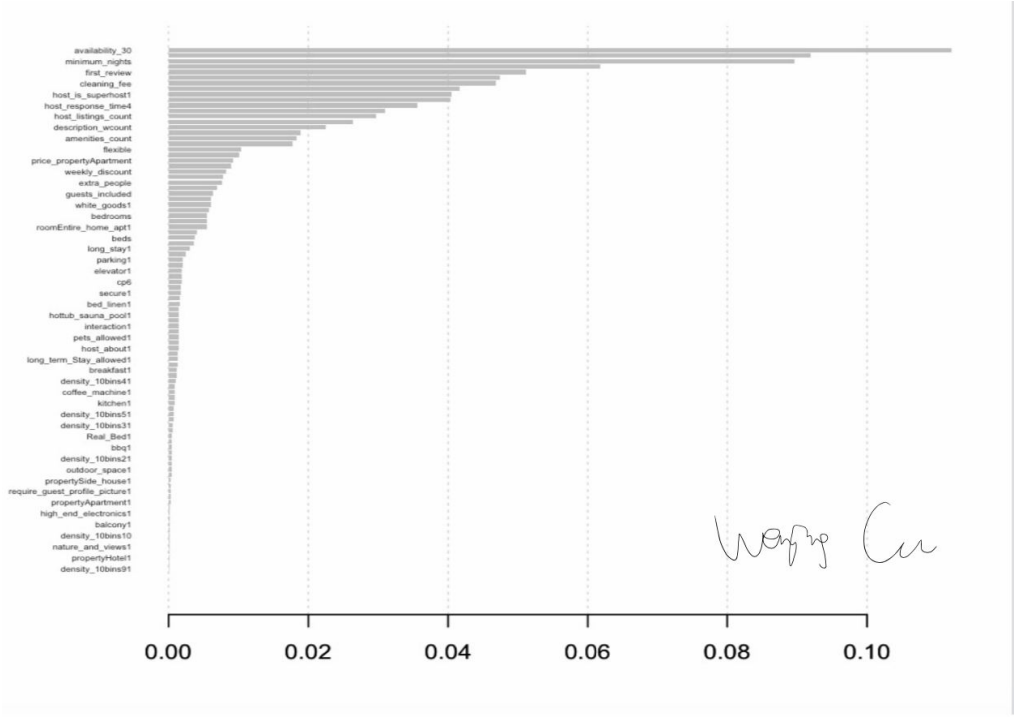
Third, to deliver higher generalization and prediction accuracy of our model, we proceed to try Boosting Algorithms. We chose XGboost algorithm over other models, specifically because XGBoost helps us achieve higher accuracy by reducing variance, and also by reducing bias since it adds trees sequentially. Another important reason for trying XGBoost is that it is known for its fast computation. Previously when we were tuning our random forest model, we were hindered by our computation capacity and ran out of RAM. So, we looked for an algorithm that does not sacrifice computation speed over accuracy.

With the application of the ROC curve, our XGboost model yielded a best validation accuracy of 0.8501 with the best cutoff (0.503). We used inbuilt xgb.cv function to tune general parameters(booster,nthread), booster parameters(nrounds,eta,gamma, max_depth, lambda, alpha) and learning task parameters(Objective, eval_metric) to finalize increasing classification accuracy.

With respect to the tuning process, following is the way in which we tuned our respective models:

- Logistic Regression : Use of library "ROCR" to find the optimal point where accuracy becomes constant and does not increase further. Plotting a graph of TPR,TNR and accuracy against cutoff yielded a clear result as to what the optimal value should be.
- Random Forest : Use of "tuneRF" method in "randomForest" library that calculates the optimal value of "mtry" variable (number of variables randomly sampled at each split)
- XGBoost : Use of inbuilt "xgb.cv" function to tune general parameters(booster,nthread), booster parameters(nrounds,eta,gamma, max_depth, lambda, alpha) and learning task parameters(Objective, eval_metric) to finalize the value of hyper parameter.

Below is our feature importance graph of the XGboost model:



We notice that certain new variables we created such as location related features, amenity related features, and description word count are very important in the XGBoost classification model. So, had we not conducted feature engineering of these variables, our accuracy might not be as high.

## 4. CONCLUSION

The model with the highest accuracy is the XGboost model. We suggest that Airbnb leverage this classification model to identify whether a new listing will have high booking rates or not. Using this information, Airbnb can rank a new listing higher in their searching system if they predict that this listing will be popular so that guests will click on potentially popular listings first. This increases the user experience of guests and in turn will make them more likely to come back to the rental platform again.

On the other hand, Airbnb also helps hosts set their pricing. If they predict a new listing to be unpopular based on multiple features, they can suggest a lower nightly price to the host. If the conditions of the property are not as ideal, at least the host can alter their pricing to stay competitive in the rental market. The prediction of a listing having a high booking rate or not is an indicator that Airbnb can use in various situations to improve the user experience for guests and hosts alike. Thus, we highly recommend Airbnb to take advantage of our prediction model in their system.

# 4.APPENDICES

**Appendix 1. List of Features Used in Final Model**

```
 [1] "density_10bins0"                 "density_10bins1"
 [3] "density_10bins2"                 "density_10bins3"
 [5] "density_10bins4"                 "density_10bins5"
 [7] "density_10bins6"                 "density_10bins7"
 [9] "density_10bins8"                 "density_10bins9"
[11] "density"                         "density_bins"
[13] "density_10bins"                  "access"
[15] "accommodates"                    "amenities_count"
[17] "availability_30"                 "availability_365"
[19] "availability_60"                 "availability_90"
[21] "bathrooms"                       "bedrooms"
[23] "beds"                            "cp1"
[25] "cp2"                             "cp5"
[27] "cp6"                             "cp8"
[29] "cp10"                            "cleaning_fee"
[31] "description_wcount"              "extra_people"
[33] "guests_included"                 "host_about"
[35] "Real_Bed"                        "host_has_profile_pic"
[37] "host_identity_verified"          "host_is_superhost"
[39] "host_listings_count"             "host_response_rate"
[41] "host_response_time"              "experience"
[43] "instant_bookable"                "is_business_travel_ready"
[45] "is_location_exact"               "long_stay"
[47] "minimum_nights"                  "price"
[49] "propertyApartment"               "propertyCommon_house"
[51] "propertySide_house"              "propertyHotel"
[53] "propertySpecial"                 "require_guest_phone_verification"
[55] "require_guest_profile_picture"   "requires_license"
[57] "roomEntire_home_apt"             "roomPrivate_room"
[59] "roomShared_room"                 "security_deposit"
[61] "CheckIn24"                       "air_conditioning"
[63] "high_end_electronics"            "bbq"
```

```
[65] "balcony"                          "nature_and_views"
[67] "bed_linen"                        "breakfast"
[69] "tv"                               "coffee_machine"
[71] "kitchen"                          "white_goods"
[73] "elevator"                         "gym"
[75] "child_friendly"                   "parking"
[77] "outdoor_space"                    "host_greeting"
[79] "hottub_sauna_pool"                "internet"
[81] "long_term_Stay_allowed"           "pets_allowed"
[83] "private_entrance"                 "secure"
[85] "self_check_in"                    "smoking_allowed"
[87] "accessible"                       "event_suitable"
[89] "rules"                            "interaction"
[91] "flexible"                         "price_propertyApartment"
[93] "price_roomPrivate_room"           "latitude"
[95] "longitude"                        "average_price_per_person"
[97] "maximum_nights"                   "num_listings"
[99] "neighbourhood_restaurant"         "city_centrality"
```

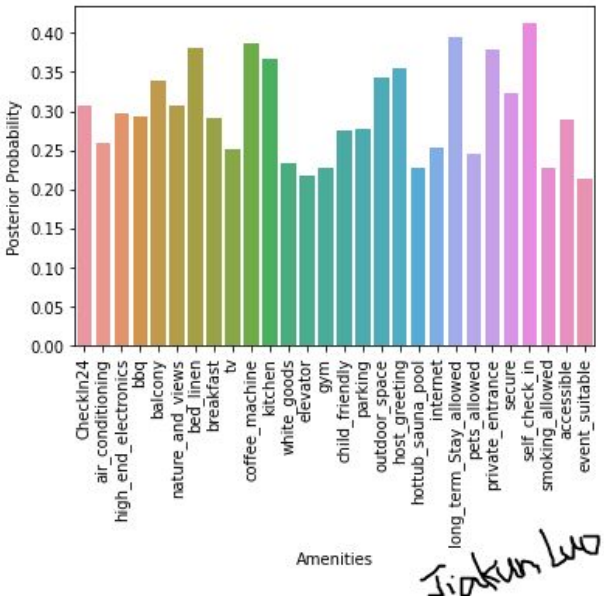
**Appendix 2. New Variables created from Property Type**

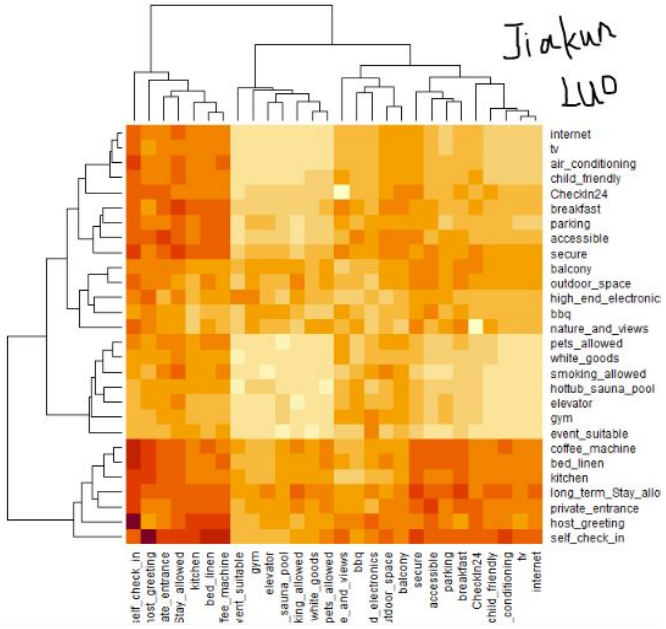
| Newly Created Features | Constitution |
|---|---|
| Apartment | "Loft, House", "Apartment", "Condominium", "Timeshare", "Aparthotel", "Serviced apartment" |
| Common house | "Townhouse", "Bed and breakfast", "Bungalow", "Villa", "Vacation home", "Chalet" |
| Side house | "Guesthouse", "In-law", "Dorm", "Guest suite", "Cabin", "Cottage", "Farm stay", "Nature lodge" |
| Hotel | "Hotel", "Boutique hotel", "Hostel" |
| Special | "Castle", "Camper/RV", "Boat" , "Treehouse", "Tiny house", "Yurt", "Cave", "Casa particular (Cuba)", "Hut", "Tipi", "Earth House", "Earth house", "Train", "Barn" ,"Island" , "Plane", "Lighthouse" |

**Appendix 3. New Variables created from Amenities**

| Type | Newly Created Feature |
|---|---|
| Service | CheckIn24; self_check_in; event_suitable; breakfast; host_greeting |
| Applicant | air_conditioning; high_end_electronics; coffee_machine; bbq; elevator; internet tv |
| Space | balcony; kitchen; gym; parking; outdoor_space; hottub_sauna_pool; private_entrance |
| Permission | smoking_allowed; pets_allowed; long_term_Stay_allowed |
| Others | child_friendly; accessible; bed_linen; nature_and_views; white_goods; secure |

Appendix 4




With Naive Bayes analysis, the bar chart exhibits the probability of listings to be high-booked given on specific amenities. The probability of high booking listings is around 25%, and except for air_conditioning, tv, internet, pest_allowed, white_goods, elevator, child_friendly and smoking_allowed, each amenity gets the higher posterior probability than the baseline, indicating that its existence increases the chance for a listing in the high booking list. In particular, if a listing is able to provide bed_linen and coffee_machine, make the kitchen accessible, allow long_term_stay, and equip private entrance and self_check_in respectively, 40% of chance that it would be taken in the waiting list of high booking listings. Besides, if a host can provide two amenities at the same time, no matter which two amenities he serves, the overall probability of being

high-booked increases greatly. Most importantly, when host_greeting and self_check_in are provided simultaneously, the probability is going to exceed 50%. In this case, we recommend that the host should provide a variety of facilities and services, and pay more attention to amenities that facilitate guests, and to services for establishing a warm and friendly host-guest relationship.