UNIVERSITEIT GENT

CONFIDENTIAL
Do not copy, distribute or make public in any way

Faculty of Sciences

Building a real estate spatial price prediction model for a web portal.

Kasper Van Lombeek

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promoter: Prof. Dr. G. Everaert
Co-promoters: Prof. Dr. D. Benoit
Prof. Dr. M. Van Meirvenne

Department of Applied Mathematics,
Computer Science & Statistics

Academic year 2014 - 2015

UNIVERSITEIT GENT

CONFIDENTIAL
Do not copy, distribute or make public in any way

Faculty of Sciences

Building a real estate spatial price prediction model for a web portal.

Kasper Van Lombeek

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promoter: Prof. Dr. G. Everaert
Co-promoters: Prof. Dr. D. Benoit
Prof. Dr. M. Van Meirvenne

Department of Applied Mathematics,
Computer Science & Statistics

Academic year 2014 - 2015

Disclaimer

Foreword

In high school, we learned for the first time to make a graph. We were drilled by our physics teacher B. Rombouts to name axes, think about the scales and add a self explaining legend. Later on in university, we had to make a report of a lab about the electricity laws. We did not have a single measurement result as we wasted the whole lab talking about student life. At home, we faked all the measurements, adding noise to the formulas according the proper physic laws, and made a nice report according to the standards learned in high school. Although we did not have a single measurement, we took the maximum. I should have realized already back then that I loved numbers.

After working five years, Caroline convinced me to take a year off and study the master statistical data analysis at UGent. Although I was working (struggling) for years in Excel making conclusions out of numbers, I did not even have a clue what a regression was. One and a half year later, the world of statistics makes so much sense, and R is a tool that I could not even imagine not having at my disposal any more. I am very grateful to her, she made me jump ship, a choice I never regretted.

Passing all the courses wasn't easy, and my questions to Koen, Adriaan and Ludger must have been endless. I can also imagine to have bugged several professors and assistents. I takes me a pretty long time to understand difficult concepts and literally can't sleep without knowing I get them.

When my parents were renting out an apartment, I could not understand that a real estate agent estimates the value by looking at the paint color and a cupboard below the sink. Via via I ended up with a dataset of online listings, and started to build an estimation engine. My promotors, G. Everaert, M. Van Meirvenne and D. Benoit made me understand the methods of econometrics, spatial statistics and the more recent data mining techniques. I also need to thank J. Meys, T. Verbeek, K. Boussauw and I. De Vos for helping me with R and the cadastral data.

And of course I would like to thank my parents and my brother for their continuous support.

Abstract

Correctly estimating the price of a house is an important step in selling or buying a house. Often the house for sale is listed online, including a text description, pictures and some mandatory features. This thesis uses the listings from Flemish real estate portal websites to estimate the transaction price of all houses in Flanders. It is the intention to use this model on a website to automatically estimate the transaction price.

First, several other open data sources are merged with the listings. With the "Grootschalig referentie bestand" (GRB), the Flemish cadastre, we were able to calculate important features such as land area or distance to street for all 2.3 million houses in Flanders. The governmental average transaction prices per postal code and neighbourhood density classifications are also used.

In a second part, data is simulated to mimic the listing dataset. On this simulated data, three models are tested and discussed:

- A two step model: first, a global generalized additive model is fit ignoring the spatial correlation. Secondly, the variogram of the residuals is used to interpolate the varying mean price over space.
- Geographically weighted regression: this model is fit to account for the heterogeneity of the feature effects. Instead of one global model, a separate weighted local model is fit for each house to predict.
- Generalized Boosted Regression: this black box model is based on consecutive simple models and does not require any statistical assumptions.

In a third part, the advantages, feature effects and goodness of fit of these three models are further discussed based on the real listing dataset. Five fold cross validation results are tabled and maps are made of the spatial effects. We find that, however the very different assumptions, all three models offer similar prediction power on the real listing dataset.

Contents

1 List of abbreviations

GRB: Grootschalig referentie bestand, the GIS of Flanders, made publically available by Agiv, the agentschap voor geografische informatie Vlaanderen

FOD: Federale overheids dienst, the belgian department of economics

AGIV: Agentschap voor geografische informatie Vlaanderen

GIS: Geographic information system, which is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of spatial or geographical data.

OLS: Ordinary least squares

EPC: Energie prestatie certificaat, a measure of the energy efficiency of the house.

GAM: Generalized additive models.

OLS: Ordinary least squares.

GWR: Geographically weighted regression.

2 Introduction

Selling or buying a house is a tedious process as we only do so once or twice in a lifetime. The first question to ask is: what is the house worth? At what price will I be able to sell the house, or what is a respectable initial offer for the house?

An obvious start when we sell or buy a house is to look at other houses that are sold or being sold in the neighbourhood of interest. We try to estimate the house based on the prices others paid for similar houses. All houses are however very different, not two single houses are equal. Hundreds of features are taken into account when we decide at what price we are interested in buying the house. Does it have a garden? If so, is it orientated towards the sun? Is the neighbourhood quiet? Is it close to a highway so I can get to work easily? Does it have two bathrooms? Is the house recently renovated? Many more features can be thought of, which makes the comparison very difficult (hence the famous saying: comparing apples and oranges).

Compairing apples and oranges is the art of statistics. Just as we use the results of very different patients (thin and thick, young and old, smoking and non-smoking, ...) when we are testing the effect of a medicine, we can take into account all the features of a house to estimate it's price. The aim of this thesis is to automate this estimation process, which can then be implemented in an online portal website to generate an estimate of a listing.

This idea is not new and already very familiar in the United States thanks to the real estate portal website Zillow (screenshot 1). Publicly available data is used to estimate the house price of each and every house across the United States. As visible on the screenshot, every single house of the neighbourhood is estimated, not only the listed houses for sale.

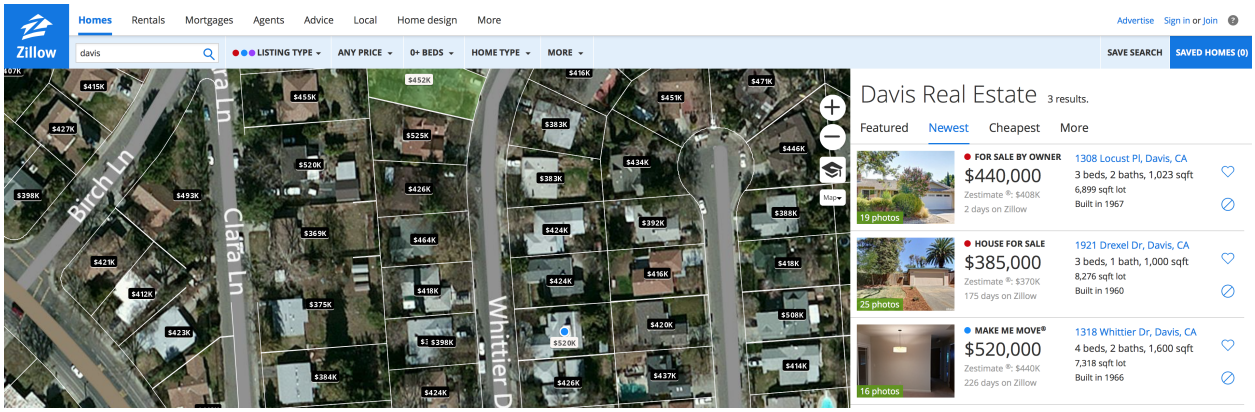


Figure 1: A screenshot of the Americian real estate portal site Zillow. On the right, we see listings of houses for sale, with the Zestimate, an estimate of the transaction price made by Zillow. On the left, we find of each house in the neighbourhood an estimated price.

The aim of this thesis is to build a similar estimation engine. The first step is to gather online publicly available data, and transform into house features. The second step is to build a statistical model, which uses these features to estimate the transaction price.

3 Data collection

A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician.

3.1 Introduction

To build the house price valuation engine, one needs a dataset with recent sold houses and their features. By combining different datasources, the aim is to find recent online listings with at least the following features:

- The sale offer price, or ideally the transaction price
- The total time the offer was online and price history
- The address including a categorizing neighbourhood code
- Features such as number of bedrooms, surface area, orientation, garden area, garage, ao.

The overview of the data collection is given (figure 2). The four data sources (web portals such as Immoweb, Grootschalig referentie bestand from Agiv, Ruimte monitor and open government data) and the algoritms reshaping the data with be discussed in the consecutive sections.

3.2 Gathering real estate listings

Most houses sold in Flanders are listed online, on either a large portal site such as Immoweb and Zimmo, or either directly on the website of a real estate agent. These listings can be gathered via a scraping techniques. When a listing is scraped, the html content of the listing has to be normalized, meaning that features such as number of bedrooms have to be recognized and filled in the proper number of bedrooms column. After the scraping and normalization of each large immo portal site, all the normalized listings have to be compared with each other, as a house for sale might be listed on different immo portal sites. This is

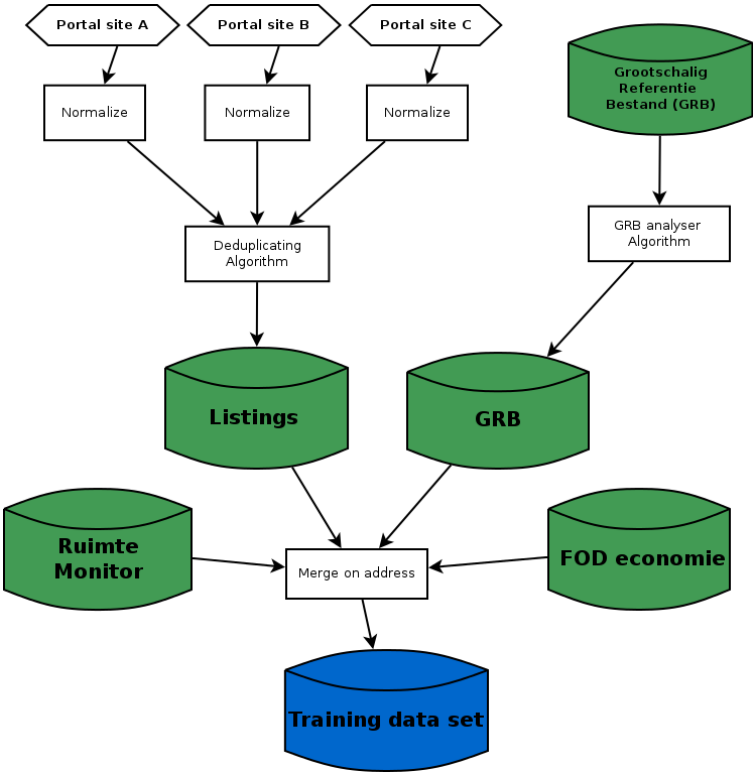


Figure 2: Overview of the data collection process. The four databases are shown in green. For some databases, pre-calculations were necessary to reshape and restructure the data. These four databases will be merged by address to build a model training dataset.

done in the de-duplicating algorithm, as pictured (figure 2). This is a crucial step in the data gathering process, as many potential double listing are found. If two listings pass all the following checks, they are considered as equal:

- Is the address of the listing the same? Issues arise in appartment blocks, where one can have several listings on one address.
- Is the price of the house the same? Issues arise when there is doubt if the price is just lowered, of it is a complete new listing of the same property.
- Is the time period of the sale of property the same? Issues arise when a property was sold, renovated and sold again at a higher price.
- Is the type of property the same? Issues arise when a house is sold, and one year later only the restaurant at the ground flour is sold again.
- Is the type of transaction the same? Issues arise when a property is for sale, and 2 months later part of the property is for rent.

3.3 Text mining of the listings

Lots of information can be extracted out of the description of a listing. An example is to use the description of a listing to check if the property includes a garage or not. This is harder

than it looks, as a description often contains an "option to buy or rent a garage". Looking only for the word garage would lead to mistakes, as listings with an option to rent or buy a garage are equally flagged. But looking for option and garage would not be sufficient either, as the word option is used as well with other features, such as "option to build another sleeping room in the attic". The following algorithm (made with the help of the text-mining R packages) was developed to tackle this issue:

- Do we find the word garage in the description?
- If so, use a tokenizer to devide the description into sentences, words and symbols.
- Use a part of speech tagger to classify each word as verb, noun, adverb, ...
- Find the word garage and its position in the list of words.
- Do we find the word "option" within five words and before the nearest punctuation before the word garage?

Lots of other information can be extracted out of the description of a listing. Very often, this is information that is hard to classify, such as the current state of a property, or the pleasant neighbourhood of the house. We decided not to spend too much time and research in this emerging field of statistics for two reasons. Firstly, it is beyond the scope of a master thesis in statistics, and secondly because houses that are not listed do not have a description. At the end, the goal is to estimate each and every house of Flanders, and of more than 99,9% of the houses in Flanders we not have any kind of description, but only a polygon and a location.

Only the following features are extracted out of the listing, mainly to detect outlying observations:

- Does the listing include a garage?
- Is the listing publically sold? If so, the asking price is only a starting offer, and of the day of the public sale the actual sale price will be a lot higher.
- Does the building require serious renovation works?

3.4 Cadastral data of 2.3 million buildings in Flanders

The second challenge is to find extra features that might explain the asking price of the listing, and that is available for every house in Flanders. The author obtained a database dump of the "Grootschalig referentie bestand" (GRB) of Vlaanderen. This GRB is what is known as GIS data ("Geographic Information System"), which contains shape files. Traditionally this type of data is managed with GIS software, such as ArcGis. To combine the GIS data with the other data sources, and to be able to use the recent statistical data-analysis techniques, the author decided to analyse the GIS data with R. An R plot of a 3 by 3 km square of the GRB GIS data is shown (figure 3).

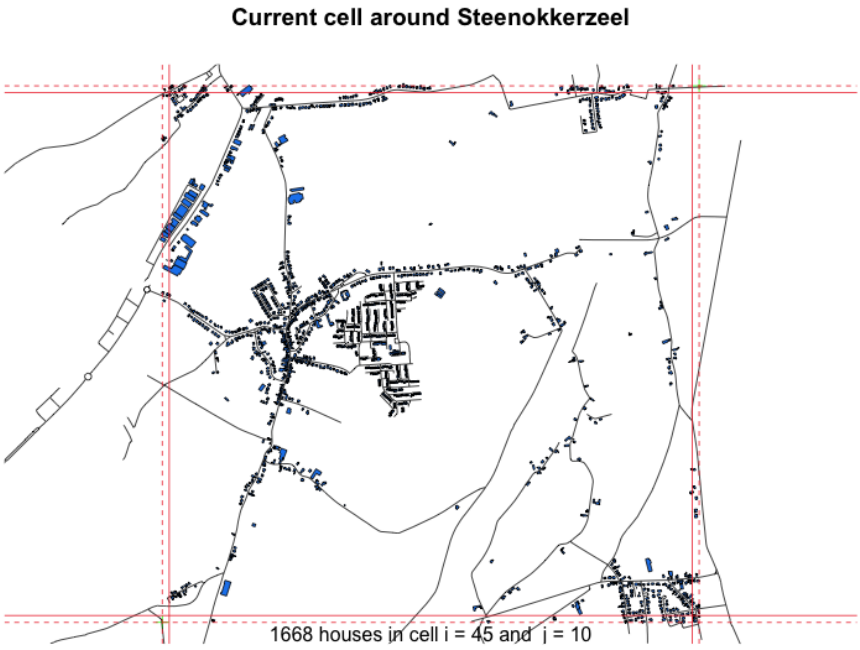


Figure 3: R plot of the cadastral data of all the buildings in a 3 by 3km square grid.

In R, the shape files are read in and converted to "Spatial dataframes". An R-dataframe is a matrix with observations in a row, and features (which can be characters, numeric values, factors,...) in the columns. As the name suggests, a spatial dataframe is a classic dataframe with a spatial object attached to each observation. Once the GRB dataset is converted to a spatial dataframe, every building in Flanders can be considered as an observation with some features and a polygon. The link between the listing and the observation in the GRB can be made with the address of the listing.

Every building in GRB has a unique ID. Some buildings have however multiple addresses attached to it, for example an apartment block. The features of these buildings are not reliable. How would we for example find out how many percent of the surface of the building

belongs to which address? For this reason, the author decided to use only the listings linked to buildings belonging only to one address.

3.5 Extract more features out of the cadastral data

Every building in the GRB has a link to a street ID in the GRB street shape file. With the combination of the building shapes and street shapes, other features could be calculated:

- Is the building touching to other buildings?
- What is facade width of the property projected on the street?
- What is the distance to the street of the building?
- What is the orientation of the building relative to the street?
- Building density: how many other buildings are in the close by neighbourhood? (within a square of 50 by 50 and 500 by 500 meter)
- Population density: how many other addresses are in the close by neighbourhood? (within a square of 50 by 50 and 500 by 500 meter)

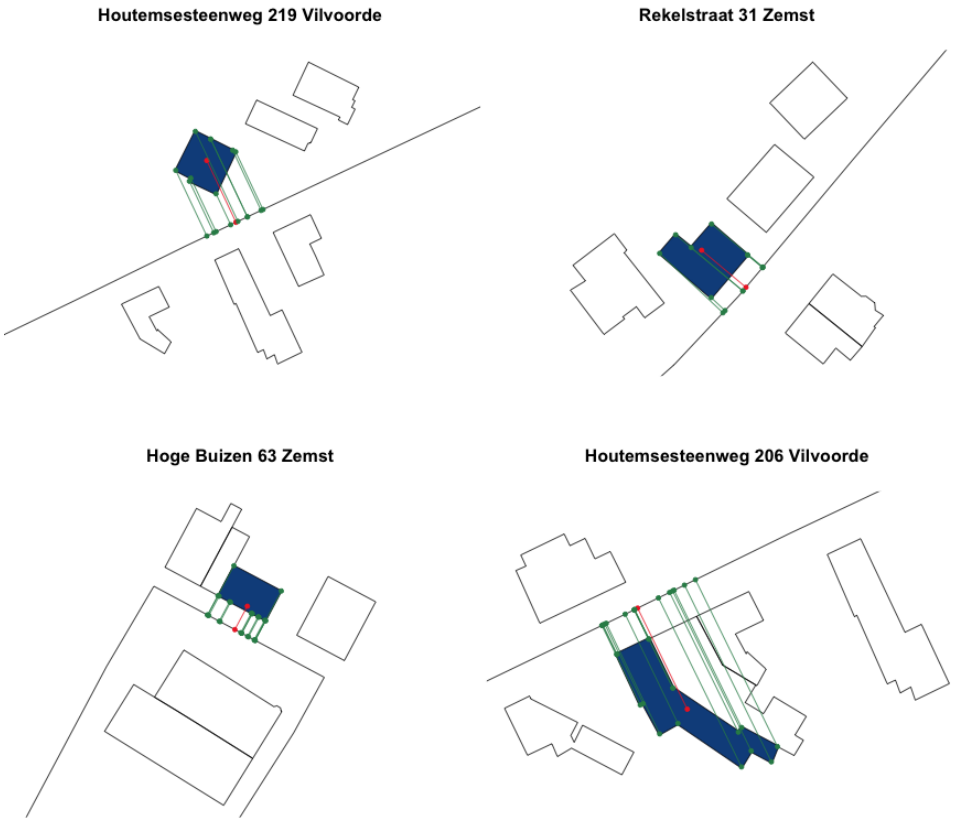


Figure 4: Demonstration of the algorithm to calculate the orientation, facade width, distance to street and touching sides of four buildings.

The algorithm to calculate these features is illustrated (figure 4). First, five nearby houses and street segments are selected. For each of the five nearby houses, we check if the building is touching another. In a second step, the x and y coordinate of the building is projected on the street segments to find the nearest street, which is used to calculate the facade width, distance to the street and orientation. The building and address density are calculated by simple counting the query results in a square around the building.

As is demonstrated (figure 4), the algorithm is not perfect. Problems arise when the building has an irregular shape, or when the nearest street is not found and the corners of the buildings are projected to another street. Overall, the algorithm still delivered good results, this is checked by manually inspecting the calculated features of 20 random selected houses in Flanders.

The calculation of these features of 2.3 million unique buildings can not be done with a simple loop over all the buildings. The most time consuming step is that for every building, the nearest neighbouring buildings and street segments have to be found in the dataset. This is speed up by nesting two loops to select in a first step all the buildings in a 3 by 3 kilometre grid plus a 250 meter buffer (figure 3). For each building in this 3 by 3 kilometre grid, the nearest neighbours are selected and used to calculate the features. When the feature calculation is done for this square, the algorithm moves on to the next square (figure 5). This looping procedure makes it possible to calculate the features of the 2.3 million buildings within 12 hours.

3.6 Open source government data

More and more data is accessible on the web as open data. Two important sources from governmental instances are used:

- Data from FOD economie or Statbel which contains the average property price per postal code (figure 6). Unfortunately, the institution distinguishes between houses and villas, a catagorization which is subjective.
- Data from ruimtemonitor.be, which contains a classification from 1 to 6 per neighbourhood, 1 being "centrum stad" and 6 being "platteland" (figure 7). This is an important dataset, as the granularity is higher than the postal codes.

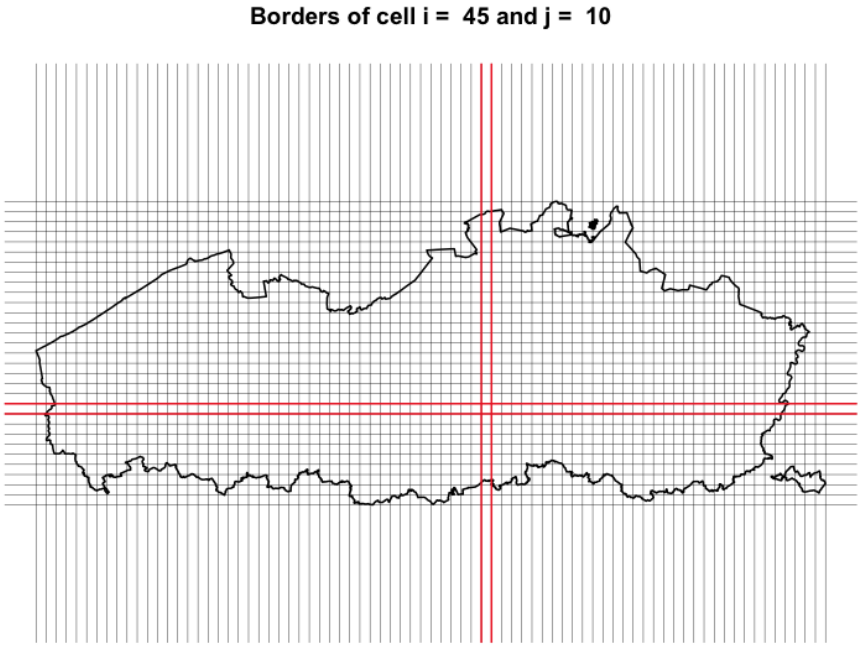


Figure 5: With two nested loops, all the houses in a 3 by 3 km square are queried. Every square in the loop contains around 100 to 5000 buildings and the calculation of all the features takes between 30 seconds and 10 minutes.

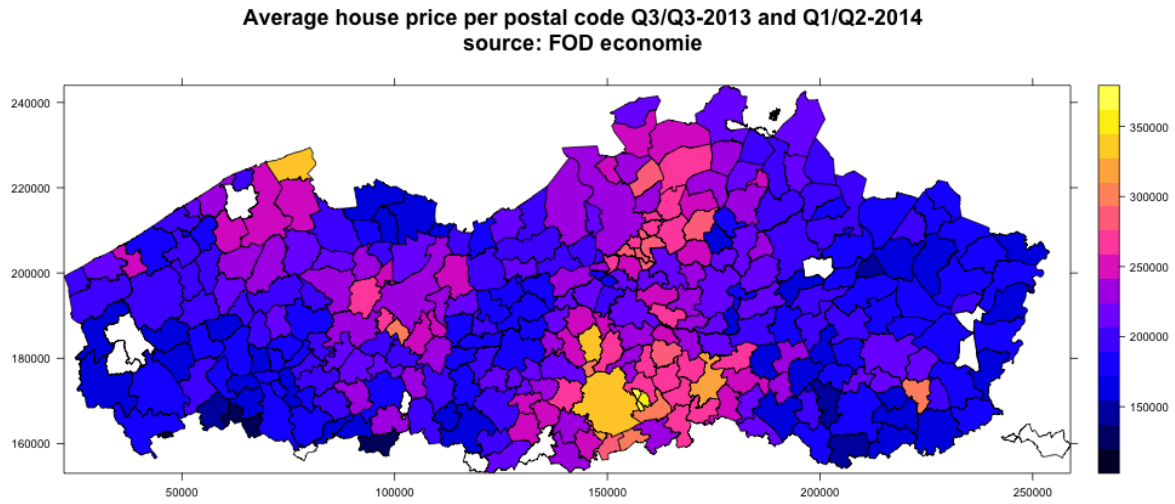


Figure 6: Average property transaction price per postal code, published by the FOD economie.

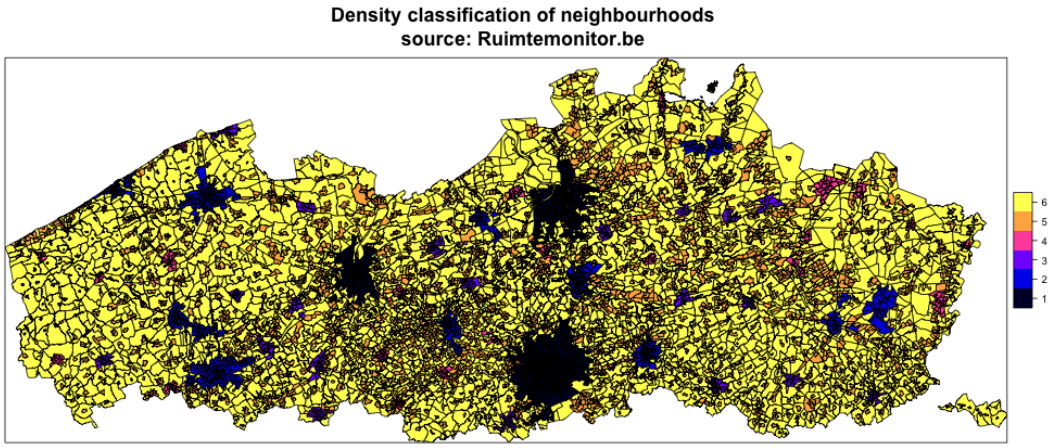


Figure 7: Every neighbourhood of Flanders with a population density classification, ranging from 1 being a central city to 6 being the country side. The neighbourhoods are more granular than the postal codes.

3.7 Remaining opportunities

Every day different open source datasets are made available on the internet. It is beyond the scope of a thesis in statistics to gather and use all these data sources. The following list of datasets could be used as well be and merged with the listings database to further expand the feature list:

- We have no feature explaining the height of a house. We could try to use the Agiv Lidar height scans to estimate this latent variable and include it in our model.
- The FOD economie is expected to publish data, such as unemployment rate, average family size of the "census 2011" on the granularity of neighbourhoods.
- The AGIV produces noise pollution maps and lists areas that might be subject to flooding.
- The GRB land parcel shapefiles do not have an address, so it is very difficult to calculate the correct land area of an address. This could be solved by combining the national cadastral data, which has a link between land and address.
- The calculations on the GRB can be expanded to include the calculations of extra features, such as "dead end street" or "rijwoning". This looks however very challenging as many decision rules have to be verified.
- The AGIV has features available about all the streets in Flanders, such as number of driving lanes and street width. These features could also be coupled to a house, as it is very likely that a house next to a calm round is more expensive than a house next to an national road with 4 driving lanes.
- We will certainly work with distances between the observations. Instead of using the Euclidean distance, it might be usefull to use travel distances between observations. With the Google maps API we could gather such travel distances. We could also work with "Manhattan distances", assuming all streets are perpendicular to each other.

4 Data exploration

In this section we will discus the exploratory data alysis. As discussed in the previous section, out of the hundred thousands of collected online house listings, a final set of listings is selected that meet the following criteria:

- The property must be a house for sale.
- The address must be known.
- The link with the cadastral field has to be found.
- The linked building of the cadastre must contain only one address.
- The price, EPC and number of bedrooms must be known out of the listing and within a reasonable range (i.e. price not equal to 1, or epc equal to 50.000).
- The listing must contain a description.

Around 15000 listings met this criteria. This is not an exact number, as the data collection algorithms are continuously running.

4.1 Feature overview

The features and their correlations are illustrated with a pairs plot (figure 8), and described in the following list with their definitions:

- **log price**: the logarithm of the ask price (out of listing).
- **log l perc opp**: the logarithm of the parcel area (out of the listing).
- **log geb opp**: the logarithm of the building surface area (out of cadastral database).
- **log bedrooms**: the logarithm of the number of bedrooms (out of the listing).
- **epc**: the value on the "energie prestatie certificaat", a measure of the energy efficient of the house (out of the listing).
- **t s**: a factor variable (0, 1 or 2) describing the number of touching sides. Zero indicates that the house is not attached to another house, one is semi-attached and two indicates that the houses is attached to another houses at both sides (calculated with cadastral database).
- **orientatie**: a number from 0 to 360 indicating the orientation of the house compared to the street. On figure 4, house one and two have an orientation around 285 degrees, house three has an orientation around 245 degrees, and house four around 100 degrees. (calculated with cadastral database).
- **log facade**: the logarithm of the line of the shape of the house projected on the street. (calculated with cadastral database).

- **log d to street**: the logarithm of the distance of the shape of the house to the middle of the street (own calculation on cadastral database).
- **month dummy**: a factor variable indicating the month and year that the listing was first gathered from an online portal site (out of listing).
- **log speed of sale**: the number of days the listing was online
- **classif**: a government made classification from one to six of every neighbourhood in Flanders, ref figure 7, one being a city, six being country side (open government data).
- **garage**: is a garage included in the house of the listing, found with algorithm described in section 3.3 (out of listing).
- **verkoop**: is the listing going to be sold on an auction (out of listing).
- **renoveren**: factor variable indicating if the house needs serious renovation works (out of listing).

The locations of the listings are shown (figure 9). Although it is clear that the listings are clustered, we see that they are distributed all over Flanders.

4.2 The listed property price

The aim of this thesis is to model the transaction price. Unfortunately, we do not know the true transaction price, we only know the listing price. We will later discuss if the listing price learns us something about the real transaction price with the governmental data, but in this section we discuss the distribution of the listing price.

Website users or real estate agents create the listings, upload photos, write a description and add a price to the listing. Just as the other features the person fills in online, it might very well be that the listing price is erroneous. These erroneous listing prices are noise and have to be removed as good as possible.

There is however an overlap between erroneous listings and outliers. A house for sale with an asking price of 1 euro is undoubtedly erroneous. But a listing with an asking price of 50k euro can either erroneous, or either a genuine cheap house for sale. We verified however manually listings below 100k euro, and found that most of these listings can not be considered as houses. Some of the listings were houses to break apart, others holiday chalets on camping sites, others were starting auction prices to draw attention. These listings, 1.6% of the total, were not considered in the rest of the analysis.

The same logic applies for a listing with an extremely high asking price. It might be either a foolish asking price not reflecting the true transaction price, either an exceptional property (i.e. a castle) sold with a transaction price close to the asking price. Due to the heavily

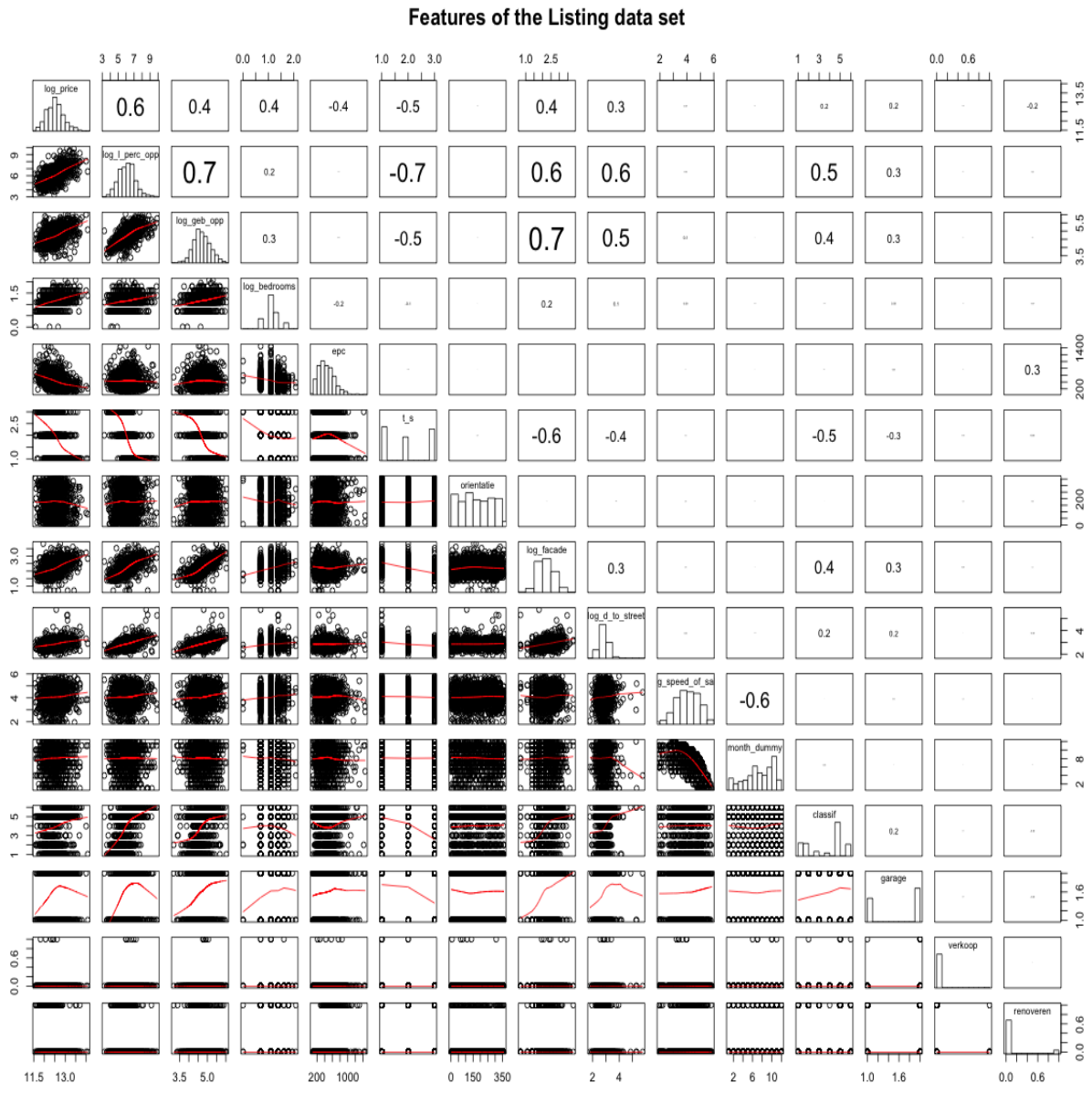


Figure 8: Pairs plot of the features. On the diagonal the Pearson correlation coefficient is given. Most of the variables are log transformed to emphasize the relationship.

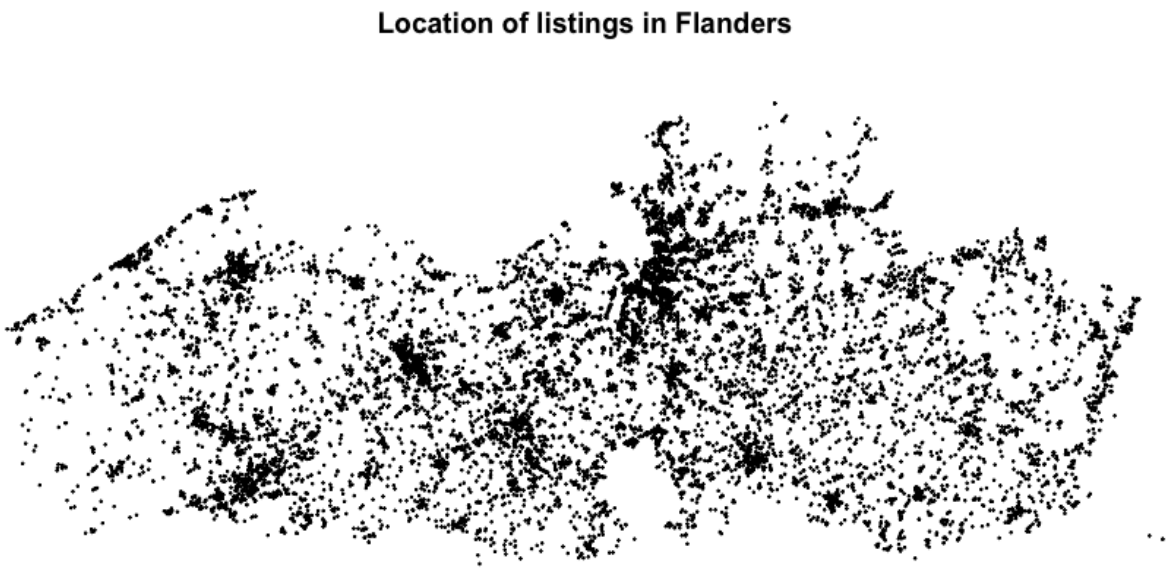


Figure 9: The x and y coordinate of each listing plotted. The listings are spread all over Flanders, but clustered in cities such as Ghent.

skewed distribution of the listing prices (figure 12), we need to remove as good as possible outliers. In section 6.2, we explain why we will model a log-log model, but some listings have such a high price that even in a log-log model, their impact is still large.

We can however not just disregard listings with an asking price above a certain threshold, as some neighbourhoods (such as Knokke and Brasschaat) have a very different price distribution than the rest of Flanders. It is indeed very likely that expensive houses are clustered around each other, and although no features can explain the high asking price, the surrounding houses do. This spatial correlation is often solved by including the postal code. Outliers are then disregarded by selecting the highest priced houses per postal code. But as the postal code is a discrete separation method, it is not used. We used the following method to select outliers:

- Plot all the listings above a certain price
- Mark (with the R command "identify") the outlying cases, being the listings with an extreme price that are not clustered.
- Repeat this procedure with a higher price.

The intermediate steps of this procedure can be seen (figure 11). By using this procedure, clustered listings with a high asking price are not selected as outliers. Overall, only 59 listings or 0.5% of the total listings are left out for further analysis due to their extremely high price. The distribution of the listings, after the univariate outlying procedure, is plot (figure 12).

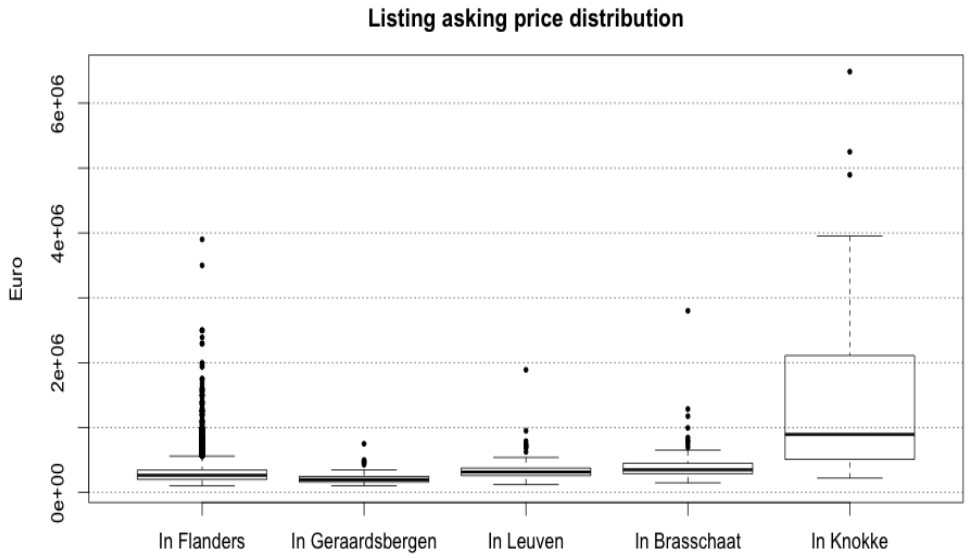


Figure 10: Price distribution of the houses in Flanders compared to the distribution in smaller cities in Flanders. As one can see, the distribution is not homogeneous over space.

In a later stage of the thesis, we decided to leave all the listings of Knokke out of the dataset. We did so because we will be fitting some global models. The distribution of prices in the postal code 8300 of Knokke is just too different compared to the other postal codes, even compared to other expensive neighbourhoods such as Brasschaat and Sint-Martens-Latem (figure 10). Knokke is also too disturbant on the later produced maps of Flanders, even on a log-scale.

4.3 Correlation between the listing prices and the governmental data

The belgian government knows the true transaction prices of all the sold real estate properties in Belgium, and publishes average transaction prices per postal code. We can use these figures to verify if the average listing price reflects the true transaction price. Unfortunately, the government publishes two tables, an average transaction price of normal houses and an average of the villas. The difference between the two is subjective.

To verify if the listing prices of our dataset reflects the true transaction price, we regressed the mean price of our listings per postal code against the transaction mean price given by the government per postal code. To compensate for the categorization the government makes, we first used the average price of only the houses (the left figure in 13), and secondly the weighted averaged price of houses and villas (the right figure in 13).

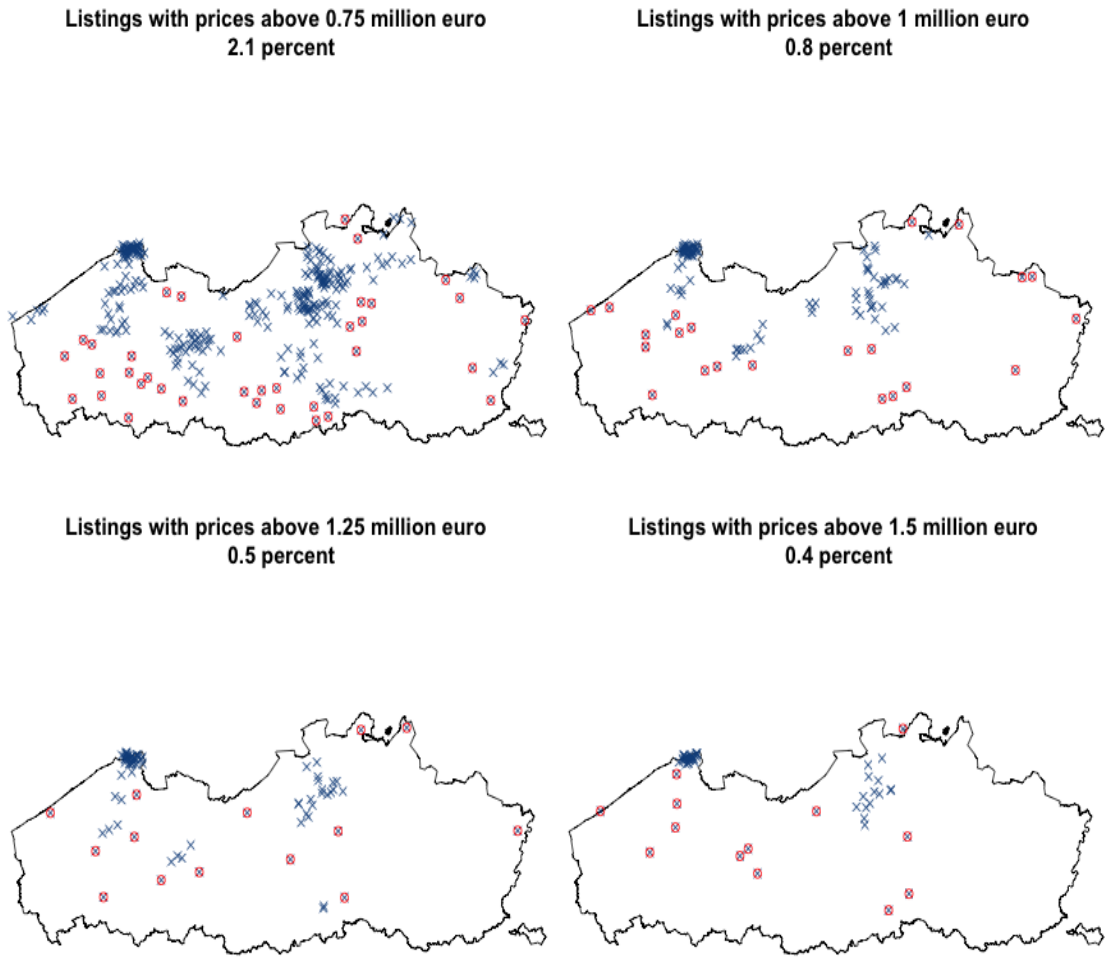


Figure 11: Procedure of selecting listings with an extremely high asking price. The blue crosses are the listings above a certain treshold, the ones with a red circle around them are the selected listings. Some jitter on the coordinates is applied to separate listings very close to each other.

The regression results are listed (table 4.3). As expected, there is a strong correlation between the average transaction price and the average listing price, reflected by the high R^2 of both models. The average house prices and average listings prices are almost directly proportional, with a slope close to one. The intercept indicates that listing price is on average 60.000 euro higher as the transaction price, but this is probably due to ignoring the villas.

When we take into account the villas, the slope is less than one, which means that not every postal code has on average the same difference between the average transaction price listed by the FOD and the average listing price. We still notice however that in most of the postal codes, the average listing price is higher than the average price listed by the FOD economie.

Data used	Intercept	Slope	R^2
Only houses	60080	1.031	0.545
Weighted average houses and villas	83770	0.804	0.660

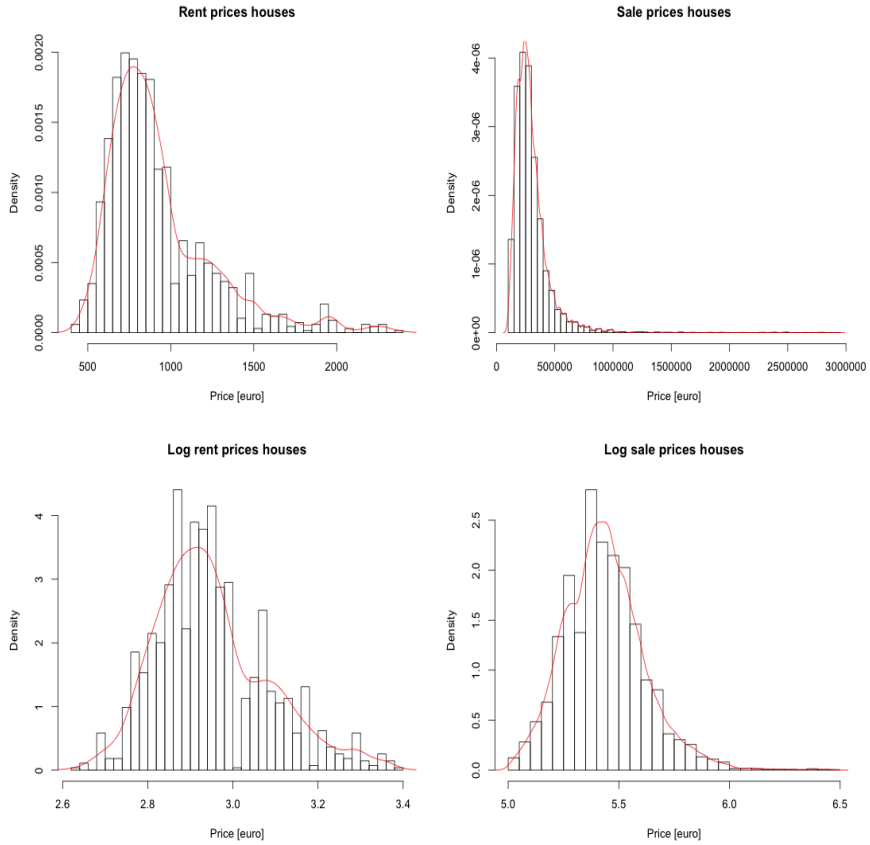


Figure 12: Asking price distributions of listings for sale and listings for rent.

4.4 Ground floor building and land area

Some of the features, such as ground floor building area and land area, have a heavy positively skewed distribution. By log-transforming these feature their distribution comes closer to a normal distribution (figure 14). This has a couple of advantages, which are explained in more detail in 6.2.

4.5 Start and end date of a listing

The listings are gathered from the large belgian immo portal websites, such as Immoweb and Immovlan. This is done by an algorithm which downloads every day the listings. The de-duplication algorithm checks if the downloaded listing was already in the database, and if the price and features match (figure 2). The first time a listing is found, the date is stored as "discovery date". Based on this "first discovery" date, the variable "month dummy" is made.

The algorithm downloading the listings notices also when a listing from the database is not online anymore. We assume that the listing is deleted from the portal website that day, and

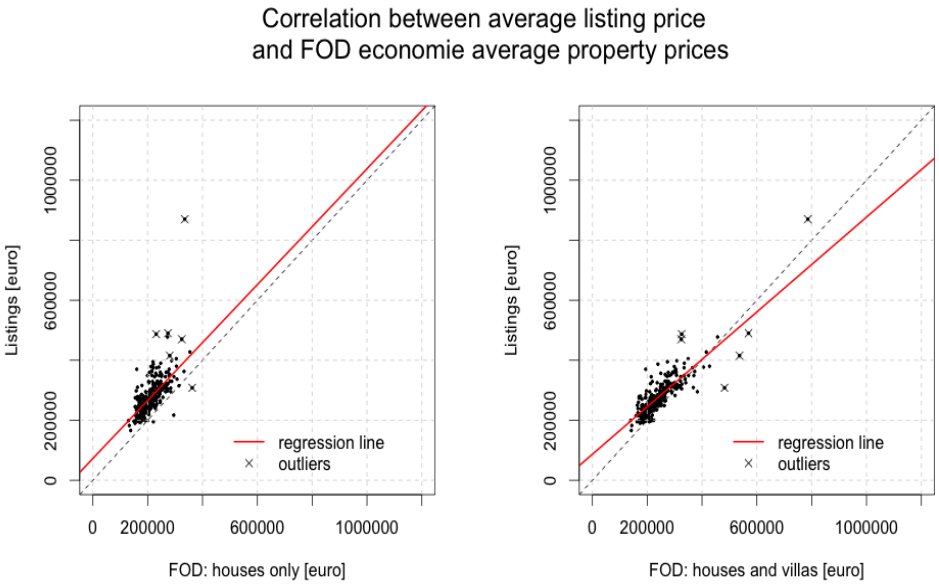


Figure 13: Is there a correlation between the average transaction price per postal code published by the government and the average listing price per postal code of our dataset?

consider the property sold. This "offline date" is added to the listing. The variable "speed of sale" is the difference between the "first discovery" and the "offline date".

The different dates and speed of sale are tabled (figure 15). As one can see, the number of houses sold per months fluctuates a lot, some months more listings are discovered as other months. This is due to modifications on the data gathering algorithms. The algorithms are maintained by programmers, and sometimes new portal websites are added (for example in May and September 2014), explaining the different peaks. On the second barplot of the "offline dates" of the listings, we notice the large amount of listings "offline" in October 2014, these are all the listings that were still online when the thesis project started. The histogram of "speed of sale" shows also a large variation, some listings are more than 100 days online, which makes it doubtful that they will ever be sold.

These uncertainties about the timing of the transaction of the house indicates that it will be very hard to estimate a time effect.

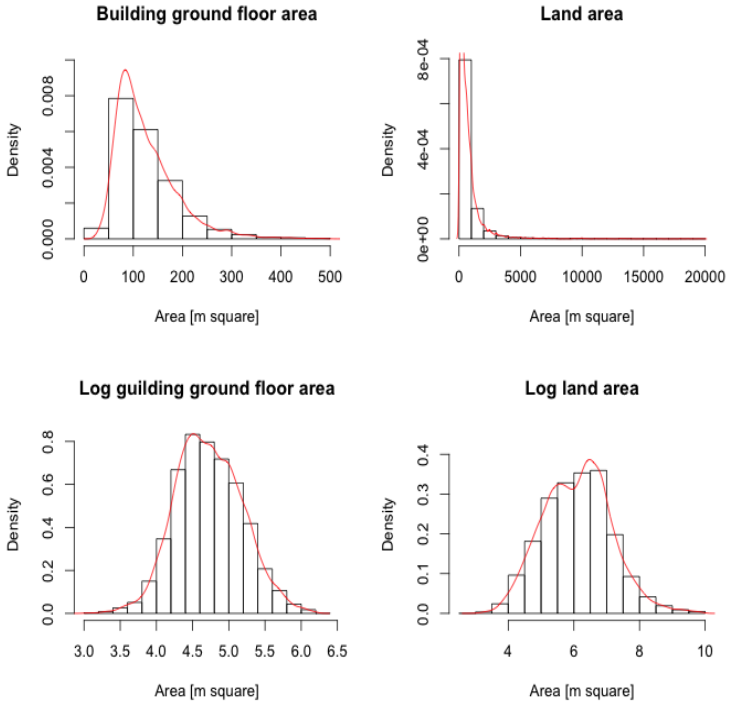


Figure 14: Distributions of features and their log transformations.

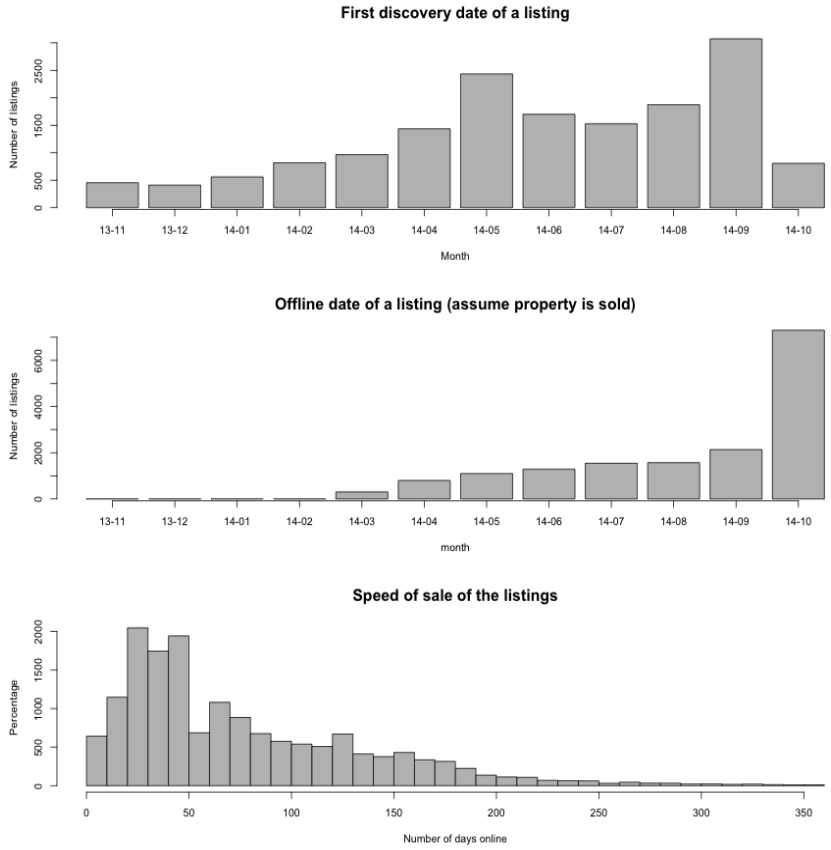


Figure 15: Barplot of the "discovery date" and "offline date" of a listing. The "offline date" is assumed to be the date the property is sold. The difference between the two dates is the speed of sale. Notice the large number of listings "offline" in October 2014, these are the listings which were still online when the thesis project started.

4.6 Spatial correlation of the listings

Everything is related to everything else, but near things are more related than distant things. Tobler's first law of geography

With a pairs plot (figure 8) we visualize the correlation between the price of a listing and its features. It is however not straightforward to visualize the correlation between the price of a listing and its location, or the correlation between the price of a listing and the surrounding listings. We do so with a variogram plot (figure 16). The theory of a variogram is explained in section 6.6.3, but in short, it visualizes how the correlation between observations decreases as their spatial separation increases.

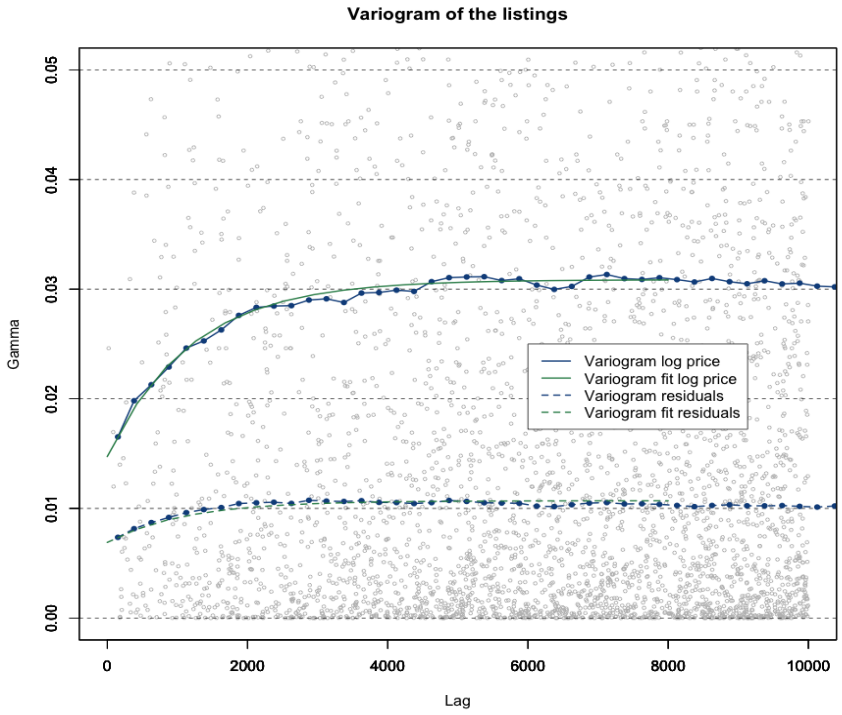


Figure 16: Variogram γ of the log prices of the listings and of the residuals of a first GAM model in function of the separation distance.

On the two variograms, we notice the following:

- There is a significant spatial correlation both in the original listing prices and in the residuals of a first GAM model. We could check this significance with a Monte Carlo simulation where the locations of the listings are shuffeld. The variograms of these simulations will be on a straight line, and not decrease when the lag becomes zero.
- A large nugget effect, which indicates that even if a measurement is made twice on the same location, we still expect an outcome with large variance
- The variance of the unmoddeled listings is greater than the variance of the residuals. The larger this variance reduction, the better the first model

5 Simulation to illustrate different methods

With a simulation, we try to mimic a simplified version of the true dataset. We generate data based on the exploratory data-analysis and on the assumptions we will make throughout the thesis. We will try all of our models on this simplified dataset to verify if the models are able to capture the true data generating process. The following data generating process is used to simulate N houses:

- Points are generated in an X, Y plane of 15 units on 15 units
- Two "city centres" are defined at (6,10) and (10, 6). The location of the houses is multivariate random with the two centres as mean values and $\sigma_1 = 3$ and $\sigma_2 = 1.5$, (figure 17). The location of the house has an effect on the price of the house according to the distance functions in the right graphs (figure 17).

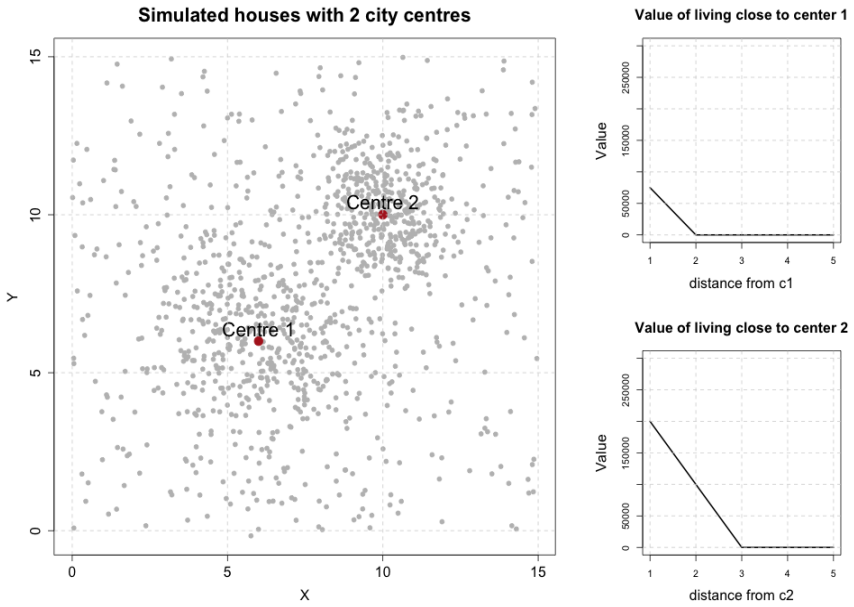


Figure 17: Location and the value of the location of the simulated houses.

- Each house has two extra features:
  - a discrete feature "number of bedrooms", ranging from 1 to 7, distributed with long right tail (figure 18). The value of an extra bedroom is non-linear according to the function.
  - 20% of the houses contain a garage. The effect of garage is not equal over the entire X, Y plane. Within a distance of 3 units from city centre 1, the effect of garage is +50.000 euros, outside that region their is no effect of garage on house prices.
- The total value of the house, defined by adding the location value, the value of the bedrooms and garage, is multiplied with a log-normally distributed noise term of mean 0 and $\sigma_\epsilon = 0.2$.

The generated data is again plot (figure 19), every house is coloured according its price.

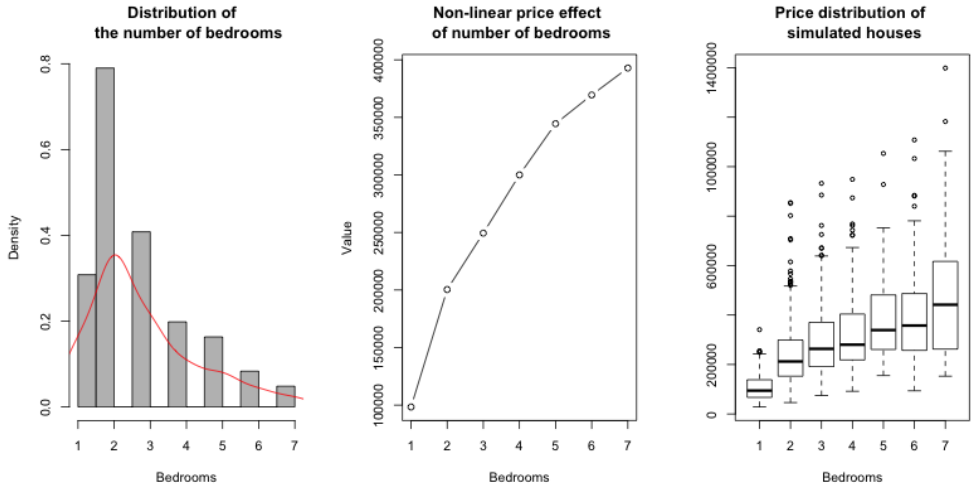


Figure 18: The distribution and value of the feature bedrooms.

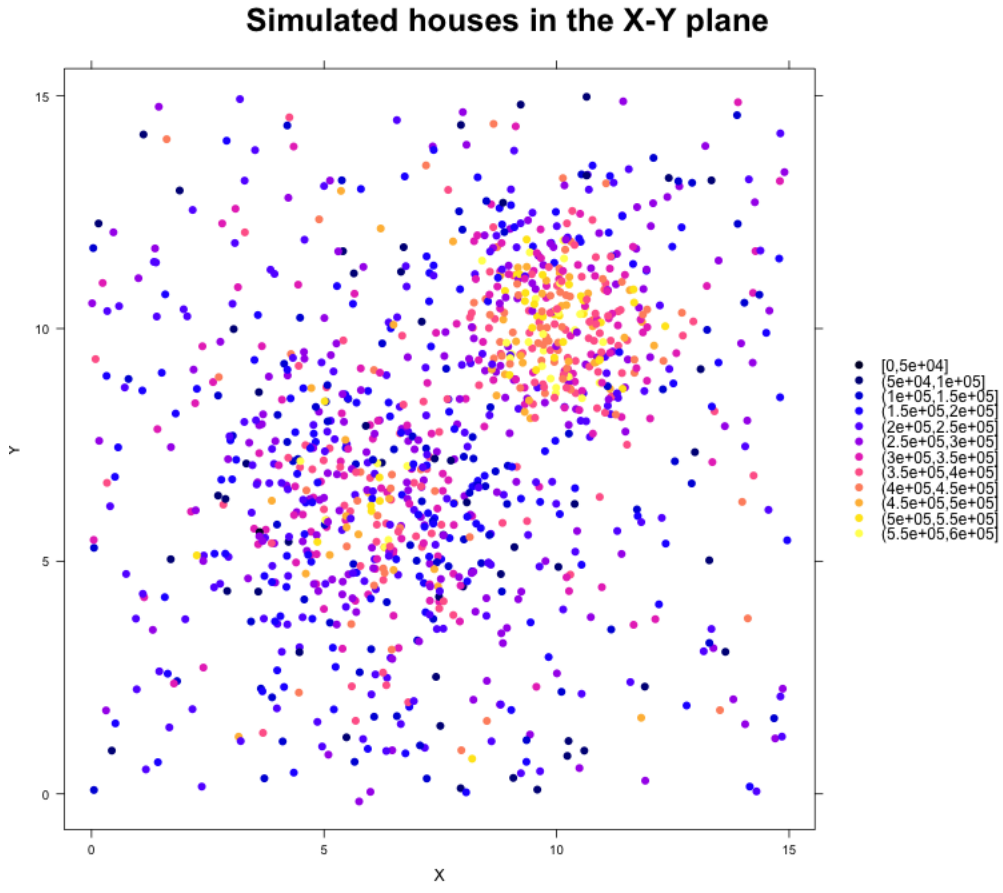


Figure 19: Plot of thesimulated houses and their price

6 Statistical methodology

In this section the statistical methodology to specify, select and fit the model is explained. Throughout the chapter, the models will be tested on the simulated data.

6.1 Introduction: hedonic regression

Modelling a house price based on its features is often one of the main examples in text books about statistics or econometrics. In these basic examples, a regression model is build based on the assumption that a house can be considered as a composite good, and hence decomposed into features. Each of these features contribute to the house price (also known as hedonic regression). The book *A guide to modern econometrics* Verbeek (2004), offers such an example.

If we assume a linear relation between the features calculated in section ... and the house price, we can indeed use the classic linear regression model,

$$Y_i = X_{i,j}\beta_j + \epsilon_i$$

with Y_i the price of the house i, $X_{i,j}$ the feature j of house i, β_j the effect of feature j on the house price and ϵ_i an independently and identically normal distributed term. The model is fit by minimizing the error terms squared, hence the name ordinary least squares (OLS).

In the examples of econometric text books, such as in the book of Verbeek (2004), the focus is more on correctly explaining the house based on its features. In examples in machine learning or data mining books, such as in the book of James et al. (2013), the focus is not on feature effects, but on prediction. It is clear that the aim of this thesis is to build a model only meant for prediction, and not on correctly specifying and calculating the feature effects. Therefore, preference will be given to the model that has the best prediction power rather than the most appropriate feature effect calculation.

6.2 Model specification

Rather than modelling the price of the houses, we model the logarithm of the house prices. Some continuous features of the house, such as the land area, are also log transformed. The

resulting model is specified as follows:

$$log(Y_i) = log(X_{i,1...p})\beta_{1...p} + X_{i,p...j}\beta_{p...j} + \epsilon_i$$

with $X_{i,1...p}$ the continuous features that are log transformed such as land area, and $X_{i,p...j}$ the features that are not transformed such as the dummy variable "garage". The following reasoning explains why we fit a log-log model:

- As demonstrated in chapter 4, the prices of the houses are not normally distributed. Unfortunately, no particular feature set is found explaining this log-normal distribution, meaning that the residuals stay log-normally distributed, regardless of the features used.

- If we would ignore the log normal distribution of the house prices and fit a linear model, we would give equal weight to an error of for example 50 keuro for all houses. Such an error is however not very large when the house is sold for 750 keuro, but disastrous if the house is sold only for 200 keuro. The distribution of the error is not the same for each house price (heteroscedasticity). By log transforming the price, we are modelling relative changes, and the same error of 50 keuro is only an error of 6% in the first case, and 25% in the second case. These relative error terms solve for a great part the problem of heteroscedasticity.

- A mathematical way of expressing the impact or influence of an observation on the model is given by the Cooks distance:

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \text{ MSE}},$$

  with $\hat{Y}_j$ the prediction from the full regression model for observation j, $\hat{Y}_{j(i)}$ the prediction for observation j from a refitted regression model in which observation i has been omitted, p is the number of fitted parameters in the model and MSE is the mean square error of the regression model. By log-transforming the data, the differences in Cooks distance decreases, and hence houses with a high price will not as fast be considered as outliers as in linear models.

- Not only is the price of a house log normally distributed, many features of the houses are also positively skewed. Without log-transforming these features, their impact on the model coefficients would be too drastic. This can be verified with the hat or leverage matrix,

$$H = X(X^t X^{-1})X^t.$$

  By log-transforming the log-normally distributed features, houses with a high hat value in feature space will not as fast be considered as outlying compared to a linear model.

We have to be cautious when we are modelling a log model to predict the price on the original scale. A prediction of the log-house is a prediction of the expected value of the log-price of the house given the features, or in a sample the mean value given the features. When we back transform a mean logarithmic value, we do not get the mean value on the original scale but we get the median value. And hence we are predicting the median price of the house. We can correctly back transform the mean value to the original scale by taking into account the standard deviation.

6.3 Model selection and expected prediction error

The aim of the real estate valuation engine is to predict as good as possible the transaction price of a new independent house for sale, and hence we have to select the model with the highest prediction power. Besides choosing the most appropriate model, we also have to make an estimate of the expected prediction power of this model.

We measure the error of a prediction of the price Y of a house, which was not in the training set, based on its features X with a squared loss function:

$$L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2$$

and select a model with the minimum expected test error

$$Err = E[L(Y, \hat{f}(X))]$$

Cross validation is a resampling method which estimates this expected test error, ref (Hastie and Tibshirani, 2005).

This is in contrast with goodness of fit statistics such as R^2, which learn us something about the error on the training data. The more complex a model, the lower the error on the training data and the better the goodness of fit. But it might be that we are overfitting the data. One can correct for this overfitting by taking into account the model complexity, and use a statisitic as R^2_{adj}, or the AIC or BIC information criteria. But as our dataset is rather large, we choose to work with cross validation.

For each model, a five fold cross validation will be performed:

- Shuffle the houses for sale, and split the data into five equal groups
- For each group i:
  - Train the model on the four other groups $j \neq i$
  - Test the model on the group i

– Store the prediction error of each house, and calculate the mean prediction error squared of loop i.

We are thus estimating the expected test error with the observed mean squared error:

$$Err = E[L(Y, \hat{f}(X))] = E[(Y - \hat{f}(X))^2] \approx \frac{\sum (Y_i - \hat{f}(X_i))^2}{N_{CV}}$$

with N_{CV} the number of observations in each fold i. By taking the square root, we get the expected standard deviation of the prediction error, which is in the same unit as the outcome variable, the price of the house.

With a linear model, the standard deviation of the prediction would be expressed in euros. As we are fitting a log-log model however, the standard deviation of the error will be in relative units. A standard deviation of the prediction error of 0.20 can be interpreted as a standard deviation of 20% of the price of the house.

The result of testing three methods (the details of the methods are discussed below) on our simulated data set with cross validation is plot (figure 20).

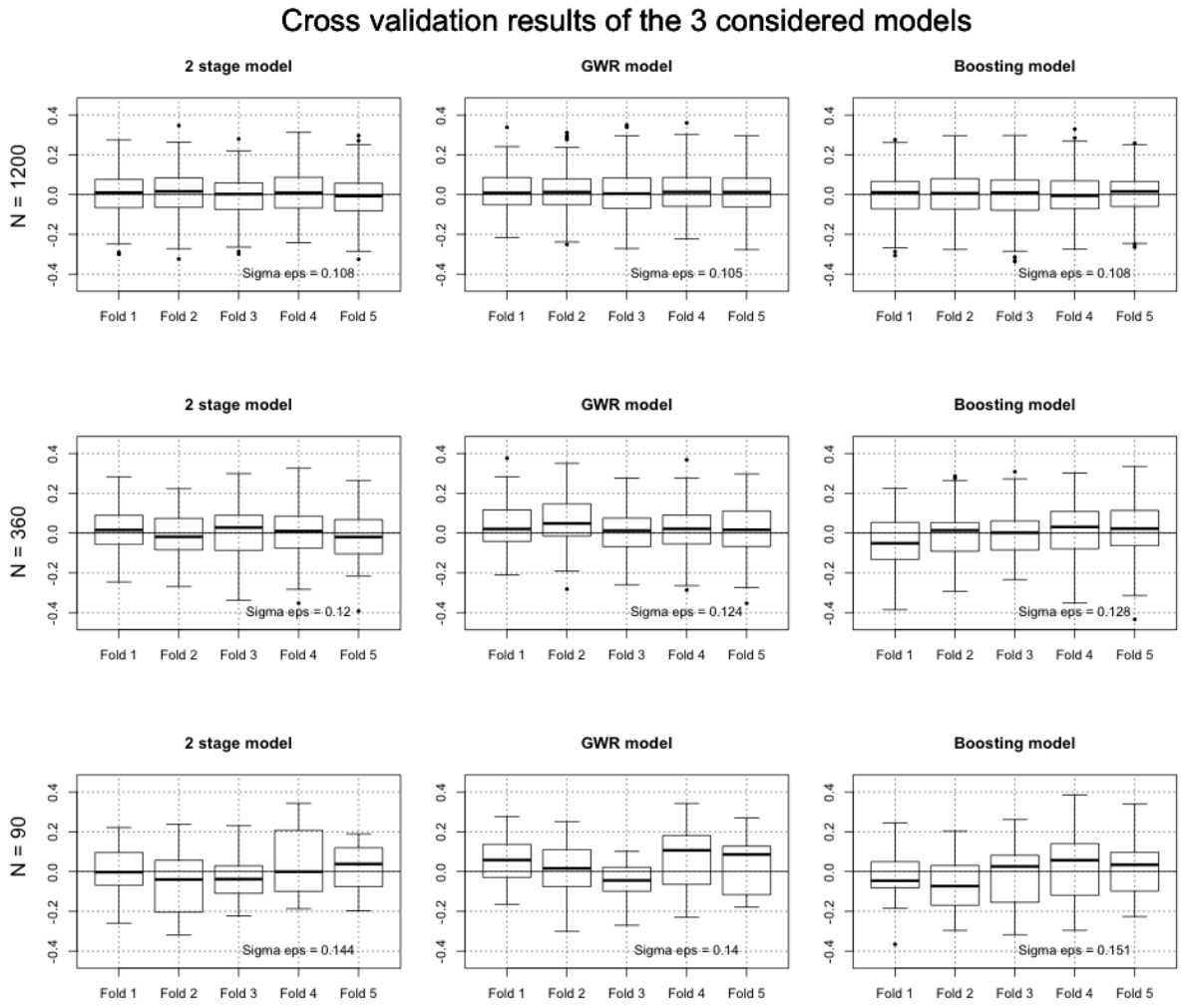


Figure 20: Error distribution of a five fold cross validation on the three models. In each column one finds the prediction errors for each of the three methods (2 step model, GWR model and a boosting model). The three rows display the results for three different sample sizes (N=1200, N=360 and N=90).

6.4 Variable selection and multicollinearity

The question arises which and how many features we should use to model the house price. It makes a lot of sense that a house with a habitable area of 200 square meters will probably be worth more than a house with a habitable area of only 100 square meters, when all the other features are similar. Habitable area is therefore an obvious candidate for a feature. Similarly, we can use the following features:

- Ground area of the house;
- Number of bedrooms;
- Land area size;
- Orientation of the garden;
- Distance to the street;
- Facade width;
- Number of touching sides of the house;
- Energie prestatie certificaat, or EPC, a measure of energy efficiency of the house;
- Construction year;
- And many more.

Not all of these features have a direct causal effect on the house price. Consider for example the EPC value of the house. Based on a linear regression we find that houses with a low EPC value are worth more than similar houses with a high EPC value. We do not know however if the EPC value causes this price effect. The strong effect of EPC value on the price of the house might be due to the correlation of EPC value with the confounding variable "condition of the house", which is unknown to us.

Care must be taken by adding too many features to the model, for two reasons. First, the valuation engine has to predict the price of all houses, also those without a listing and hence with no listing features. It might be that the "number of bathrooms", "sauna", "floor heating" and other details explain the house price, but even though some listings do contain that level of information, most of the listings don't. Let alone houses that are not even listed! The training dataset would also be reduced too significantly if we would try to use all such features, as houses with missing values can not be used with regression models. A trade off has to be made between selecting features to be used in the model, the reduction of the training set and the prediction power of listings without a lot of features.

Secondly, we have to take the multicollinearity between the features into account. Although the multicollinearity makes the effect of features doubtful as its standard errors increase, the goodness of fit of the model will only increase by adding features. The prediction power might decrease however due to overfitting, therefore cross validation goodness of fit statistics

are used rather than the more common R^2.

6.5 Non-linear effects

The assumptions of the linear regression model might be too strong. Firstly, it is doubtful that the effect of continuous features such as habitable area or EPC value is linear. To capture the non-linearity of these effects and still keeping the additive effect of the feature effects, we will use generalized additive models (GAM models). These models are specified as follows:

$$Y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + ... + f_p(x_{ip}) + \epsilon_i \tag{1}$$

With f_1, f_2, ... f_n non-linear functions of the features $x_{i...p}$, and ϵ_i a Gaussian distributed error term. The algorithm minimizes the sum of square errors in (1) in an iterative way (backfitting). During each step only one function f_j is fit. This function f_j is fit on the residuals of a model with all contemporaneous functions $f_{j\neq 1..p}$ of all the features except x_j. The calculated f_j with the other functions $f_{j\neq 1..p}$ is then used in the next step to calculate residuals for the calculation of f_{j+1}.

Any kind of function can be used for each f_j, for example polynomials, local regressions or splines. We choose to use smoothing splines. A smoothing spline is fit by minimizing a penalized least squared, such as in:

$$RSS(f, \lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int (f''(t))^2, \mathrm{d}t$$

with λ a tuning parameter to define the smoothing. It can be shown that the unique minimizer of this penalized least squares is a natural cubic spline (James et al., 2013) with knots at all different values of x_i. This might seem counter-intuitive as the high amount of knots might over fit the data, but the penalization makes sure the spline is not too complex.

Although we build a solid understanding of the GAM model in combination with smoothing splines, we have used an R-package (package GAM) to fit them. It would have taken us too far to implement first the smoothing spline algorithm, and over it the iterative GAM algorithm for more than ten features. Having all these complex packages available is one of the advantages of working with R compared to a GIS system, and one of the valuable tools learned in a master of statistics.

6.6 Spatial autocorrelation

Spatial autocorrelation is neither a magical phenomenon nor a statistical nuisance. It is a specific consequence of the failure to accurately measure or specify variables or relationships. Miron J.

One of the main assumptions of OLS is that the observations, in our case the houses, are independent. This assumption is obviously not met, as houses close to each other are correlated. This is due to a couple of reasons.

First of all, houses near to each other are often similar. A small house in a city without a garden will very likely be surrounded by other small houses without a garden. The size of a garden is however a feature of a house, such as distance to street, touching sides and ground floor area and directly modelled.

Second and more difficult, are neighbourhood effects such as "criminality rate", "proximity to school", "amount of green" and others. The neighbourhood or the location of a house might even be the most important feature of all. Two extreme approaches exist to model the impact of a neighbourhood on the price of the house:

- We try to find features that do explain the impact of the location. Examples are distance functions such as distance to city, distance to railway station or distance to school, neighbourhood features such as unemployment rate-, pollution-, density of the neighbourhood or stratifications per postal code.
- We ignore the neighbourhood effects, and consider them as latent variables. The neighbourhood effects are pushed into the error term, and causes the error terms to be autocorrelated. We can take this autocorrelation into account with several methods, such as two stage least squares.

In this thesis, we opted rather for the second approach. The only neighbourhood feature we used is the average house price per postal code, listed by the government. We choose to spent time and effort in the calculation of the house features of the two million homes in Flanders, rather than calculating neighbourhood features, assuming we would find the neighbourhood effects in the residuals. In the two following sections we further explain our choice.

6.6.1 Modelling spatial autocorrelation with location features

Many studies seem to model the impact of the location by using the location features (Helgers et al., 2013). Although it might intuitively make sense to use deterministic methods to model the impact of the location on the price of the house, the following cases explain why we choose a stochastic approach.

Consider for example the feature "distance to Brussels", the capital city of Belgium. It looks obvious that houses close to Brussels are worth more than similar houses far way from Brussels. The feature "distance to Brussels" is however not able to capture the true anisotropic effect of distance to Brussels, as house prices do not drop the same amount per kilometre from Brussels for each direction. In the north of Brussels are many factories, leading to a lot of employment but also to noisy and polluted neighbourhoods. East of Brussels one finds an airport. South of Brussels are nice neighbourhoods close by the European institutions. If the variable "distance to Brussels" is added to the model, other features such as "distance to airport", "distance to industrial zone", and many more should be added as well. The author of this thesis knows Brussels well and could think about these features, but how would we find the important features of cities unknown to us?

The consequence of falsely specifying a distance function is illustrated with the simulated dataset. On the exploratory plot (figure 19), we located the city centre 1 and included a distance function in a GAM model, but did not know of the existence of city centre 2. The result is a completely erroneous distance function (figure 21). The increase in average house price at 6 units away from city centre 1 has nothing to do with city centre 1, but is due to city centre 2. Although an exaggeration, we faced the same problems when we tried to model the house prices with distance functions.

One other method to capture neighbourhood effects is to use stratifications, such as postal codes. In these kind of models, dummy variables for each postal code are added. Not only can the intercept or average house price vary per postal code, these dummy variables can also interact with the feature effects. We decided also not to use this approach as:

- Postal codes are discrete separation methods, and the postal codes borders are drawn already fifty years ago. It is very unlikely that discontinuities occur exactly at the administrative boundaries.
- The within postal code variation can be very large. The city of Ghent has one postal code, but contains over 100 neighbourhoods, ranging from very exclusive authentic neighbourhoods to industrial houses close to factories.
- Although the dataset is fairly large, some postal codes have only a handful observations, which will lead to overfitting.

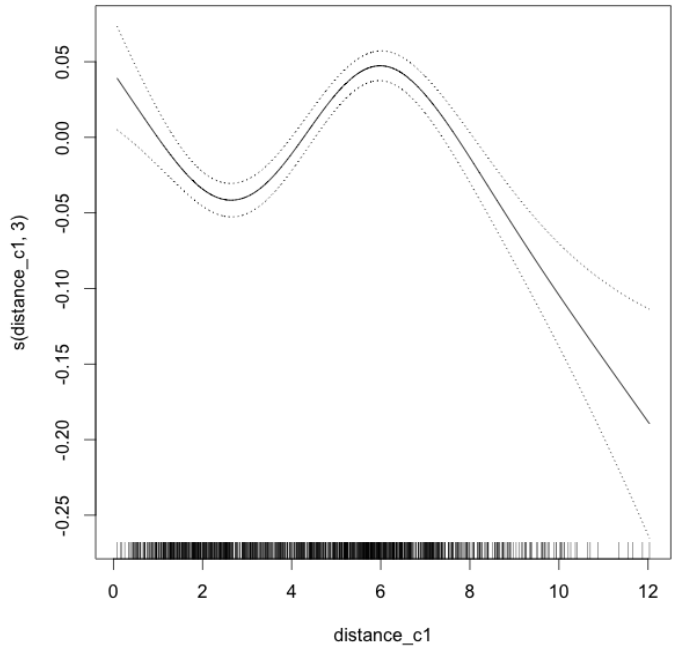


Figure 21: The effect of distance to city centre 1 on the house prices predicted with a GAM model, ignoring the distance to city centre 2. The effect is erroneous, as the sudden increase of the effect on price at 6 distance units from the centre 1 is not due to city centre 1 but to city centre 2.

6.6.2 Modelling spatial correlation with generalized least squares

When we do not add neighbourhood features into our model, we push the locational information into the error term of the model, and hence the error terms will be correlated. One way of dealing with this spatial correlation is with a generalized least squares estimator (GLS) of our non-linear model:

$$Y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + ... + f_p(x_{ip}) + \epsilon_i$$
$$E[\epsilon_i|X] = 0$$
$$Var[\epsilon_i|X] = \Omega$$

With Ω the variance covariance matrix of the residuals. As we do not know the variance covariance matrix of the error terms, we will have to estimate it. One method is a two staged generalized least squares estimator. We first estimate the house prices ignoring spatial autocorrelation with the GAM. In a second stage, we use the residuals to estimate the spatial correlation. An advantage of taking into account the variance-covariance matrix in the second step, is that we derive correct standard errors of our coefficients.

To model the variance-covariance matrix of the residuals Ω we need the covariances between

the residuals on the off diagonal. Just as in a longitudinal or panel setting, one makes assumptions modelling this variance-covariance matrix. In a spatial setting, it is assumed that the covariance between two residuals is dependent on the distance between the two observations:

$$\Omega = \begin{bmatrix} \sigma_\epsilon & f(d_{i,i+1}) & \cdots & f(d_{i,N}) \\ f(d_{i+1,i}) & \sigma_\epsilon & \cdots & f(d_{i+1,N}) \\ \vdots & \vdots & \ddots & \vdots \\ f(d_{N,i}) & f(d_{N,i+1)} & \cdots & \sigma_\epsilon \end{bmatrix}$$

with $f(d_{i+1,i})$ an arbitrary function of the distance between observation i and $i+1$. The covariance between two observations is however seldom modelled in a spatial setting, as the sample covariance calculates fluctuations around a global mean. The mean of the residuals will however be varying locally over space (the average of the residuals of an expensive city will be higher as in an industrial neighbourhood) and hence not spatial stationary. For this reason, which is also pointed out in (Pace, 1998), we will not try to model the variance-covariance matrix Ω and not use two stage least squares.

We also did not find any papers explaining how to include a fitted variance-covariance matrix into a GAM model with smoothing splines and how to make Best Linear Unbiased Predictions with this model, this could be subject to further research.

6.6.3 Kriging the residuals to account for spatial autocorrelation

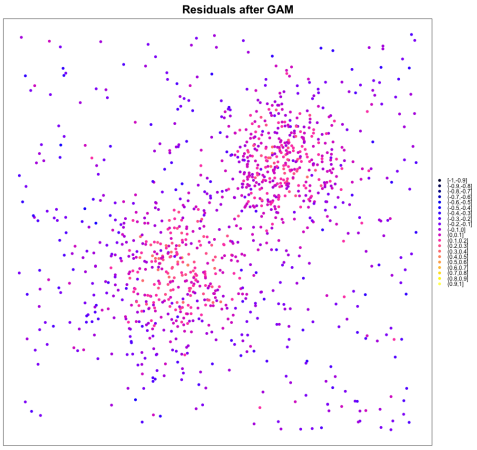
We can however make the assumption of intrinsic stationarity, meaning that the average of the residuals of the first GAM is locally constant, or that the expected value of the difference between two house prices is locally zero:

$$E[\epsilon_i(z) - \epsilon_j(z+h)] = 0$$

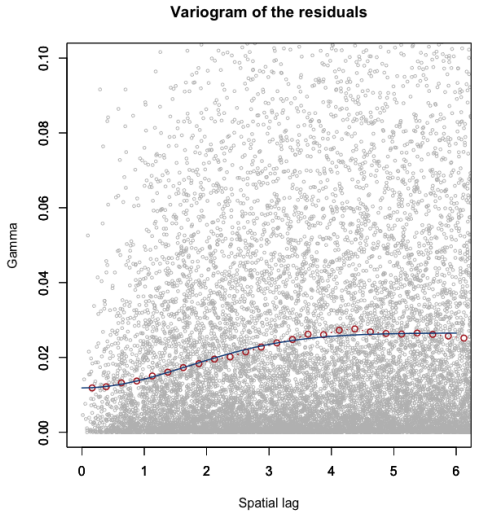
with ϵ_i and ϵ_j two residuals at location z and z plus the separation distance h. We now write their spatial dependence as the semi-variogram γ:

$$\begin{aligned} 2\gamma(z, z+h) &= \mathrm{Var}\left[\epsilon_i(z) - \epsilon_j(z+h)\right] \\ &= E\left[\left(\left(\epsilon_i(z) - \mu(z)\right) - \left(\epsilon_j(z+h) - \mu(z+h)\right)\right)^2\right] \\ &= E\left[\left(\epsilon_i(z) - \epsilon_j(z+h)\right)^2\right] \end{aligned}$$

An example of such a variogram is given (figure 22). The grey dots represent all the pair-wise squared differences against the pair separation distance. We modelled this particular variogram with a Gausian model.



(a) The residuals of a first GAM model ignoring spatial correlation are plot on a map. The residuals are clearly correlated over space. The variogram γ is calculated for each of pair of residuals.



(b) The grey dots represent the variogram γ of each pair of residuals, in function of their spatial lag. This experimental variogram is averaged per separation distance (red line) and modelled (blue line).

Figure 22: Illustration of the variogram fitting process

We can now use an interpolation algorithm that takes into account the modelled variogram of the residuals to produce a map of the expected residual values over space. An interpolation algorithm will simply calculate for each point in space a weighted average of surrounding points, in our case surrounding residuals. Such an interpollation technique is kriging. By imposing that the prediction variance is as small as possible taking into account the modelled variogram, one derives a set of kriging equations to estimate the interpollation weights.

This technique is applied on the simulated dataset. A simple GAM model is fit, with two predictors "number of bedrooms" and "garage". The variogram of the residuals is made (figure 22) and a map is produced based on kriging (figure 26). Although no spatial feature is included in the model, we were still able to capture the effect of a particular location on a house price.Note that due to the misspecification of the GAM model (the effect of "garage" is not equal over the complete X-Y plane), the value of living close to city centre one is somewhat overestimated due to the extra value of the garage.

6.6.4 Using the government average transaction price: simple kriging

When modelling the spatial variogram of the residuals of the first step GAM model, we could go a step further and use the average transaction price published by the government. As there is a significant relation between the average listing price and the average transaction price

published by the government, we could regress the residuals against the average transaction price of the postal code X_{avgtp}:

$$\epsilon_i = \beta_0 + \beta_1 * X_{avgtp} + \mu_i$$

By doing so, we would spatially demean the residuals ϵ_i. The residuals of this model, μ_i, would hence be spatial stationary with $E[\mu_i] = 0$, an assumption we could not take earlier to use GLS. (Instead of modelling the covariance, we had to model the variogram). We can now model the covariance in function of the distance:

$$\sigma(\mu_i, \mu_j) = E[(\mu_i - E[\mu_i])(\mu_j - E[\mu_j])] = E[(\mu_i - 0)(\mu_j - 0)] \approx f(d_{\mu_i,\mu_j})$$

which can be directly implemented in Ω to use GLS.

6.7 Spatial heterogenity of features effects

Another main assumption of the models discussed so far is that the feature effects are constant over space. This assumption might be too strong. It is for example unlikely that the value of an extra square meter of habitable area is equal in dense city areas and on the country side.

One approach to deal with this spatial un-stationarity is to use a semi-parametric regression. Instead of using all observations to fit a global model, we will fit a model for each house, using only neighbouring houses. The number of neighbouring houses that are selected to fit this local model is a tuning parameter, and will be selected with cross validation. This approach known as Geographically Weighted Regression (GWR) is well documented in the book Geographically Weighted Regression by (Brunsdon, 1998).

Feature effects to estimate house i are calculated with a generalized least squares model:

$$\hat{\beta}(u_i, v_i) = (X^T W(u_i, v_i) X)^{-1} X^T W(u_i, v_i) y$$

with $\hat{\beta}(u_i, v_i)$ the feature effects at location u_i, v_i. The weights $W(u_i, v_i)$ of N neighbouring houses around house i are typically calculated by a distance function, such as the Gaussian curve:

$$w_{ij} = exp\left[-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right] \tag{2}$$

with b the bandwidth, being the maximum distance between house i and the selected neighbouring houses. As the number of houses selected is a constant, the kernel area will depend on the density of the area. We can further expand this distance function by using the neighbourhoodcode ($nhcode_i$) and classification of the neighbourhood ($nhclass_i$):

$$w_{ij} = \begin{cases} \text{If } nhcode_i = nhcode_j : & exp\left[-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right] * A \\ \text{Else If } nhclas_i = nhclass_j : & exp\left[-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right] * B \\ \text{Else :} & exp\left[-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right] \end{cases} \tag{3}$$

With A and B tuning parameters. With this weight function, houses in similar neighbourhoods will be given more weight. The prediction of house i is then given by:

$$\hat{y}_i = \hat{\beta}_0(u_i, v_i) + \sum_{k=1}^{K} \hat{\beta}_k(u_i, v_i) x_{ik}$$

The number of houses N selected for each local regression is an important tuning parameter, and can be thought of as a trade-off between bias and variance. If we select only a few observations around house i, we expect the model to predict a value close to the real price of house i. But if we select too few houses, the model will be unstable. Adding one house to only a few N neighbouring houses would indeed change the model significantly, hence a high variance. The MSE of a five fold CV procedure on the simulated data is shown for different N selected houses (figure 23), the minimum of $N = 60$ is the optimum between bias and variance.

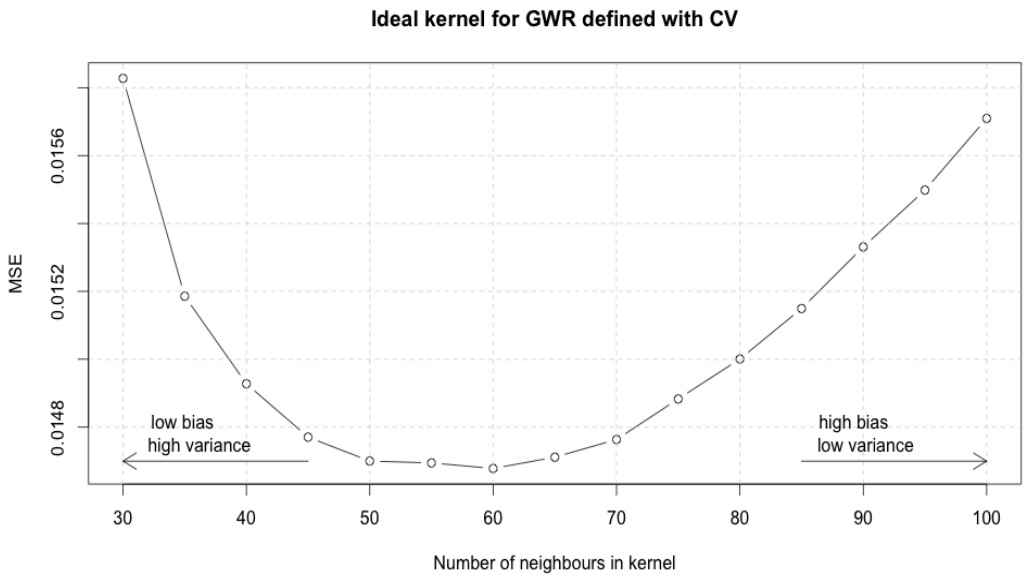


Figure 23: Illustration of the bias-variance trade-off when selection the ideal number of houses for GWR.

We can use GWR again to make a map of Flanders, by predicting at each grid point the price of a house. We can make two kinds of maps:

- A map of the house price conditional on feature parameters. This map is interpreted the same way as the map produced by kriging the residuals, the price of a similar house is shown for all locations on the map.
- A map of the average house price for all locations. The average price is not conditional on the features, but the features are integrated out, meaning that for all locations, the expected value of the features is used to calculate the average house price.

6.8 Time effect

Just as there is a strong correlation between houses that are close to each other, there might be a correlation between houses that are sold in the same time period. We have however seen in the section data exploration (figure 15), that the timing of the transaction is unclear to us.

In both discussed models, we can take into account a correlation over time, they are basically an expansion of the distance function:

- We could add in the variogram calculation the time t as third dimension, and create a spatial-time lag between observations.
- Instead of only taking into account the distance between observations in the weight function of GWR, we can also calculate the weights based on both distance and time difference.

If we would do so, we are dealing however with three dimensions, $X - Y - t$. The extra dimension makes it a lot harder to find similar observations in feature space. This effect is also known as the curse of dimensionality. As we are both not sure of the date of the transaction and we do not have sufficient observations, we just calculate a global feature effect of time by including a dummy variable of each month, month_dummy.

6.9 Boosting

At the end you have to say wauw as these are such powerful ideas I wonder if nature has discovered them to. Is there a good engineering in the brain based on good science? Or given the nature of evolutions is it just random junk that isn't the best way for doing anything, who knows. But today we are going to talk about an idea that I'll bet is in there somewhere, because it is easy to implement, it is extremely powerful in what it does, and it is the essential item in anybody's repertoire of learning mechanisms
Patrick Winston in an MIT open courseware Youtube video about boosting.

Thanks to the exponential increase in computer power, new prediction models emerged based on not one complex model, but thousands of combined simple models. Although these models are very hard to interpret, they are very powerful in making predictions. They are also known as black box models, as the actual prediction process is impossible to interpret. One of these models, called Random Forests, is used to predict the real estate property prices on the American real estate portal website Zillow, the pioneer in showing estimates on real

estate listings.

Boosting is another example of these black box models. It is based on consecutive simple decisions trees. After each decision is made, every data point receives a new weight, based on the error of the previous decision. It is trained by choosing three parameters, the depth of each decision tree d, the number of consecutive decisions B, and the learning rate λ. These parameters will be chosen with cross validation. The algorithm is as follows:

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, ...B$, repeat:
   (a) Fit a tree $\hat{f}_b$ with d splits (d+1 terminal nodes) to the training data (X,r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(x) \texttt{ <- } \hat{f}(x) + \lambda \hat{f}_b(x)$$

   (c) Update the residuals

$$r_i \texttt{ <- } r_i - \lambda \hat{f}_b(x_i)$$

3. Output the boosted model

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}_b(x)$$

The final boosting model is built upon more than 1000 decision trees, making it impossible to understand based on what combination of decisions a prediction is made. We have however two model summaries available, being the relative importance of each predictor and the partial dependence of each predictor:

- In every tree b, d decisions are made. Each decision is based on a predictor. The relative importance of each predictor is the sum of the squared improvements over all internal nodes $B * d$ for which it was chosen as decision maker.
- We can calculate the partial dependence or the marginal effect of each predictor. They represent the effect of a predictor taken into account the other predictors. The partial dependence of feature X_p is given by:

$$f(X_p) = E_{X_{j \neq p}} f(X_p, X_j \neq p)$$

  with X_p the range of the predictor we want to plot, and is estimated with:

$$\overline{f}(X_p) = \frac{1}{N} \sum_{i=1}^{N} f(X_p, x_{i,j \neq p}) \tag{4}$$

  with $x_{i,j}$ the observed values of the other predictors. The difference between the

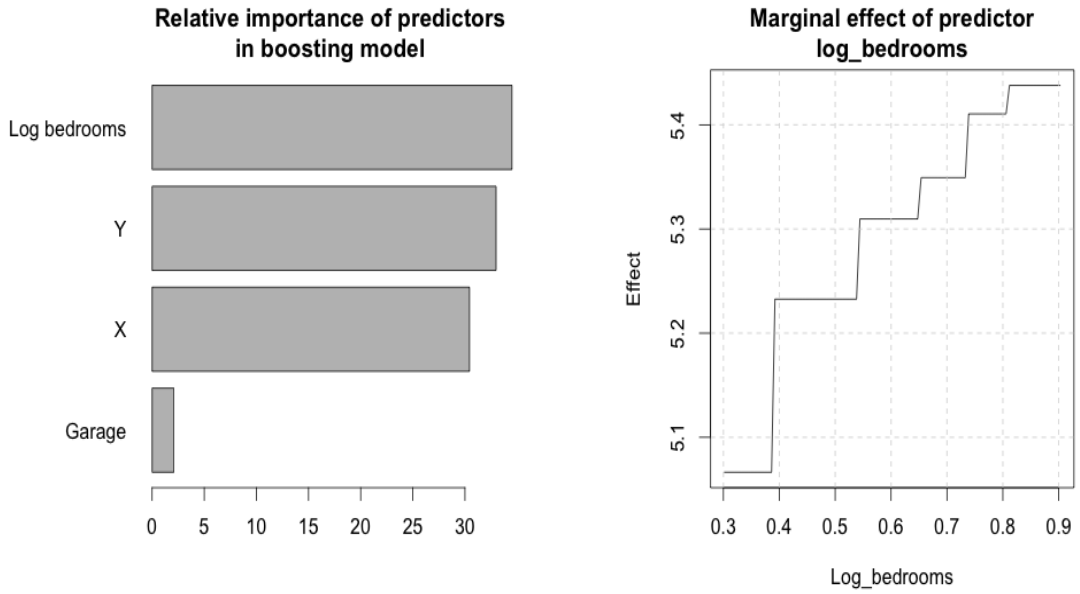


Figure 24: Boosting model summary plots. The left plot indicates the relative importance of the four predictors in our simulated data set, the right plot visualizes the marginal effect of the predictor of log bedrooms.

marginal effect and the conditional expectation is important. Take for example the effect of location $f(x, y)$. If we would not average over all the other feature values, it might be that the effect of the location is not due to the features x and y, but due to the different feature values $X_j \neq p$ at location x and y. In the our case, it might be that houses at location x, y are not more expensive due to the location, but due to that houses at that location x, y often have large garden. By averaging or integrating out the other feature values, we derive a marginal effect. In the case of the average partial effect of location, one can interpret this as "the average extra value one would pay for a house on location x, y".

The two summaries of the boosting model fitted on the simulated data are shown (figure 24). In the left figure, one can see the importance of the predictors, on the right plot the marginal effect of the predictor log bedrooms. Due to the nature of a boosting model, the feature effects are not smooth, they are discontinuous piecewise constant. The same effect will be seen in the marginal effect of location, instead of a smooth map, we will obtain a piecewise constant map over space (figure 28). This demonstrates that although the boosting model has good prediction power, the produced models are clearly not made on statistical assumptions but on a machine learning process.

6.10 The effect of location: which maps can we produce?

If one is interested in the effect of a location on an outcome variable, we typically make a map. We can as well do so for the effect of the location on the house price. We have to be cautious however, as the maps produced by the three different methods have a different meaning:

- In the first model, we kriged the residuals of a global GAM model, and hence we are mapping the conditional expectation of the residual given the location: $E[\epsilon|u, v]$, with u, v the space coordinates. This conditional expectation is independent of the other features of the house, and the kriged map can thus be interpreted as "the average value one pays extra for a given house on location A compared to location B".
- As discussed with formula (4), we can calculate the marginal effect of the features of a boosting model. The marginal effect is calculated by averaging the other features out. The produced map has a similar interpretation as the map produced with kriging, it can be interpreted as "the average one would pay extra for an average house on that location". Note that the calculation of forumula (4) would take computationally a very long time, but due to the nature of the boosting equations, the marginal effect can be calculated from all the B trees , ref (Hastie and Tibshirani, 2005).
- With the geographically weighted regression we could try to calculate something similar as a marginal feature effect. But integrating out all the features at each grid point would be computationally to heavy and we will have to work with another kind of map. The map we will produce is the local weighted average of the house prices on that location. It is a very different map as the other two maps, as we produce a map of the average one pays extra for the average house on that location:

$$E[Y_p|u, v, E[X_{(u,v)}]]$$

These maps are made for each model (figure 26, 28 and 28) for different sample sizes of the simulated data. As one can see, for the three methods, the maps become more and more informative as the sample size increases.

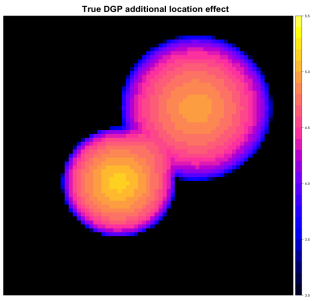
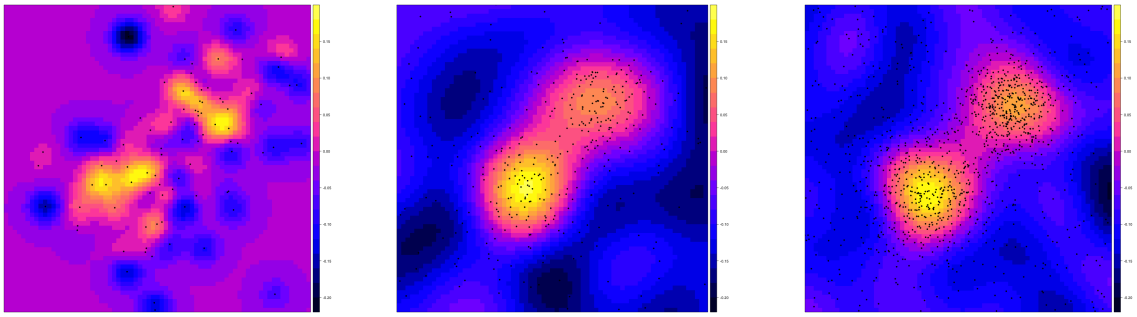
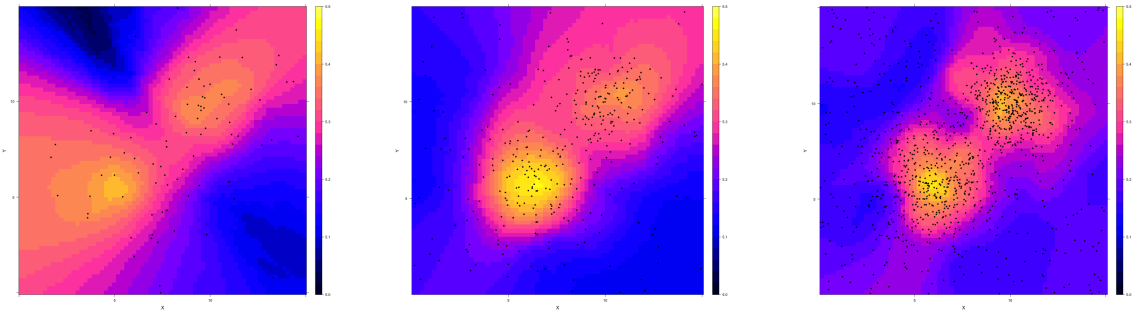


Figure 25: The true data generating procuess of the additional effect of location (the distance functions are specified in figure 17).



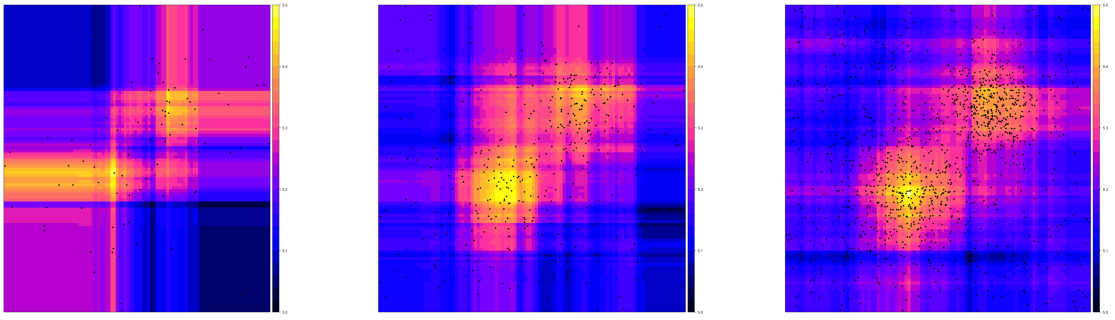
(a) N = 90 (b) N = 360 (c) N = 1200

Figure 26: The additional effect of location on the house price modeled by kriging the residuals of a first GAM model, with different sample sizes N



(a) N = 90 (b) N = 360 (c) N = 1200

Figure 27: The geographically weighted average house price calculated at each grid point for different sample sizes N



(a) N = 90 (b) N = 360 (c) N = 1200

Figure 28: The marginal effect of the feature u_i, v_i calculated by boosting for different sample sizes N.

6.11 Overview of the three considered models

In this section, we have discussed the three different models that we will test on the data to estimate the transaction price. The list below summarizes the three models, with their advantages and disadvantages:

1. A two step model, with step one a generalized additive model and in step two ordinary kriging of the residuals.
   - All observations can be used to model smooth non-linear feature effects
   - Features effects are homogeneous over space
   - An efficient interpolation algorithm is used to calculate the location effect
2. Geograhically weighted regression, with tuned distance functions
   - Smooth non-linear features effects are only based on neighbouring observations, and hence heterogeneous over space
   - Correlation between observations is taking into account by distance
3. Generalized boosting regression modelling.
   - Black box algorithm based on consecutive simple models
   - Feature effects are heterogeneous over space and can have strong interactions
   - Predictive model, the feature effects are less clear

The goodness of fit statistics, being the average of the cross validation σ_ϵ are listed (figure 20) and in the table below. The goodness of fit statistics are similar to each other, only the boosting model seems to be the weakest when the sample decreases.

Model / σ_ϵ	N = 90	N = 360	N = 1200
GAM + Kriging	0.144	0.120	0.108
GWR	0.140	0.124	0.105
Boosting	0.151	0.128	0.108

6.12 Remaining opportunities

- Throughout the thesis, we use cross validation to assess the expected prediction error of a new observation. Cross validation produces however a global prediction error. But there might be variation over feature space of the prediction error. The prediction interval of a house with 2 bedrooms will be for example smaller than the prediction interval of a house with 11 bedrooms, as there were only a few houses with 11 bedrooms in the dataset. This is also true for the location features u_i, v_i. When we are for example interpolating a map with kriging, we get on each point in space a different confidence interval. We could sum the prediction interval of the first step GAM model and second

step kriging interpollation, assuming that these standard errors are independent, to produce prediction errors varying over (feature) space. We could do something similar with GWR, use the goodness of fit statistics of the local models to make a map of the varying prediction intervals over space.

- If we would have more data available and if we would know the date of the transaction, we should also model the correlation over time of houses. And if we would have hundreds of thousands of observations, we could try to model this correlation heterogeneous over space. If this would work, we could see the location effect of neighbourhoods changing over time.
- All the models are based on the assumption of the additive effect of features. As the name says, generalized additive models (GAM) are very good in capturing additive effects, but they are not able to capture interaction effects.
- And extension of GWR would be to include spatial correlation in the weight matrix. As we are fitting a global model, we can assume that the residuals are locally stationary and use the covariance function instead of the variogram to model the spatial correlation. This is done in Wu et al. (2014).
- When we are modelling a map of the house prices in Flanders with kriging, we could make block predictions instead of point predictions.
- When looping over the houses to predict and fitting for each house a geographically weighted GAM model, we could detect outliers and refit a model without them. This might lead to a better goodness of fit of the model. Outliers can be automatically detected with measures such as the Cooks Distance.

7 Results

7.1 Model 1: Global generalized additive model plus kriging

The first fitted model is the two step model. In the first step, we fit a global general additive model (GAM) with smoothing splines. The selected features and their effect on the house price are visualized (figure 29), and the R-output with the feature effects and goodness of fit statistics are tabled (table 39). As the model is global and takes all observations into account, we can use several degrees of freedom to model non-linearities without overfitting.

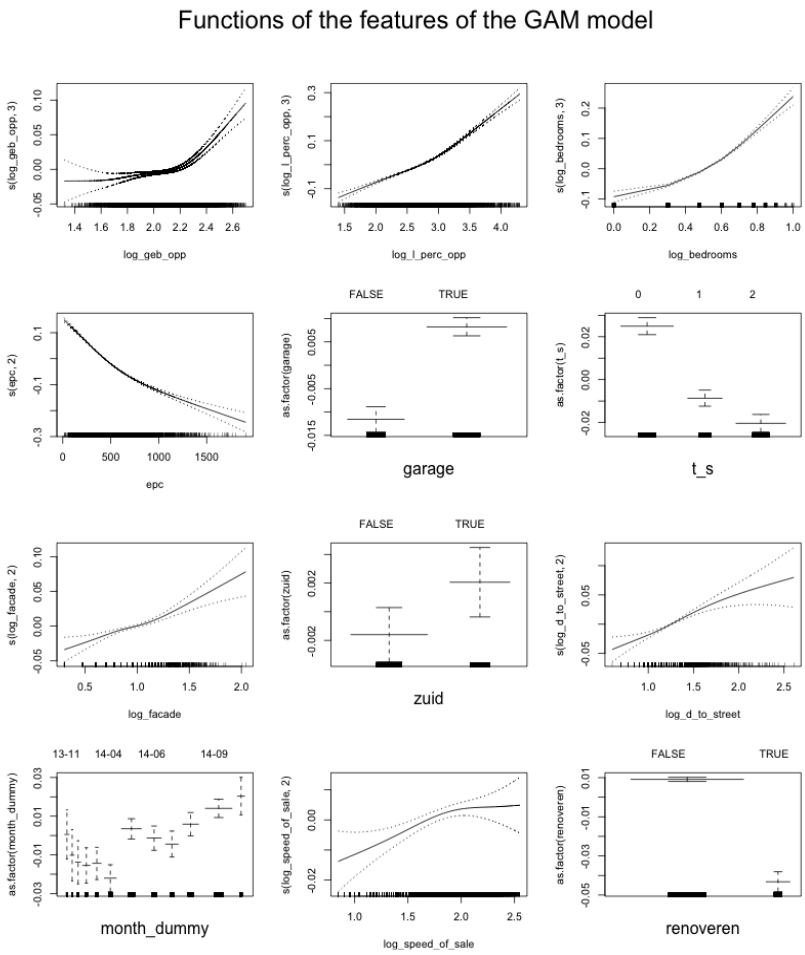


Figure 29: Plots of the relationship between each feature and the effect on the house price in the model with their standard errors.

The feature selection and the smoothing of the GAM model is done with a backward elimination process. First, all the features are added with a smoothing spline with several degrees of freedom. We can use the standard errors and the p-values to verify if the feature effect is significant, and use an anova test in case of doubt. If we, for example, doubt the significance of the spline with three degrees of freedom of the feature log_l_perc_opp, we fit both a model with two and three degrees of freedom of the smoothing spline and compare the goodness of

fit statistic the two nested models:

Degree of freedom spline	Extra dof	Reduction SSE	F-value	P-value
2 spline log_l_perc_opp				
3 spline log_l_perc_opp	1	0.37148	22.824	1.792e-06 ***

The feature functions make intuitively sense. The price of a house for example increases with building area (feature log_geb_opp), and a detached house is worth more than an attached one (feature touching sides t_s). We also see a time effect appearing (feature_month_dummy), but it looks like this time effect is taking into account the addition of another data source in the data gathering process in May 2014 (figure 15). We also notice the positive effect of total time online of the listing (feature log_speed_of_sale), something that was pointed out in the book Freakonomics by (Levitt and Dubner).

The model diagnostics of the first step GAM model are plot (figure 30). Based on these plots, we selected a handful of clearly outlying values. The distribution remains however skewed, we see still some values with an very low prediction compared to their true price. We leave them in the dataset, as we hope to improve these predictions in the second step.

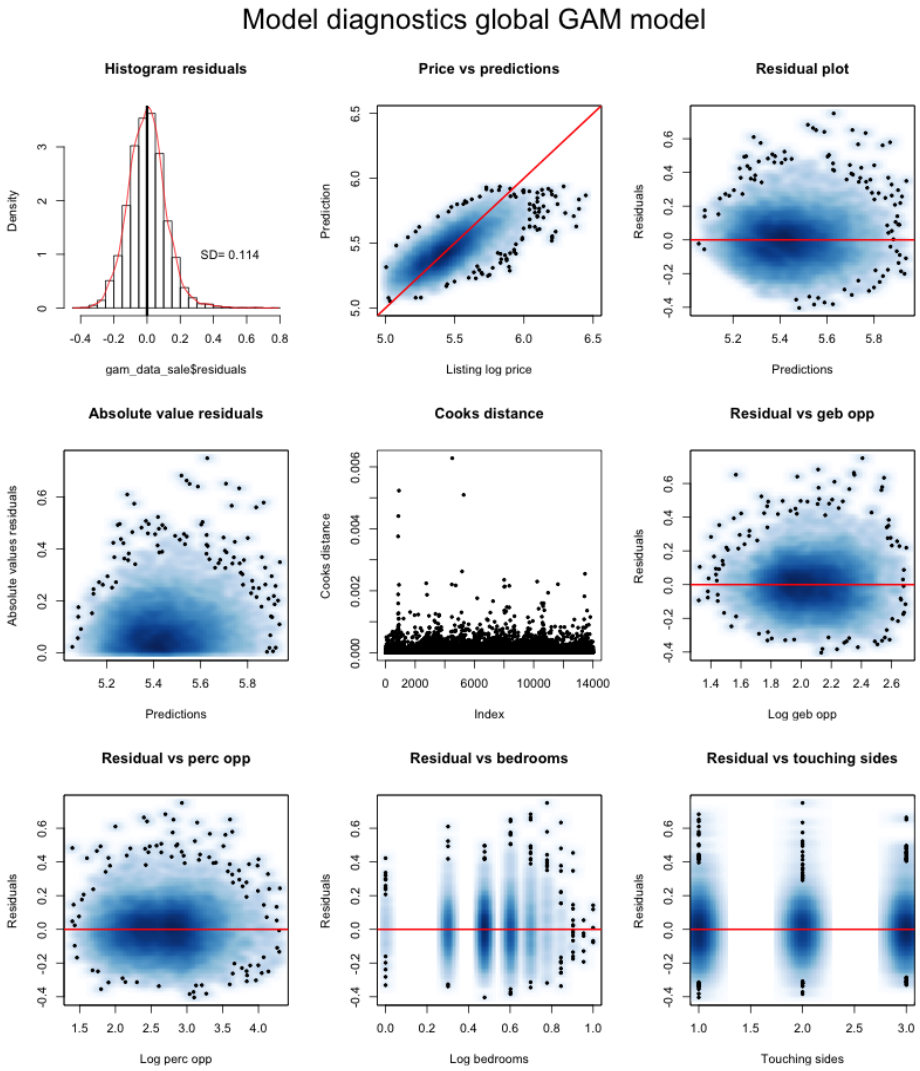


Figure 30: Model diagnostics of the first step GAM model

The second step is to model the variogram of the residuals and use kriging to interpolate them to the grid of Flanders. The modeled variogram of the residuals is already shown (figure 16). The interpolated map of the residuals is shown (figure 36). This is calculated on a grid of 10000 pixels, but it seems the map is not smooth. When we zoom in on a 10000 grid on the postal code of Ghent (figure 31), we however find a smooth map. Of all the three considered models, it is clear that this is the one capturing the most spatial variation.

As mentioned earlier, we have to interpret this map as a marginal effect, it is an extra premium one would pay for an average Flemish house on that location. Without using any neighbourhood features, we were able to capture neighbourhood effects such as the city centre of Ghent and the expensive suburb Sint-Martens-Latem, or the industrial zone around the factory site of Arcelor Mittal.

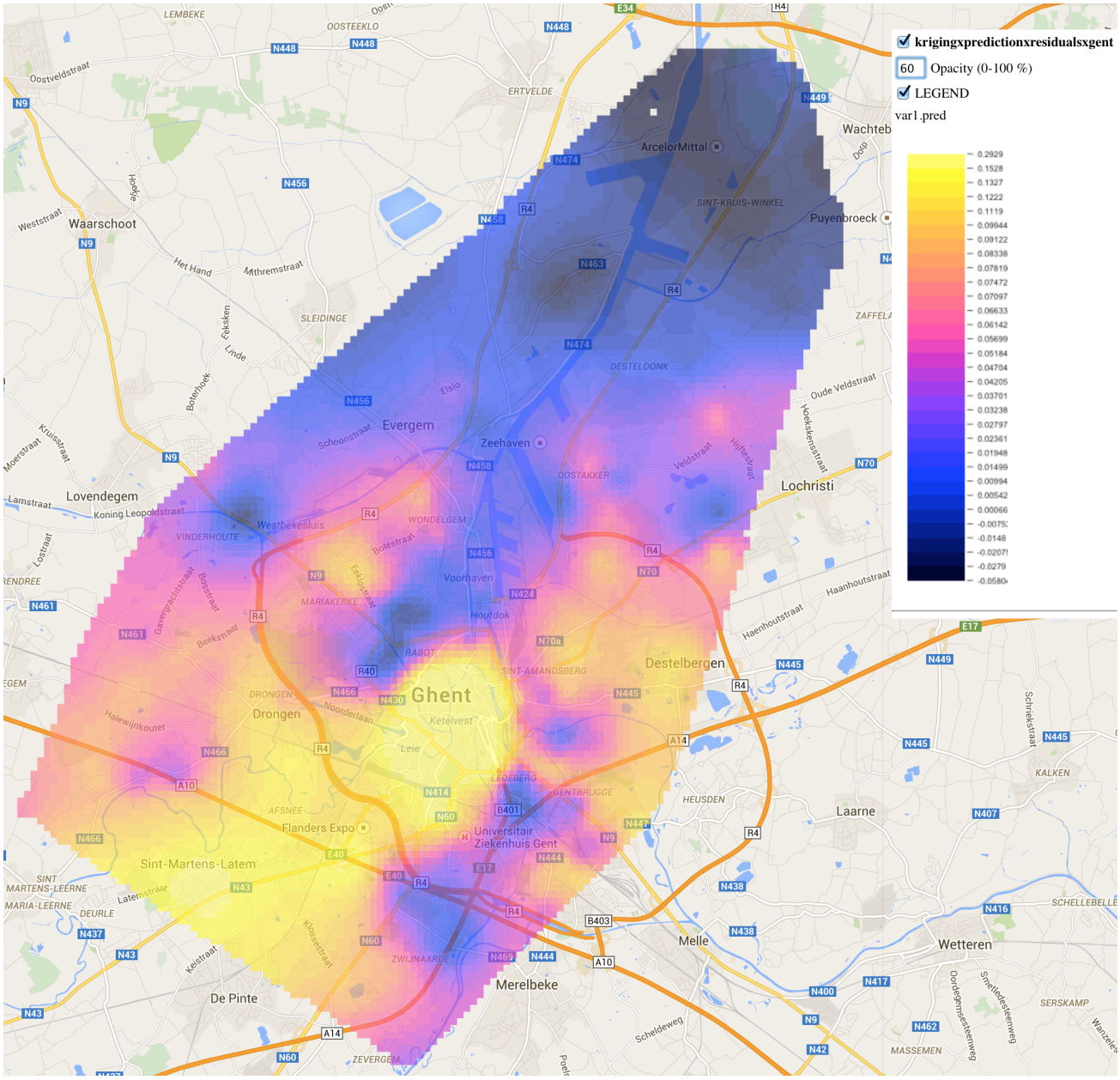


Figure 31: The effect of location calculated with the GAM and kriging method. We can interprete this effect as the average extra one would pay for an average Flemish house on that location. The units are in percent, so in Sint-Martens-Latem one would pay on average around 35% more for a house than in the North of Ghent.

7.2 Model 2: Geographically weighted regression

The second fitted model is the geographically weighted regression model. As explained in section 6.7, we do not make a global model, but fit for each unknown house a local weighted regression model. We use two different distance functions, the Gauss kernel distance (formula (2)), and an expansion of this distance function with the neighbourhood classifications (formula (3)). A plot of these two weighting systems is made (figure 35).

As we are fitting a local model, we are using only a fraction of the total amount of data, and hence it would be foolish to use all features if we are only using around a hundred observations. That is why we only took the most important ones of the GAM model of the first section. The final local model we fitted at each regression point was a weighted GAM model with the following features:

- a smoohting spline of log_l_perc_opp with 2 degrees of freedom
- a smoohting spline of epc with 2 degrees of freedom
- a linear effect of log_bedrooms
- the factor variable t_s (touching sides)
- the factor variable renoveren

With cross validation we select the ideal numbers of observations in the kernel (figure 32). As is demonstrated on the graph, there is a clear optimum of observations to include in the local weighted regression. It is also clear that by rescaling the distance function, giving more weight to observations that are in the same classification of neighbourhood, we need less observations in the kernel to get the same result.

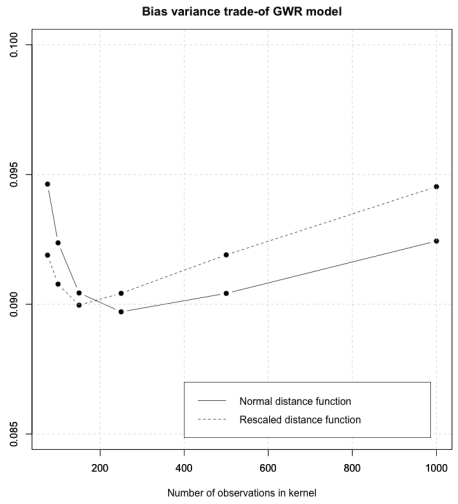
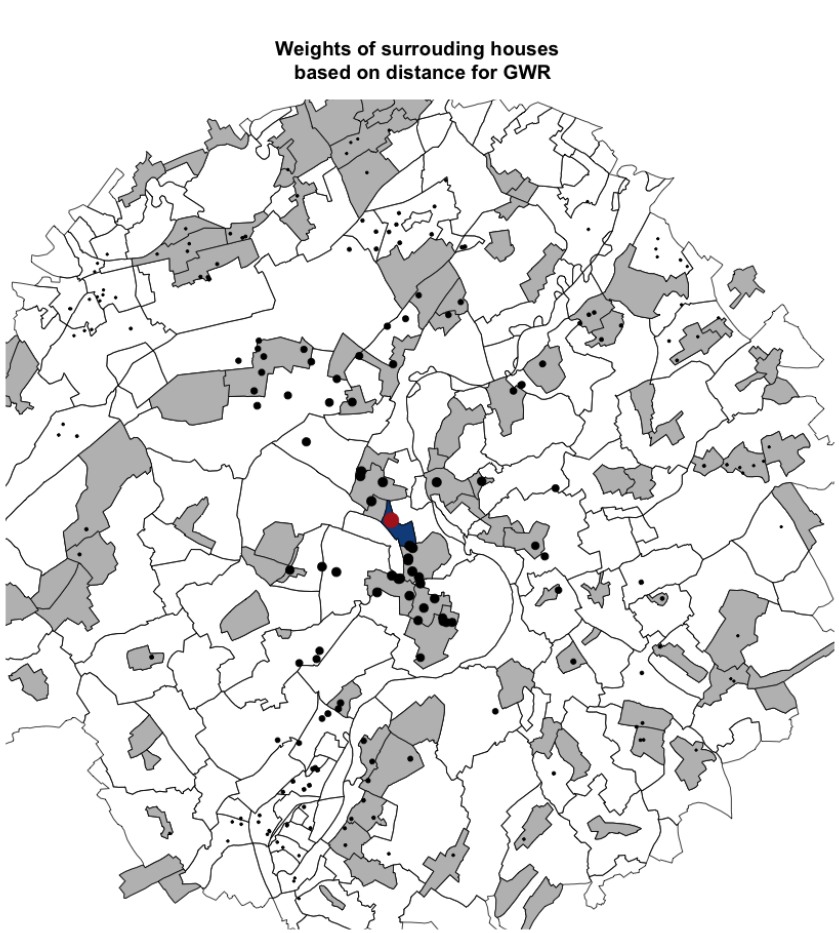
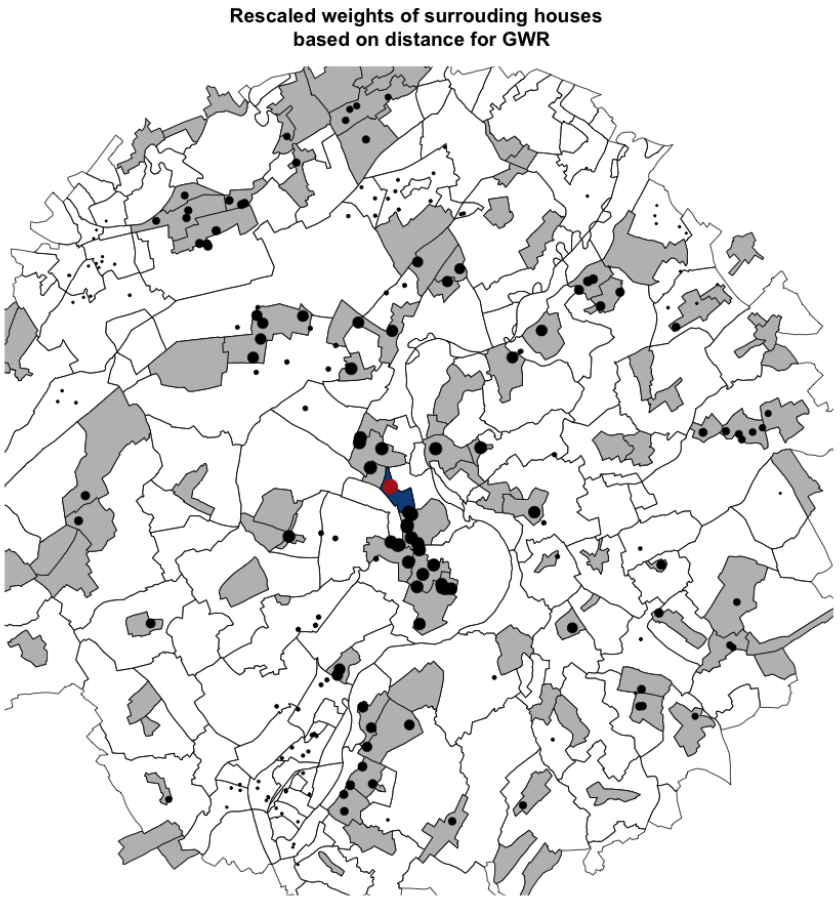


Figure 32: The bias variance trade-off of GWR for two distance functions. There is an optimum of the number of observations to be used in the local kernel.



(a) The neighbouring houses receive a weight in function of the separation distance



(b) The weights of the left figure are rescaled if the house is in a similar neighbourhood (all neighbourhoods with the same density classification of the red house are in gray).

Figure 33: Illustration of the weighting algorithm of geographically weighted regression. We are predicting the price of a house (red dot) by using houses in the neighbourhood (black dots). The larger the black dot, the more weight it gets.

7.3 Model 3: Boosting

The third fitted model is the black box boosting model. We have demonstrated earlier that we expect the model to fit the data reasonably well, although it has a few pitfalls. Firstly, the model is based on thousands of consecutive simple models. As we are dealing with a large dataset with more then 10 predictors, we have to keep an eye on the model complexity (tuning parameters). We have not only to make a trade-off between model complexity and goodness of fit, but as well between model complexity and the time it takes to fit each model. On the cross validation plot (figure 34), we displayed the goodness of fit in function of the tuning parameters. We also plotted the time it took to fit the model. It is clear that both increasing the three depth and the total number of trees increases the time it takes to fit the model. (Notice that the time plotted on the graph is the time it took to fit on of the five folds of Cross Validation. To test a model with 8 leaves in a tree and a total of 6400 trees it would thus take 124 seconds multiplied by 5, which is more than 10 minutes). Given the tuning parameter graph, we fitted the final model with a tree depth of 12 and a total number of trees of 12000.

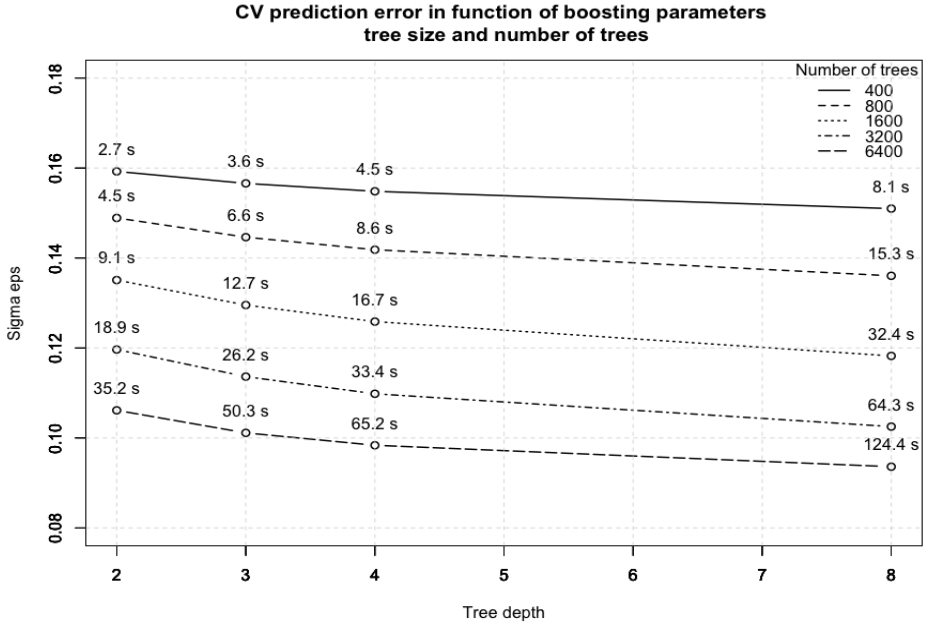
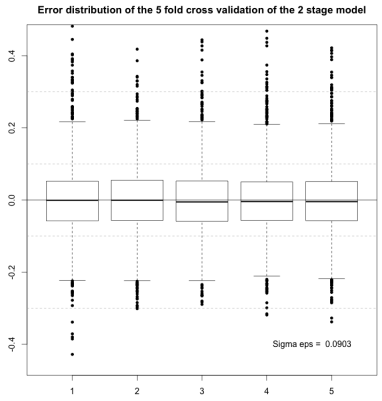
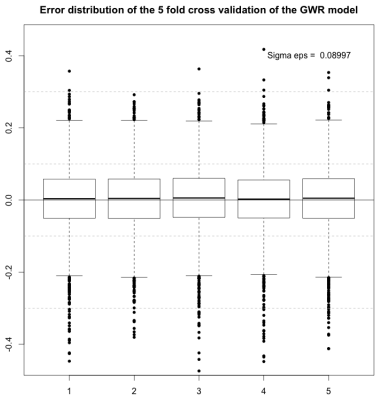


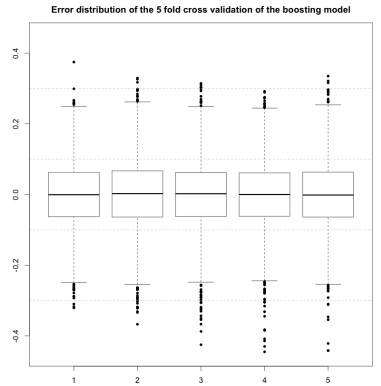
Figure 34: Average σ_ϵ of five fold cross validation for boosting models with different tuning parameters. Each dot represents the average σ_ϵ of a boosting model with tuning parameters tree depth d and total number of trees B



(a) Cross validation results of the 2 step model. σ_ϵ=0.0903



(b) Cross validation results of the GWR model. σ_ϵ=0.0899



(c) Cross validation results of the Boosting model. σ_ϵ=0.0934

Figure 35: Goodness of fit statistic of the three models of the true listing dataset

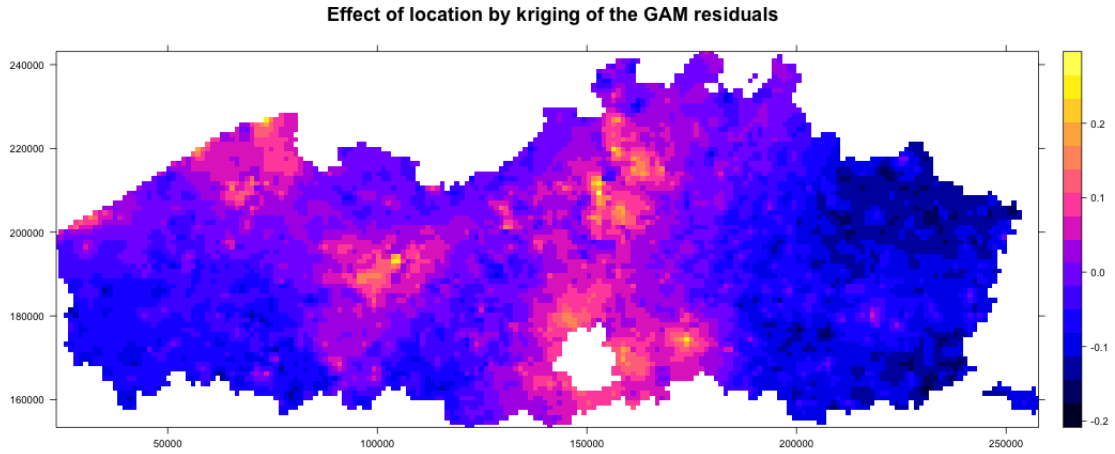


Figure 36: Map interpolated with kriging. The map can be interpret as "the average one would pay more for an average Flemish house on a particular location". Notice that this map is in relative units.

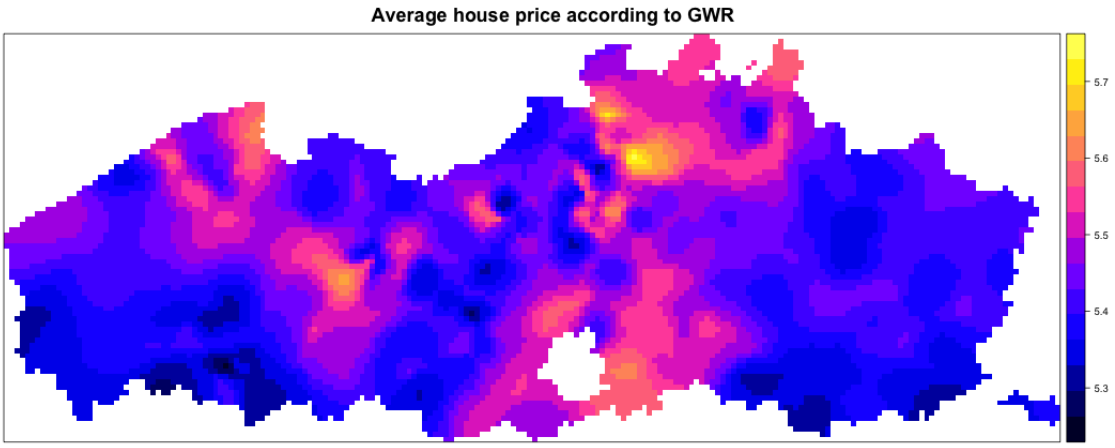


Figure 37: Map of the weighted average house price produced by geographically weighted regression. This map represents the local average price of a house on a particular location.

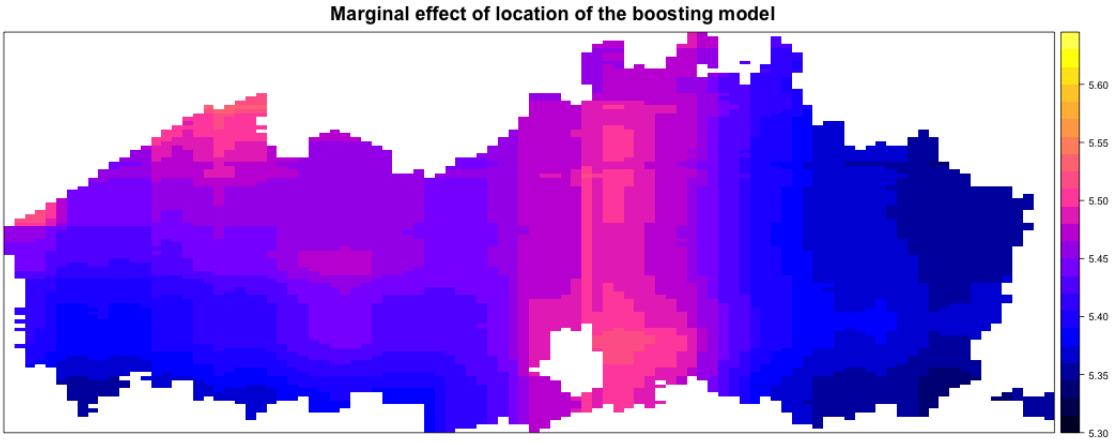


Figure 38: The marginal additional effect of location calculated by boosting. This map can be interpreted as the extra premium one pays for an average house on a particular location.

8 Discussion

All three models have advantages and disadvantages, but all three were very valuable throughout the thesis and for the final valuation engine.

Although the boosting model has the least prediction power and the produced graph is clearly not able to explain the effect of location on the house prices, its value can not be underestimated. It is a black box "machine learning" model, and can be fit without making any assumption or without understanding anything about econometrics or spatial statistics. That is why this model is often fit first. The goodness of fit of the two other models, based on complex statistical assumptions, is compared the boosting model. It is clear that the two other models outperform the boosting model, but a lot more time was spent understanding and modelling these.

The geographically weighted regression is the most intuitive model. When we estimate a house, we use only houses that are nearby. Due to the bias-variance trade of, we can however not take into account too many features, as the local models would need too many observations. This is where the global GAM + kriging model comes into play, as we assume global feature effects we use all the observations, and we can capture strong non-linear feature effects. A serious disadvantage of the global GAM model is that it is not able to capture interactions between effects.

The produced maps of kriging and GWR are very different. The kriging interpollated maps show a lot more variation. This is because the interpolation by kriging does not only take into account the distance between observations, but as well the correlation between all the observations. Clustered houses receive all an equal weight in GWR, where in kriging a cluster of houses receive less weight compared to an other house that stands alone. This effect is known as shielding. It is due to this effect that the interpolated map by kriging is able to capture a lot more variation.

9 Conclusion

The estimation of a house price based on online gathered listings and open data sources is proven possible and meaning full. The effect of features such as number of bedrooms or building area can be calculated, and maps can be made of the effect of location on the house price.

Correctly gathering the data is a crucial step. Time needs to be spend investigating the characteristics and noise of each feature. Each listing of a house could be matched with independent objective extra features, such as the governmental cadastre data.

A first black box machine learning model, generalized boosted regression, was fit to the data. The advantage of boosting is that it requires no statistical assumptions, and hence it can be fit in a very early stage of the modelling process. Although the calculated feature effects are stepwise and hard to interpret, the goodness of fit is close to the other statistical methods.

If one takes the assumption that the house prices is a composition of goods (or feature effects), regression models can be used to calculate the feature effects and predict the house price. The regression models can be expanded with smoothing splines to capture non-linearities. A generalized additive model is able to fit these smoothing splines with back fitting for several features. A disadvantage of these models is that no interaction between feature effects can be calculated.

In case the feature effects are assumed heterogeneous over space, one can fit a global model. In a first step the spatial correlation between the houses is ignored, and taken into account only in a second step by interpolating the residuals of the first step. The interpolation algorithm kriging is used, based on the modelled experimental variogram of the residuals. The produced map can be interpret as the average extra one pays for an average Flemish house on a particular location.

When one does not take the assumption of spatial heterogeneity of the feature effects, local models have to be fit. Geographically weighted regression is such a local model, that gives more weight to nearby observations when predicting the price of a house. The number of local houses used is a bias variance trade-off. The distance functions can be expanded with the neighbourhood classification of the houses. A map of the locally weighted average house price can be made based on the listings.

Cross validation estimates the expected prediction error and is used on the three different models. The three different models have all three similar prediction power.

10 Appedix

10.1 List of R-files

The names of the R-scrits are meaningless.

- **2014-10-23 read Statbel to polygon.R**: R file to clean the raw csv files from FOD economie, and convert them into spatial dataframes.

10.2 Figures

```
Call: gam(formula = log_price ~ s(log_geb_opp, 2) + s(log_l_perc_opp) +
    s(log_bedrooms) + s(epc) + as.factor(garage) + as.factor(t_s) +
    s(log_facade, 2) + s(log_d_to_street, 2) + as.factor(month_dummy) +
    as.factor(zuid) + s(log_speed_of_sale, 2) + as.factor(renoveren),
    data = gam_data_sale)
Deviance Residuals:
       Min         1Q     Median         3Q        Max
-3.0511175 -0.0745639  0.0005365  0.0724030  0.9388962

(Dispersion Parameter for gaussian family taken to be 0.0162)

    Null Deviance: 523.0645 on 14217 degrees of freedom
Residual Deviance: 230.0565 on 14181 degrees of freedom
AIC: -18209.23

Number of Local Scoring Iterations: 2

Anova for Parametric Effects
                          Df  Sum Sq Mean Sq   F value    Pr(>F)
s(log_geb_opp, 2)          1 115.226 115.226 7102.7018 < 2.2e-16 ***
s(log_l_perc_opp)          1  48.097  48.097 2964.7804 < 2.2e-16 ***
s(log_bedrooms)            1  37.675  37.675 2322.3562 < 2.2e-16 ***
s(epc)                     1  64.037  64.037 3947.3186 < 2.2e-16 ***
as.factor(garage)          1   1.879   1.879  115.8204 < 2.2e-16 ***
as.factor(t_s)             2   3.955   1.977  121.8851 < 2.2e-16 ***
s(log_facade, 2)           1   0.391   0.391   24.1222 9.142e-07 ***
s(log_d_to_street, 2)      1   1.262   1.262   77.8203 < 2.2e-16 ***
as.factor(month_dummy)    11   1.251   0.114    7.0128 5.705e-12 ***
as.factor(zuid)            1   0.046   0.046    2.8259  0.092775 .
s(log_speed_of_sale, 2)    1   0.182   0.182   11.2026  0.000819 ***
as.factor(renoveren)       1   4.716   4.716  290.6752 < 2.2e-16 ***
Residuals              14181 230.057   0.016
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova for Nonparametric Effects
                        Npar Df  Npar F     Pr(F)
(Intercept)
s(log_geb_opp, 2)             1  38.027 7.165e-10 ***
s(log_l_perc_opp)             3  58.117 < 2.2e-16 ***
s(log_bedrooms)               3  43.277 < 2.2e-16 ***
s(epc)                        3 100.472 < 2.2e-16 ***
as.factor(garage)
as.factor(t_s)
s(log_facade, 2)              1  15.628 7.748e-05 ***
s(log_d_to_street, 2)         1   4.539   0.03315 *
as.factor(month_dummy)
as.factor(zuid)
s(log_speed_of_sale, 2)       1   4.105   0.04278 *
as.factor(renoveren)
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 39: R output of the first step GAM model

References

C Brunsdon. Geographically weighted regression. *Journal of the Royal . . .*, 1998.

Trevor Hastie and R Tibshirani. The elements of statistical learning: data mining, inference and prediction. 2005.

Roel Helgers, Eric Buyst, and Frank Verboven. De relatie tussen woning- karakteristieken en woningprijzen: een nieuw licht op de recente prijsevolutie in Vlaanderen. (november 2011):472–479, 2013.

G James, D Witten, T Hastie, and R Tibshirani. *An introduction to statistical learning*, volume 102. 2013. ISBN 9781461471370.

Steven D. Levitt and Stephen J. Dubner. Freakonomics: A Rogue Economist Explores the Hidden Side of Everything.

R Kelley Pace. Spatial Statistics and Real Estate. 17:5–13, 1998.

Marno Verbeek. *A Guide to Modern Econometrics*. 2004. ISBN 0470857730.

Bo Wu, Rongrong Li, and Bo Huang. A geographically and temporally weighted autoregressive model with application to housing prices. *International Journal of Geographical Information Science*, 28(5):1186–1204, January 2014. ISSN 1365-8816. doi: 10.1080/13658816.2013.878463.