DSF Group Project:
Rent Price Predictions for Apartments in Switzerland in 2019

3,580: Workshop Fundamentals of Data Science

Maximilian Georg Hans Bisges (18-606-533)
Alessandra Elivia Eugenia Cerutti (18-607-622)
Amélie Carla Hoa Madrona (20-613-816)
Jason Roger Rosenthal (19-614-965)

St. Gallen, 5th of December 2021

Abstract

The purpose of our data science project is to analyse and predict rental prices of apartments in Switzerland in 2019 based on various features. For this purpose, we use a dataset that we received from Novalytica AG, a real estate data analytics company based in Berne, Switzerland. The dataset was constructed by web-scraping the popular apartment listing platform Comparis. The raw dataset comprises about 100'000 observations of 139 variables describing features of the listed accommodation. We first clean the dataset and choose the most meaningful variables. Then we improve on the existing dataset using text analysis. Next, we present the dataset through exploratory data analysis before we use different regression models to predict the rental prices. Finally, we compare and discuss the results obtained.

Contents

1 Cleaning

The first step of our analysis is to select meaningful variables. To form a first impression of the dataset, we inspect the number of missing values in each column and we have a scan through what each column contains. We then proceed to remove columns for which there are too many missing values, that are meaningless (for example the heating system used) or that contain almost the same information as other columns. The result of this is a dataset containing 40 variables.

We noticed that the dataset contained information about the MS (Spatial-Mobility) region in which the apartment was located, MS regions being homogeneous regions defined by the Federal Statistical Office of Switzerland (BFS). The BFS constructed these MS Regions in order to separate regions in Switzerland with similar economic and societal characteristics[1]. This variable (which was probably obtained by matching the MS Regions to the given Communes) allows us to spatially group the apartments. This is vital as, for example, Zurich has a different housing market compared to Appenzell. Given that the BFS, however, introduced "labour-market regions" (*Arbeitsmarktregionen*, AMR) in 2018 to replace these MS Regions, we proceed to append the AMRs to the dataset. The use of AMR in our analysis has one great advantage: if needed, we can find a great amount of economic and social data based on these AMRs in the BFS databases. We also decide to keep several variables constructed by Novalytica called Micro Ratings[2]; though we don't have insights on how exactly they were created, we are still interested in knowing whether they are useful for our analyses.

The way we then filter observations is twofold. First, we filter out housings with missing values in the surface area and only select listings with an area higher than 25 m^2, as we consider that this is the minimum area for an apartment to be inhabitable. Then, we use the 1.5*IQR method (adapted from the boxplot system) on rent prices per m^2 grouped by AMR to remove bottom outliers[3]. This is a way to eliminate listings such as hangars, WGs and parking spots (offering low prices for large areas) while considering regional differences in price. Finally, we assume that columns containing 1s and NAs such as balcony or furnished are binary observations, so we assign 0s to missing values. This assumption was necessary since we have only limited information on the metadata of the dataset.

2 Text analysis on the descriptions

In a second step, we investigate whether it is possible to improve our dataset by performing text mining on the apartment descriptions. We conclude that five variables could be extracted from the text: the area of the apartment, the number of rooms, whether it is furnished, whether it has a balcony and the home type (if it is a loft or a studio for example). Generally, the risk of performing unsupervised text analysis to extract missing values is the fact that it can provide incorrect outcomes without possibility of checking. We therefore keep in mind the trade-off between obtaining more insights and possibly including incorrect entries before incorporating the results of text analysis in the datasets.

Working with the area meant working upstream on the raw dataset, as the cleaned version had missing values for rooms filtered out. After using bigrams to detect numbers higher than 25 followed by indicators of an area such as m^2 or qm for observations where the area is missing (5089 in total), our text analysis detects an area for 520 observations. However, after running the algorithm on the observations for which the area is available, we have two findings: (1) our method is quite effective, as the results match with 74% the actual observations, (2) we obtain a Mean Absolute Error (MAE) of 12.9 m2, representing 16% of the mean area. In this case, we consider that including 520 additional observations in the dataset (which already has 77851) has limited benefits compared to the risk of including areas with large deviations. For this reason, we decide not to use text mining to extract values for the missing areas.

Therefore, we restrict our text analysis to the clean dataset, and start with the number of rooms. For this purpose, we look for bigrams containing numbers between 1 and 10 (including halves) followed by "rooms" (and equivalents in all official Swiss languages). To bypass the problem of halves written as "1/2" appearing as "1" "2", we also had to use trigrams. These methods allow us to capture 474

[1] See https://www.bfs.admin.ch/bfs/de/home/statistiken/raum-umwelt/nomenklaturen/msreg.assetdetail.415729.html

[2] These variables assess the attractiveness of apartments based on specific factors and grade them on a continuous scale from 1 to 10 for each commune; there is also a general Micro Rating which seems to summarise the grades from the specific Micro Ratings

[3] The IQR equals the 3rd quartile minus the 1st quartile. We consider observations as outliers if the squaremeter price is below the 1st quartile minus 1.5*IQR. Quartiles are computed for each AMR separately.

observations and it is particularly effective when tested on observations with known rooms. It has an accuracy of 83% and a MAE of 0.15 rooms (representing 4.5% of the mean number of rooms). Though this method is interestingly more accurate than what is entered in the column "rooms" for a few observations, we decide to incorporate the results of text mining only where there are NAs for the number of rooms.

We also decide to include the result from text mining for the "balcony", "furnished" and "home type" features, without checking the efficiency of our method. We make this choice because these variables are of lesser importance compared to "area" and "rooms" and the checking process requires that the description writers include those in the text, which is not always the case for these apartment details. We find 8086 balconies and 1518 furnished apartments, making sure that those are not preceded by a negation (such as "no balcony"). We also look for descriptions of the housing type to see if we can replace generic apartments (*Wohnungen*) by more precise categories (*Attika, Dachwohnung, Maisonette, Ferienwohnung, Terassenwohnung, Studio, Loft*), and we are able to replace the housing type for 1842 observations. We proceed to remove the *Ferienwohnungen*, as we are only interested in apartments rented for the entire month for our analysis.

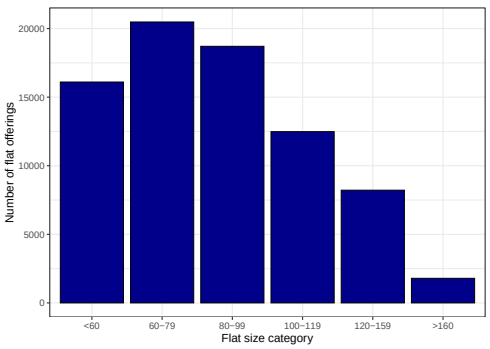
3 Data exploration and analysis

We now explore our dataset in more detail with the help of summary statistics as well as graphical illustrations. Furthermore, we run some simple linear regressions on selected models. A selection of our results is shown below.

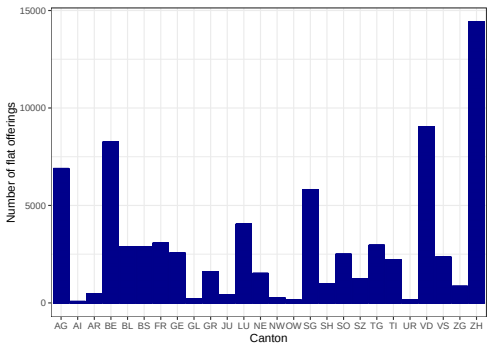
Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
rent_full	77792	1758.91	678.09	290	1330	2030	7800
area	77792	84.17	31.83	25	62	102	360
rooms	76790	3.4	1.11	1	2.5	4.5	15
year_built	29993	1985	49	1087	1970	2015	2021
Micro_rating_new	77792	5.429	1.193	1.055	4.595	6.265	8.885

Table 1: Summary statistics of selected variables

In a first step we want to get an impression which flat sizes are offered in our dataset and how many flats are offered in each canton. Apparently, most flats are offered in the canton Zürich (ZH) while the most occuring flat size is 60-79 m^2.

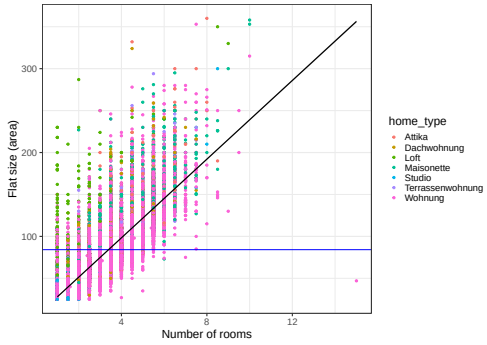


(a) Flat offerings in 2019 depending on flat size

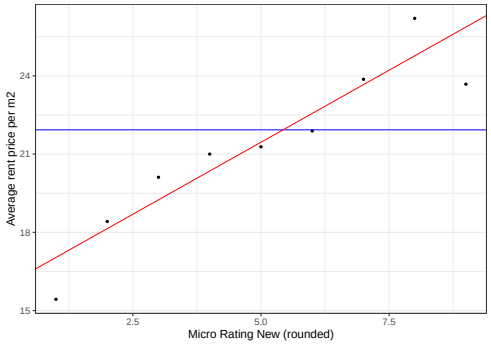


(b) Flat offerings per canton

Knowing this, we want to get an impression about the home types of the flats offered while we are also interested in identifying which home type is associated with which number of rooms and flat size. We can see that for example lofts are often associated with less than 3 rooms but high flat sizes ($>$ 100 m^2). Furthermore, an illustration of the correlation of the general Micro Rating (rounded) shows that a higher Micro Rating is associated with higher rent prices per m^2.

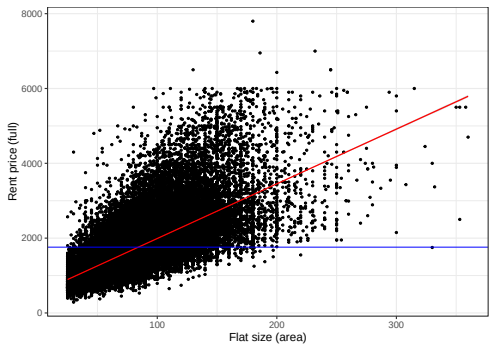


(c) Flat characteristics (rooms, area, home type)

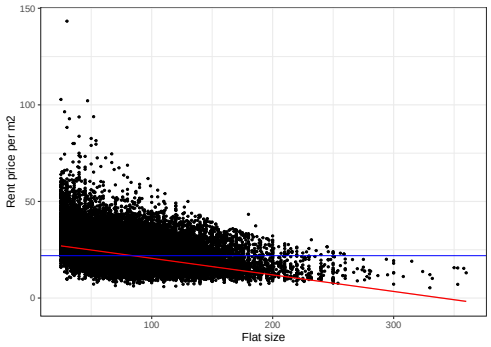


(d) Average rent price per m^2 depending on general Micro Rating (rounded)

In a next step, we show the relationship between flat size and the rent price (full) as well as the development of the rent price per m^2 depending on the flat size. In accordance with market knowledge, the absolute price of a flat is increasing with the flat size while the rent price per m^2 is decreasing with an increasing flat size.

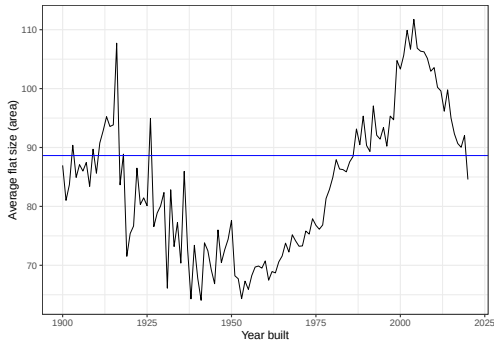


(e) Rent price (full) depending on flat size (area)

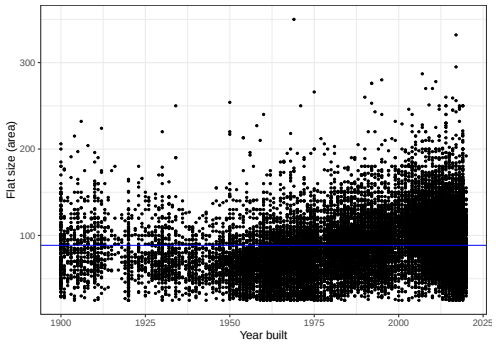


(f) Rent price per m^2 depending on flat size (area)

Finally, as an add-on, we inspect whether we can identify a trend in the development of the average flat size over the years. Of course, we only have the listings available in 2019 but this is still useful to approximate a trend. Due to the little number of observations before 1900 in the dataset, we made a plot from that date to 2020. Although the values are not accurate, the development itself is in line with the data of the BFS[4]. This means that the average flat size decreased until the 1960s, then increased until roughly 2005 followed by a constant decrease since then. Plot (h) shows us that our data points are mostly concentrated after 1950, which explains the larger variance before that date.



(g) Average flat size depending on building year (1900 - 2020)



(h) Flat size (area) depending on building year from 1900 to 2020

[4]See https://www.bfs.admin.ch/bfs/de/home/statistiken/bau-wohnungswesen/wohnungen/groesse.html

4 Models

We now turn to the modelling part. Based on the data analysis of the previous section, we consider the following variables to be essential for predictions: area, number of rooms, balcony (yes/no), furnished (yes/no), distances to places of interest, AMR, home types, and Micro Ratings. The variables used in each ordinary least squares (OLS) and Random Forest (RF) models and Gradient Boosting Modelling (GBM) can be seen in the table below. In general, we partitioned the dataset into a training and testing dataset. On the training set, we computed the OOB or cross-validation errors. Using the trained model, we then computed the MAE of the predictions on the test data.

Variables included	OLS No.	RF No.	GBM No.
area, rooms, home_type, furnished, Micro Ratings, balcony, distances to places of interest, AMR	1	1, 4, 5	1
area, rooms, furnished, Micro Ratings, balcony, distances to places of interest	2	2	2
area, rooms, general Micro Rating, AMR	3	3	3
area, area squared, rooms, rooms squared, home_type, furnished, Micro Ratings, balcony, distances to places of interest, AMR	4	-	4
area, rooms, home_type, furnished, Micro Ratings, balcony, distances to places of interest, AMR, cantons, AMGR	5	-	5

Table 2: Model Characteristics

4.1 Linear Models

The linear model of choice was the OLS. The reasons are straightforward: (1) OLS is a simple yet powerful model with a fast implementation (important for our size of the dataset). (2) It is used widely within the econometric and statistical literature, and it is very easy to interpret (without making causal inferences), (3) It serves as a baseline model with which we can assess the predictive performances of the other models. We estimated five OLS models computed the MAE and Root Mean Squared Error (RMSE) through cross-validation (10-folds) and the validation set (predicting test data) approach. The results can be seen in the table below.

Type	No.	RMSE (CV)	MAE (CV)	MAE	MAE/mean
OLS	1	361.77	248.56	250.42	0.1417
OLS	2	476.45	332.99	335.64	0.1909
OLS	3	373.12	257.99	256.50	0.1458
OLS	4	361.59	249.06	249.16	0.1415
OLS	5	360.00	247.69	244.55	0.1387

Table 3: Results of Linear Models

Please note that the MAE of the models may slightly vary due to randomness. In general, we see that the last model with both cantons and the greater labour market regions included results in the lowest MAE. Model 2 without labour market regions included has significantly worse results. This indicates the importance of geospatial separation when comparing rental prices across Switzerland. While the MAE of the test data shows some differences, the cross-validated MAE shows very similar values across model 1, 3, and 4. Therefore, a consideration of overfitting for the latter two models is of importance. As the cross-validated RMSE is an important indicator for flagging overfitting, we consider this metric. As the differences of the cross-validated RMSE for the three best-performing models are only marginal, we stick to the simplest of them (model 1). Nonetheless, cantonal differences may occur as labour market regions do not need to adhere to cantonal borders, and may stretch over multiple cantons (therefore not taking cantonal differences into account).

4.2 Random Forests

The first non-linear model we use is the Random Forest (RF). The principle is the following: we take bootstrapped samples of the training data and fit a decision-tree with randomly chosen variables to it. The mean of the predictions of all trees is then the prediction of the model. The number of chosen variables, the bootstrapped sample size, and the size of each tree are relevant parameters to optimize the performance of the model. We start with a baseline model (model 1) from the randomForest package made by Liaw and Wiener. We use the default parameters of the package to get a first impression of the results. The next four models used the ranger package from Wright and Ziegler. We then trained RF model 2 and model 3 which consider the same variables as the OLS model with the same model number. This allows for a comparison across different model types. We then attempt a hyper-parameter tuning with all variables and different starting parameters. The model specifications with the lowest OOB RMSE in the tuning were used to train model 4. The last random forest model tried the optimal model specifications from the tuning, but with a sample size equal to the total number of observations of the dataset. The results of the five random forests can be seen in the table below.

Type	No.	mtry	Nodes	Sample size	RMSE (OOB)	MAE	MAE/mean	Time
RF	1	40	5	1	319.1	213.69	0.1209	96.78
RF	2	4	5	1	396.6	270.33	0.1532	1.13
RF	3	33	5	1	339.58	229.53	0.13	7.47
RF	4	45	3	0.8	318.94	215.12	0.1225	8.85
RF	5	45	3	1	319.68	208.82	0.1184	10.51

Table 4: Results of Random Forest Models

The results of the RF models highlight some notable aspects. Firstly, we work with the OOB errors which also flag potential overfitting. Firstly, modelling with the randomForest package took considerably longer than with the ranger package. Secondly, the RFs with the same variables as their OLS counterpart all delivered considerably better results. In absolute terms, we decreased the test data MAE by roughly CHF 40 for the best model, which means we increased the predictive performance by roughly 15%. The OOB RMSE of the first, fourth, and fifth model are very similar again. This result is mildly disappointing as it essentially means that the hyper-parameter tuning did not result in notable performance boosts. Although we decreased the node size of the trees in the latter two models, we also increased the number of variables used in each tree (mtry). Perhaps 45 variables is too many, resulting in possible overfitting. A hyper-parameter tuning across more parameter combinations may lead to better performing models. Increasing the complexity of the models, however, means a trade-off in both time and possible variance of the predictions.

4.3 Gradient Boosting Modelling

The second non-linear model we use is the GBM, which sequentially adds new models to the ensemble. For every iteration, a new weak, base-learner model is trained with respect to the ensemble learnt so far.

First, we adapt the hyperparameters used for GBM. Due to time and computational costs, we cannot use a full grid search for hyperparameter tuning. To find reasonably optimal parameters by running only few tests, we use constraints optimization to lower cross-validated RMSE and the test set's cross-validated MAE, taking the execution time into account. Since the errors are squared before being averaged, the RMSE penalises large errors more than the MAE. The latter measures the average magnitude of the errors in a forecast set. Since the RMSE is always larger or equal to the MAE, it is useful to examine the difference between both values. The more significant the difference between MAE and RMSE, the greater the variance in the individual errors in the sample.

We start by changing one hyperparameter while holding the others constant. To screen a wide range of values for the hyperparameter in a few runs, we either double or halve the value depending on the RMSE and MAE of the previous test. We continue to adjust the hyperparameter as long as the errors (RMSE and MAE) decrease. We then change the hyperparameter after finding a plausible value for the local minimum on the error curve. Then we apply the same procedure for the other hyperparameters, minimising both RMSE and MAE each time.

For our GBM, after evaluating twelve tests, we set our number of trees to 600, with an interaction depth (the number of splits performed on a tree) of 6 and shrinkage (learning rate) of 0.1, cross-validation folds of 5. It has an RMSE of 313.67, a MAE of 216.46 and a computing time of 7 minutes.

We then proceed to train our models using the optimal hyperparameters found previously. We train five models with the same selected variables as in the linear models, to be able to compare GBM with OLS and RF. The results are shown in the table below.

Type	No.	Time	RMSE (CV)	RMSE	MAE	MAE/mean
GBM	1	7.25	321.41	313.67	216.46	0.1231
GBM	2	1.28	424.19	426.56	296.79	0.1683
GBM	3	5.92	338.47	341.04	232.56	0.1321
GBM	4	6.84	320.06	318.47	216.41	0.1232
GBM	5	6.90	320.32	317.53	215.72	0.1228

Table 5: Results of the GBMs

We observe that model 1 gets similar results to the models 4 (which includes on top area squared and rooms squared) and 5 (which includes Arbeitsmarktgrossregionen (AMGR) and the cantons). This suggests that the added variables in models 4 and 5 have little significance in the performance of the rent price prediction. Model 2, which does not include AMRs, gives the worst prediction of rent prices; this was also the case with the linear models. Thus, it confirms our assumption that the geospatial separation is useful to predict the rent price.

When comparing the result of GBM to OLS, we notice that GBM is much more accurate in predicting rent prices; we see in all five models a lower MAE compared to their OLS counterparts. RF compared to GBM has similar results in terms of MAE (e.g. RF 1: 213.69 vs GBM 1: 216.46). One difference between the two non-linear models is the approach to tuning hyper-parameters. While we tuned the hyper-parameters of the RF using a hyper-grid method, we adjusted the hyper-parameters of the GBM by adapting the parameters sequentially and then trained the models with selected variables.

5 Discussion of results & Conclusion

We begin this section by reviewing the process of our data science project again. From an initial raw dataset, we first cleaned the data and selected the meaningful variables for our analysis. In order to extract information from the descriptions of the listings, we used text mining, identified new insights and thus improved about 10'000 observations. We are content with the performance of our text analysis and regard it as an important component of our data science project. Subsequently, we use the improved dataset to gain insights through exploratory data analysis. We created summary statistics and plotted graphs to highlight important relationships between variables. This step prepared us for our last task, modelling and predicting the rental costs of apartments in Switzerland in 2019. Through three different models using various combinations of variables, we obtained models with a MAE of about 210 CHF. This means, on average, these models' prediction were within a 210 CHF range from the actual rental prices. We think this performance is a success, considering the spatial, attractiveness, and size differences across all apartments. Local real estate agents are certainly also capable of such estimation performances. However, how many real estate agents can estimate rental prices accurately across different regions of Switzerland and across apartments with very different features?

There are still, however, some areas where this project could be improved on. Considering the data cleaning, we faced a trade-off between removing irrelevant observations (e.g. parking spaces, holiday homes, etc.) and not increasing the bias of the data. We tried to overcome this challenge by removing outliers through a heuristic rule which we adapted from the boxplot system. There may, however, be more effective ways of cleaning the data in this respect which we did not take account of. Next, although we deem the text mining of the description as a success, there may again be an increased bias through extracting wrong information from the descriptions. Further procedures and more advanced methods could be applied in this respect. With respect to the modelling, we have always attempted to gain unbiased error samples in order to guard the models from overfitting. We, however, never analysed the variance of the prediction errors, which could provide further insights in the nature of the models.