IEOR 4650 Project Report

Group members: Haodi Liu, Jiani Lu, Chenfei Hou, Yuxing Chen, Jie Chen

1. Introduction

Having a popular rental market plays a significant role in the Airbnb corporation's development. The main sources of revenue for the Airbnb corporation are service fees from hosts and visitors and advertising fees from merchants who want to utilize the platform to advertise themselves. Of course, a popular rental market is a foundation for all of the sources of revenue.

Thus, we are considering helping advertise the unpopular houses by recommending them to users using recommendation system and providing the hosts of unpopular houses with constructive instructions on how to improve their quality of service and popularity with the help of predicting models. This initiative is both lucrative and civic-minded because it helps to improve the businesses of hosts of unpopular houses and the Airbnb corporation itself can make much more profit from the overall more popular rental market.

The Methodology part introduces how we clean our dataset, built recommendation models and prediction models that could help hosts to measure and improve the quality of their houses.

All the results and analysis are presented in the Results section.

The Discussion section contains a brief summary of our findings.

2. Methodology

2.1. Data

This data file includes all the needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

Data Manipulation

The original dataset contains 50599 New York City Airbnb housings and 106 parameters that are related to each property.

All the parameters that have more than 35 percent of missing values and constant parameters are excluded. All the descriptive features, such as the name and the summary of the property are also dropped from the dataset.

Data Splitting

In order to test and compare the performance of different predicting models, the dataset is split into three parts. 80 percent of data is used as training data, 10 percent as validation data, and the rest 10 percent is the testing data.

2.2. Recommendation Model

Unsupervised KNN

We created a recommendation engine using unsupervised KNN. While users browsing the website of Airbnb, the website can keep track of the behavior of each user and record which houses each user has viewed. The houses viewed by a user are considered as the houses that the user might be interested in. Based on the record, the recommendation engine can recommend to the user the houses that are similar to the houses he/she has viewed.

The houses are divided into a “popular” group and an “unpopular” group based on average reviews per month and the time of the last review. Houses with average reviews per month less than the median or those whose last review was acquired before 2019 are considered as “unpopular” houses that would need help in advertising. After obtaining the historical record of a user, the website can feed the IDs of the viewed houses into the recommendation engine and specify how many houses from the “popular” group and “unpopular” group respectively to be recommended. As a result, a specified number of similar houses from the “popular” group and “unpopular” group respectively will be recommended to the user. Of course, the ratings of the houses have to be high enough (no less than 80) so as to be qualified to be recommended. In this way, many “unpopular” houses can be seen more frequently by the users and thus their popularity can be considerably increased.

2.3. Prediction Model (Rating Score)

Since our recommendation system will recommend houses based on their rating scores, we first intend to build models that enable the Airbnb hosts to predict the rating score for their properties, which can help them to get more customers and make more profits. Linear Regression, KNN, Random Forest, and Neural Network are used in this project to predict the rating score. By comparing the RMSPE of different models, we select the best model that has the smallest RMSPE on the validation dataset.

Linear Regression

Before performing linear regression, we first considered the correlation between each pair of variables to see if variables are highly linearly correlated. To avoid collinearity problems, all the other review scores, such as review_scores_cleanliness and review_scores_accuracy, are excluded from the dataset since they are highly correlated with our dependent variable. Date type variables are also dropped.

Stepwise forward selection is applied to select a set of important features for the linear model.

Random Forest

In this Random Forest model, we choose 30 features including ‘price’, ‘neighbourhood_group’, ‘room_type’, etc. For the random forest, the max depth we choose is 3, the number of trees we use in the forest is 1000. For the best size of the random subsets of features, we run a loop and find that the best size is 27.

KNN

In this KNN model, the total features are the same as those in KNN. We try different numbers of neighbors to fit the model. From 1 to 31, we find that 7 neighbors perform best on the validation dataset.

Neural Network

In order to find the best prediction model, we also fit a Neural Network regression model. The features stay the same. For the parameters, we try different amount of hidden layers in [0,1,2,3,4,5] and we also try different number of neurons in [16,32,64,128]. Finally, we find that the best combination is 5 hidden layers each with 128 neurons(The final value may inflate, because of the random initial value of parameters, but the results are very similar.)

2.4. Prediction Model (Number of Reviews)

Modeling the timing of the first review:

Similarly, the second type of prediction model tends to predict the popularity of the houses. In order to model the occurrence of the first review, we apply Timing Models. We observe that the earliest host_since is 2008-08-22, but the first review of the dataset is 2009-04-20. Therefore, it is reasonable to assume that the Review function started after some business had run for a while. We only take listings with host_since later than 2009-04-20 into account (49977 obs). If first_review is NaN, we censor the first_review to last_scraped. With results

from timing models, we are able to compute the hazard of the first review. For newly opened or non-review stores, they might care about how to increase the likelihood of receiving their first reviews.

3. Results

Prediction Model (Rating Score)

Model	RMSPE
Linear Regression	0.268
KNN	0.271
Random Forest	0.279
Neural Network	0.277

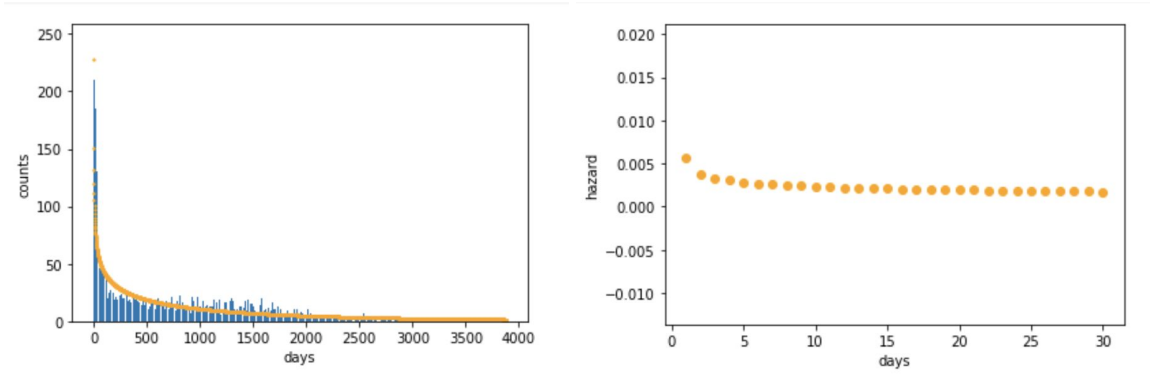
Table 1: Summary of RMSPE of Prediction Models

From the table, we can conclude that all of these models have similar performance, and Linear Regression is slightly better than others. By running Linear Regression on the testing dataset, we get its RMSPE is about 0.343.

Prediction Model (Number of Reviews)

Modeling the timing of the first review

For the one-segment Weibull model, we get lambda: 0.00091, c: 0.7370, indicating the hazard is small and decreasing over time. We plot out the distribution of the number of operating days until the first review and the distribution of the hazard of the first review's arrival in the first 30 days. The plots are shown below.



Besides the one-segment Weibull model, we make some more exploration. We assume there are two types of listings: higher λ (popular) and lower λ (not popular). However, we get

a p of 0.9999999, implying there is less than one listing belong to type II. Therefore, two-segment is not that necessary. We also assume continuously observed heterogeneity and try the Weibull-Gamma model. (see Jupiter notebook for reference)

At last, we fit the Weibull model with covariate by incorporating price, bed_type, room_type to λ in one-segment. The βs for the price, bed_type, and room_type are negative. According to our results, if a host wants to improve his/her probability of receiving the first review, he/she should reduce the price and consider changing bed_type and room_type. Among the top 30 listings that are most likely to receive their first review, we see a price range from $19 to $35 and the bed type of airbed, couch, pull-out sofa, and futon. Most of them are shared rooms.

4. Discussion

For the “unpopular” houses with a good rating (>= 80), our recommendation system can help recommend them to users so that they will become better known. In this way, their popularity will be greatly improved by showing up more frequently in front of customers. The houses with no rating so far will also be recommended to customers so that they can get the chance to earn a good reputation.

For the prediction models, the hosts can use the prediction model to predict the rating scores for their new houses. If the predicted score is higher than a threshold(We think the medium of the scores is a reasonable threshold), then the new house is supposed to be profitable. If the predicted score is low, the host should not rent out the house.

For new Airbnb houses with no review records, the likelihood of the first review decreases over time, so it’s important to make sure your features are attractive to customers at first glance. Weibull model with covariate tells us low-priced shared rooms with airbed, couch, pull-out sofa, or futon are the most attractive type. New hosts can start from economical room type, providing a favorable price to win their first customer. Also, rather than wait to be reviewed, those non-review hosts should take the initiative to transform their houses to more affordable types, i.e. by dividing entire apt to shared rooms or adding a pull-out sofa in the living room to lower the cost per capita.

5. Links

video link: https://drive.google.com/open?id=1FUjTh7ScPAbAu-QCPRpn3w8IHGIGjGh2
dataset link: https://drive.google.com/open?id=1yfXwOOaOd_EpZaKAF6atAtjjj7YeoTd1