**Budapest apartment for rent analysis**

**Introduction**

My friends and I often discuss the hardships of finding a great apartment to rent on a student budget. It is a difficult process in every large city, but I came to understand how outstandingly hard it is in Budapest to find a nice place. The issue in this capital is, that while wages are increasing YTD ~10%, the apartment prices are increasing on an even larger extent. Therefore, while parents are supplying their student children with as much as possible, they cannot keep up with the market prices. Let’s take a look at some numbers to uphold this argument: The Central Bureau of Statistics in Hungary (KSH) reported an average net income of ~700 EUR in Hungary. According to Numbeo, which is a crowd-sourcing based cost-of-living calculator website, an average rent per month for a 1-bedroom apartment outside of city centre is 375 EUR while in the city centre it is approximately 500 EUR. This means that the average Hungarian, if s/he wants to rent an apartment in Budapest, will spend ~54% to ~71% of wages on the rent. Madness! The rental fees for apartments are highly correlated with the real-estate prices, which also saw an exponential increase since the 2008 crisis. At the beginning of 2020, experts stated that the real estate market is going to normalize by rising proportionally to the extent of wage growth. This seems more and more unlikely due to the economic effects of the novel coronavirus, which will inevitably decrease the demand for, and thus, decrease the price of real estate, allowing the population to take a breath from the massive price increases. Nevertheless, it is still going to be a tough job to find an affordable place to rent in Budapest for Hungarians.

In this project, I aimed to research and understand the drivers of rental fees for real-estates. My focus is on providing both descriptive and predictive results in showing the relationship between price and attributes of an apartment listing. As an end product, I want to help renters and rentees alike to understand the market better and to be able to calculate the costs of renting depending on personal needs about the apartment (e.g. it needs to be close to my university, has to be furnished, etc.). So to summarize **why should you keep reading:**are you thinking about renting a place in Budapest? Or are you thinking about renting out your place instead? I will show you**how much it will cost you to rent or how much you can earn by renting your place out!**

To support my analysis, I will be using data from ingatlan.com, which is arguably the largest real-estate sales platform in Hungary. Instead of using a snapshot of data from the website, I decided to collect the data over a month to have more predictive power. I also decided to flag the data based on when it was inserted and deleted from the website. The latter is especially interesting because it can mean two things: either the place was rented out or the owner decided to take the listing down. What does that mean for us? We will see the last price the apartment listing before a potential contract, making us able to infer the actual value of the apartment (however, we will not be able to consider the price negotiations between the tenant and the owner…).

**Methodology**

           I will try to not bore you to death with the methodology, but I think it is important for you to understand the steps I followed to conduct this research. I will try to keep it as short as possible though.

I have started with creating a pipeline for scraping the data, applying change data capture (CDC) mechanism and loading the data into a DB2 database on cloud. The CDC is just really a fancy word for identifying the newly inserted, updated and deleted data from the website. I used the beautifulsoup4 package to scrape the data from ingatlan.com, then I stored the data in a .csv file which was picked up by the CDC script to make the transformations and store that in a .csv. In the extraction part, I removed duplicates based on the address, nr. of rooms and price variables. Lastly, this file was picked up by a pyspark script to load the data into the DB2 database. Before loading everything into the database though, I translated everything to English. The data from the website is mostly handwritten with some loosely defined categories. This means that inside one category you can add as many categories as you want (e.g. when describing the heating system, some people just added everything in there..) To avoid having awfully lot of subcategories I defined some umbrella terms. To make things a bit more fancy and automatized, I used Apache Airflow to schedule and run these scripts in the order in one job. With Airflow, I was able to set a schedule to run the job once a day. That is all about the data collection, let's move to the analysis part.

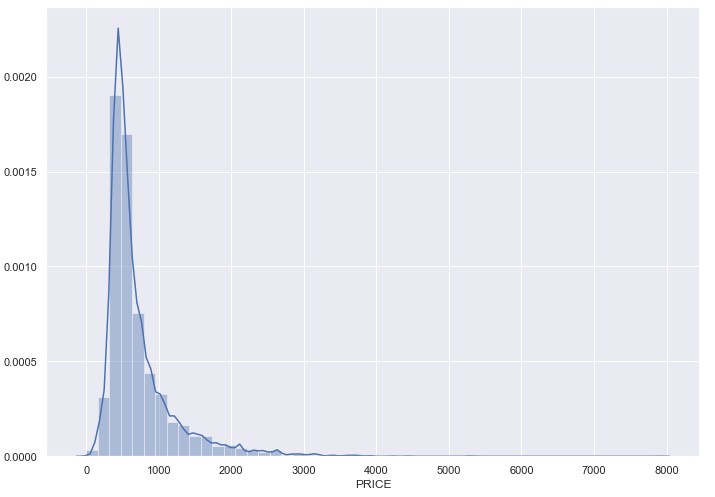
Remember when I was talking about the data being handwritten? This issue is going to make a comeback here. There were plenty of cases where because of this the data quality is, let’s just say, questionable. Therefore, I started the data wrangling with an extensive feature engineering. But before we dive into that, let me acquaint you to the variables:

|  |  |
| --- | --- |
| Variable name | Short description |
| sTREET | *The address of the apartment* |
| dISTRICT | *The particular district in Budapest* |
| sIZE IN m2 | *The total area size of the apartment in m2* |
| rOOMS | *Total nr. of rooms* |
| pRICE | *Rental fee* |
| cONDITION | *The shape of the place (e.g. newly built)* |
| cOMFORT | *In Hungary this refers to the nr. of bathrooms a place has (weird, right?)* |
| eNERGY uSAGE | *Refers to the EU energy performance certificate* |
| fLOOR | *On what level the apartment is on* |
| lIFT | *Does the flat have an elevator or not* |
| bUILDING hEIGHT | *The height of the building, pretty straight forward* |
| iNNER hEIGHT | *In other words: ceiling height* |
| aIR CONDITIONING | *Does the apartment have AC or not* |
| fURNISHED | *Is it furnished or not* |
| uTILITIES eXPENSE | *Total amount of utilities cost* |
| wHEELCHAIR aCCESSABILITY | *Is it wheelchair accessible or not* |
| oRIENTATION | *Is the apartment facing to north, east, west or south* |
| vIEW | *are the windows looking to a garden or to the street or to the corridor, etc.* |
| bALCONY | *Balcony size* |
| hAS gARDEN | *Does it have access to a garden* |
| hAS aTTIC | *Does it have an attic* |
| hAS eQUIPMENT | *Is the kitchen equiped* |
| pETS aLLOWED | *Are pets allowed or not* |
| sMOKING aLLOWED | *Is inside smoking allowed or not* |
| pARKING | *Is there parking possibility, if so is it paid or free* |
| pANEL pROGRAM | *Did the flat recently participate in a renovation program. This refers to flats built during Communism.* |

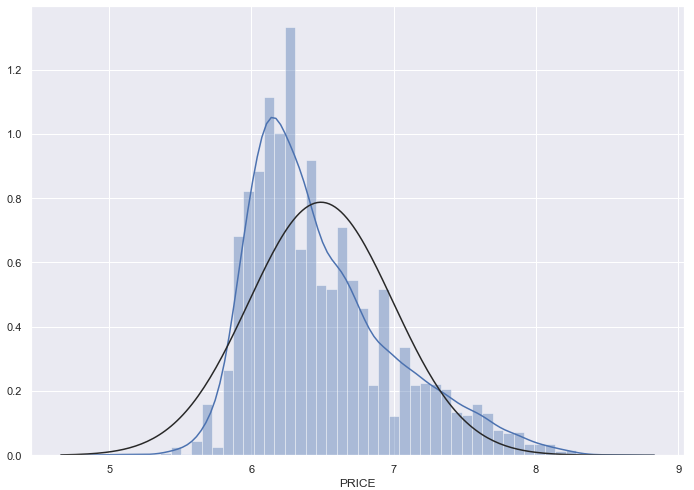
Now that you are somewhat familiar with the variables, let’s dig into the feature engineering. In some cases, the utility expense was incurred within the selling price, meaning that the seller left the field without a value. Also, some sellers indicated that the apartment hasn’t got a balcony by not adding a value to that field. In both cases, I swapped the null values with 0 instead. Also, I created a new column indicating if the price contains the utilities or not. Moreover, I changed some seemingly nominal values into ordinal ones, such as the energy usage variable, condition and comfort. Next, I converted the price and utility currency from HUF to EUR using the daily conversion rate. Lastly, categorical variables were one-hot encoded.

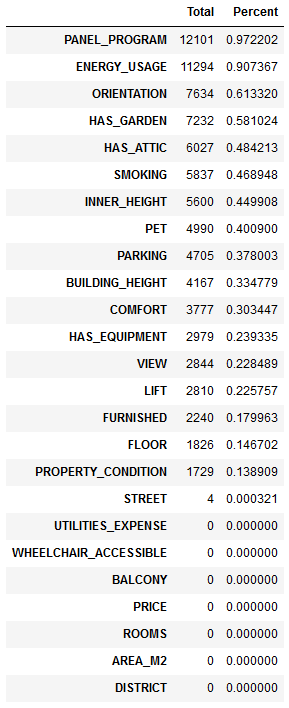
At this point, I started looking into the data. I was still suspicious about the quality of my data so the first thing I did was an outlier analysis. I used the Z-score method, which finds the distribution of the data where the mean is 0 and the standard deviation is 1. I used the standard z-score threshold which is between 3 and -3, meaning that the data points outside of this threshold are considered outliers. Ok, now let’s check the maximum price of an apartment with and without outliers: EUR 422,859 with and EUR 7,901 without. Let’s be honest, the first value is completely unrealistic, no one would rent an apartment for such a price. I further checked some other variables to confirm my suspicion: some of my data is unrealistic and their existence would probably harm the predictive model, thus, I removed the outliers from the dataset. There were 497 removed rows, which reduced our observations to a total of 12,447.

Let’s further analyze our predictive variable. Keep in mind that we want to create a linear regression model. One of the assumptions of linear regression is that the value that we want to predict is normally distributed. When modelling variables with non-linear relationships, there is a higher probability to produce errors as well. Ok, now that we know this, let’s check the distribution of the Price variable:



Hmm.. that doesn’t look normally distributed to me. In fact, it has a long right tail with a skewness of 2.83 and kurtosis of 13.04. This makes total sense: there are many listings with similar prices, and there are some more luxurious apartments that cost more. But what do we do about the normality assumption? Well, there’s a quite handy trick called log transformation. Logarithm improves the fit of the model by transforming the distribution to a more normally-shaped bell curve. Let’s apply this in practice and see the results:

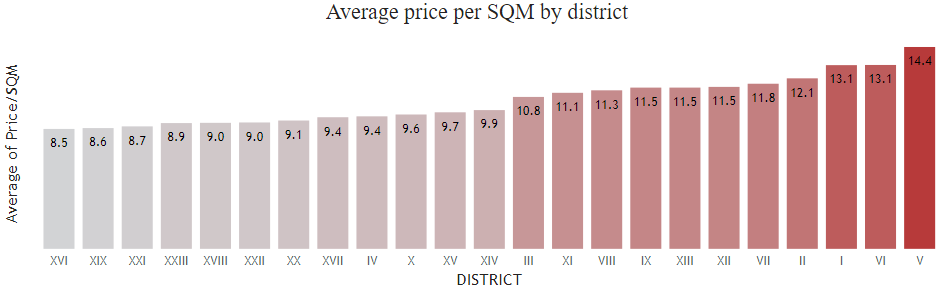


The black curve indicates a perfect normal distribution whereas the blue curve indicates the distribution of our data after log transformation. Compared to the baseline distribution, this looks much better and workable.

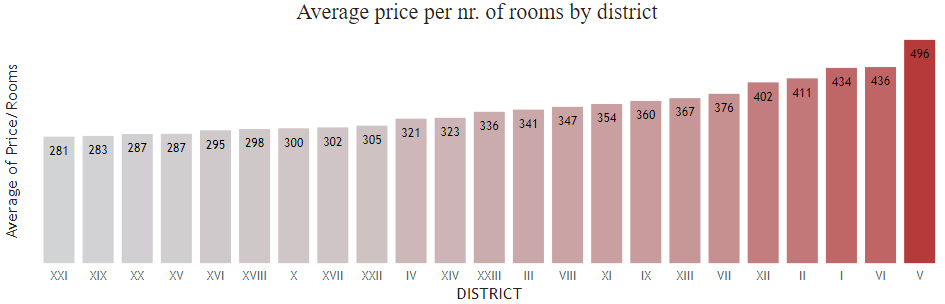
Now we know a bit more about or dependent variable but we still didn’t deal with a huge issue: missing data. As you might have guessed already, this dataset has a lot of missing data. In the table to the right, I put together the exact number of missing values per column. As you can see, some of the variables are completely useless as we would never be able to ‘guess’ the values for them and in these cases, it is also not advisable to swap them with 0 or with the mean value. Therefore, I decided to remove all columns that have more than 25% of the data missing. The remaining null values were dropped as well, meaning that the final number of observations is 6914 rows with 15 predictor variables and one dependent variable. These are the variables used to predict the rental fee: District, area size, nr. of rooms, condition, floor level, lift, air conditioning, furnished, utility expense, wheelchair accessible, view, balcony, has the equipment and has utilities.

To build the model(s), I first split the data into training and testing sets and used the sci-kit learn package. You are going to see three-four models: the baseline, which is a linear regression model and three with some regularizations, ridge, lasso and ElasticNet regressions. Without going into too much detail this is what I mean by regularization: ridge regression adds a penalty term to your base linear function and penalizes the model for the sum of squared value of the weights. In lasso, the model is penalized for the sum of absolute values of the weights. Also, this model automatically detects the variables that have the least impact on your model and it doesn’t use them. Lastly, ElasticNet is a combination of Lasso and Ridge. I used the cross-validation method to select the best alpha for these models. The regularizations also help us tackle the issue of multicollinearity and heteroscedasticity. These are especially useful if you have lots of data with lots of variables.

**Results**

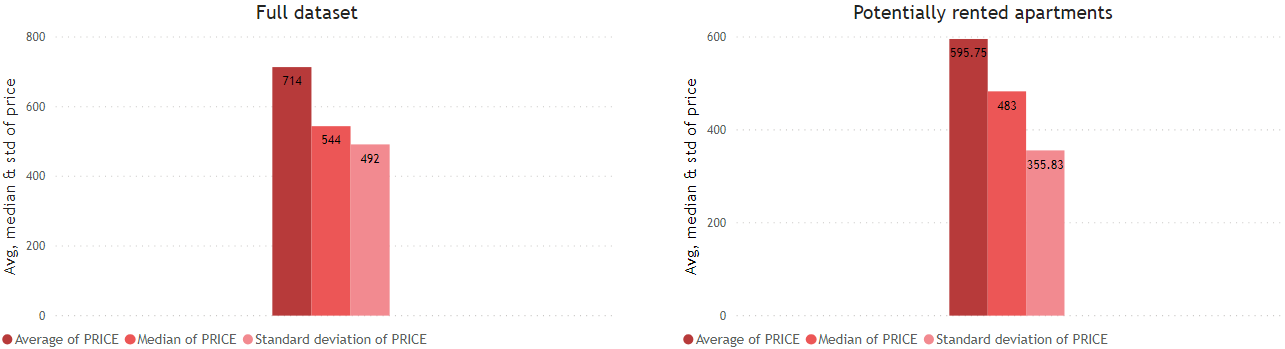
Let’s start with something simple yet quite interesting: checking the average price per m2 by the district. This way we can have a quick look at which district is the most expensive to rent in so we will have a rough idea about the market. Also, we already leverage the most intuitive correlation in our dataset: price with area size. I am using the whole dataset for the following graphs.

Not surprisingly, the highest average rental fee per m2 is in the city centre with €14.4 per m2, which is followed by some other highly touristic districts. Price/m2 is great when you plan to buy a real estate, but in case of renting, you will probably end up renting a room for yourself. So, let’s also check the average price/room by the district.

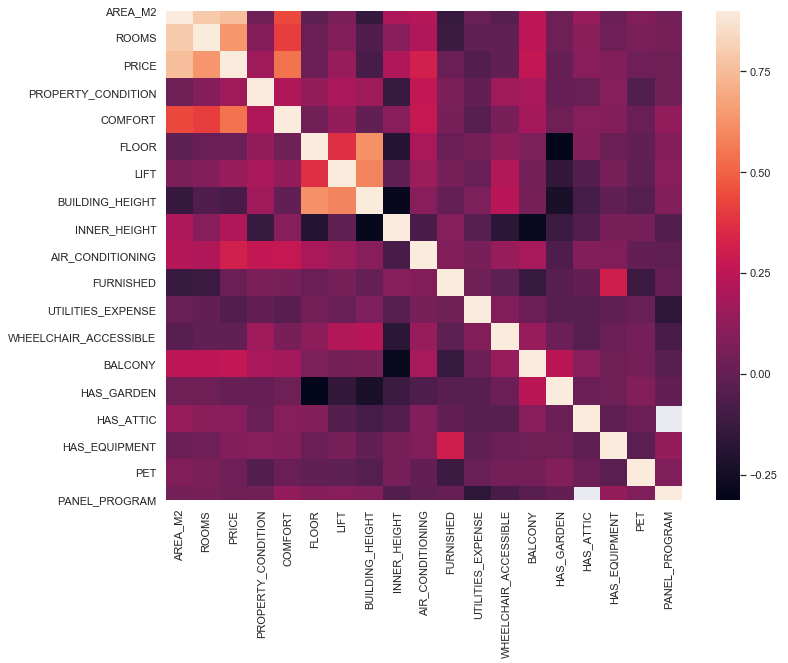


Now that’s something more tangible for renters! It looks like the trend continues with highly touristic districts being the most expensive ones to rent in. We can also see that the cheaper options are (obviously) far from the city centre, however, as a tradeoff people renting there must cope with quite long transit hours to get to the centre. I suspect that the variables used here are going to be important for our regression model too.

Let’s take a quick look on the apartments that are potentially rented. As I told you before, at the data collection part I implemented the CDC mechanism for check for data that isn’t on the website anymore. The total amount of that data after outliers is 4701 observations. At the graphs below you can see an average, median & standard deviation comparison between the potentially rented apartments and the total dataset. The former gives us a quite good estimation on what is being sold on the market compared to what is advertised. Whereas the average price on the total dataset is EUR 714, the potentially rented apartments are more than EUR 100 cheaper. This shows us that, while there is supply for more expensive, and therefore, more luxurious apartments, the demand is looking for a lower price.



Onto the correlation between our variables! From this time on, we are going to focus our analysis on the total dataset. Remember, we are interested in which variables affect the most the rental fee. This means that we are looking for variables that are correlated to price. Below, you can see a correlation matrix which is an extremely useful graph to visualize correlations (and also to detect multicollinearity, which can hinder the results of our model). The lighter the color the higher the correlation.

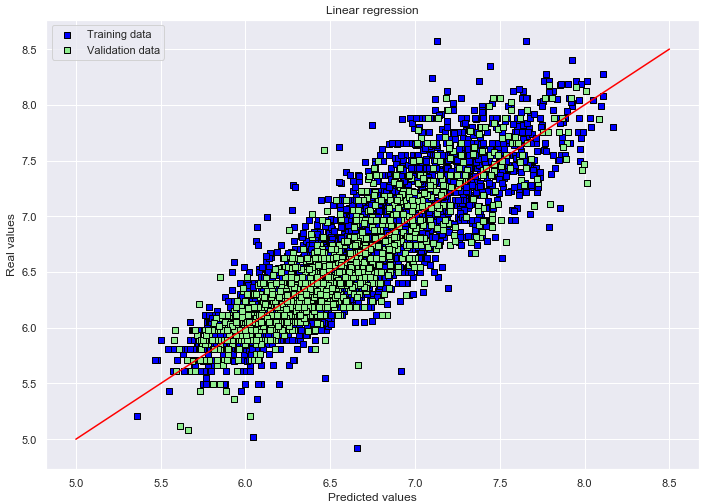


As suspected, area and rooms are highly correlated with price. Comfort seems to be correlated as well and to some degree having an air conditioner and the size of the balcony seems to be correlated with price too. Interesting. However, area, rooms and comfort seem to be highly correlated with each other. This potentially poses a problem, but we can solve that by regularizing our model later on. We will not be able to use all the variables presented in the matrix, though, because of missing values. After removing null values as described in the Methodology section, I started building the regression model.

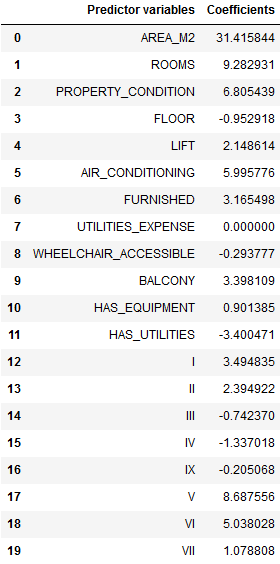
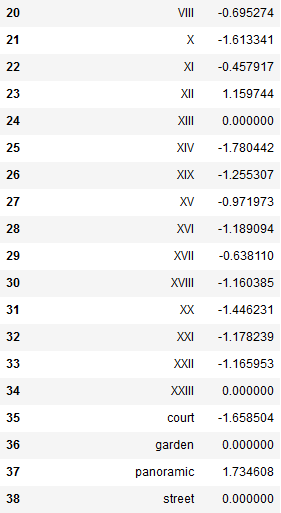
You can check the below table to see which variables we ended up using. I have indicated whether the variable is numerical or categorical. Categorical variables were one-hot encoded. I also indicated if the numeric value is originally ordinal data or nominal with two values. The latter was converted into either 0, indicating the non-existence, or 1, indicating the existence of something.

|  |  |
| --- | --- |
| Predictor Variables | Variable Type |
| Size in M2 | *Numeric* |
| Nr. of rooms | *Numeric (ordinal data)* |
| Property Condition | *Numeric (ordinal data)* |
| Floor | *Numierc* |
| Lift | *Numeric (0 or 1)* |
| Air Conditioning | *Numeric (0 or 1)* |
| Utilities expense | *Numeric* |
| Furnished | *Numeric (0 or 1)* |
| Wheelchair Accessibility | *Numeric (0 or 1)* |
| Balcony Size | *Numeric* |
| Has Equipment | *Numeric (0 or 1)* |
| Has Utilities | *Numeric (0 or 1)* |
| District | *Categorical (one-hot encoded)* |
| View | *Categorical (one-hot-encoded)* |

After splitting our data into training and test, let’s start building our baseline model. Our base model is just a multiple linear regression model using all the above independent variables to predict price. I have used the work of [juliencs on Kaggle](https://www.kaggle.com/juliencs/a-study-on-regression-applied-to-the-ames-dataset) to plot the residuals and predictions. He defined an RMSE calculator with 10-fold cross-validation which is used for assessing how accurately our model performs together with a traditional R2 score. With our baseline model, we achieved an **RMSE of 0.2339** and **R2 of 0.7938**. Let’s map the actual values to the predicted ones:



Here you can see the training data in blue and test in green. The red line shows the intersection of actual values with predicted ones, where we aim to be as close to this line with our predictions as possible. It looks like with higher values we have higher variability and more dispersed from the actual values. The ridge, lasso and elasticnet models barely improved the RMSE of our baseline model and in some cases, it even decreased the R2. This is because we are working with a small number of predictor variables. However, in my data pipeline, I automized the variable selection based on the available variables. Once a variable has less than 25% missing data, then it will automatically become a predictor variable in our model to enrich it. Nevertheless, I achieved the best **RMSE** with the ElasticNet model with a score of **0.2338**and a **R2 of 0.7937**. You can see the coefficients of variables below. Since our dependent variable was log-transformed I exponentiated the coefficients, subtracted one from the numbers and multiplied by 100 to give the percentage increase/decrease for every one-unit increase in the predictor variables. You can see a couple of 0 values for coefficients, that means that our ElasticNet regularization shrinks the least significant variables down.



**Discussion**

This is the part you’ve been waiting for. While the model is not perfect, it does give you a quite accurate prediction about rentable apartments in Budapest. As we suspected, the size of the apartment weighs the most in determining the price followed by the number of rooms the place has. This means that, all else stays equal, a 1 m2 increase in the area size of the apartment leads to a 31.41% increase in the price of the apartment. Another interesting relationship to look at is the effect on the location of the apartment based on districts. As we saw at the beginning of the Results section, clearly the touristic districts were more expensive. This is represented in the model as well. If you look at the coefficient of district V, you can see that if the place is located in it, that will automatically increase its price by ~8.7%, all else stays equal.

So, as a person who would like to rent an apartment or rent out his/her apartment, keep in mind the following: area size increases the price of the apartment by the most; having more rooms will also increase the price by quite a lot, with each additional room increasing it with ~9.3%; The higher the apartment is situated in the flat the more decrease in the price, however, having an elevator in the flat can offset this issue; having air conditioning increases the price by 6%, which actually comes quite handy in the extreme summers in Budapest; if the apartment is furnished, the price increases by ~3.16%; if it is wheelchair accessible then it is worth ~0.29% less (quite counter-intuitive); the balcony size has a ~4% effect on the price with each additional m2; if there is kitchen equipment has a 0.9% effect and if the utility is not included in the price it decreases the fee by 3.4%; lastly, keep in mind that the location has a great effect on the price too!

Now that we successfully quantified the effect of different variables on price, it is time to reflect on the process, on what could be improved. To achieve even better predictive power, it is important to have the best data quality possible. Therefore, having only one data source might be a problem, especially in our case. The people who share their apartments on the website have in their best interest to showcase the apartment in the best condition possible, e.g. showcasing a larger m2 size, to earn as much as possible for it. Thus, it is possible that some data points do not represent the reality and that is why other sources should be added to our dataset to validate our data. One such source would be from a tax office, because the incentive for the people here is to showcase their apartments in such a way to minimize taxes, e.g. smaller m2 size. Another source could be a different real-estate website that has other variables that we didn’t incorporate into our model.

Let’s talk about the predictive power of our model. This depends to a large extent on the data available to us. In this case, we had lots of missing data, so we decided to use those columns only that have less than 25% of data missing. We could feed more variables into our model by increasing this threshold, however, that would decrease the amount of data our model is using due to null value removal. This is a trade-off to be considered: are we interested in better predictive power on a smaller and more concentrated amount of data? Or are we interested in less powerful but still viable model with less concentrated data.