Data Analysis 3 - Assignment 2

Airbnb Price Prediction - Montreal, Canada

Xibei Chen

09 February 2022

Introduction

The purpose of this project is to help a company set price to their new apartments that are not yet on the rental market. The company is operating small and mid-size apartments hosting 2-6 guests in Montreal. To help them set reasonable price, my approach is to analyze the data of airbnb listings in Montreal on 11th Dec 2021 (the most recent date available) from Inside Airbnb. Then I build several price prediction models with methods, such as OLS, Lasso, and Random Forest. Eventually with prediction performance, interpretability, and external validity taken into consideration, I will decide which prediction model to use for helping the company set price to their apartments. (Please note that this is a summary report, please find more technical details on the technical report)

Data Preparation

Data preparation includes mainly three parts: Data Cleaning, Label Engineering, and Feature Engineering. For Data Cleaning, I filtered observations meeting the following criteria: with valid Airbnb's unique identifier for the listing, maximum capacity is between 2 and 6, room type is entire apartment or flat, property type is entire place (boat and RV excluded). I have also change format of some variables and handled missing values with different methods such as imputing, adding flag, etc.

Label Engineering & EDA

Descriptive Statistics of Daily Price (CA$)

	Mean	Median	SD	Min	P25	P75	Max	N
price	113.34	95.00	78.96	15.00	68.00	130.00	999.00	8390

Our target variable is the daily price in CA Dollars, which is very straightforward. The Mean and Median is not too far way from each other, but Mean is larger than Median, indicating there are some extreme values at right side. For now I decide not to exclude them, because there is no proof that they are errors.

Feature Engineering

I create many dummy variables for amenities and host_verifications. And I created following variable groups for further analysis about variable importance. And I put the symbols such as "d_", "f_" "n_" and to indicate what type the variable is, dummy, factor or numeric. I have also added following features to capture potential non-linear patterns.: squared_number_of_reviews, cubic_number_of_reviews, ln_host_since_to_today. Besides, Interaction terms have also been created for modelling with OLS and Lasso, as I see that for different property types, the patterns between price and if pets allowed, and if there is pool change. In addition for different neighbourhoods, the patterns between price and if there is free parking and if there is lake access also change. (Here I did not create interaction terms with amenities, because there are so many, it would be super hard to compute because of the lack of computing power)

Modelling

Preparation

Mange different samples: Create a holdout set with 20% of observations by random sampling, and the left 80% would be used as work set; Create multiple predictor levels: basic_lev, basic_add_lev, reviews_lev, poly_log_lev, amenities_general, and two interactions levels X1 and X2.

(1) OLS

I build following 7 models with increasing model complexity.

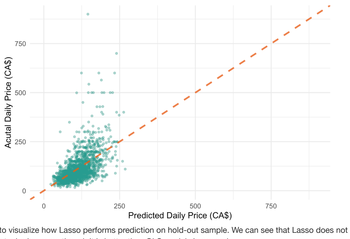
- Model 1: n_accommodates;
- Model 2: basic_lev;
- Model 3: basic_lev, basic_add_lev;
- Model 4: basic_lev, basic_add_lev, reviews_lev, poly_log_lev;
- Model 5: basic_lev, basic_add_lev, reviews_lev, poly_log_lev, X1;
- Model 6: basic_lev, basic_add_lev, reviews_lev, poly_log_lev, X1, X2;
- Model 7: basic_lev, basic_add_lev, reviews_lev, poly_log_lev, X1, X2, amenities_general.

(2) Lasso

OLS and Lasso Performance Summary (5-fold CV)

Model	Coefficients	R_squared	BIC	Training_RMSE	Test_RMSE
M1	2	0.0767814	77285.11	76.47644	76.29090
M2	52	0.1757602	76964.52	72.26070	72.92222
M3	57	0.2213935	76626.29	70.23190	70.89325
M4	64	0.2267550	76641.60	69.98967	70.68750
M5	75	0.2322180	76690.94	69.74199	70.57547
M6	114	0.2516933	76862.14	68.85178	70.32217
M7	152	0.2884578	76858.84	67.13913	69.13331
LASSO	106	0.2492095	NA	NA	68.94066

For Lasso modelling, we still use the OLS M7 predictors. From the above summary table, we can see that M7 with 152 coefficients is the best among OLS Models 1-7. However, Lasso with 106 coefficients picked from M7 gets us slight better Test_RMSE than OLS M7 by around 0.2$. I also evaluated the prediction performance of both OLS M7 and Lasso on our hold-out sample. OLS M7 with 64.32827, and Lasso 63.9918342. Hence, Lasso also wins OLS M7 on hold-out set, by roughly 0.4$.



I created the above graph to visualize how Lasso performs prediction on hold-out sample. We can see that Lasso does not perform prediction well at extreme values of actual price, even though it is better than OLS models in general.

(3) Random Forest

Another method for prediction is regression tree, which can capture interactions and non-linearities automatically. However, it is prone to overfit data, even after pruning. Therefore, I choose to use Random Forest directly here after OLS and Lasso. With the method of Bootstrap Aggregation, 500 trees are created based on similar but not the exact same samples, and then the mean of predicted price is calculated. As I have 177 predictors in total, I arbitrarily set for each split, only randomly picked 13 variables (closest to square root of 177) are taken into consideration. In this way, the trees are decorrelated kept independent from each other, and more chance is given to all the predictors. Both bootstrapping and decorrelating trees mean using random sets of information (observations and predictors). By tuning I also arbitrarily set 50 observations for minimum size per terminal node to avoid each tree overfitting to some extent. It turns out Random Forest does better prediction than both OLS and Lasso. In term of Holdout RMSE, Random forest with around 61$ outperforms Lasso with around 64$ by around 3$, which is quite a good improvement.

Model Performace Comparison

	CV RMSE	Holdout RMSE
OLS (M7 predictors)	69.13331	64.32827
LASSO (M7 predictors)	68.94066	63.99183
Random forest (all predictors)	66.16580	61.06271

Robustness Check

Even though random Forest does quite a good prediction job compared to OLS and Lasso, the RMSE on the hold-out set is still quite large, above 60$. Recall that Lasso does poor job at prediction extreme values, leads me to think about whether excluding apartments with extreme values of price could make us get a model with better prediction. Therefore, I decided to exclude apartments with price higher than 75th percentile plus 1.5*IQR which is 223$ to rerun the above three models.

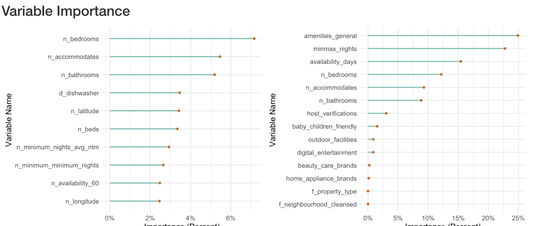
	CV RMSE (without EV)	Holdout RMSE (without EV)
OLS (M7 predictors)	34.86540	35.12591
LASSO (M7 predictors)	34.78986	35.01581
Random forest (all predictors)	33.16948	33.57070

We can tell from the above table that without extreme values taken into consideration, Random forest is still the best for both CV RMSE and Holdout RMSE. What is new is that the RMSE is only half of the RMSE compared to when we did not exclude extreme values earlier. This is huge improvement regarding prediction performance. Therefore, if the properties our company is operating has no unusual features, for example neighbor of celebrities, collections of famous paintings in the house, or any other luxurious facilities, if they are just normal properties, I would use the data without apartments with extreme values of price to train a Random Forest model for much better prediction.

External Validity

Random Forest gives better performance than OLS and Lasso at predicting quantitative target variable, but this comes at the cost. It is a black box, we do not have coefficients for hand-picked variables to interpret. Therefore, it is very hard to tell what patterns of association between daily price and the predictor variables look like. Besides, we are not aware of which variables are more important for the prediction. Patterns of association and variable important are extremely necessary for us to evaluate external validity. Therefore, I applied some diagnostic tools such as variable importance, partial dependence and performance across subsamples to dig deeper into the random forest to find out important variable and the pattern of association between them and price, and how performance varies among different subsamples. Here I keep using the data without extreme values of price.

Variable Importance



Variable Importance is a sum of gains in terms of MSE reduction by splits involving the variable. Since there will be too many variables on full variable importance plot, and it would be impossible to see clearly. Here I only zoom in to see only top 10 on the left side. Here we can see that number of bedrooms, the maximum capacity and the number of bathrooms are the top 3, which can pass our sanity check. On the right side, after grouping, we can see that the sum of importance of all the general amenities variables is actually quite large. And it is the same for minimum and maximum nights and availability variables. They are more importance than the original top 3 after grouping together the variables. Another thing we can see is that variable groups such as host verification, baby children friendly, outdoor facilities, digital entertainment and beauty care brands do have quite some effect on price prediction as groups.

Partial Dependence

Next I examine the pattern of association between price and some of the predictor variables. I picked two variables that I would like to know more about and check the pattern: the number of guests to accommodate and property type. It turns out there is a pretty linear and positive relationship between predicted price and number of guests. In terms of property types, entire townhouse, entire loft and entire residential home are the top 3 expensive types, whereas entire rental unit is the cheapest type.

Performance on Subsamples

To further investigate the performance of the random forest model, I also look at subsamples by three predictor variables: apartment size, property types and districts. For apartment size, I divide apartments into two group, ones with 2 or 3 guests capacity, and other ones with 4-6 guests. It turns out prices of small size apartments are slightly harder to predict than medium size apartments. For property types, I arbitrarily pick two that I am more interested in, namely Entire condo and Entire loft. There is not much difference regarding predictive performance among these two property types. For districts, I choose 5 districts that are located in the inner city, where most investment properties are located. It turns out Côte-des-Neiges-Notre-Dame-de-Grâce and Westmount, the two districts at the west part of city center, are harder than other inner city districts to predict price.

Conclusion

Among 7 OLS models, Lasso and Random Forest, Random Forest has the best prediction performance. By robustness check, I found out by excluding apartments with extreme value prices, prediction performance can be improved significantly. So given that the properties of our company are normal apartments without extraordinary features, I would use Random Forest on data without extreme values of price for prediction. Besides, we should keep in mind that depending on the location and features the properties of our company, we might have different prediction performance. Last but not least, improving amenities, host verification, availability, and other features might enable us to set a relative higher price to generate higher profit for our business.