CAPSTONE PROJECT

REAL ESTATE ASKING PRICE ESTIMATION IN BUDAPEST

PUBLIC PROJECT SUMMARY

By

Tamás Burghard

Submitted to
Central European University
Department of Economics and Business

In partial fulfillment of the requirements for the degree of
Master of Science in Business Analytics

Supervisor: György Bőgel

Budapest, Hungary

2019

Table of contents

The original business goal of the project

The aim of my project was to develop an automated, deployable real estate price estimator model based on the given database of previous ads listed on the website ingatlan.com. The ideal outcome was declared as a complete machine learning pipeline with a scalable and deployable model. That means, to implement a software and a machine learning model that performs high-quality price estimation on real estate data, periodically refreshes the model, and provides performance data to follow the accuracy of the predictions. There are several **business applications** using such model, here I specify two:

1. **Helping the advertiser with providing a correct asking price**. It's easy to understand the usefulness of this as most people has no relevant market information and a bad decision has great impact on their wealth.
2. **Automated reporting** to the content moderation team about new **ads with extreme low / high prices** (better content with less effort). The quality of the content is one key factor of the long-term survival.

The client company

Ingatlan.com is a market leader company in Hungary in the industry of online housing listing with around 250.000 active ads. It provides additional services like high quality moderation and financial intermediation, and has more than 20 years of experience in developing software. Although it uses data-driven business decisions and some data science internally, has no expertise in successfully implementing a complete machine learning pipeline at state-of-the-art level.

The question to answer

During the first period of the cooperation we defined the real question that our model should answer, and it is adapted to the first mentioned application and the available data of the company:

"At what price would an average person put that ad on the site today?"

This question implies **two choices** about the price (we estimate the **asking price**) and the **validity** of the price (**the first day of the ad**). Naturally there are other interesting questions like "the price on one could sell the property in a certain amount of time with a high confidence", but to answer this one needs transactional data – that was not available.

We also agreed on that this industry calculates **the price differences in percentages** and using the **m^2 prices** instead of the whole prices. They told me that they prefer having several smaller errors than having a big one, and they are aware of the results of the most famous American site, the Zillow. This site has data of more than 100 million American homes, with a top-notch algorithm being the etalon in the estimation industry. Fortunately, they also share some of their ideas and the precision metrics as well.

Previous work

This project had an unsuccessful predecessor at the company, with methodological errors that caused delays and irreal expectations about the possible outcomes. Finally, the developer left the company. I had a tough situation communicating the invalidity of the former project. Basically, I only kept the base type of the model, and the idea about focusing on flats in Budapest, as this is a dense area that consist the majority of the ads.

Transforming the data

It is a commonplace, but the process before modelling takes about 80% of the time in a data science project – this project was not an exception. **My model relies only on data can be found in the ads**, I haven't used other in-company data, nor external sources. The following several paragraphs intended to demonstrate some interesting aspects of the data by data type:

Textual data: Although it is quite possible to train a deep learning model to 'read' the text description of the ad, at this phase I simply marked the occurrence of some keywords, that could have been found in highly unpredictable ads. This field contains more data than the others, and the utilization is very important, because the users tend to be lazy and not filling in other variables if it is in the text.

Tabular data (numbers like size and room numbers, and categories like 'has elevator' or the condition is very good or poor): This kind of data had severe quality problems due to the low fill out ratio. This could be better with a more conscious data policy later (users should be urged to fill out fields, or utilizing a process with processing the text data and prefilling the matching fields)

Pictures: Deep learning models could find useful features of the uploaded photos, but it was not in the scope of this project.

Spatial data: The 3 most important factors are: location, location, location… says another commonplace in the real estate business – it is easy to understand how important the data quality is here. The main problem was the **lack of consistency of the spatial data**. Moreover, coordinates (if exist) and the stored location info were imprecise, and inconsistent with each other. I wrote an algorithm that tried to solve the conflicts forcing a clearer hierarchy than the client has now– and this was the most important step toward a better precision. However, precision still remained a problem, as less than 40% of the ads had house number info. (= precise data)

In data science, **duplicated data** could cause unwanted bias. Luckily, my client uses a duplicated property finder solution that works well (it is based on matching uploaded photos). I selected the ad with the most information of each duplicated group. There are lots of them as different agents could advertise the same property at the same time. I haven't used further duplicate detection, seeing the results.

Building the model

I created **a 2 step model**, where at the **first step** I imputed the **mean prices** to every ad from the previous 3 months of the same area, and at the **second step a boosting model** had to **estimate the difference** between the local mean and the actual price – with this, the model learned features that more or less independent of time, avoiding the time series difficulties with boosting. To find the optimal **hyperparameters** for the model, I used a Google Cloud computer instance with GPU, funded by the client. During this long process I took care of the proper time-based cross validation. Around 35% of the cases the error of my final model is less than 5%, and half of the cases had less than 7.56% deviance.

The milestones

My **first milestone** was an **interactive application** written in R with Shiny, that let the client to navigate on a map, to select a location, to set all the parameters that the model used, and get back the estimation. It is very useful to have an application like this, because one can see how the model behaves changing the parameters. It runs locally on a computer, using a specific version of the model.

The **second milestone** was a **deployable** script, that every day gets the fresh ads, transforms them and makes predictions with *different* models and store all this information in the cloud, to let the client monitor the daily performance of different models. A Tableau dashboard was made on top of this by a data analyst of the firm.

Benefit to the company

The main benefit for the client is being participated in a data science product development. This expertise will be crucial for the long run. Some of the business decisions and even the structure itself should be (and will be) changed: the proper production and storage of the data, involving data people into the core development and planning procedure, create a cooperative environment between the data team and the programming team, identifying the potential lucrative data projects.

Data and data science-based services (especially with data not available to others, like a monopoly) might be the cash cow of the company within years, giving an advantage on competitors that haven't share this vision. Continuing this project could yield a positive feedback for the company makes it establish a special data science team.

Topics to improve

I have already mentioned some, but there are four topics to improve: **using different type of models** (like K-nearest members, close to the appraiser logic), **using additional data** (like Deep learning NLP features on the ad descriptions, real estates outside Budapest), **using better data** (the client should incentivize the users filling out more variables, NLP text extraction. location info for all ads) and **finalizing the project** (automatic model update, complete Docker and Github compatibility)