From: Airbnb Insights and Analytics team

Attention of: Executive Grant Case

Re: Planning for Earnings through Prediction of Rental Prices

Airbnb is a platform that allows people to rent their property. In order to do so, people must share information about themselves and their property and people who do rent this property can share their experiences by writing a review. However, pricing the property has always been a struggle and a crucial factor in the overall satisfaction.

For Airbnb to be able to manage its earning plans and expectations, I recommend improving our analytical methods to perform more accurate predictions. We should look into predicting the retail price of the properties since it directly impacts our profits and sales. This can be done by developing an optimal model with the lowest Root Mean Square Error (RMSE).

In order to get the lowest RMSE, I began with data prediction and exploration and then used various modeling approaches. In this report, I explained the process of the project highlighting the most important insights starting with the data prediction and exploration, feature selection and finally the various analysis methods used. I was able to attain an RMSE of ~ 325 in the final results, using the ranger model.

For more details about the model, you can refer to the R code provided under a different file.

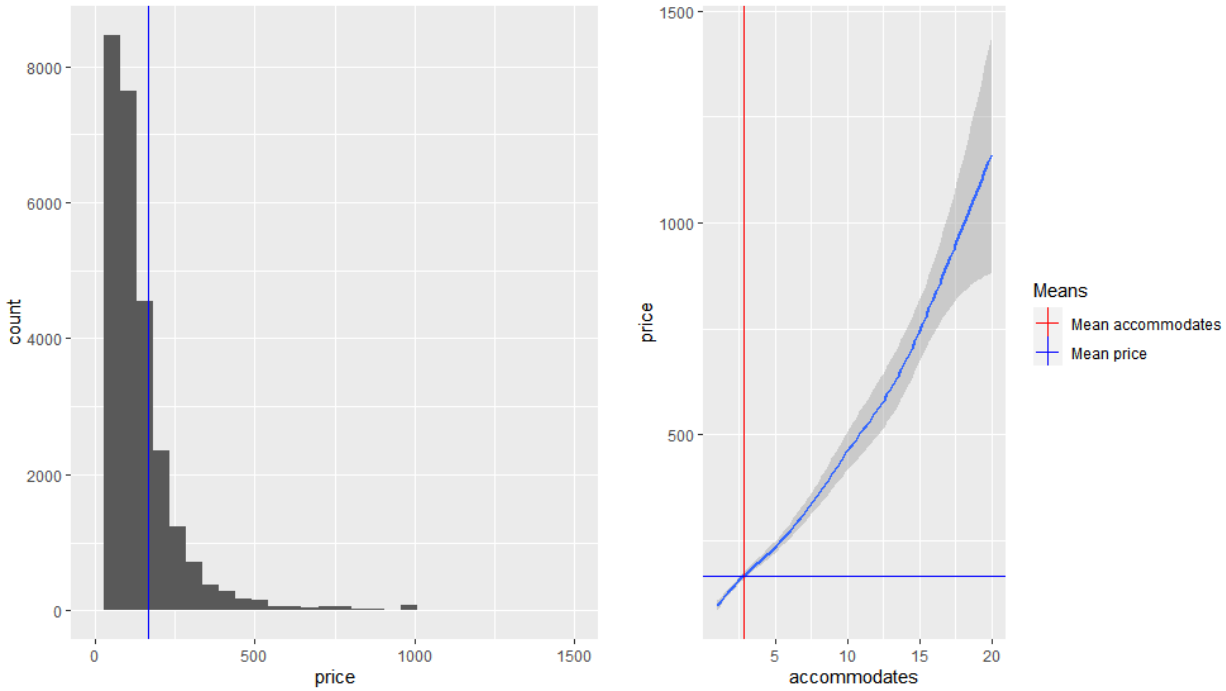
Data Prediction:

Before kicking off the analysis, we started by investigating the data collected. Like any other analysis project, the Airbnb dataset used for this analysis was not perfect and needed to be adjusted accordingly. The data preparation was done prior to splitting the data into train and test. The data set consists of different type of variables such as integers, freeform texts, lists, binary and so on. The important variables such as number of reviews, bed type, price, review scores rating and bedrooms had missing values. After splitting the data and combining the train, test and scoring data sets, we replaced these missing values by their average or by predicting the value based on other variables such as bedrooms' value was predicted by accommodates' value. As for the freeform texts and lists, we created a count of the number of characters with dummy variables.

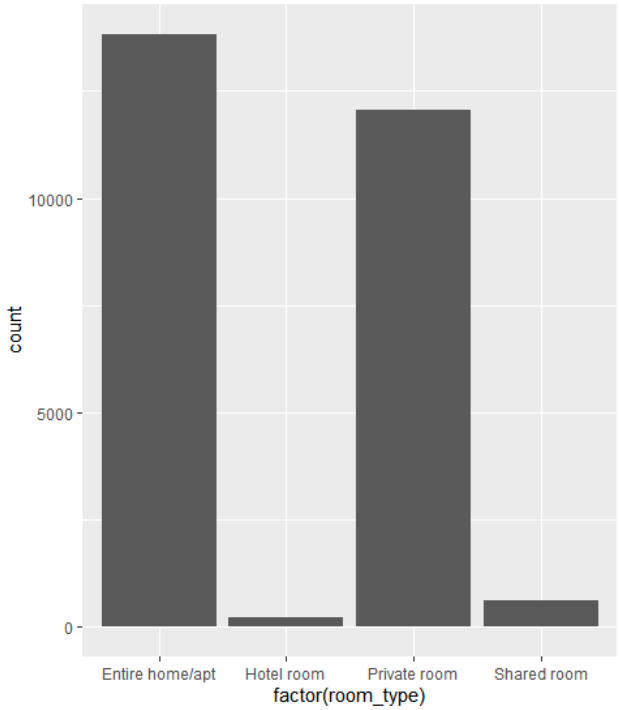
Data Exploration:

This Kaggle Competition made me understand the importance of this step and the next one. Unfortunately, I was not able to explore and clean the data before starting with the project, it was a back-and-forth process. Therefore, I started with the variable that made sense to me. In order to further understand the analysis data collected, I started looking into the main numeric variables such as price, accommodates, bedrooms, bathrooms and beds. I noticed that most of the rental properties are below $250 with an average price around $166 (Exhibit 1). Exhibit 2 shows that the units that can

accommodate less than 3 people consist of the majority of the population. Also, as the number of accommodations increases, the price increases (Exhibit 2).



Let us look into the room type. From Exhibit 3, we realize that most properties are entire home or apartments or shared rooms.



However, the dataset contained 90 variables. Not until later during the process that I discovered that I have missed some important variables that I did not use in my models.

While writing this report, I proceeded with my discovery and learned that all the variables related to the location (zipcode, longitude, lattitude, etc.), the host response rate and some others would have lowered even further my RMSE.

Feature Selection and Data Cleaning

The analysis data consists of 90 variables. We first started by the most important step, identifying the variables that are crucial to our study. I picked variables that I found to be mostly useful for the lm function. I started by cleaning the data accordingly. Then I started adding more variables to test for the RMSE. After submitting the project, I realized that using one of the feature selection methods would have been more efficient and helpful and less time consuming.

Modeling Analysis

After exploring, cleaning and selecting the data, we were now able to proceed with the analysis. I used different modeling approaches such as linear regression, trees, boosting, random forest and ranger. I will briefly go through the various methods used and their respective results. I started with an RSME of approximately 420, getting to ~325.

Linear Model

The first prediction with the highest RSME (~420) was done using two basic variables: minimum_nights and number_of_reviews. By adding bathrooms, bedrooms, beds, room_type, accommodates, security_deposit, guests_included, property_type_upd and cleaning_fee, we were able to reduce the RSME to ~382. From running summary(modelLinear), we get a further understanding of the impact of the chosen variables with price.

```
modelLinear = lm(price~minimum_nights+number_of_reviews + bedrooms+ bathrooms +beds+
                   accommodates + security_deposit +
                   guests_included + host_response_rate + room_type + property_type_upd, data=train)
```

Boosting, Random Forest and Ranger

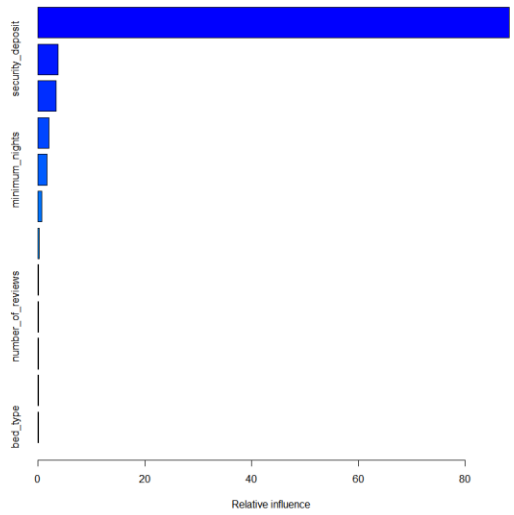
After trying for many variables in the linear regression and noting that the RMSE is not dropping anymore, I changed my modeling approach and started using the boosting, random forest and ranger.

The most effective one seemed to be the ranger model which dropped the RMSE of my final model to ~325 (on Kaggle) and lower on R.

Boosting

The RMSE of my gbm model was ~355. The drop in the RMSE made me realize that I am getting closer to attaining my goal. However, I decided to look further into the chosen variables.

The model resulted in the graph on the right:



Random Forest

I then decided to try a different model to validate my gbm results. The RMSE of the random forest with the same variables as the boosting was of ~ 350.

My Random Forest led to the results on the right:

	Overall
calculated_host_listings_count_private_rooms	657189158
property_type_upd	509951645
minimum_nights	280760680
security_deposit	271873676
number_of_reviews	217751605
review_scores_rating	207718839
bedrooms	132328908
room_type	130868325
review_scores_cleanliness	116168600
neighbourhood_group_cleansed	90728971
instant_bookable	81013321
calculated_host_listings_count_shared_rooms	41249684
bed_type	5107629

From the above models, I decided to remove the least important variable: **bed_type.**

Ranger

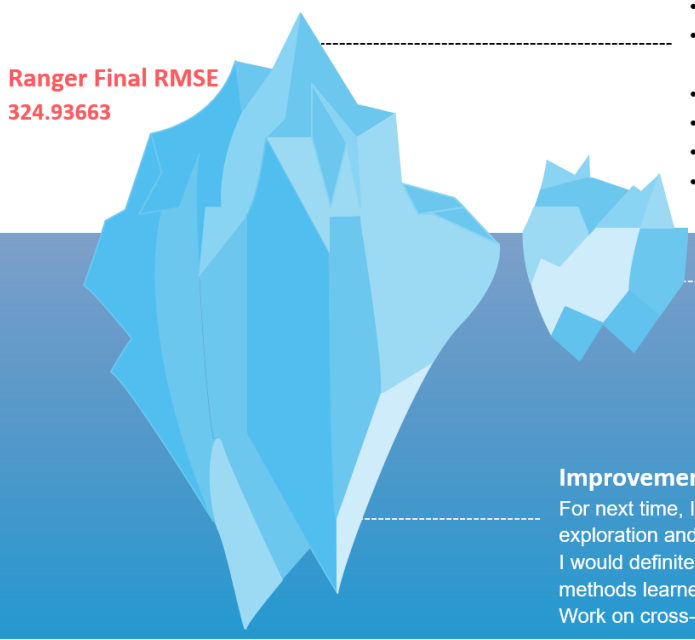
After removing unnecessary data that was jeopardizing the RMSE and choosing the ones that would fit best my model, I used the ranger technique. The final model proved to be the most effective with an RMSE of ~ 325. Unlike gbm and random forest, ranger did not take a long time to run.

Final Thoughts of Rights, Wrongs and Redo's:

The Iceberg principle states that "We cannot detect most of a situation's data". I was able to realize this even more during this Kaggle competition which taught me that no matter how much I know, I still don't know. Also, exploring, predicting and cleaning the data from the beginning would form a strong base for the model and would give us more flexibility in analyzing the data.

airbnb

Airbnb Rental Price Prediction



The Rights

- Starting with the data exploration
- Doing the data wrangling process using the means, predictions and so on (even though it was back-and-forth)
- Starting with the Linear Regression to have a brief overview on the model
- Having the curiosity to look into various modeling approaches
- Being greedy to minimize as much as possible the RMSE
- Setting the goal to learn and enjoy the competition instead of simply competing

The Wrongs

- Not giving enough time for data exploration and cleanup before starting with the process; the back-and-forth was time consuming
- Not using the methods learned for the feature selection process
- Using a minimal number of variable and dropping some important ones
- Overfitting the model

Improvements

For next time, I would give even more time for the data exploration and cleaning process.
I would definitely perform one of the feature selection methods learned (e.g. Lasso, stepwise, …).
Work on cross-validating the models