DA3 - A2 - Summary Report

Haaris Afzal Cheema

Introduction

This analysis will focus on predicting the price of airbnb apartments in Porto to help a company price their new apartments. These apartments accommodate between 2 to 6 people. Different prediction models of varying complexity will be analyzed and compared.

Feature Engineering and Sample Design

The initial data contained 10747 observations, which contained the price for the Airbnb apartments for 9th December 2021. The price per day was maintained in Euros. The distribution of the daily price was strongly skewed and the distribution of its log was close to normally distributed. In this case, we will proceed with taking level price for clearer interpretability of the results. As far as predictor variables were concerned, the intuitively more important ones included property type, room type, neighbourhood, along with numerical variables containing information regarding the number of beds, bedrooms and bathrooms. Furthermore, review-based columns pertaining to the number of reviews, reviews per month and the review scores based on criterion such as cleanliness, location etc were also included. For categorical variables the categories with sufficient representation in the data were kept. In the case of property type, an additional criteria was that categories which come under term of 'apartments' were kept and the rest were filtered out. The discrete quantitative variables such as the number of beds, bathrooms, bedrooms etc were analyzed and converted to categorical variables where categories with few values were clubbed together. Next, for the review-based variables, roughly 13% percent values were missing. An interesting find however, in this case was that the values for all these variables were missing in the same rows. So from a sample design decision standpoint, two distinct datasets were created. One, where a combined flag variable was included, and the missing values for each variable were imputed with the median. In the second data, all these rows were dropped (N=1062). The resultant data in this contained 7236 observations as opposed to the other data which contained 8338 observations. This was done, so that the two datasets could be tested in the prediction models separately, and the ones with the better predictions could be kept, whereas the other data would be discarded from the further analysis. Lastly, we had binary variables relating to whether the host was a superhost and whether the apartment was instantly bookable along with a set of amenities which were parsed out of a text variable from the data. The list of amenities was analyzed, and using domain knowledge, 11 amenities were added as binary variables. For interactions, conditional distributions of price were looked at based on different values for the categorical variables. Some interactions were included for the neighbourhood variables based on domain knowledge via external research.

Model Building

Three regression models were specified for the prediction analysis. The three models differed in terms of model complexity (based on the number of variables) where the first model contained the key predictor variables. The second contained the review-based variables as well as the binary variables in addition to the basic key predictors. For the last, most complex model, interaction terms of the neighbourhood with property and room type along with number of people accommodated were added. These regression models can be seen below:

M1: *Number of guests accommodated(squared term also),number of beds (squared term also), number of bedrooms (squared term also), Days since first review (squared and cubic also), Neighbourhood, Minimum nights of stay*

M2: *M1 + log of all review based columns + relevant amenities*

M3: *M2 + interactions +selected interaction terms with prop type and room type + interaction terms with neighbourhoods*

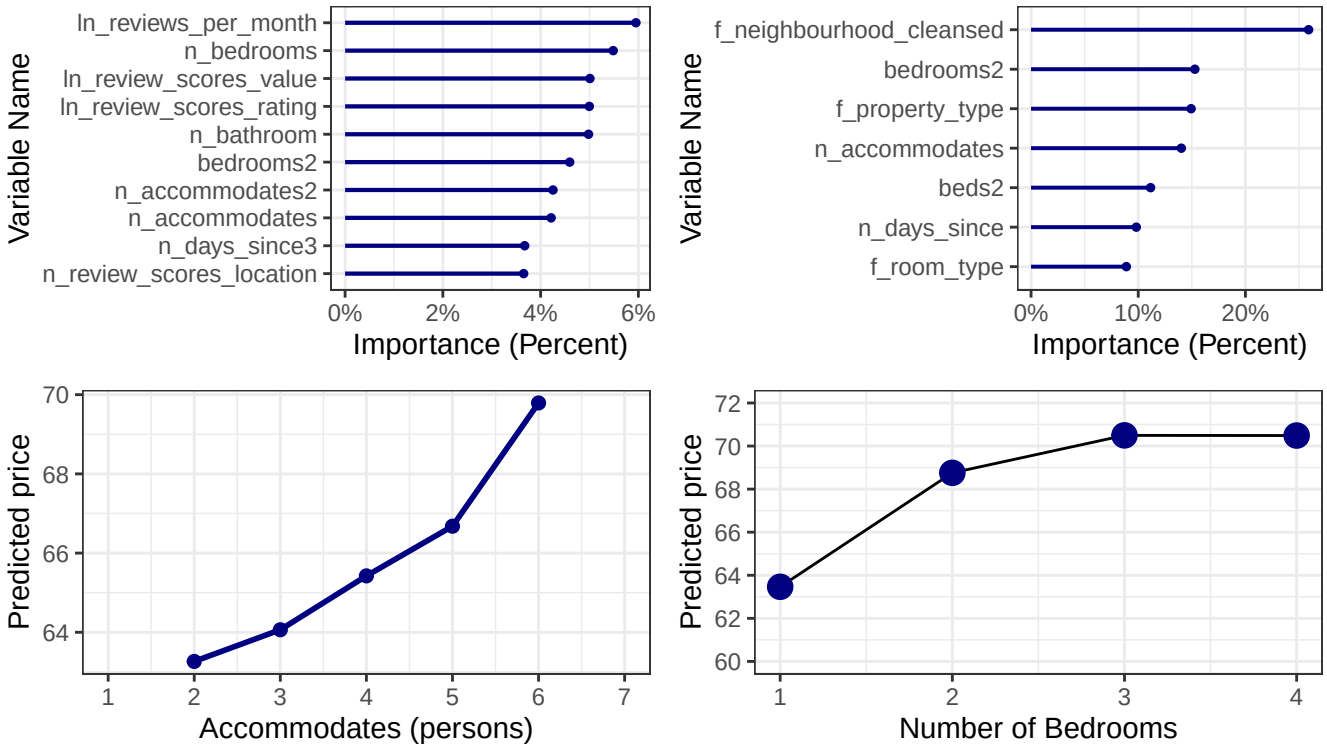
We looked at patterns of association between the numerical predictor variables and the target variable. It was found that the price is approximately linearly related to the number of people that the apartment can accommodate.

However, due to a slightly non-linear trend for higher values of this predictor variable, it was specified as a quadratic. Parabolas were observed for the number of beds, bedrooms and the days since the first review. These were modelled as quadratics as well. For review-linked variables, non-linear trends were observed for which their logarithms were included.

For model training and testing, the sample was divided into two sub-samples, namely the work sample and the holdout sample. The work sample contained 70% of randomly selected observations from the initial sample whereas 30% were kept in the holdout set. 5-fold cross validation was done on the work set, where 5 further random splits were made (after replacement)for the training set and test usage.

Model Evaluation

Firstly, the random forest algorithm was used in the prediction exercise. Prior to running the algorithm, tuning parameters were defined. The number of predictor variables to consider for each split when growing the decorrelated trees was kept at 8. This was done by following the standard practice of using the square root of the number of predictor variables (I rounded up in this case). Next, the stopping rule for each tree was kept with a minimum of 50 observations in each node. Finally, for the number of bootstrap samples, the default option of 500 was chosen. Not a lot of experimentation was done with the tuning parameters due to the robustness of random forest towards the tuning parameters. Due to the sample design dilemma faced earlier, the algorithm was run on two models (model 1 and 2) for each of the two distinct datasets (one with imputation and the other with the dropped rows). It turned out that the average RMSE was lower (47.55) in the case where the rows were dropped as opposed to the data where imputation was done (49.52) when comparing the more complex model (model 2). A similar trend was observed in the simpler models. Therefore, all subsequent predictions were also evaluated on this dataset.Because of the 'blackbox' nature of the algorithm, model diagnostics were performed to identify some of the most important variables in RMSE reduction. Based on the top 10 RMSE reducers, the number of bedrooms, reviews per month and the number of bathrooms turned out to be the most important variables. In the case of grouped variables, it turns out that neighbourhood cleansed is the most important in improving the model fit.



OLS linear regression was run on model 2. It was run with a 5-fold CV. 159 variables were used in this case. Similarly, for LASSO, the most complex model (model 3) was used, which had the highest complexity, in terms of the number of variables as well as the functional forms assigned. This algorithm was also run with a 5-fold cross validation for selecting the optimal value of lambda. In the end, the algorithm picked a model with 218 variables. An overlap is observed between the LASSO coefficients and the OLS coefficients, and the values are also not too different. However, different variable and functional form selection is also seen in the case of LASSO.In the case of the CART algorithm,

Table 1: Horse Race of Models CV RSME

	CV RMSE
OLS	48.54922
LASSO (model w/ interactions)	47.97784
CART	49.98323
Random forest 1: smaller model	48.56969
Random forest 2: extended model	47.50522
GBM	48.24007

resampling results were obtained for 10 values of the complexity parameter (in the order of increasingly stricter stopping rules). RMSE was the metric that was chosen to obtain the optimal model out of the 10 samples. The final value used for the model was with the complexity parameter equal to 0.0157, indicating that the stopping rule for the splitting would be where the R-squared in the test sample would increase by less than 1.57%. Lastly, GBM was used for which multiple tuning parameters had to be specified. The tree complexity was set at 5 and the number of trees made were kept at 250. The learning rate was specified at 0.1 and the minimum samples were 20.

For all models (Table.1), the cross validated RMSE was calculated. Random forest on the extended model has the lowest RMSE in this case, indicating the best fit for the prediction. That is followed by the GBM method which also produces a good fit for the prediction. The last contender in this case is the CART algorithm with an RMSE equal to 50.

Conclusion

Due to the best performance in the CV RMSE, the random forest extended model is now evaluated on the holdout set. The holdout set RMSE is 50.84 as opposed to the CV RMSE of 47.55. There is a little difference in the CV RMSE and holdout set RMSE, which can partly be attributed to the difference in sample sizes of the train and holdout sets. This means that we can expect to make an error of 50.84 Euros when using our model on live data in case we have high external validity.