Prediction with Machine Learning for Economists: Assignment 2

Ramzi Chariag

February 10, 2023

Paris Airbnb apartment price prediction models

Intro

In order to support a company operating in real estate rental in Paris, by building a model that predicts the price of small to mid-sized apartments hosting 2 to 6 people, which is the niche in which the company specializes. Using data from *insideairbnb.com*, I train four different predictive models that manage to predict quite accurately the prices of apartments comparable to the ones that the company is pricing. The models are trained on a sample that includes housing options that can accommodate for 2 to 6 individuals. The models' target is log price, so that the models in can be interpreted in percentage terms, which is more comfortable for the clients. The average log of price in our training sample was 4.736 while the best model had a root mean squared error of 0.356.

Data and Modelling decisions

The data used includes many variables that provide the models with information that carries predictive power. These include information about the host(like the experience), and also about the apartment. Some modeling decisions were such as not including interaction terms in the models. This is because most of the variables are categorical, and they happen to be unbalanced. This means that when taking a sub-sample while performing cross-validation, we are very likely to end up with variables with zero variance, which is not good for the models. Some of them will not be able to provide any predictions and the cross validation loop will break.

Models and Results

- **OLS**
  The first model is the most straightforward of all. We put all of our variables in a regular OLS regression and cross-validate that to mitigate overfitting.

- **LASSO**
  The second one is slightly more complicated, but not too much. Instead of counting only on cross-validation, we use a penalty term to shrink the terms. Then we do a grid search for the coefficient on the penalty term, the hyperparameter λ. The following figure shows the result of the grid search.

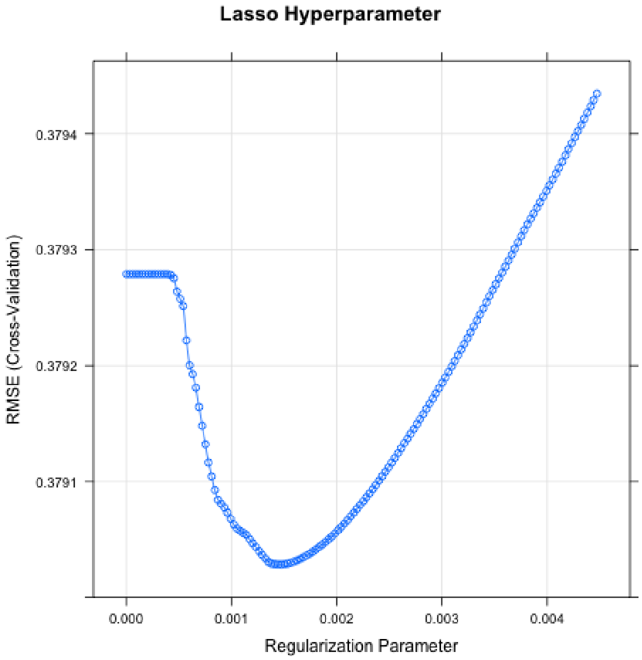

Figure 1: λ grid search

- **CART**
  The third model is a classification tree. I allowed it to optimize automatically over 15 values for the tuning parameter which defines the stopping rule of the tree. The following figure shows the resulting tree. It does not run deep, but it does show the variables that the most important, which are for the most part the number of bedrooms and of people the house can accommodate.

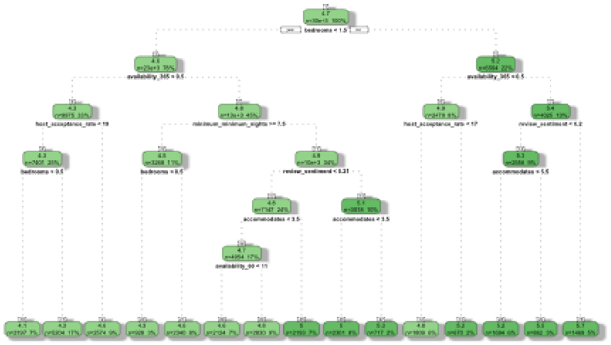

Figure 2: Regression Tree

- **Random Forest**
  The fourth and last one if the random forest which decides based on an ensemble of trees. The stopping rule was fixed at 10 observations per node, but the RMSE was minimized with respect to the number of variables each tree takes. The random forest is also implying a pattern of variable importance similar to that implied by the tree.

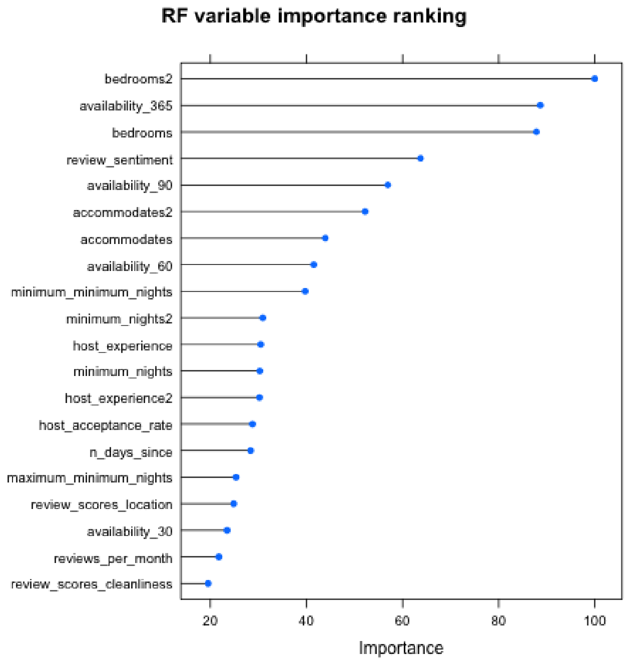

Figure 3: Regression Tree

Running the models on the holdout set gives very similar RMSE scores, which is good news. This provides another layer of affirmation that our models are well specified. The ranking of the models is to be expected. LASSO does not really bring much to the table over OLS, due to the fact that dimensionality reduction is not the issue in this situation. CART performs horribly. That is in part, due to the model being not so capable, and in another part due to me specifying a strict stopping rule. As for the random forest, it provides the best results, and it captures the main drivers of the price change in a way that makes sense.

Table 1: *Summary of results*

$Model$	$RMSE$	$RMSE_{Holdout}$
OLS	0.380	0.380
$LASSO$	0.379	0.379
$CART$	0.426	0.427
RF	0.358	0.356

github repo: https://github.com/RamziChariag/Prediction-ML