Balint Thaler - Data Analysis 3
Github repo: https://github.com/balint-thaler/DA3-Assignment-2

Technical report

Introduction

The aim of the current report is to present some of the technical choices made during the modelling process. The main results and findings are summarised in the summary report. The current document is designed as a complement of the summary report.

Feature engineering

The raw data that was downloaded from Inside AirBnB's website contains 29,537 observations and 74 columns in total. After inspecting the dataset, many columns were found to be either in the wrong format (e.g. prices containing the $ symbol) or irrelevant to the analysis (e.g. IDs, urls, names etc.). First, a total of 27 columns were dropped due to irrelevance, and further 9 were transformed. The latter operations included removal of symbols (e.g. "$", ",", "%"), converting t/f values that represent true/false to 1/0 and extracting the numerical value from a few-word-long text value for bathrooms.

Regarding the target variable, all observations (2156 in total) with missing prices were dropped.

Missing values were handled differently depending on the data type of the specific column. For categorical variables they were replaced with "Missing", which applied to host_location (6,169 observations) and host_neighbourhood (14,534 observations). For number of beds, the missing values (313 observations) were imputed based on the number of accommodates. Lastly, all other observations with missing values were dropped from the dataset. This decreased the size of the data by 5,417 observations.

As the aim of the analysis is to price apartments that accommodate between 2-6 people, the data was filtered so that it only contains apartments from this range resulting in further 2,365 observations being dropped. Furthermore, given the focus of the exercise, all non-apartment-like properties were dropped (4,915 observations).

Lastly, the assumption is made that we are aiming to price a regular apartment which is not in the super high-end price range. Given that less than 2% of the dataset is above 450 EUR, anything above this range was also dropped from the dataset.

The result of all the feature engineering is a clean dataset with a total of 14,422 observations and 46 variables.

These decisions will likely increase the precision of the models but will limit their external validity to apartments outside of these ranges.

The different x variables can be grouped based on what they refer to: there are accommodation-related variables (location, number of beds, property type etc.), host-related variables (host rating, host response time etc.) and review-related variables (number of reviews, review scores on specific aspects of the apartments etc.). Based on intuition, the accommodation-related variables would have the strongest influence on price, followed by the review-related ones. Compared to these, host-related variables are posited to be less important.

Exploratory data analysis

To get a clearer understanding of the target variable, its histogram is charted – see figure 1 below. We observe that prices have a long right tail with some extreme high values, while most of the apartments are priced between 70-130 EUR. Its mean is 125.5 EUR with the lowest value being 8 EUR, the highest 449 EUR. The variable is not converted any further despite the non-symmetric distribution as the aim is to predict absolute prices.

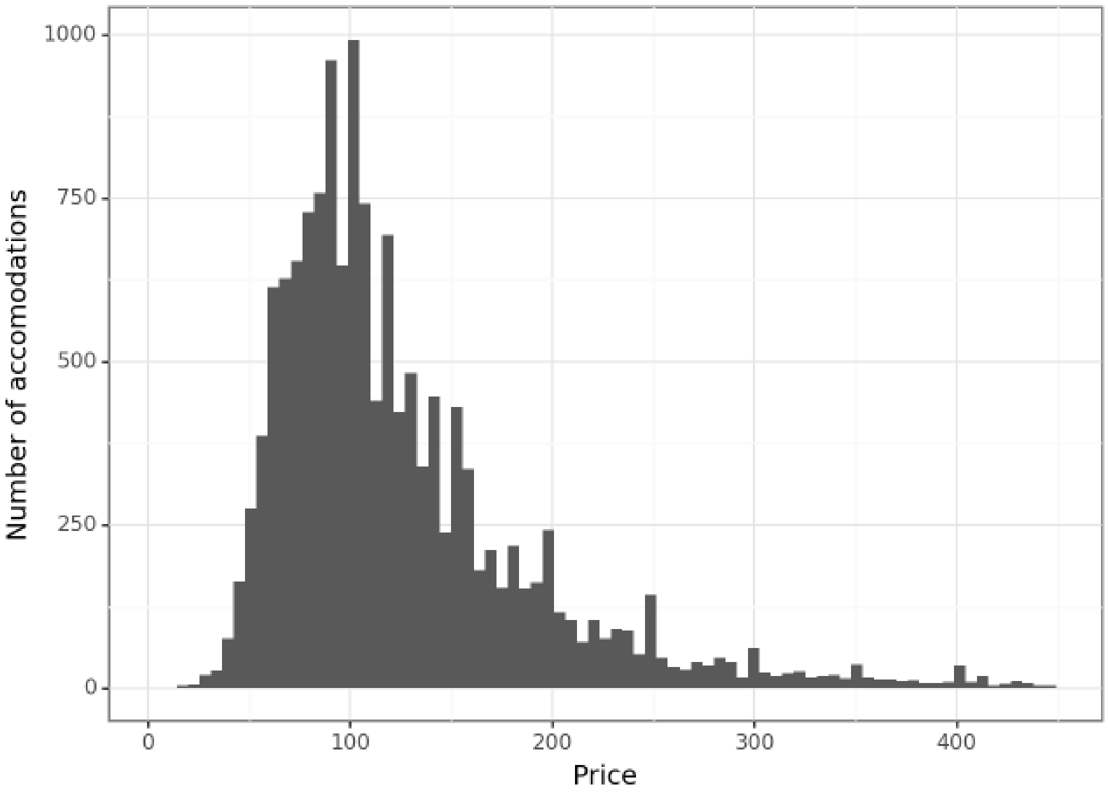


Figure 1: histogram of prices

Further EDA steps are described in the summary report.

Modelling

1. OLS

For the first OLS model the accommodation-related variables are used only. This model is chosen to be simplistic on purpose. To understand some more about these variables, scatter- and boxplots are charted against price (for details, see the appendix). Based on these, two polynomials are added to the dataset for minimum_nights and bathrooms. As mentioned in the summary report, the OLS model performs with an RMSE of 55.38.

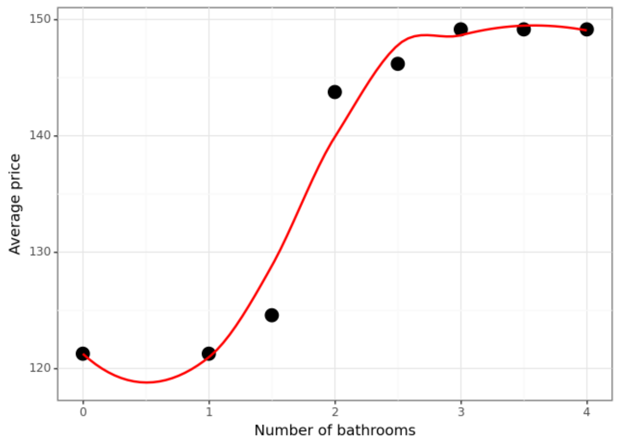
2. Random forest

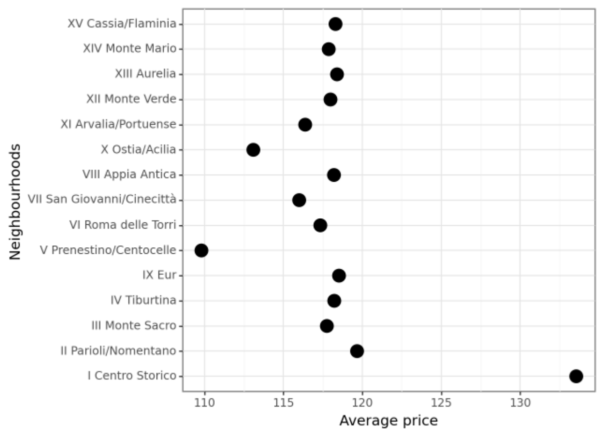
For the second model, random forest, first the theoretically optimal number of variables are computed for maximum number of features to be used, this was 21. The tuning grid for the GridSearch uses a set of maximum number of features (19, 21, 23, and 25) and a set of minimum number of samples in a leaf (5, 10 and 15). After the 5-fold cross validation, the optimal model uses 25 features and 5 minimum variables in each node. Theoretically it would be possible to further optimise this, however the potential gain from this is likely outweighed by the extra time and effort required to identify the best set of parameters for tuning the algorithm, therefore no further optimisation was done.

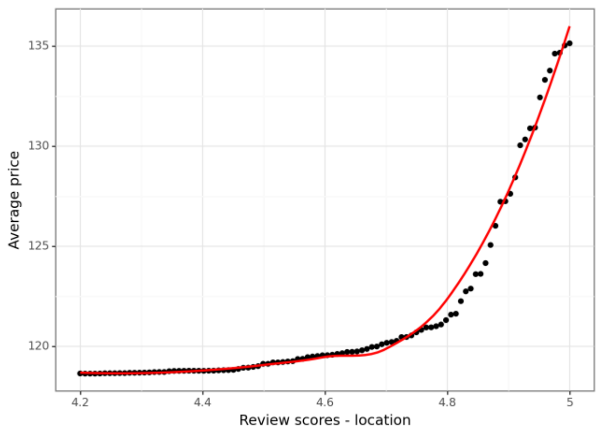
To get the best picture of the variable importances we group the categorical ones together with OneHotEncoder. Once this is done, we see the variable importances for all features. To get a clearer picture of what the patterns of association are between the most important features and the price are, the partial dependence plots are charted for the top 8 features with all other features set for the average of the respective categories.

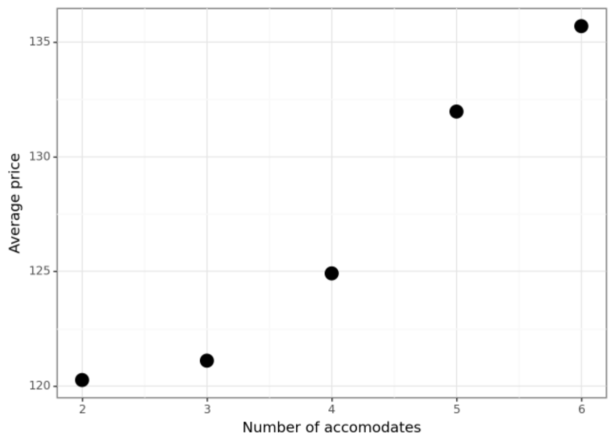
Based on these we find an S-shaped association between number of bathrooms and price, a rather varied picture based on neighbourhood, a non-linear and positive association for reviews on location, a linear-like positive association for number of accommodates, a non-linear negative association for all review number related variables and lastly, a positive but flattening relationship between number of beds and prices.

The charts suggest that both for bathrooms and beds, after a certain number (two and three respectively) the features do not really change the price of the apartments. We can also see that apartments with fewer reviews are priced considerably higher. A possible explanation is that these apartments are simply overpriced, and people are less likely to rent them. As a result, fewer reviews are written about them. Lastly, we observe that association between review scores on location and price blows up above a score of 4.8 suggesting that a good location in Rome is highly valued by the customers. However, it is worth noting that there are relatively few points on the scatterplot above 4.8, so this is a rather exclusive group.









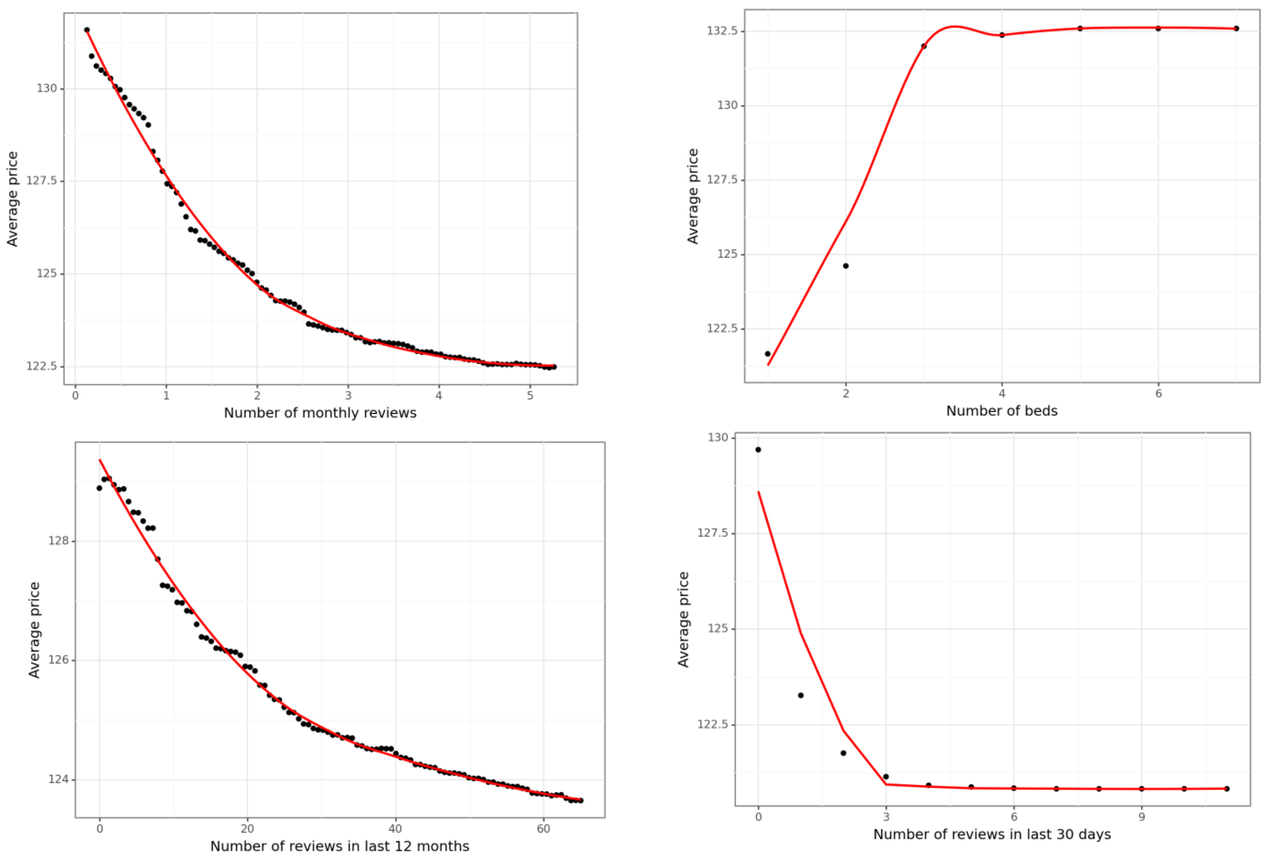


Figure 3: partial dependence plots of the top 8 features

3. Gradient Boosting (GBM)

For tuning the GBM model – similarly to the random forest – a number of tuning parameters were supplied to find the most optimal model. First, its learning rate is set at 0.1, the minimum sample split at 20, the maximum number of features at 10 and the number of estimators used at 50. Then the tuning grid is set up for number of estimators at 200 and 300, and max depth at 5 and 10. Similarly to the random forest, it is possible that we will be unable to identify the absolute best set of tuning parameters, however the potential gain from this is likely outweighed by the extra time and effort required to identify the best set of parameters for tuning the algorithm.

The optimal model has a maximum depth of 10, maximum features of 10, minimum sample split of 20 and number of estimators of 200 resulting in an RMSE of 48.46. The same way as for the random forest, the feature importances are observed both in a non-grouped and grouped setting.

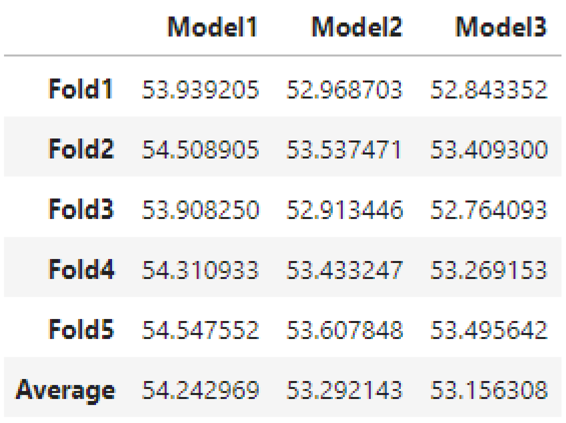
The GBM model identifies more or less the same variables as the most important ones as the random forest, however their weights differ: the random forest's top 3 have a cumulative feature importance of 0.3 while the same figure is only 0.21 for the GBM.

4. Refined OLS

The last refined OLS model is constructed using the outputs of the ML models. For bathrooms, beds, accommodates and review-related variables some polynomials were added based on the shapes observed on the partial dependence plots. Furthermore, three interaction terms are also considered between beds and neighbourhoods, beds and accommodates, and finally beds and bathrooms. The three models considered here contain:

- Model 1: the top 7 variables from the feature importance list
- Model 2: on top of model 1's, the polynomial terms are added
- Model 3: on top of model 2's, the interaction terms are added

To find the best model a 5-fold cross-validation was performed. The best model was model 3 (the most complicated one), but only with a rather small margin – see table 1 below.



	Model1	Model2	Model3
Fold1	53.939205	52.968703	52.843352
Fold2	54.508905	53.537471	53.409300
Fold3	53.908250	52.913446	52.764093
Fold4	54.310933	53.433247	53.269153
Fold5	54.547552	53.607848	53.495642
Average	54.242969	53.292143	53.156308

Table 1: Cross-validated RMSE scores for the OLS models

As mentioned in the summary report, based on the RMSE scores of all models considered, the GBM model performs the best with an RMSE of 48.46. However, it is also important to mention that the refined OLS model which was built using some of the outputs of the ML models has successfully outperformed the random forest one. It is still a relatively simple model with only 7 features, but it produced lower RMSE scores than a highly sophisticated machine learning algorithm nonetheless. However, it would have been extremely difficult to identify the most relevant and important features using domain knowledge only. The great benefit of the OLS model is its rather simple nature (no tuning parameters, relatively simple mathematics behind it) and its ability to be interpreted simply.

Shapley values

Lastly to understand the best performing model, the Shapley values are calculated and visualised for the top 7 features based on the training set. It is noteworthy that this is an extremely computation-heavy calculation, the time required to calculate them on a dataset of 11,537 observations took 1 hour and 26 minutes in total. The most important findings about the Shapley values can be found in the summary report.

Appendix

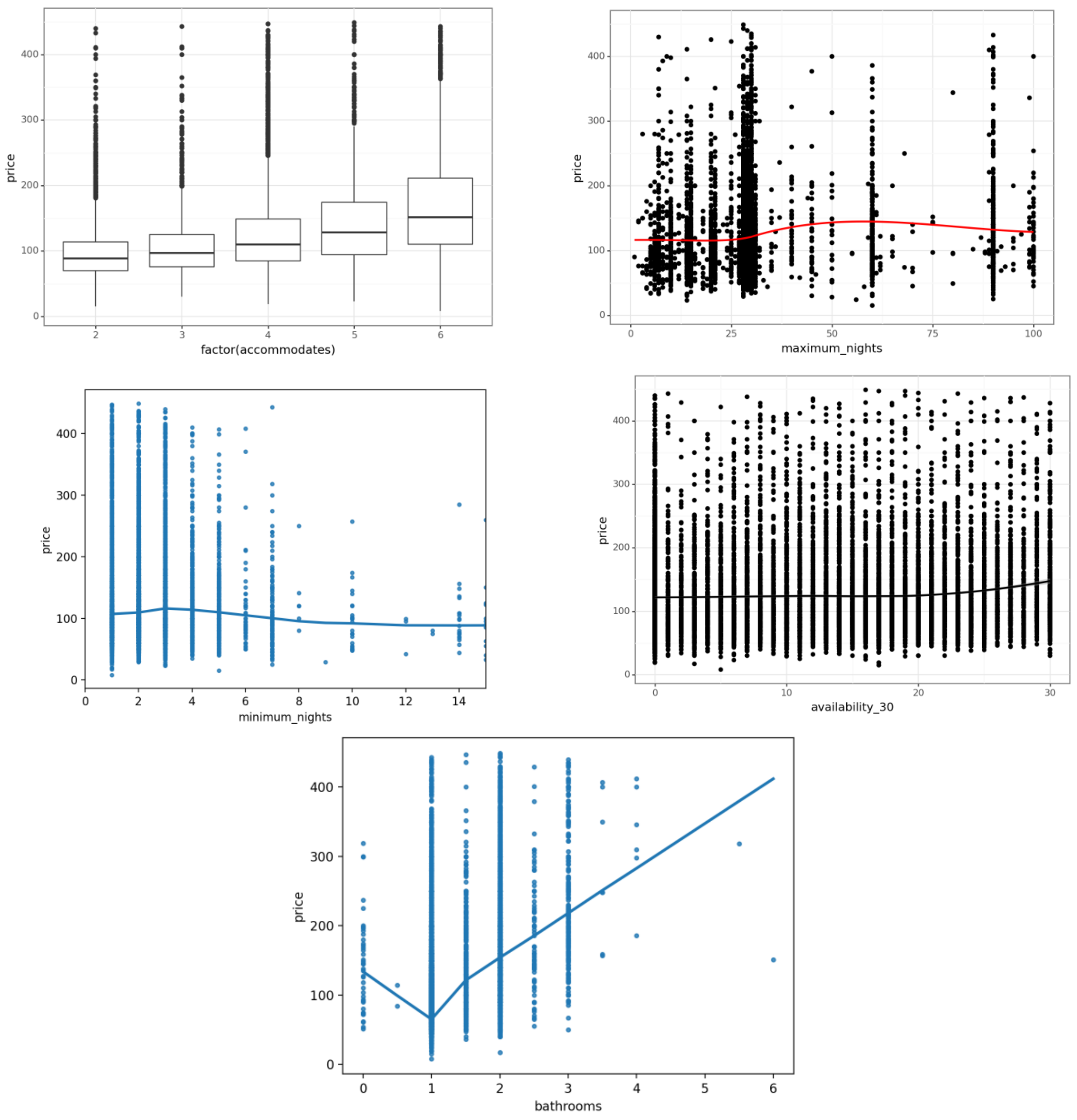


Figure 2: scatter- and boxplots of some of the variables in the OLS model