Airbnb Price Prediction - Montreal, Canada

Xibei Chen

09 February 2022

Executive Summary

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, I analyzed the data of airbnb listings in Montreal on 11th Dec 2021 (the most recent date available) from Inside Airbnb, and build several price prediction models with OLS, Lasso, and Random Forest. While OLS and Lasso are helpful to investigate the patterns of association with interpreting coefficients of variables, Random Forest generates the best prediction performance. Therefore, the final model of choice in my project is Random Forest. Through robustness check, I found the prediction performance can be improved significantly if extreme values are excluded, given that the properties of the company do not have unusual fancy features. Eventually, as Random Forest is a black-box model, I also applied diagnostic tools to evaluate variable importance, partial dependence and performance on subsamples to help the company make better decisions regarding setting price for apartments with different features and investing in what features to generate higher profit.

Data Preparation

Data preparation includes mainly three parts: Data Cleaning, Label Engineering, and Feature Engineering. For Data Cleaning, I filtered observations meeting the our business logic, for example maximum capacity is between 2 and 6. and room type is entire apartment or flat. I have also changed format of some variables and handled missing values with different methods.

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

In terms of label engineering, our target variable is the daily price in CA Dollars. From the above descriptive statistics of daily price, we can see the Mean and Median are 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 decided not to exclude them, as they are not proved to be errors.

Feature Engineering I create many dummy variables for amenities and host_verifications, and the variable groups respectively for further dignostics. And I put the symbols such as "d_", "f_" and "n_" to indicate what type the variable is, dummy, factor or numeric. I have also added some features with polynomial or log-level to capture potential non-linear patterns. Furthermore, interaction terms have also been created for modelling with OLS and Lasso, to capture different patterns of association between price and if pets allowed or if there is pool for different property types, and different patterns between price and if there is free parking or if there is lake access for different neighbourhoods. (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

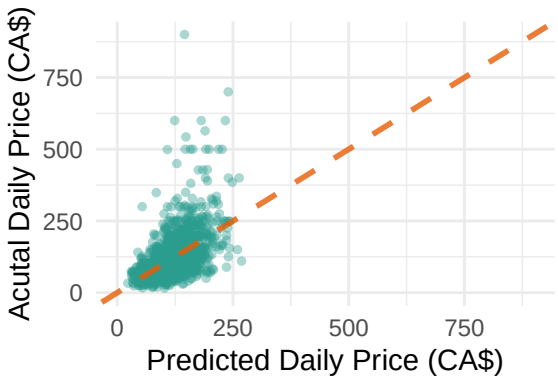
Preparation Firstly, I manged different samples by creating a holdout set with 20% of observations by random sampling, and the left 80% would be used as work set. Secondly, multiple predictor levels are created: basic_lev, basic_add_lev, reviews_lev, poly_log_lev, amenities_general, and two interactions levels X1 and X2. Moreover, number of folds for cross validation is set as 5 for all the models.

(1) OLS & (2) Lasso I built 7 models with increasing model complexity, in terms of number of predictor levels and number of coefficients. The most complex model 7 has all the predictor levels and interaction terms created previously.

Table 1: 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, I still used the OLS M7 predictors. From the above summary table, we see M7 with 152 coefficients is the best among OLS Models. However, Lasso with 106 coefficients picked from M7 predictors gets us slightly better Test_RMSE than OLS M7 by around 0.2$. I also evaluated the prediction performance of both OLS M7 and Lasso on our hold-out sample. OLS M7 with 64.33$, and Lasso 63.99$. Hence, Lasso also wins OLS M7 on hold-out set, by roughly 0.4$.



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

(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. 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 and kept independent from each other, and more chance is given to all the predictors. By tuning I also arbitrarily set 50 observations for minimum size per terminal node to avoid overfitting each tree. It turns out Random Forest does better prediction than both OLS and Lasso. In term of Holdout RMSE, Random forest outperforms Lasso by roughly 3$, which is quite a good improvement.

Robustness Check

Even though random Forest does the best prediction compared to OLS and Lasso, the Holdout RMSE is still quite large, above 60$. Recalling that Lasso does poor job at prediction extreme values, leads me to try excluding apartments with price higher than 75th percentile plus 1.5*IQR which is 223$, and to rerun the above three models and re-evaluate the performance.

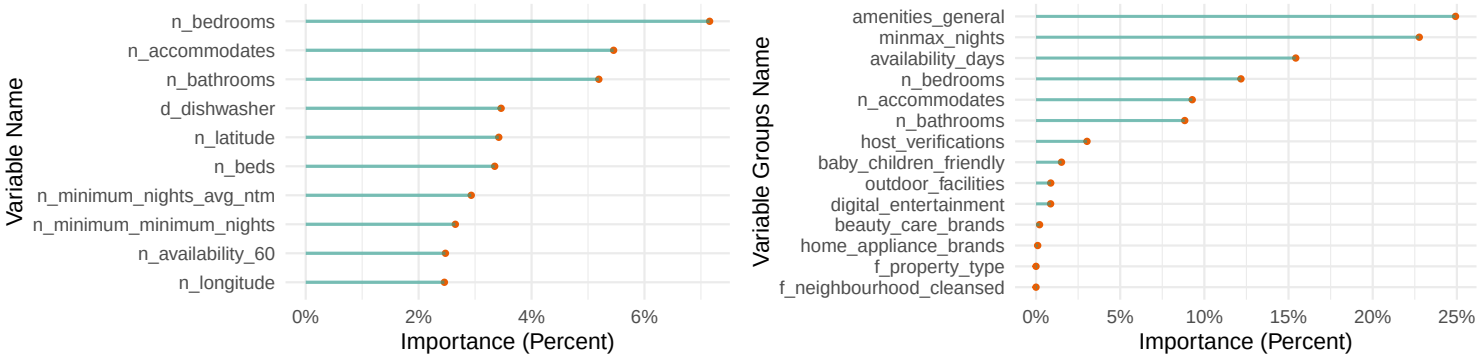
We can tell from the below table that without extreme values, Random forest is still the best for both CV RMSE and Holdout RMSE, but only half of the RMSE when we did not exclude extreme values. This is huge improvement regarding prediction performance. Therefore, if the properties our company is operating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

	CV RMSE	Holdout RMSE	CV RMSE (without EV)	Holdout RMSE (without EV)
OLS (M7 predictors)	69.13331	64.32827	34.86540	35.12591
LASSO (M7 predictors)	68.94066	63.99183	34.78986	35.01581
Random forest (all predictors)	66.16580	61.06271	33.16948	33.57070

External Validity

Random Forest gives better prediction performance than OLS and Lasso at the cost of being a black box model, we do not have coefficients to interpret. Therefore, I applied some diagnostic tools to dig deeper into the random forest to find out important variables and the patterns of association between them and price, and how prediction performance varies among different subsamples.

Variable Importance



Here for clear visualization, I only zoomed in to see the top 10 on the left side. Number of bedrooms, the maximum capacity and number of bathrooms are the top 3 important variables, which can pass our sanity check. On the right side, after grouping, the sum of importance of all the general amenities is actually quite high, the same for minimum and maximum nights and availability variables. They are of more importance than the original top 3 after grouping together the variables. Plus, 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 as well, which is very hard to be discovered without grouping variables.

Partial Dependence Next I examined the pattern of association between price and two variables that I would like to know more about: 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.

Performance on Subsamples I also looked at subsamples by three predictor variables: apartment size, property types and districts. For apartment size, I divide apartments into two groups, small with 2-3 guests and medium 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 picked two that I am more interested in, Entire condo and Entire loft, where there is not much difference regarding predictive performance. For districts, I chose 5 inner city districts, where I assume most investment properties are located. It turns out Côte-des-Neiges-Notre-Dame-de-Grâce and Westmount, the two districts on the west side of city center are harder than the rest to predict price.

Further Research

In terms of external validity, we need to expect higher prediction errors if we change cities or times. We can further evaluate the prediction performance of our model on Montreal but for different past times. In this way we can have a better idea whether our model will perform well for the future. For different cities, I would expect larger prediction errors, as there are different features for different city and patterns with price are highly likely to be different than Montreal, therefore I would recommend to build models based on the data of that specific city to lower the prediction error.