Jahnavi Rati
Robert Simione
Luke Lawson
Jessica Meyer
APAN 5200 – Frameworks & Methods 1
8 Dec. 2021

KAGGLE PROJECT: AIRBNB RENTALS

PURPOSE

The focus of our Kaggle project is to predict Airbnb price rentals in New York using ninety variables. The datasets collected include information and factors that potentially impact price rentals. I have information regarding most of the people who are allowing property rentals at a home or at an apartment. The intent behind exploring the datasets is to identify significant predictors that have impact on Airbnb price rentals.

THOUGHT PROCESS

Without looking at the dataset, I have basic knowledge about some potential factors. I believe that reviews have an impact on price rental because if the reviews are good on the property share, then the price would be higher, and vice versa when the reviews are bad. When I would like to rent an Airbnb, I look into the additional fees. The lower the cleaning fee, the more preferable it is to reserve, but potentially, the higher the price. Another important factor which I look at when reserving a space is whether the host is a super host. Super hosts are renters who have a long-time experience or have had huge success rates with number of paying guests being satisfied by the property share, so the price negotiations can occur and thus, have an effect on price rentals.

Accommodations would also impact the price rentals as does the basic requirements for renting an Airbnb including the number of bedrooms, bathrooms, and number of beds. Even the amount of space offered for rent would matter, so that would mean the amount of space shared and private measured in square feet. Additionally, price rentals can depend on the location for property share. With a basic start in identifying potential price predictors, I began exploring the datasets and classify all those ninety variables as potential predictor or not.

EXPLORING DATA

We gathered two different datasets: analysis dataset and scoring dataset. The analysis dataset was assumed to be used for exploratory analysis with total 91 variables including our outcome variable, price and to formulate a predictive model. The scoring dataset was assumed to be used for running the model and obtain price predictions. The scoring dataset did not include our outcome variable, price, so the reliance was obtaining low root mean square error. These two datasets are completely different. The intention behind having different datasets is to train our analysis data first, identify the important features or factors that impact Airbnb price rentals, and then, run the model on the scoring data which is considered our test data set.

Since there are ninety variables that can count as price predictors, identifying significant predictors was essential. Ninety variables is too many predictors, and some of these predictors are too descriptive which makes modelling predictions challenging. When I open the analysis dataset, I see that each column represented one of the ninety predictors, so column by column and row by row I look for: 1) any missing information, 2) type of variable (categorical or numerical), and 3) types of information provided. An open-ended response such as summary, description, neighborhood overview, and many other variables that makes data tidying and cleaning more difficult was disregarded from the features list. Open-ended responses are based on personal reviews and perspectives involving lots of variation in responses so the focus would be on numerical and categorical variables.

LEARNING EXPERIENCE

Throughout this process, I learned that data tidying and data cleaning can get messy which is why it is essential to look at every possible detail and variation to the outputs it can have. As I continued to select features for prediction model, I discovered that the zip code shows interesting outputs. I considered the zip code because the numerical information can be easily categorized. However, I noticed that some of the zip codes are not five-digit format. After reformatting, I discovered there were 347 missing zip codes, so I started associating each of the neighborhoods to the should-be zip code. During this neighborhood and zip coding matching process, I discovered different neighborhoods in New York had been record with the same zip code. Normally, zip codes are different for every neighborhood in the state, so having the same zip code for multiple neighborhoods does not make sense, and thus, had to exclude zip code from our potential features list.

There were total seven features from the potential features list that had missing information. The predictors are as in the following:

host_response_rate host_acceptance_rate host_identity_verified
host_has_profile_pic host_is_superhost security_deposit
cleaning_fee

The categorical variables about the host having identity verified, profile picture, and is a super host were simpler to impute for the 6 missing values. I replaced the missing value with the variable’s median after conditioning “f” as 0 and “t” as 1. The other four variables are numerical, and to impute predictions for 1000’s of missing information was challenging. I thought that I could impute the missing value with the median, but before deciding to impute, exploring the distribution of each of the variable’s outputs was a must. If the distribution was skewed, I would impute using median, and if the distribution appeared normal, then I would use mean. Each of those numerical variables have a skewed distribution, so I used median function to impute.

Another strategy for managing dataset with missing information is to eliminate all the missing values. In this particular scenario with ninety predictors and over 40,000 user information, eliminating all missing values will drastically reduce the number of user information and poses a chance of underfitting the model. I learned that if there is a small proportion of missing values from the entire dataset, then there is an option of completely removing the missing values and information associated with it or impute the missing values. However, when there is a large proportion of missing values from the entire dataset, then imputing the missing values or disregarding that particular variable would be more reliable.

ANALYSIS TECHNIQUES

Once the data was cleaned and ready for analysis, the first step was to create a linear model with all of the identified features that potentially predict price. I identified features that are potential predictors listed in the following:

host_response_rate
host_acceptance_rate
host_is_superhost
host_has_profile_pic
host_identity_verified
room_type
neighbourhood_cleansed
neighbourhood_group_cleansed
is_location_exact
property_type
accommodates
bathrooms
bedrooms
beds
bed_type
square_feet
security_deposit
cleaning_fee
guests_included
extra_people
minimum_nights
maximum_nights
number_of_reviews
last_review
days_since_last_review
review_scores_rating
instant_bookable
review_scores_accuracy
review_scores_cleanliness
cancellation_policy
review_scores_checkin
review_scores_communication
review_scores_value
review_scores_location

The output of the model indicated how each of the variables are related with price, and how significant each relationship is. From the same list of variables above, variables that have insignificant relationship with price were crossed out below and the rest of the variables remained as final predictors of price:

host_response_rate
host_acceptance_rate
host_is_superhost
host_has_profile_pic
host_identity_verified
room_type
neighbourhood_cleansed
neighbourhood_group_cleansed
is_location_exact
~~property_type~~
accommodates
bathrooms
bedrooms
beds
~~bed_type~~
~~square_feet~~
security_deposit
cleaning_fee
guests_included
extra_people
minimum_nights
maximum_nights
number_of_reviews
~~last_review~~
days_since_last_review
review_scores_rating
instant_bookable
review_scores_accuracy
review_scores_cleanliness
cancellation_policy
review_scores_checkin
review_scores_communication
review_scores_value
review_scores_location

Then, I constructed different kinds of models that involve cross-validations and tuning to help obtain better predictions of the model. I created eight different predictive modelling out of which one predictive model performed the best with the lowest RMSE. The eight models tuned are as in the following: tuning cross-validation model, tuned random forest, xgboosting, boosting with cross-validation, boosting model, bag model, default tree model, and random forest model. The model that output the best RMSE was from the random forest model. The other models were not a big success but it taught me how different models provide different predictions.