Data engineering for house price prediction

van der Burgt, E.J.T.G.

Award date:
2017

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science
Data Mining Group

Data Engineering for house price prediction

Master's Thesis

Erik van der Burgt

Supervisors:
prof. dr. Mykola Pechenizkiy (TU/e)
ir. Koen Vrijdag (Royals@Work)

Assessment committee:
prof. dr. Mykola Pechenizkiy (TU/e)
ir. Koen Vrijdag (Royals@Work)
dr. ing. Marwan Hassani (TU/e)

Eindhoven, May 2017

Abstract

Estimation of house prices can be done in various ways. In this thesis two approaches are presented. All the presented solutions and methods are implemented in a framework. This framework is also capable of, estimating the house price given a certain input, based on calculated models and input data. The machine learning approach uses regression models to predict the price of a real estate. A basic model is a model which is trained with data from one region and one property type. Feature selection and imputation are performed on various data sets in order to create a correlated and complete set to use in training, validation and testing. Because there are neighborhoods which do not contain that many transactions per property type, two neighborhood clustering methods are introduced with the goal to merge neighborhoods with the same characteristics, in order to create models that are trained with more data. Ensemble learning is used to merge basic models that are trained on different regions in order to decrease the error. Where a region can be a cluster of neighborhoods, municipality or the Netherlands. The solution being a stacked regression model for each neighborhood or fallback on municipality, resulting in an accuracy of 85.90%.

Contents

Glossary

ensemble learning Process with multiple (regression) models that are strategically generated and combined to solve a particular machine learning problem. Ensembling tends to increase the accuracy of the model and reduce the error. 2, 3

level-0 model Standard machine learning model that is used for prediction. In ensemble learning, this is called level-0 because it is the starting point for improvement.. 3, 10, 42–44

level-1 model Machine learning model that uses the output of level-0 models as an input, the first solution of ensemble learning. 3, 10, 42–44

neighborhood District or area with distinctive characteristics that is a geographically localized community within a larger city or municipality.. 1, 3, 4, 7, 8, 14, 21, 22, 27, 28, 37, 39–41, 43–45, 47, 49, 50

Price Index A normalized average of a property's value with a given type in a given region during an interval of time. 2, 10, 13, 15–20, 26, 49

regression Statistical process for estimating the relationships among a set of features and given output. 2, 3, 15, 23, 24, 33, 34, 41, 42

transaction A transaction represents a sale of a real estate, with the price, date and real estate information. 13–15, 21, 28, 31

WOZ-value Official estimation of a property's value, estimated by the municipality. WOZ is also used for the amount of municipality taxes the property owner needs to pay. (Dutch: Waardering Ontroerende Zaken). 1, 14

Acronyms

CBS Central Office for Statistics. 8, 13

CS Repeated Sales price index by Case and Shiller. 15, 16, 18, 19, 25, 26, 34

NVM Dutch Real Estate Agent Association (Dutch:Nederlandse Vereniging voor Makelaars). 13, 14

RSS Residual Sum of Squares. 15, 34

SPAR Sale Price Appraisal Ratio. 14, 16–20, 25, 26, 34

WOZ Waarde onroerende zaken. 14, 20

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter the problem statement is presented along with domain knowledge, used methodology and outline of the thesis. First domain knowledge is presented in Section 1.1. Secondly the problem is presented along with the related questions in Section 1.2. Next the research methodology in Section 1.3 and a outline of the thesis is Section 1.4.

1.1 Domain

The real estate domain holds many activities like acting as an intermediary between a buyer and seller, arranging or advising with buying or selling real estate and the valuation of (residential) property. The valuation results in a valuation report that can be used for mortgage requests or as an objection to the indicated WOZ-value. A validated valuation report that is used for a mortgage request, requires a good indication of the property value. It is created by the real estate agent and validated (depending on the type of the report) by an external institute that validates if the price is a good indication for the dwelling. However, this report requires the real estate agent to inspect the valuated dwelling and other (mostly previously valuated) dwellings. The valuation process in total takes about a week if everything is scheduled well and costs the applicant of the report about 300 euro. The result of a valuation is always a subjective one and therefore always not accurate, however the average accuracy is that two third has an error of less than 20%, where 20% is within a 5% error range [Schekkerman, 2004].

1.2 Problem statement

The valuation of a dwelling costs time and money, however for some cases it is feasible to have an indication of the property price instead of a validated valuation report. In preliminary work, (auction) price prediction is researched by Vrijdag in [Vrijdag, 2016]. Although the result for sales seems accurate it is too inaccurate to serve as a uniformly and / or consistently reliable price indication. Therefore the goal of this project is to introduce a price prediction approach that has a more consistent and reliable methodology. Reliability can be achieved if the prediction for each region is within a certain limit and the global standard deviation is improved. Therefore the general problem statement is: Given an address in the Netherlands with additional properties, estimate the price of the real estate on that address (Figure 1.1). If the address is already sold once and therefore the characteristics are known, validation is needed to account for changes of the real estate or neighborhood.

This problem can be solved in various ways. By looking at it from a real estate agents' perspective one would come up with a model like Figure 1.2. For this strategy, the real estate agent would find a similar house in the neighborhood and then uses a weighting to accommodate for the differences between the two. In this case there are two options, the similar dwelling is sold less than a month

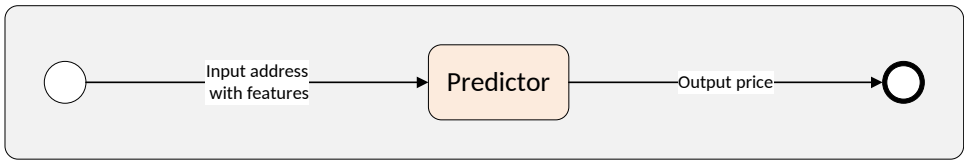


Figure 1.1: Simple price prediction flow, related to problem statement.

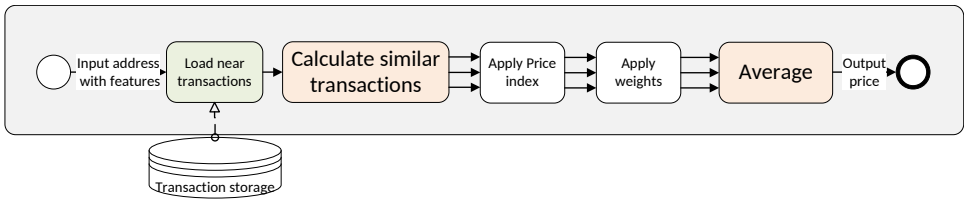


Figure 1.2: Real Estate agent price prediction flow. Where the input is an address with additional properties of the real estate and the output is the average price over the near transactions.

ago or the price needs to be weighted again by a Price Index to accommodate for market changes. A minimum of three reference properties is required on a validated valuation report. By taking the average of the three, or more results in a price for the query dwelling.

Another way to look at the problem is from a machine learning approach Figure 1.3. Using regression to create models that represent an area. With the addition to create a combination of these models with ensemble learning.

In order to make these two methods more accurate, extra Real Estate characteristics (other than type and size) are a good start. For the first approach of the Real Estate agent an adjustable Price Index is needed along with a method of determining similarity, to search for similar dwellings. This leads to the following research objective with questions associated to it: *We would like to have a Price Index that can be calculated on a specific region, different time ranges and for specific property types, such that it follows the trend of the market changes.* The research questions that relates to this objective is: "Is it possible to create a Price Index that behaves the same or better as the government issued one" and "What is the difference of a specific Price Index in comparison to a general Price Index". From Data Engineering perspective it is desired to know how the index is calculated. To calculate a Price Index, it is desired to know what the alternative methods are that can be used to calculate a Price Index with the available data. If the Price Index is calculated, then the difference between a specific and general Price Index can be calculated as well as the accuracy between the two.

The first approach is from the real estate agents' perspective, where the research objective is: *We would like to be able to predict a value of a dwelling in a way similar to real estate valuation.* We want to measure the accuracy of the prediction model in order to relate this to the real world

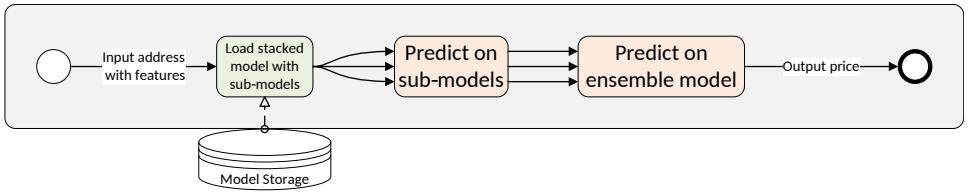


Figure 1.3: Price prediction flow using multiple regression models. Where the input is an address with additional properties of the real estate and the output is the result of the ensemble model.

scenario. In order to measure the accuracy of this method it is necessary to know what technique is similar to the technique the real estate agent uses.

The more robust and accurate approach is machine learning with the use of regression and ensemble learning models. A level-1 model, the result of ensemble learning, requires other machine learning models (level-0 models) as input. Therefore the data input in the level-0 models models is essential, this leads to the following objective: *We want to gather more Real Estate characteristics in order to have more correlated Real Estate features which will improve the prediction.* The main question is if it is possible to gather a large amount of correlated real estate data. To answer this question it is valuable to know what correlated data is and how features are correlated. Because data is gathered from external sources the chance is large that not every entry is present, therefore it is necessary to know how to deal with the missing features or entries.

The second objective is: *We want to predict a price of a real estate with a regression model.* Now that we have an extended data source and the default one, we want to know what the accuracy of a regression model is, default compared to extended. The other interesting point is the accuracy of a model that is trained on all properties types compared to a set of models that are each trained on one property type. For both of these challenges the accuracy of the regression model needs to be calculated as well as what the requirements are in order to train a regression model with certain features.

The next step is to refine the prediction in order to have a lower error, therefore the following goal is set: *We want to combine different regression models in order to create a model with higher accuracy and lower error rate.* From the previous goal the accuracy and error of the default model is known. The questions that arise form this goal are how to create a model based on other models. Therefore it is necessary to know how to combine models that are trained with different data sets.

1.3 Research methodologies

This thesis contains four main research methodologies, they do not directly align to the thesis structure. Therefore a brief overview is listed below.

Price Index construction: In Chapter 4, different types of price indices are discussed. Sale Price Appraisal Ratio, Repeated Sales and standard statistical techniques are explained and evaluated on the data set.

Data Engineering: Data Engineering is conducted on data sets in order to make the features fit better for machine learning, Chapter 5. Different methods for missing or incomplete data are discussed as well as different methods for measuring correlated features.

Regression: Regression for predicting real estate prices is used in Chapter 7. The algorithm and different implementations are listed in the same chapter. Also basic models are trained in order to get some insights over the different data sets and property based models.

Sale and Neighborhood comparisons: For the Real Estate Agent prediction comparisons are made between different sales (transactions) in Chapter 6. In Chapter 8 comparisons between neighborhoods are made based on an existing framework and data set. This in order to combine them into clusters.

1.4 Thesis outline

In Chapter 2 the backgrounds of this thesis are presented with the Dutch housing market and the previous work. The next section, Chapter 3 lists the basic foundations of the developed framework. This framework is capable to calculate and validate various regressions models with different types and amounts of data. The ultimate goal of this framework is to predict a house price for an existing dwelling in the Netherlands, based on a group of models that covers the whole of the Netherlands. Different price indices explained in Chapter 4 are used for translating transaction prices overtime. These price indices can be calculated and validated by the framework. The data that forms the input for the regression models is gathered from different sources. With the use of different feature selection methods the best properties are selected to form a presentation of a dwelling which is described in Chapter 5. A method similar to how a real estate agent performs a valuation but without visiting the real estate is presented in Chapter 6. The K-Nearest Neighbor algorithm is used to simulated the real estate agent and to find similar transactions in the neighborhood. Regression is used as a machine learning approach to predict a price with higher accuracy in Chapter 7. To improve this approach neighborhood clusters are constructed by two different approaches. These clusters are used to create ensemble models that will form a model for a region Chapter 8. The conclusions and future work are listed in Chapter 9.

Chapter 2

Background

In this chapter the basic background knowledge for the rest of the thesis is covered. First a general introduction about the Dutch housing market is presented in Section 2.1. Secondly the previous work done by Koen Vrijdag about auction price prediction in Section 2.2.

2.1 Dutch housing market

The dutch housing market contains about 7.6 million properties. Only 57% of these properties are owner-occupied, the rest is in rented sector [Capital Value, 2016]. In 2016 a new record was set for the amount of sold properties a year namely, 214,793 [Vrieselaar et al., 2017]. This means that around 5% of the whole market is sold within a year. This year growth tends to increase even more, however the prediction is that shortages will supress the growth. The prediction is that between 220,000 and 230,000 properties will be sold [Vrieselaar et al., 2017] in 2017.

Not only the amount of property transactions will increase but also the transaction prices. In the year 2008 there was a financial crisis in the Netherlands which dropped the Gross domestic product (GDP) by a large amount. Currently the GDP is again at the same level of 2008, however the growth is still less as in 2008 [Bhageloe-Datadin, 2016]. Therefore the market is making a steady recovery. Because of the growth in transactions, economy and the shortages the increment of prices is inevitable.

The fact that the amount of transactions is increasing also means that the amount of valuation reports increases. Almost every property that is sold will require a valuation report for the mortgage. However there will also be valuation reports for the special management branch of banks and for other reasons. The special management branch regulates the finances of people who cannot afford to pay their loan anymore. This branch is a special division within the bank itself, every year they would like to have an update report of all these special properties.

2.2 Previous work

The work of [Vrijdag, 2016] introduced a framework as shown in Figure 2.1 that predicts auction prices based on transfer learning by sale prices. The resulting accuracy of the sale prices is not bad with a weighted average accuracy of 85.97%. The basic sale prediction part has a good basis however, there are improvements that could make it more accurate and more reliable across all the regions of the Netherlands. Which should also improve the accuracy of the auction price prediction.

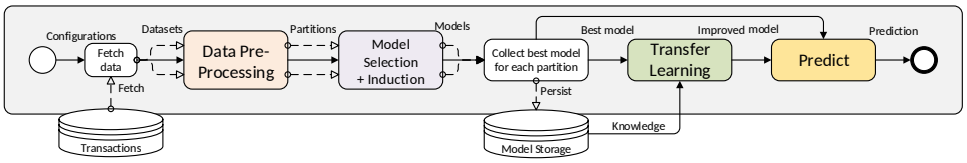


Figure 2.1: Highly abstracted view of framework by Vrijdag. Models trained through this framework can predict regular sales and auctions (with or without transfer learning). [Vrijdag, 2016]

The parts “Fetch data“, “Data Pre-Processing“ and “Model Selection“ and “Induction“ that are the base sale prediction part that are going to be improved. This is however not done in a the existing framework, because of the big changes. The “selecting model“ part in the framework Figure 2.1 will need refactoring when changing the first part. While the “transfer learning“ part should remain the same.

Chapter 3

Prediction Framework

In this section the revised framework (build from scratch) is presented. First the data inputs Section 3.1 are listed. Secondly the supported tasks of the framework are explained Section 3.2.

3.1 Available Data

At the start of the project there are already data sets available that can be useful. A table filled with all the addresses of real estate in the Netherlands, see section Subsection 3.1.1, Transaction data of real estates that are sold in the past Subsection 3.1.2, statistical data of each neighborhood in the Netherlands Subsection 3.1.3 and a data set of the trainstations in the Netherlands Subsection 3.1.4.

3.1.1 Address data

The address data called BAG (Registration of Address and Building) contains the addresses of all the real estate, type of building, construction year of the property and living area in meters squared. The dataset originates from Kadaster which is the official party responsible for registering real estate information in the Netherlands. The dataset contains about 8.8 million entries and receives updates every month.

3.1.2 Transaction data

The transaction data set contains all the officially registered transactions of real estate properties since 1993. This is data has the properties: property type, construction year, building type, lot size, living area size, transaction date and price. The set contains more than 5.4 million records and originates also from Kadaster and is monthly updated. Some statistics are gatherd in tables Table 3.1 and Table 3.2 to get a better understanding of the types of properties.

Table 3.1: Statistics of the Transaction data set for each property type.

Property type	Amount	Average price (€)	Average Area (€)
A (Apartments)	1,413,074	179,273	313
C (Corner)	626,525	191,758	236
SD (Semi-Detached)	502,151	219,622	505
T (Terraced)	131,990	182,391	172
D (Detached)	577,792	309,641	1,806

Table 3.2: Percentages of the amount of property types on the Transaction data set for the four largest municipalities.

Municipality	A	C	SD	T	D
Amsterdam	86.20	2.41	0.80	9.86	0.74
Rotterdam	71.79	6.30	1.57	19.16	1.18
's-Gravenhage	83.69	2.66	0.80	12.39	0.46
Utrecht	46.90	7.50	1.33	43.18	1.10

3.1.3 Neighborhood data

For all the neighborhoods, around 180 statistical attributes are collected that contain information such as the amount of stores, schools but also about the people that live in the neighborhood. The Netherlands counts around 12 thousand neighborhoods. This data originates from CBS and is very sparse, this is because not all the ingredients for the statistics are available for every neighborhood. Therefore some missing properties need to be handled, Subsection 5.1.1.

3.1.4 Train stations

A list of all the train stations in the Netherlands and the directly connected ones in Belgium and Germany. For each train station the name, location and type are available. The type of the station depends on the amount of trains and the type of trains that go through the station. The data set contains the three types of trains: Stop, Speed and Intercity. There are 8 different types of stations, Normal and Node for each type of train gives 6. The other 2 are Mega (only the largest stations) and Optional. The optional stations are mostly located next to large stadiums and are not available on a normal day. Therefore these stations will be filtered out the list of stations. For prediction we will use 5 different properties:

1. The distance in meters to nearest station
2. The distance in meters to nearest Stop station
3. The distance in meters to nearest Intercity station
4. The distance in meters to nearest Speed station
5. The distance in meters to nearest Mega station

3.2 Tasks

The Framework works with a start up configuration that indicates the task and some additional parameters for the task. Each task is related to a research objective where an objective can have one or more tasks. Currently these tasks are supported:

- Calculate price index (Figure 3.1)
- Validate price index (Figure 3.2)
- Calculate feature selection (Figure 3.3)
- Construct and validate price prediction model (Figure 3.4)
- Construct neighborhood cluster (No figure, rather straightforward flow)
- Construct stacked prediction model (Figure 3.5)

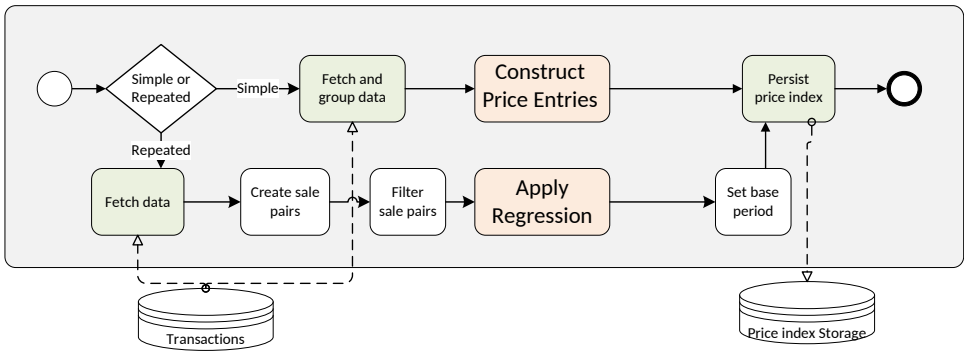


Figure 3.1: Detailed view of the price index flow in the framework.

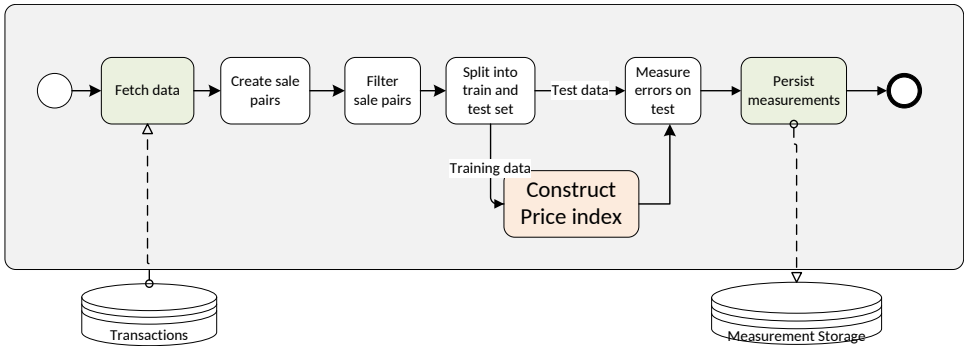


Figure 3.2: Detailed view of the price index validation flow in the framework.

The price index flow in Figure 3.1, is able to calculate a price index for a given region with a given time interval of the types: Simple (Average and Square meter) and Repeated sales. For the simple price index, transactions are loaded in groups from the store so no pre-processing is required for calculation. For the repeated sales approach the pairs require pre-processing before regression is used to calculate the index values. Post-processing is required in order to calculate the correct base year. All price index entries are stored in the price index store.

The price index validation flow (Figure 3.2), can only be used for the repeated sales index. This task constructs the repeated sales price index. From the test data the first price is used as input while the second price is calculated by the index. This is done on both the constructed index and the government index. The price and predicted price are stored in a measurement object.

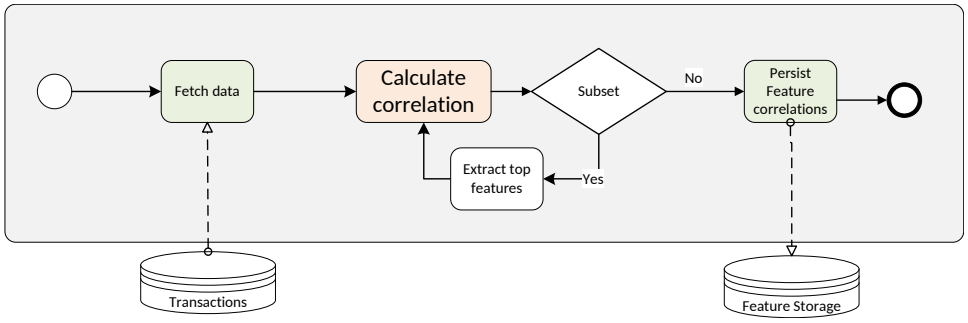


Figure 3.3: In depth view of the Feature Selection configuration lane of the framework.

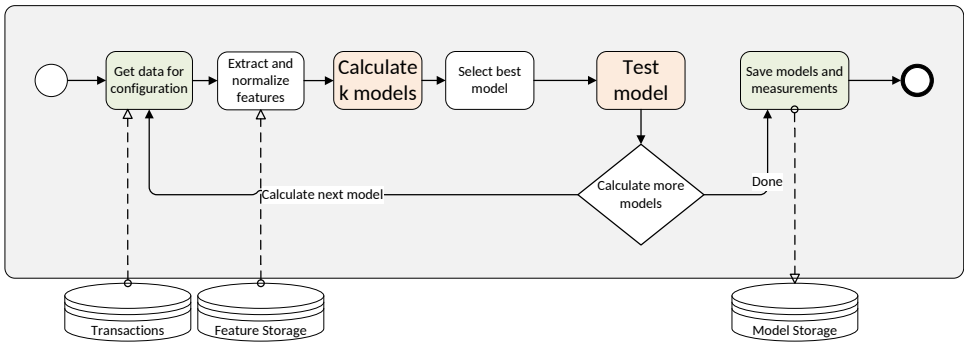


Figure 3.4: In depth view of the Prediction configuration lane of the framework.

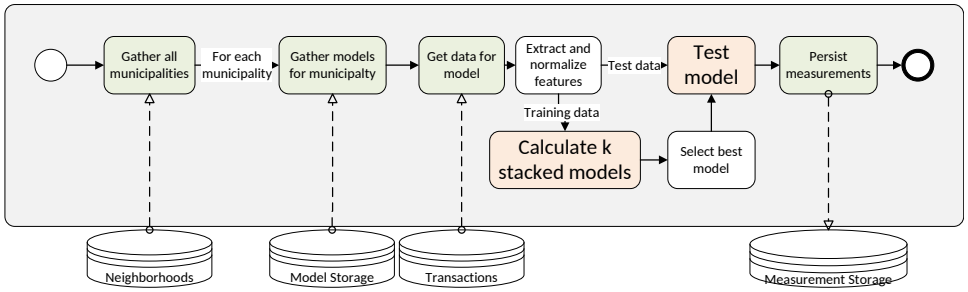


Figure 3.5: In depth view of the stacked model prediction configuration lane.

The feature selection task (Figure 3.3), calculates the correlated features for all of the features that are in the data set. A special feature store is created to store the results of this task.

The prediction flow (Figure 3.4) is used to create a simple model on data (level-0 model). Each of the k models are created with the use of features that are selected in the feature selection task. After creation the model with the lowest measurement parameter (RSS) is tested and stored with the measurement results in the model store.

The stacked prediction task (Figure 3.5) groups the data for each sub region with the according models. For each group k models are predicted and with test data the model is tested. The resulting model (level-1 model), is stored in the model store along with the measurement and a list of the used sub models.

Each of these configurations saves one or multiple entries in the database, where the validation lanes construct a Measurement object Subsection 3.2.2 that contains use full metrics. The calculated Price Indexs are represented in a price entry object Subsection 3.2.1.

3.2.1 Price entry

The price entry object always exists as a list of objects, where each object represents a period with a certain region. The price index with time interval year, over The Netherlands contains 24 entries (since 1993). Each price entry representing one year with the percentage according to the base year for that entry, the base year is also listed (default is set to 2010).

3.2.2 Measurement

The Measurement object with input original and predicted value, calculates the following statistical variables, where P is the output set and $\hat{P}$ the predicted output defined as $P = \{p_1, \ldots, p_n\}$,

$\hat{P} = \{\hat{p_1}, \ldots, \hat{p_n}\}$ and $E = \{e_1, \ldots, e_n\} = abs(P - \hat{P})$, where n is the size of the set (equal for all sets):

- Minimal Error: $min(E)$
- Maximum Error: $max(E)$
- Mean Error (μ): $\frac{1}{n}\sum_{i=1}^{n} e_i$
- Standard Deviation: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \mu)^2}$
- Residual Sum of Squares (RSS): $\sum_{i=1}^{n}(p_i - \hat{p_i})^2$
- Mean Squared Error (MSE): $\frac{1}{n}\sum_{i=1}^{n}(p_i - \hat{p_i})^2$
- Rooted Mean Squared Error (RMSE): $\sqrt{\sum_{i=1}^{n}\frac{(p_i - \hat{p_i})^2}{p_i - \hat{p_i}}}$
- Mean Absolute Percentage Error (MAPE): $\frac{100}{n}\sum_{i=1}^{n}\frac{p_i - \hat{p_i}}{p_i}$
- Mean Absolute Accuracy (MAA): $100 - \frac{100}{n}\sum_{i=1}^{n}\frac{p_i - \hat{p_i}}{p_i}$

Chapter 4

House Price index

Properties that are sold from one owner to the other result in a transaction, as listed in Subsection 3.1.2. The price that is listed in these transactions can not be compared meaningfully with each other due to inflation over the years, different date, different region, saturation and desire. A solution for the date, region and inflation is to use a Price Index. The transaction prices will be normalized according to the transaction date. Therefore every price that is adjusted with a price index can be compared without having to deal with inflation at the time of the sale. The research objective in this chapter is the following:

- We would like to have a Price Index that can be calculated on a specific region, different time ranges and for specific property types, such that it follows the trend of the market changes.

To achieve this objective, different types of price indices are researched. The Average Property Price is discussed in Section 4.1 and Square Meter Price in Section 4.2. More advanced price indices like the Sales Price Appraisal Ratio in Section 4.3 and Repeated Sales method in Section 4.4.

4.1 Average Property Price

The most basic indicator for comparing dwelling values is the Average Property Price, calculated by the average price of all the sold dwelling in a certain time frame (4.1). This can be calculated with different periods, regions and for different types of dwelling.

$$\rho_{avg} = (\sum_{i=0}^{N} T_i^p)/N \tag{4.1}$$

In Equation 4.1 ρ_{avg} is the average price for the given set of transactions T. The T_i^p indicates a price of a transaction at the index i, N is the amount of items in the set. This can not be considered a valid Price Index as also stated in a memo by the CBS [van der Wal and Wiebe, 2008]. This is due to the fact that every period different houses are sold with different attributes. If the Average Property Price is used in combination with different property types and regions it would be more accurate.

4.2 Square Meter Price

One of the important features of a real estate is the size. One metric available for the size is the amount of square meters. Although this characteristic does not represent the whole house it has a high correlation with the price. This Price Index is used by NVM and The Finnish government [Finland, 2017], to provide an overview of the real estate market. The calculation itself is straight forward Equation 4.2 but there are some constraints. A threshold of 50 on the

amount of transactions for time and region is implemented to have a good representation. In a neighborhood per month this is quite a lot. Therefore the index NVM provides, uses regions instead of neighborhoods, 76 in total ranging from one till ten municipalities each.

$$\rho_{smp} = (\sum_{i=0}^{N} T_i^p / T_i^m)/N \tag{4.2}$$

In Equation 4.2 ρ_{smp} is the (average) square meter price for the given transaction set T. The T_i^p indicates a price of a transaction at the index i, T_i^m the square meters. The N indicates the amount of items in the Transaction set.

4.3 Sales Price Appraisal Ratio (SPAR)

The method used in the Netherlands by the government is the Sales Price Appraisal Ratio (SPAR) method presented in [de Haan et al., 2009] [Bourassa et al., 2006]. The SPAR method makes use of the WOZ-value and the transaction price. The WOZ-value is the value that indicates the amount of taxes that needs to be paid for the Real Estate.

$$\hat{p}{}_W^t = \frac{\sum_{i \in S^t} p_i^t / n^t}{\sum_{i \in S^t} \hat{p}{}_i^0 / n^t} \tag{4.3}$$

In the Netherlands a homeowners pays taxes based on the WOZ-value of their property. Homeowners often complained that the values were not correct, politicians reacted and changed the law [1]. October 2016 is the starting date of the new law that lists that the WOZ must be publicly available. This way homeowners can see if the WOZ-value of their home is correct with respect to similar houses. Although the data is publicly available, it is not allowed to download the data automatically or more than 10 properties a day. On the release day only 45% of the municipalities had their data online. Therefore it is not possible to gather the data and create the index.

The SPAR index is also publicly available and is calculated by the Central Bureau of Statistics in association with the Kadaster. However it is not as detailed as required, it starts in the year 1995 and the following combinations are available:

1. The whole of Netherlands year based.
2. The whole of Netherlands quarter based.
3. The whole of Netherlands month based.
4. The whole of Netherlands year based by property type.
5. The whole of Netherlands quarter based by property type.
6. The four largest cities year based.
7. The four largest cities quarter based.

The four largest cities of the Netherlands are Amsterdam, Rotterdam, 's-Gravenhage and Utrecht.

[1] http://wetten.overheid.nl/BWBR0007119/2016-10-01

4.4 Repeated sales

The repeated sales Price Index uses the transactions that occur twice on the same dwelling. The basic technique that uses this is the SP/Case-Shiller [Case and Shiller, 2006], therefore also called (CS). The Case-Shiller method forms the basis again for other indices like the House Price Index (HPI) used by the Federal Housing Finance Agency (FHFA) in the USA [Calhoun, 1996], [Bailey et al., 1963].

The Case-Shiller technique starts by filtering all the transactions based on the following criteria:

1. A transaction must be arms-length.
2. A property must have two recorded transactions.
3. A transaction must take place 6 months after the previous transaction.

An arms-length transaction indicates that both the buyer and the seller act in their best economic interest when agreeing upon a price. The transactions from the Kadaster indicates regular, auction and forced auction types and also family transactions and leasehold. Only regular transactions are considered arms-length as auction prices vary from normal sale prices. Family and leasehold are also excluded from the calculations.

After the filtering, weights are applied to the sale pairs that have a large interim time. Default the weight is between zero and one. Sale pairs with a interim time of more than 10 years will get a weight of 0.8. Because of the weighted part in the construction the index is called a Weighted Repeated Sale (WRS) index. The next step is the construction of the matrices and the calculation part Subsection 4.4.1.

4.4.1 Matrix construction

The representation of a sale pair is different when the initial sale occurs in the base period. This is due to the fact that the base index is 100%. Therefore two equations are possible Equation 4.4 and Equation 4.5.

$$\frac{I_t}{I_0} = \frac{200}{195} \Rightarrow \log\left(I_t\right) = \log\left(100 \cdot \frac{200}{195}\right) \tag{4.4}$$

$$\frac{I_{t_2}}{I_{t_1}} = \frac{200}{195} \Rightarrow \log\left(I_{t_2}\right) - \log\left(I_{t_1}\right) = \log\left(100 \cdot \frac{200}{195}\right) \tag{4.5}$$

The construction of the index is based on Least Squares Regression Equation 4.6. This is the closed form solution that minimizes the Residual Sum of Squares. For this we need a matrix X and a vector Y. Matrix X has a sale pair on each row while each column represents a index period. The values of X represent the left side of the equations Equation 4.4 and Equation 4.5. At the specified index column a 1 is placed (mind the minus sign!). The Y vector represents the other side of the equations. An example is listed in [Silverstein et al., 2014].

$$\hat{\beta} = (X'X)^{-1}X'\hat{y} \tag{4.6}$$

The regression results in a vector $\hat{\beta}$ with the length of the amount of index periods. To calculate the percentage changes the power function $f(x) = e^x$ can be used. The base entry (where the index is 100%,) of the index is on the first index this can be easily changed by the formula $f(I_t) = I_t/(I_b/100)$. Where I_t is the entry in the vector and I_b is the new base entry.

4.5 Experiment

In this section the goal of the experiment Subsection 4.5.1 is explained and the setup of the different configurations in Subsection 4.5.2.

4.5.1 Goals of the experiment

The goal of this experiment is to find a good reliable Price Index that can be used as an alternative for the SPAR index. This relates to the research questions:

- Is it possible to create a Price Index that behaves the same or better as the government issued one.
- What is the difference of a specific Price Index in comparison to a general Price Index.
- What is the added value of constructed price index vs already available indices for house price prediction.

4.5.2 Experiment design

For each the following price indices: Average, Square meter and Repeated Sales, the following combination of configurations are generated: (Netherlands, Region, Municipality, City and Neighborhood) with (Year, Quarter and Month). This resulting in 90 different price indices.

For the largest cities in the Netherlands (Amsterdam, Rotterdam, Den Haag and Utrecht) the SPAR index can be used to evaluate the calculated index. For smaller cities there is a change that there is not enough data to compute a good index.

4.5.3 Evaluation

The Price Index must be evaluated to check if it gives a good indication of the price changing over time in the real estate market. This is achieved by extracting 20% of the sale pairs from creation. These will be used after construction to measure the MAA between the real second price of the pair and the calculated price. The MAA is calculated over the Netherlands and the four largest cities for the Repeated sales and SPAR index.

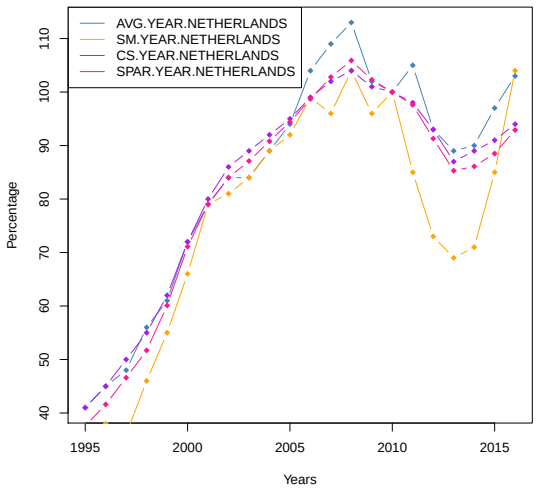
4.6 Results

In this section the different price indices are plotted against each other to indicate differences. In 4.1a the Netherlands year based indices are compared. During observation the AVG and SM index are respective up and below the SPAR index, therefore an average of both is computed and plotted next to SPAR in 4.1b called AVGSM.

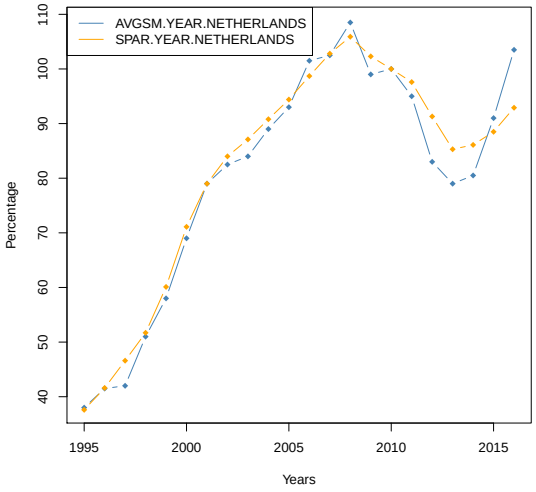
The created indices are also computed on smaller time intervals in order to see their behavior. In 4.2a the AVGSM, CS and SPAR indices are shown with a quarter time interval. It is clearly visible that the AVGSM index fluctuates to much. Hence in 4.2b only the CS and SPAR index are plotted with a monthly time interval. These two indices follow a similar trend.

Next the difference between the Price Index for different property types is plotted. The CS and SPAR index for property types Terrace (4.3a) and Detached houses (4.3b). Where the terrace property type index tends to behave similar between CS and SPAR. While for the detached versions there is more difference.

The four largest municipalities also give a good indication of the usability and adaptability of the Price Index. For the CS and SPAR index each of the four municipalities are plotted against each other in Figure 4.4. For these municipalities also a validation is performed, the results are listed in Table 4.1. In the graphs it is not visible which Price Index has the best accuracy, but in the table the CS index has the highest accuracy. The municipality graphs (Figure 4.4) show a slightly

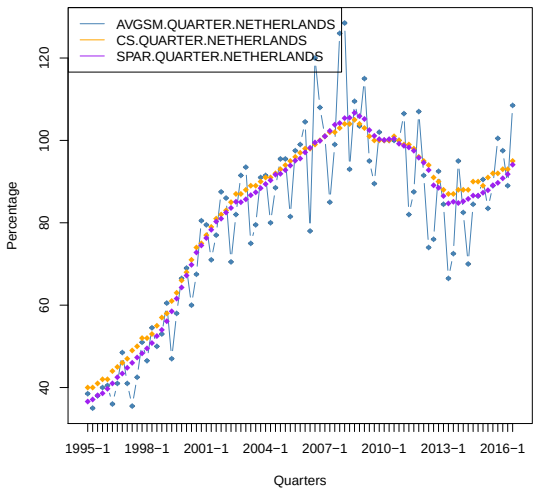


(a) All indices (year based) in one graph

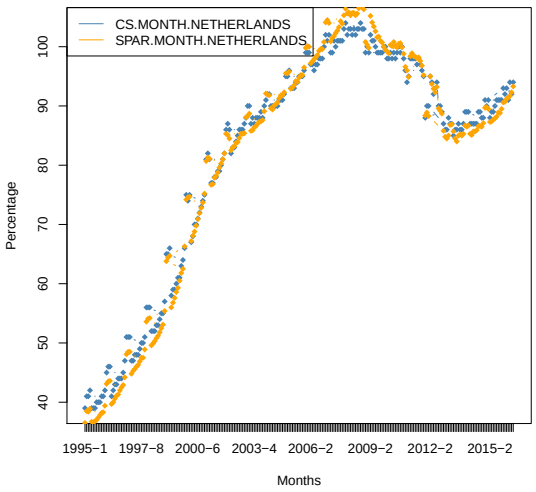


(b) Average from AVG and SM compared with SPAR

Figure 4.1: Difference between year based Price Index

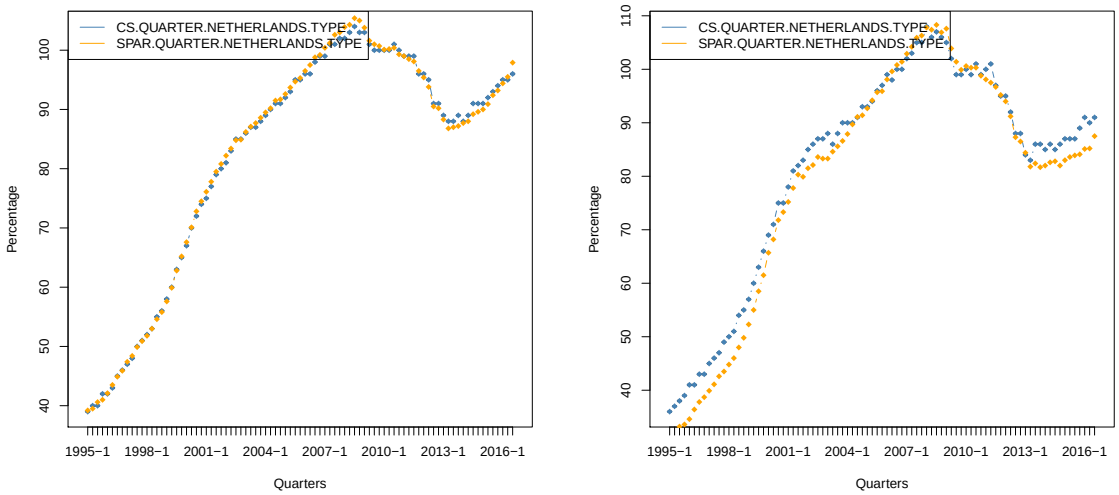


(a) Comparison between AVGSM, CS and SPAR price indices, montly based.



(b) Comparison between CS and SPAR index, montly based.

Figure 4.2: Difference between quarter and month based Price Index.



(a) Comparison between the CS and SPAR Price Index where the property type is Terraced house.

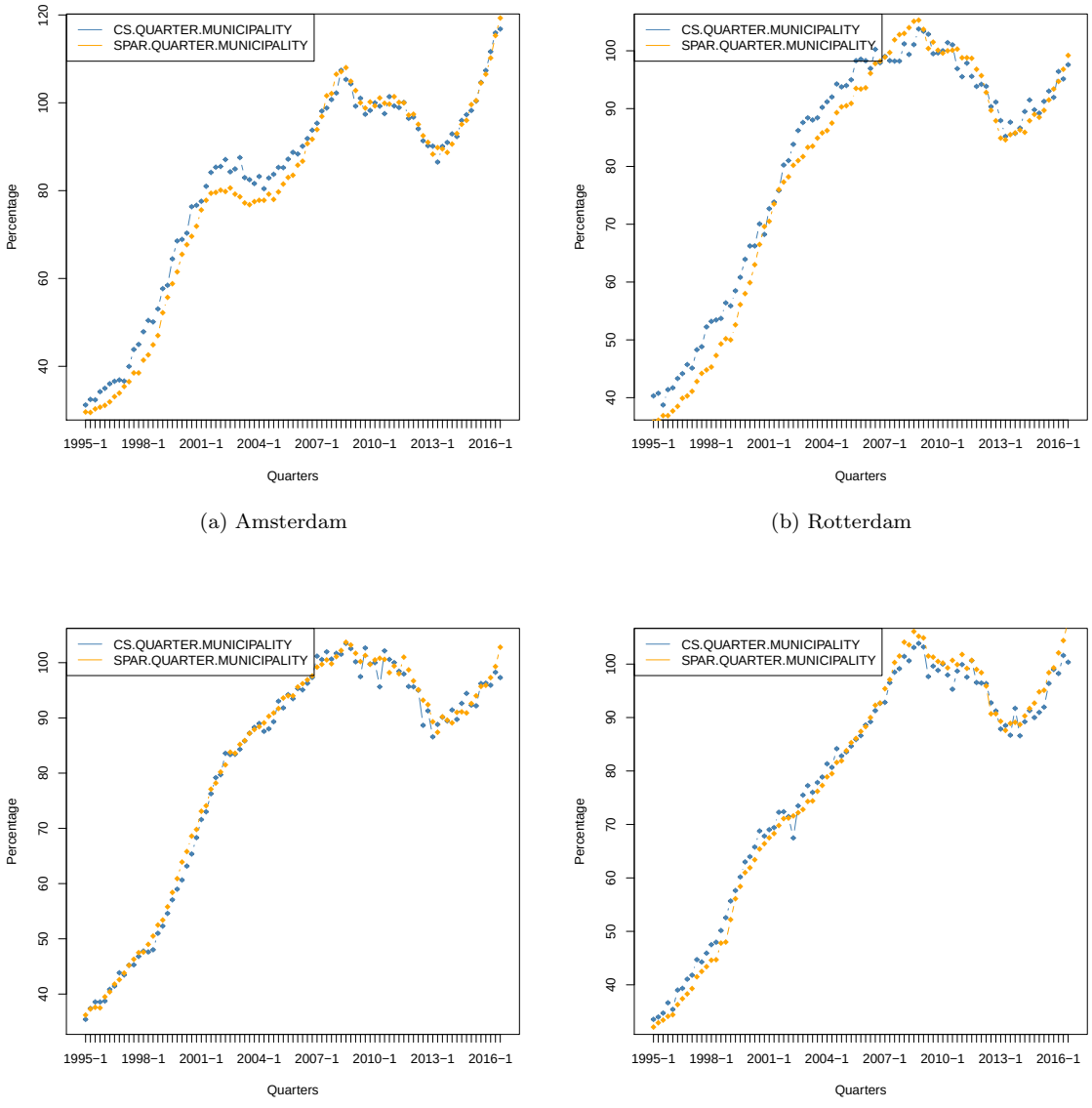
(b) Comparison between the CS and SPAR Price Index where the property type Detached house.

Figure 4.3: Difference between CS and SPAR Price Index with different properties.

different curve for each city. This has to do with the different types of property that exist in each municipality as listed in Table 3.2.

Table 4.1: Validation results of the four largest municipalities, all price indices have a quarter time interval.

Municipality	Amount	CS NL	CS Municipality	SPAR NL	SPAR Municipality
Amsterdam	43292	76.80	**80.20**	75.35	75.31
Rotterdam	49182	77.78	**77.80**	75.64	73.63
Utrecht	28853	82.54	**84.16**	81.37	80.09
's-Gravenhage	59668	77.87	**78.25**	77.33	76.85



(a) Amsterdam

(b) Rotterdam

(c) 's-Gravenhage

(d) Utrecht

Figure 4.4: Difference between CS and SPAR Price Index the four largest cities.

4.7 Conclusions

In the results section are some interesting observations visible. The Average index with time interval year gives a good indication of the housing market. Although around the years 2008 and 2012 it fluctuates a bit more. However the quarter and month case are useless because of the amount of fluctuations. Therefore we can conclude that the Average Price Index is not a real Price Index as also mentioned in [van der Wal and Wiebe, 2008].

In the year overview the SM Price Index Figure 4.1 indicates a much lower index during the financial crisis. This is explainable due to the fact that only smaller houses were sold in that period. The combination of the average and square meter Price Index is a better measurement as one of them alone. Because both measure an average of what is sold, however the fluctuations increase when the amount decreases.

In Table 4.1 is visible that the repeated sales index will result in a more accurate price than using the SPAR index. The repeated sales municipality Price Index will also preform better than the SPAR municipality Price Index.

In the graphs for the property types Figure 4.3 there is an interesting observation, the Terraced house index is closer to the SPAR while the Detached house index has some larger difference on a few points. From the data set statistics in Table 3.1 it is visible that most of the houses are of the type Terraced house and least of the houses are Detached. Also we can add that the average price of all the Terraced houses is the lowest and the average price of the Detached houses is the highest. Another interesting point is the surface Detached houses have by far the largest surface while Terraced houses have the lowest amount. Because the data set has no indication when or if a property is rebuilt we can conclude the following. The differences in the Detached houses price index relates to the average price and average surface of the detached houses. The SPAR index uses the WOZ to correct for renovations while repeated sales does not.

Chapter 5

Engineering Real Estate Characteristics

Real Estate characteristics are features identifying a real estate. These features are important in order to increase the accuracy of the prediction, therefore the goal of this chapter is:

- We want to gather more Real Estate characteristics in order to have more correlated Real Estate features which should improve the prediction.

These features can be location related as the neighborhood Subsection 3.1.3, about the insides of the real estate or the surroundings of the property Section 5.1. Feature selection is used to calculate the best identifiers. Different Feature selection algorithms are listed in Section 5.2. The experiments that are conducted are listed in Section 5.3 with the corresponding results in Section 5.4 and conclusions Section 5.5.

5.1 Funda attributes

For extra real estate attributes the popular real estate website Funda is spidered. Only the sold properties are collected because these can be matched with the transaction entities. The downside of the website is that they keep their sold properties only online for a year. This means we only have properties that have been sold since sept-2015.

The data from Funda includes different types of information about the real estate, but also about the garden and the garage. The full list of properties is visible in Table A.1. These are the kind of properties that are related to the price according to [Sirmans et al., 2005]

Not all of the properties are always available an overview is listed in Table A.1. Prediction algorithms do not work well on sparse data, therefore a technique is required to handle this missing data Subsection 5.1.1.

A large part of the values in the Funda data set are text based, meaning they can have different variations of a word list as the result of an input option list. These text properties are not useful for machine learning, therefore some translations are preformed to generate extra combinations or extract features Appendix B.

5.1.1 Missing data

Missing data can be categorized into three categories [Batista and Monard, 2003]:

1. Missing Completely At Random (MCAR)

2. Missing At Random (MAR)
3. Not Missing At Random (NMAR)

The real estate attributes are spread out over all of these categories. For example an apartment does not contain a garden in 99% of the cases, while a normal (not rental) house has no monthly costs to an owners association. These cases are in the category NMAR. The category MAR is for example with apartments or houses that have no bathroom listed, because every apartment or house has a bathroom. MCAR cases are all non explainable empty values and therefore hard to account for. These missing data properties issues can be solved in different ways.

1. Ignore or discard data.
2. Parameter estimation.
3. Imputation.

Each solution as well as the applicability on the Neighborhood and Funda data set is explained in detail in the subsections below.

Ignore or discard data

Ignore or discard the data that has missing properties, essentially means that it is not used in calculations. This can be both row and column based [Batista and Monard, 2003]. Meaning that incomplete entries can be ignored but also a feature that is only present in a small amount of entries. In case of very sparse data set this leads to a very small amount of usable data. In case of dense data set this technique will have a good effect on the data because all the entries are complete.

In the case of the Neighborhood data, which is very sparse, removing is not an option because this removes parts of the Netherlands. Therefore for this data set it is not possible to ignore or discard data.

In the case of the Funda data set where there are not a lot of entries and all the categories of missing data occur multiple solutions are needed. Discarding is the best solution for the MCAR cases, entries that have a large amount of missing properties, because the credibility of the entry can not be guaranteed.

Parameter estimation

Parameter estimation, fills the unknown fields by using a prediction algorihtm based on the known values. In [Grzymala-Busse and Hu, 2000] the C4.5 algorithm, also used for the generation of a decision tree, is used. According to this paper this provides a good solution to missing items in comparison with other imputation strategies. This problem can also be solved by using a Maximum likelyhood estimator as presented in [Dempster et al., 1977].

For the Funda data set this would be applicable for the entries that are in the MAR and NMAR category.

Imputation

There are different imputation techniques to fill up the missing values [Batista and Monard, 2003]:

1. Avg, take the mean value of all the known values.
2. Hot and Cold deck, divide the value into two cases.
3. Min, take the minimum value of all the known values.

4. Zero, substitute all empty values with 0.

5. Random, take a random between the min and max of all the known values.

Imputation on the Funda data set is usefull for the category NMAR where the min value of the data set can be used as a baseline. For example when a toilet amount is missing for a data entry, this is always at least one, which is the minimum that can be imputed. Using min in the MAR case is also applicable and its a lightweight solution. For the neighborhood data set using the minimum or zero are both applicable because when a statistic is not present it will zero or between zero and the minimum value.

5.2 Feature selection

Using all the property attributes (around 120) would be insufficient and lead to the Curse of dimensionality and overfitting of the model. Therefore Feature Selection is applied on the data to find the most depending properties. However it can be the case that the feature selection solution is too large for computation as stated in [Liu and Motoda, 2007]. Feature selection on houses is already researched in other countries, Turkey [Selim, 2009] and Singapore [Fan et al., 2006].

The easy but computational intensive method is Subset Evaluation Subsection 5.2.1. The Pearson correlation coefficient Subsection 5.2.2, a statistical approach. A regression approach is to use the Least Absolute Shrinkage and Selection Operator (LASSO) Subsection 5.2.3. Another machine learning approach is a decision tree Subsection 5.2.4.

5.2.1 Subset Evaluation

Subset Evaluation can be done by using all possible subsets and compute the error on the validation set for each of them [Kira and Rendell, 1992]. It starts with a set containing only one feature. For each set a model is trained with training data. The error is computed against validation data. Validation data is different form training data. For each subset of features the error is computed. The subset with the lowest error contains the properties that are best suited according to the training and validation set.

Additions to this algorithm that are a less computational intensive are Backward elimination and Forward selection presented in [John et al., 1994]. Backward elimination starts with the full set and eliminates one features so that the resulting error will decrease. If the error does not decrease the optimal set of features is found. The Forward selection algorithm starts with the empty set and adds features while the error decreases.

5.2.2 Pearson Correlation Coefficient

The Pearson Correlation Coefficient [Benesty et al., 2009] is the measure of the linear correlation (dependence) between two variables X and Y. The general formula is represented by Equation 5.1, where X and Y are populations. The coefficient is the relation between the Covariance of the two random variables and the Standard deviation multiplied.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sqrt{E[X^2]-[E[X]]^2}\sqrt{E[Y^2]-[E[Y]]^2}} \tag{5.1}$$

In calculations the sample version of the formula Equation 5.2 is used. The Coefficient results in a number in the range $[-1, 1]$, where 1 is positive linear correlation, 0 is no linear correlation and -1 is a total negative correlation.

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{5.2}$$

This technique is also used by other data scientists for feature selection in [Tsymbal et al., 2005] and [Guyon and Elisseeff, 2003].

5.2.3 The Lasso

The lasso method presented in [Tibshirani, 1996] and [Liu and Motoda, 2007] is a regression based feature selection method. This addition of this method is that it sets coefficients that are less than a constant to 0, which eliminates the features. Let $\hat{\beta} = (\hat{\beta}_i, \ldots, \hat{\beta}_p)^T$, the lasso estimate is defined by Equation 5.3. In Equation 5.3 the absolute signs around β_j make that the coefficients are set to 0.

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_1 - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \sum_j |\beta_j| \leq t \tag{5.3}$$

A drawbacks of the Lasso algorithm is that it can only select as much features as there are data entries [Liu and Motoda, 2007]. However this is not an issue in our case where we only have 200 features and more data entries. The Lasso algorithm is used in all kinds of areas to define relevant features [Zhou et al., 2010].

5.2.4 Decision trees

Decision trees can be used for price prediction (Random forests), however also for finding relevant features. Each of the nodes in the tree represents a choice of a variable. When given an input a tree walk is done from the top node to a leaf, where in each node a choice is made depending on the input. All the leaves contain results. The decision tree relates to feature selection because the algorithm uses only features that best relate to the output.

Decision trees can be build by different algorithms, well known are the Iterative Dichotomiser 3 (ID3) [Quinlan, 1986] and the successor C4.5, presented in[Salzberg, 1994]. In [Kira and Rendell, 1992] the ID3 algorithm is used for feature selection and compared with other methods. A domain related study is the of [Fan et al., 2006] where housing data of Singapore is used to find which features are relevant to the price.

5.3 Experiment

In Section 5.2 three feature selection methods are presented. With these feature selection methods experiments are conducted. In Subsection 5.3.1 the goals are listed and the design of the experiments in Subsection 5.3.2.

5.3.1 Goals of the experiment

The feature selection experiments need to indicate which of the features are a good representation of the price. This needs to be calculated for the extended data set with 117 features and about 300.000 entries and also the standard data set which only contains about 10 features and 4 million entries. This needs to result in 4 sets of features for Apartments and Houses with the Extended and Default data set. Secondary goal is to indicate if there is a difference between different price indices and imputation strategies.

The main research question we need to answer in this experiment is the following:

- Is it possible to gather a large amount of correlated real estate features?

Other questions related to the main research question are:

- Is there a large difference in the correlation of features?
- Are all of the gathered real estate features correlated?

5.3.2 Experiment design

For each of the above methods an experiment is conducted. For apartments and houses each with 2 different price indices, CS and SPAR. These indices are quarter based for each property for the Netherlands as region.

The Pearson Correlation Coefficient (PCC) method is calculated on the extended data set and not on the default set. This is because the correlation of one feature does not depend on one of the other features. The next two experiments will use the PCC features as an input for a subset evaluation. The forward selection algorithm will start with the 40 highest correlated features and selects the features that decrease the residual sum of squares. Backward selection will start with all non zero features and removes features that decrease the residual sum of squares. Next the Lasso algorithm is used to find the best subset of features from both data sets. Where the data set with small features will likely list all features. The last method that is used is the Decision tree, this experiment is conducted on the full data set only. The last experiment differs from the rest because it will indicate if there are differences between the property types for the house features. The Lasso algorithm is used on the full data set that is split up into 4 parts.

5.3.3 Experiment validation

The experiments all use different methods, therefore we want to verify the outcome of the experiments by evaluating them with each other. One could have a few features more or less but the base must be the same. Where the base are the top correlated features. It is not possible that a high correlated feature at method x is not a correlated feature at method y. The differences must be in the low correlated features.

5.4 Results

In table Table 5.1 all the results of the experiments are summed up. For each method indicating the amount of features that was found relevant and the used price index for both apartments and houses.

Table 5.1: Feature selection results separated by Apartment and House for all techniques with CS and SPAR price index.

Method	Price Index	#Features	
		Apartment	*House*
PCC	CS	84	82
PCC	SPAR	84	82
PCC and Forward selection	CS	45	48
PCC and Forward selection	SPAR	46	48
PCC and Backward elimination	CS	84	82
PCC and Backward elimination	SPAR	84	82
LASSO	CS	86	84
LASSO	SPAR	86	84
Decision tree	CS	84	82
Decision tree	SPAR	84	82

The last experiment was to indicate if the different type of houses are supported by the same

features. The results of this experiment are displayed in Table 5.2.

Table 5.2: The amount of correlated features for each house property type.

Type	#Features
D (Detached)	84
T (Terraced)	85
C (Corner)	84
SD (Semi-Detached)	84

5.5 Conclusions

The first observation from Table 5.1 is that there is one small difference between the SPAR and CS Price Index, one feature is not correlated while using the another price index. This is because the variable has a low correlation in SPAR and the entries supporting this correlation have an other calculated price. The other correlations all have less difference between the two methods.

Another interesting observation is that there is a large difference between PCC with Forward Selection and the other algorithms. This is explainable because the regression applied by the Forward selection algorithm has a limited amount of iterations making it not very accurate. The reason for the low accuracy of the regression is the amount of iterations. The regression has a small amount of iterations because of the large amount of data. Using only a partial data set with more iterations on regression would yield a better result. The reason for using a small amount of iterations is to minimize the computations.

Not visible in Table 5.1 but important to mention is that the specific types of apartments or houses are not correlated in any case. These types are translated from the Funda type feature in Appendix B (15 and 16). There are a few reasons that could cause this. The first is that the feature is split into two parts and therefore is not correlated anymore. Another reasons is that the values are not reliable.

The differences between apartment and house listed in Table 5.1 show a difference of two features. More specific for apartment the following properties are correlated: The floor where the apartment is located, Surface of extra storage and Costs for the Owners Association. While the Lot size is only correlated at house. The Costs for the owners association are not correlated with houses because a house owner does not have to pay to the Owners Association. The extra storage of an apartment is listed as specific type, storage units of houses are listed under garage storage. The lot size of a house is not present at apartments because owner has no ownership over the floor in the apartment, this has to do with regulations.

The floor where the apartment is located is at first not required at a house, however the results of the last experiment (Table 5.2) that count the features between the different property types indicate that it is required. Some Terrace houses are split up into two, ground floor and first floor as separate house. Therefore the floor location is relevant.

Chapter 6

Real Estate valuation

Real Estate valuation is the process of developing a value (price) for a certain real estate. The real estate agent (valuator) determines his value mostly on comparable real estate in the neighborhood and accounts for differences between them. These differences can be the size of the property but mostly it will be the conditions of certain elements of the property. In this chapter we want to predict the value of a dwelling with the same technique as a real estate agent but instead of visiting a house only the characteristics are used. The goal of this chapter therefore is:

- We want to simulate the work of a human real estate agent in order to have a baseline to compare against the regression / ensemble models.

In Section 6.1 the K-nearest neighbor algorithm is explained. The experiments are listed in Section 6.2 with the corresponding results in Section 6.3. The conclusions can be found in Section 6.4.

6.1 K-nearest neighbor

Nearest neighbor can be used to predict a result of an object by comparing the difference with other objects where the output is known. In terms of Real Estate, the query is the house which value is yet to be determined and the known items are houses in the neighborhood that are recently sold. In [Bourassa et al., 2010] K-nearest is also used to predict house prices although they implemented additions. To measure the nearest house a distance algorithm is used.

The distance is calculated by measuring the difference between all the characteristics of the query and the characteristics of all the items in the neighborhood. This can be calculated in different ways see Subsection 6.1.1. The k in the nearest neighbor algorithm is also an important factor because it determines the amount of neighbors that is collected for the average. Therefore if the value is too small it could be to specific or give a to high or too low result. The implemented algorithm has some additions over the simple K-Nearest algorithm (Subsection 6.1.2).

6.1.1 Distance

The distance between two real estates can be seen as the distance between two points in multiple dimensions. For this problem are many solutions already, for example the Euclidean distance Equation 6.1 and the Manhattan distance Equation 6.2. In both equations the p is the known house and q is the query real estate, D is the set of properties.

$$distance(p, q) = \sqrt{\sum_{i=0}^{D}(p_i - q_i)^2} \tag{6.1}$$

The Euclidean distance takes the difference of all properties and squares it, after that it takes the square root of the result.

$$distance(p, q) = \sum_{i=0}^{D} |p_i - q_1| \tag{6.2}$$

The Manhattan distance takes the sum of the positive difference of all the properties.

6.1.2 Additions

As an addition to the default average a weight is introduced that increases properties with a smaller distance. This weight is applied on the prices of the properties so small distance properties will have a larger impact on the result price. The weight is calculated by the formula: $w_i = 1/d_i^2$, where w_i is the weight of the property and d_i is the distance from the property to the query. The total price is then calculated by Equation 6.3. Where k is the amount of near transactions, w_{total} is the sum of weights and p_i is the price of the near distance found transaction.

$$\hat{p} = \sum_{i=0}^{k} p_i / w_{total} \tag{6.3}$$

The other improvement is that only properties from the same neighborhood, city, municipality or region are matched for distance. This can have positive or negative effects. The positive effect can be that because the location-based properties of the real estate are the same, the nearest distance real estate is very accurate maybe even the same property located in somewhere in the same street. The negative effect can be that because the range is limited similar real estates are not found. For example a large Mansion exists only once in a certain region. However this technique is strongly related to the work of the real estate agent because he also applies weighing[1].

6.2 Experiment

The goal of this experiment is to measure the error of the prediction as it was done by a real estate agent. More clearly we try to full fill the following goal.

- What is the accuracy of a simulated Real Estate agent prediction with the full data set of features.

6.2.1 Experiment design

We randomly choose an amount of previous transactions from the database to use as query objects. First experiment is with 1000 entries of data while the second takes 10000 query objects. To check if the weight addition makes it more accurate both experiments will be ran with and without weighting.

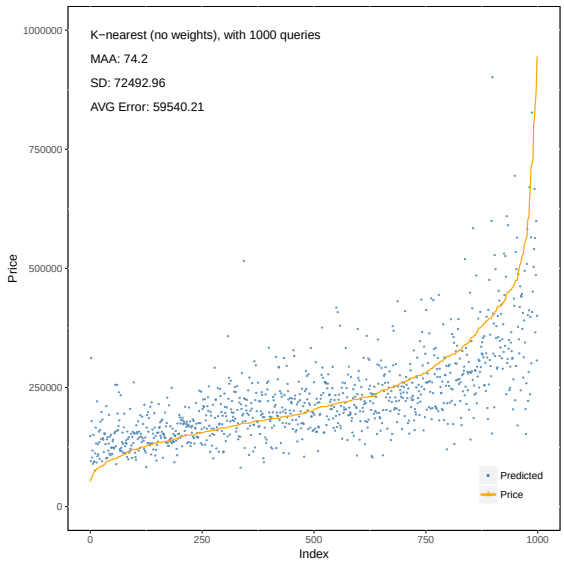
6.3 Results

The results of the two experiments are listed in Table 6.1. For all experiments a graph is drawn in Figure 6.1. The MAA in the 10,000 sample case is listed as 0.00 because MAA is a value between 0 and 100. The real values are negative. This can also be seen in the correlation graph Figure 6.2, as some houses have an error as high as the actual price.

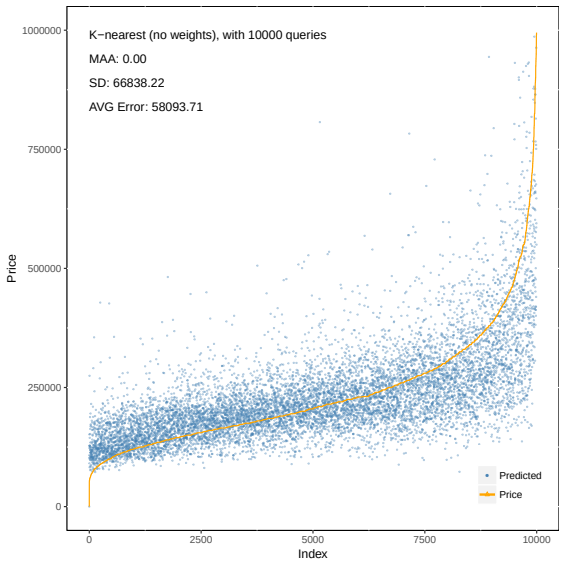
[1] https://site.nwwi.nl/paginas-nwwi-website/hoe-werkt-het-nwwi-2/

Table 6.1: Results of K-nearest neighbor prediction with different sample size and weights.

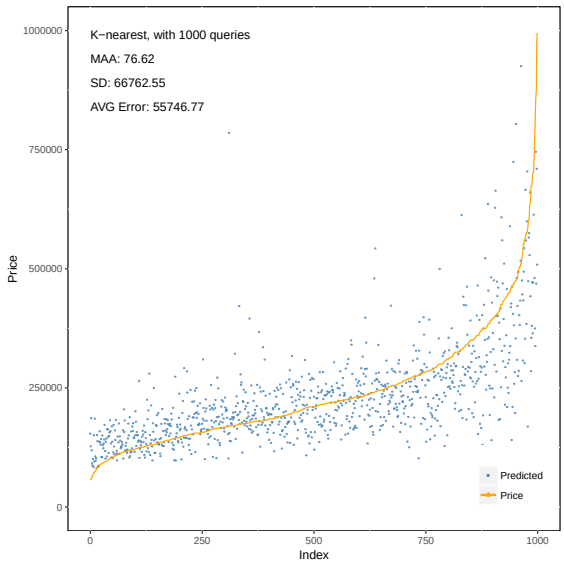
Samples	Apply weights	Average Error	Standard deviation	MAA
1000	No	59540.21	72492.96	74.20
10000	No	58093.71	66838.22	0.00
1000	Yes	55746.77	66762.55	76.62
10000	Yes	60521.92	71523.97	0.00



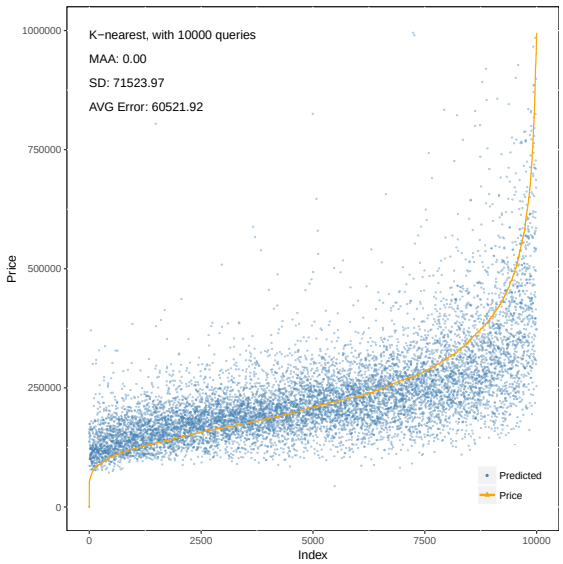
(a) Not weighted 1000 samples.



(b) Not weighted 10000 samples.



(c) Weighted 1000 samples.



(d) Weighted 10000 samples.

Figure 6.1: The results of 1000 and 10000 samples, with and without weighting.

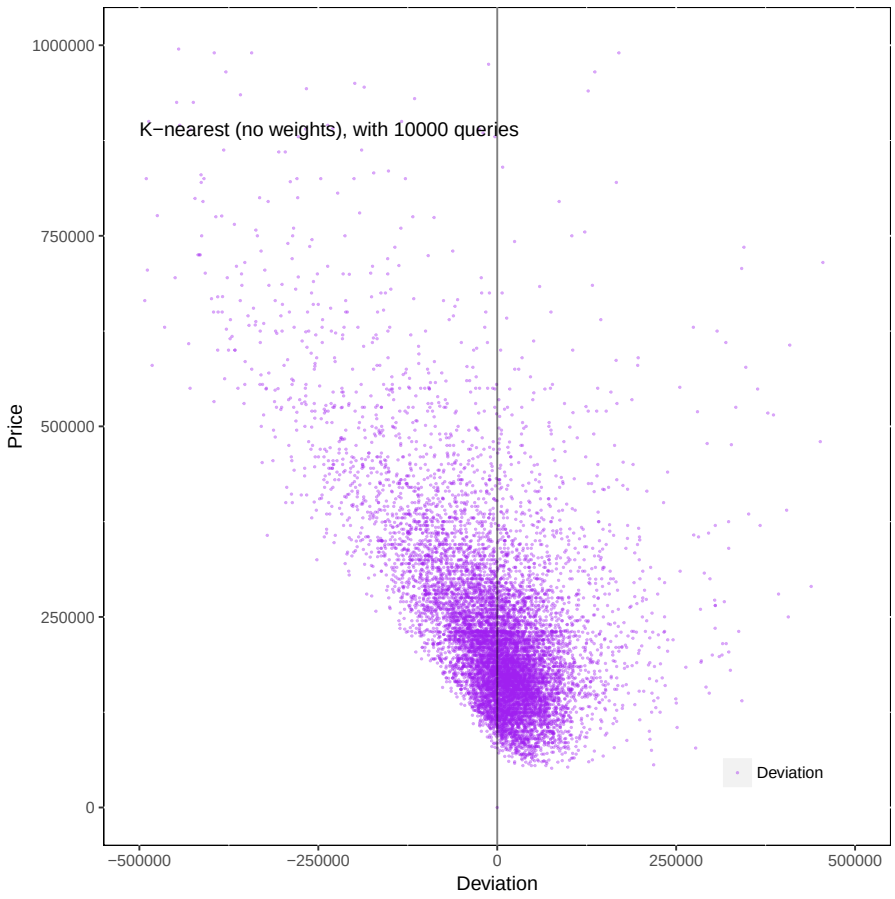


Figure 6.2: Deviation graph of the K-nearest neighbor prediction without weights, with 10000 samples.

6.4 Conclusion

From the result in Table 6.1 it is visible that the k-nearest prediction with the extended data set is not accurate. The average error and standard deviation are both too high. The main reason for this is maintenance, the extended data set contains no values for maintenance. Where the exact same dwelling that is renovated can have a large difference in transaction price to the other that is not renovated.

Another reason for this result is the equality in weighting of individual variables. For the algorithm all variables have the same weight. A difference in lot size has the same relation to the price as having a sauna or not.

The graphs in Figure 6.1 give an indication that the output is somewhat tending to be the average price, because in the low section the prediction is up while in the high section the prediction is under. Also in the middle segment it tends to do the best of all (around 250,000). There is not much difference between the normal and weighted solution. However the small case of the weighted solution has better result but this could also come due to the random selection of the testing data.

Chapter 7

House Price prediction

This chapter forms the basics for machine learning prediction part. The goal of this chapter is to predict the price of a house based on its characteristics. More specifically:

- We want to predict a price of a Real Estate with a regression model.

In Section 7.1 the regression algorithm is explained. Cross validation which is used in the experiments is explained in Section 7.2. The experiments are listed in Section 7.3 with the corresponding results in Section 7.4. The conclusions are listed in Section 7.5.

7.1 Regression

Linear regression is a statistical approach for modeling input variables X with an output variable y. Where the simplest version is just a linear equation as in (7.1) called linear regression.

$$p(x) = a_0 + a_1 \cdot x + \epsilon \tag{7.1}$$

Linear regression can represent more than one variable as input but all the variables are linear correlated. This is in line with the Hedonic Price Theory where the assumption is that the price of a house is an aggregation of individual components or attributes [Griliches, 2013] as also used in [Limsombunchao, 2004]

7.1.1 Ridge regression

Ridge regression also known as Tikhonov regularization, invented by Andrey Tikhonov in [Tikhonov, 1966] solves the linear regression problem differently as the least squares method in order to reduce overfitting.

This is done by adding an extra weight based on a parameter, often called l2-parameter or α. If α is zero then it is the same as linear regression. The process is called L2 regularization and adds a matrix to the equation to add smoothness. This matrix is formed by αI. Where α is the parameter and the I is the identity matrix.

In section Subsection 4.4.1 Least Squares Regression Equation 4.6 is used to solve the equation. Because Computing the inverse of the $(X'X)$ has a complexity of $O(D^3)$ it can take very long. Another approach can be used that is less computational intensive is Gradient descent.

The goal of the gradient descent algorithm is to find the minimal error between the fitted line and the solution. A minimum of a line $f(x)$ can be found by solving the equation $f'(x) = 0$, computing the gradient of the function. Gradient descent preforms a number of iterations with a step size.

Each iteration all of the weights (coefficients) are updated to minimize the error Equation 7.2. Where w_j is the selected weight, η is the step size, h_j is the feature corresponding to w_j and the last part is the gradient.

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + 2\eta \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(w^{(t)})) \tag{7.2}$$

7.2 Cross validation

In order to create a model with low training error, Cross validation is used. Cross validation is a method that splits the data into subsets for training and validation.

There are two types of Cross validation, Exhaustive and Non-exhaustive. For the best result the Exhaustive one is obviously better but this takes more time and computation power. The Non-exhaustive ones are not that intensive and give an average to good result. K-fold cross validation is the basic cross validation type where the k, a number between 1 and N, depends how exhaustive the algorithm is. The total data set is split into k equal parts. The training set contains $k-1$ parts and the validation set the last part. Therefore the algorithm runs k times so that every part becomes a validation set.

In data mining and machine learning $k = 10$ is mostly used [Refaeilzadeh et al., 2009]. If $k = N$ then the algorithm is called Leave-one-out cross validation (LOOCV). The estimate obtained is almost unbaised but has high variance [Refaeilzadeh et al., 2009]. This technique is mostly used on small data sets.

7.3 Experiment

In this section the experiments are presented with the desired goal in Subsection 7.3.2, the design in Subsection 7.3.1 and the validation in Subsection 7.3.3.

7.3.1 Goals of the experiment

The goal of the following experiments is to test different setting with different sub sets to have a calculated model with high accuracy. The accuracy can be measured by different measures. For this the measurement object Subsection 3.2.2 from the framework is used. The research goals are the following:

- What is the accuracy of a regression model trained on the default features set compared to the accuracy of a regression model that is trained on the extended feature set.
- What is the accuracy of the result of a combination of models that are trained on specific property types, according to one (general) model for the same instances.

7.3.2 Experiment design

All experiments use default 10-fold cross validation. Because the feature selection shows no real differences between the CS and SPAR price index Section 5.5, we will only use CS in this experiment. The 10-fold corss validation selects the best model based on the RSS. The RSS gives a good indication of the training error because all are squared en summed up, large errors will produce high RSS, while low errors produce only little.

Different models will be trained in order to compare these. A general model and one for each property type will be calculated with all the data of the Netherlands on both of the data sets, this

will results in 12 different models.

Next the four largest municipalities in the Netherlands (Amsterdam, Rotterdam, 's-Gravenhage and Utrecht) are used with the same configuration with both sets, with or without property type distinction. This also results in 12 different models per city. In total this produces 60 different models for 10 datasets.

7.3.3 Training, Validation and Testing

To make sure the model is preforming well it needs to be validated and tested. Three sets of data are used for this, a training, validation and test set. The training set is used to train the model, while the validation set is used on the trained model to calculate the error and select the best model, if multiple are calculated. The test set is then used to measure the accuracy of the selected model. If only one model is trained with the training data then a train and test set will suffice.

The error in training, validation and testing can be measured by the Residual Sum of Squares (RSS) Subsection 3.2.2. Where p_i is the desired output of item i and $\hat{p}_i$ is the predicted output of item i. The RSS is a measure that can be used to check whenever a setting preforms better than another. By squaring all the error terms the RSS is always possible and increases quadratic. In the results the quality measure MAA will be used to indicate the accuracy of the model.

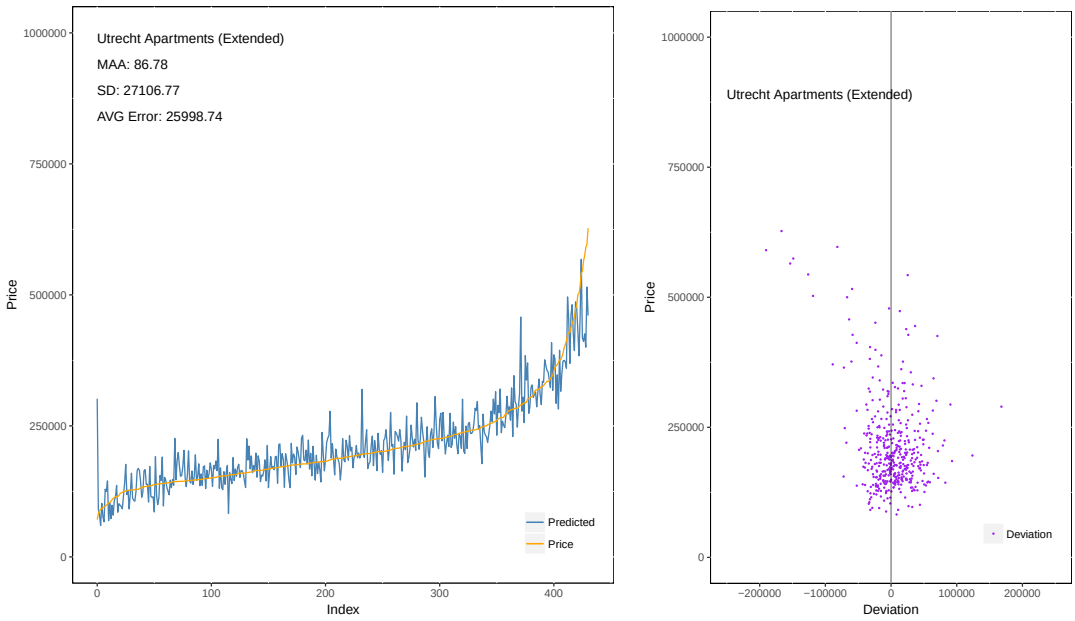
7.4 Results

All the different model configurations are visible in Table 7.1. The lowest standard deviation is 23,780 ('s Hertogenbosch Semi Detached Extended data set) while the average is around 85,000. The model with the lowest mean at 25,998 is Utrecht Apartment Extended data set, where the average is around 83,000.

Table 7.1: Feature selection results with Apartments (A), Corner (C), Semi-Detached (SD), Terraced (T), Detached (D), all these results combined in Combined and a total model in All.

Region	All	A	C	SD	T	D	Combined
The Netherlands (Extended)	78.45	78.31	73.60	80.93	85.96	72.30	83.80
The Netherlands (Default)	00.00	68.68	81.36	74.91	80.58	67.85	74.33
Amsterdam (Extended)	84.37	84.85	79.10	78.68	90.60	60.20	85.27
Amsterdam (Default)	63.44	65.46	74.18	52.07	72.82	55.07	66.64
's-Gravenhage (Extended)	83.36	83.96	78.47	88.92	85.06	83.89	84.04
's-Gravenhage (Default)	60.66	65.94	71.75	75.80	68.93	61.79	66.63
Utrecht (Extended)	86.33	80.67	82.14	67.14	71.25	62.87	87.68
Utrecht (Default)	71.90	75.96	75.12	71.34	76.56	50.63	76.03
Rotterdam (Extended)	81.89	83.15	81.39	78.61	83.59	58.04	83.17
Rotterdam (Default)	61.94	66.26	72.70	59.83	73.74	52.80	68.21
Other (Extended)	82.40	82.84	84.27	81.34	87.19	76.44	83.56
Total (Extended)	82.61	83.37	84.17	81.36	87.15	76.36	83.72

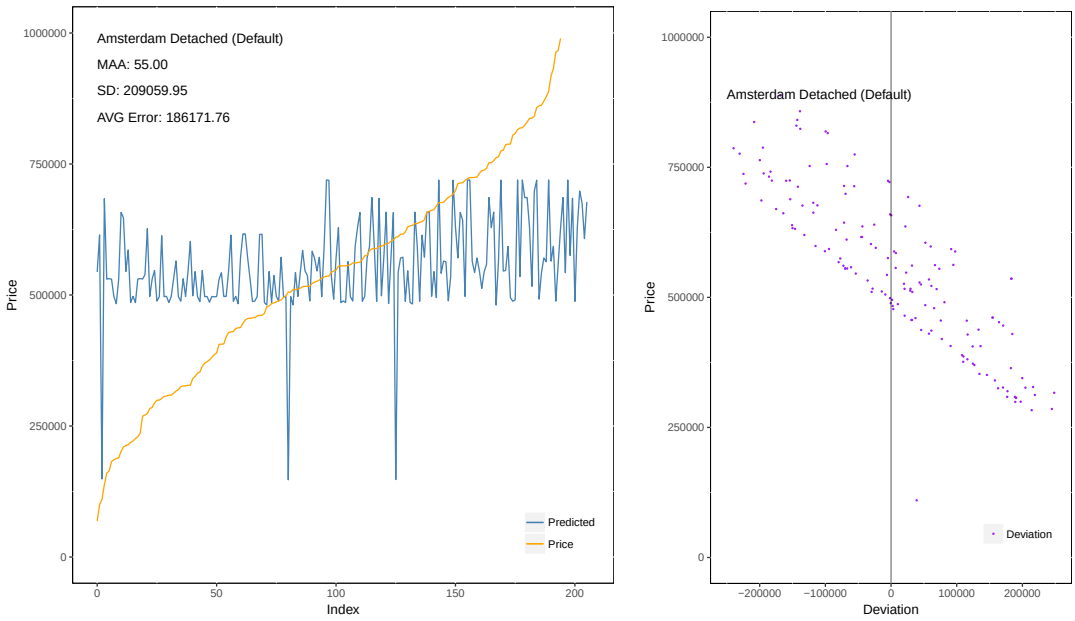
In Figure 7.1 a example from a prediction with a high accuracy is visible, while in Figure 7.2 an example with low accuracy is visible. Both images consist of a prediction graph and a deviation graph. The example (Figure 7.1) with high accuracy is a decently fitted model with average error and standard deviation that is desired. While the example with low accuracy (Figure 7.2) has high average error and standard deviation. In Table 7.1 is visible that the total accuracy from all the municipality models with different property types is 83.72 this combination of models has an average error of 40750.39 with standard deviation of 48581.22.



(a) Prediction graph, output vs. predicted output.

(b) Deviation graph, error vs. the output.

Figure 7.1: The results of prediction with a good accuracy of the extended data set over apartments in Utrecht.



(a) Prediction graph, output vs. predicted output.

(b) Deviation graph, error vs. the output.

Figure 7.2: The results of prediction with a bad accuracy of the standard data set over detached houses in Amsterdam.

7.5 Conclusions

From the results in the previous section some conclusions can be drawn. Overall the accuracy of all the models can be considered not bad, meaning that the models can predict a price. However the desired accuracy needs to be higher for real world usage. This is also visible with the average error and standard deviation, the best example (Figure 7.1) has a desirable outcome. While one of the worst solutions (Figure 7.2) only produces an average with high error. This is also visible in the deviation graphs, 7.1a and 7.2a.

However there are also some positive conclusions, in Table 7.1 it is visible that the models trained with the Extended data set mostly have a higher accuracy as the models trained with the Default data set. This concludes that the extra data fields that where added have a positive effect on the accuracy of the prediction.

From the Table 7.1 can be concluded that the property type models do not always perform better than general models. This is an interesting observation and can be related to one of the following problems. The amount of data, looking back at Table 3.1 it is visible that there are not more than 2% dwellings of the type Detached and Semi-Detached. The other issue is the location of the dwelling. In Amsterdam for example the price of an apartment can differentiate a lot, in the city the price can be twice as high as outside the city, for the same size, rooms and building year. It is very difficult for the algorithm to predict this, because the parking facilities are better outside the city but in the center everything is close by, while the neighborhoods are not that different.

Chapter 8

Neighborhood comparison

This chapter is an extension on the prediction in the previous chapter. By merging data of equivalent neighborhoods into a set, more data is available for a model. That will most likely lead to less error in prediction. Therefore the goal of this chapter is:

- Combine models in order to create a model that has an accuracy that is the same or better with a lower error than the input models.

In this chapter neighborhoods are compared to each other in order to define the equality. First an example from a neighborhood rating is discussed in Section 8.1. Next the neighborhood comparison is explained in Subsection 8.2.1. In Section 8.3 Ensemble learning is explained to merge the different prediction models. The experiments in Section 8.4 with corresponding results in Section 8.5 and conclusions in Section 8.6.

8.1 Liveability meter

The dutch government owns a website [1] that list the Liveability in a region, ranging from a 100 meter block till the size of a region. The definition of liveability is listed as a total score with 5 subscores. Where the total score is the sum over all subscores. The subscores are the categories: Houses, Residents, Facilities, Safety and Physical environment. The category scores are based on variables that are specific to that category. The values are compared against the average and based on that a positive or negative score is given. An overview image of the website Figure 8.1.

With this score it is already possible to calculate equal neighborhoods based on the categories. However a neighborhood can have sub values for these categories that contradict each other.

[1] http://leefbaarometer.nl/

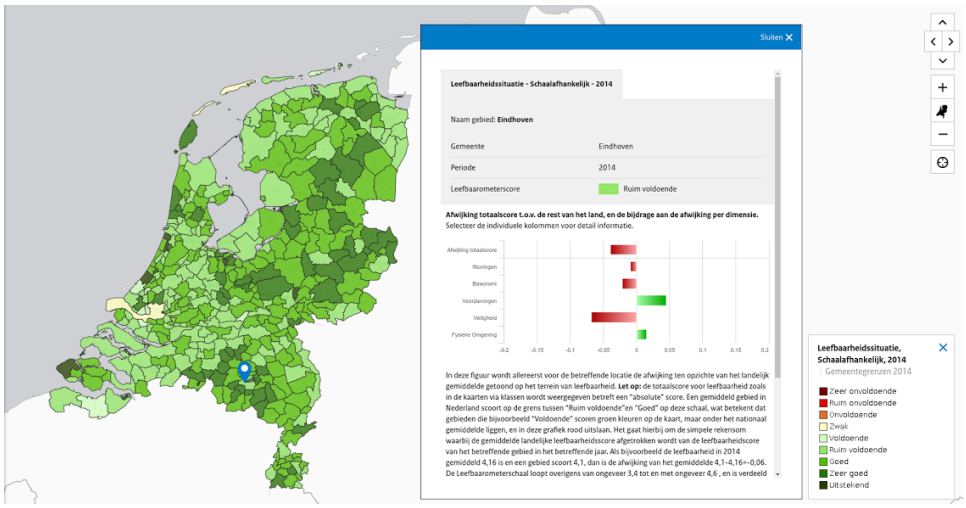


Figure 8.1: This is a screenshot of the website `http://leefbaarometer.nl/`, on the left the netherlands is displayed with different colors for liveability and in the center the municipality Eindhoven is selected with the given scores.

8.2 K-nearest neighborhood

With the inspiration of the Liveability meter the idea for the cluster neighborhood is formed. Instead of using the categories for comparison, the variables behind these categories are used. To make sure that the equality is based on the characteristics of the neighborhood and not so much on the size, only percentages are used.

These variables are based on four categories namely: Houses, Residents, Services and Physical environment. In our neighborhood data set Subsection 3.1.3 are however no safety variables present, in total 94 different variables are used.

Near neighbors can be formed in different ways, two approaches are presented. A cluster approach is used in Subsection 8.2.1 while a K-nearest approach is used in Subsection 8.2.2. Another approach used in [Ng and Deisenroth, 2015] is to use the geographically neighbors of the neighborhood. However is our case means we have to manually make this grid because the neighborhoods have no geographical attributes.

8.2.1 K-means clustering

With the K-means clustering algorithm Subsection 8.2.1, clusters are formed of similar neighborhoods. The K-means clustering algorithm, also know as the Lloyd-Forgy algorithm [Forgy, 1965], splits a set S with D dimensions into k partitions so that it minimizes the distance between the items in the cluster. More formal explained in Equation 8.1.

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} |x - \mu_i|^2 \tag{8.1}$$

The downside of the k-means algorithm is that the amount of clusters k is an input parameter. Therefore the amount of clusters must be known before calculation.

8.2.2 K-nearest neighborhood

The K-nearest neighbor algorithm presented in Chapter 6 can also be used to find the nearest neighborhoods based on the neighborhood characteristics presented in the previous section. Instead of a weighted price the adjusted algorithm returns a list of near neighborhoods. For each neighborhood the data is collected in order to form a model starting with the nearest neighborhood.

In comparison with the clustering this algorithm always delivers ten near neighbors in order of distance. Thresholds and limits on the data are used to make sure that a neighborhood with large data has a small influence from others.

8.3 Ensemble learning

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions [Dietterich, 2000a]. The most simple one is if the models are split on property type and when an input is given only one of the models calculates it. However in this case the models are not combined to predict the output together. With ensemble learning it is possible to merge the models so that the output is based on all of them.

There are mainly three different types of ensemble methods: Bagging Subsection 8.3.1, Boosting Subsection 8.3.2 and Stacking Subsection 8.3.3.

8.3.1 Bagging

Bagging stands for Bootstrap Aggregation and is used to combine simple predictors to increase prediction and decrease overfitting and variance [Breiman, 1996], [Dietterich, 2000b]. Statistics are used to construct a new set of data. The training set is split up into k new training sets with size n'. These training sets are known bootstrap samples, with $(1 - 1/e)$ of unique items, the rest are duplicates.

Testing occurs on all the k models that are trained and the solutions are averaged into a linear solution. Therefore the drawback of this technique is that it does not work on linear regression models. Bagging relies on the instability of the prediction model. Neural networks and decision trees are claimed instable models [Breiman, 1996].

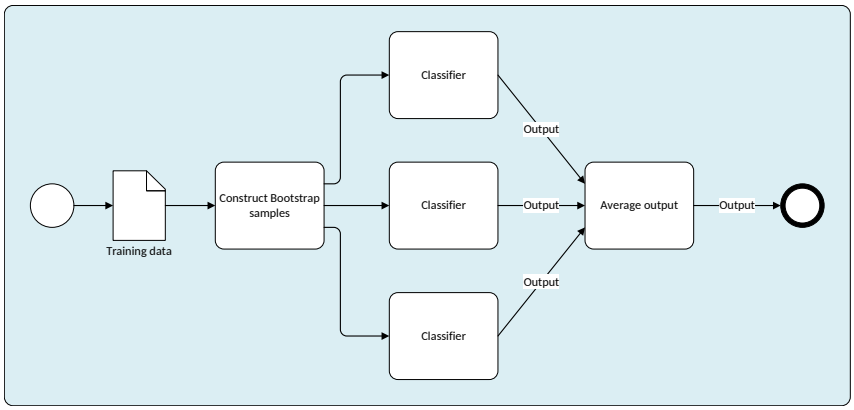


Figure 8.2: Global overview of the stacking algorithm.

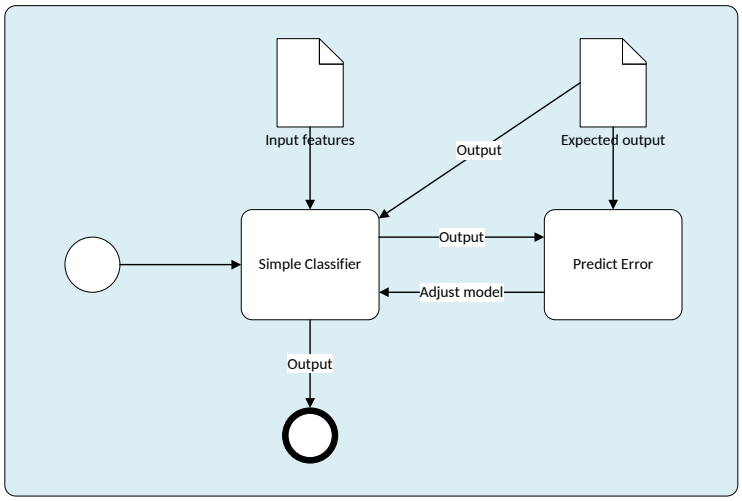


Figure 8.3: Global overview of the boosting algorithm.

8.3.2 Boosting

The Boosting ensemble method is an iterative procedure that starts with a simple model. In each iteration the goal is to predict the errors from the previous iteration. Resulting in a new model that is combined with the previous model. This step is very similar to the gradient descent weights update. An overview of this process is visible in Figure 8.3

Boosting is most likely been used in combination with decision trees [Freund et al., 1999], [Drucker and Cortes, 1995] and [Dietterich, 2000b], but it is also possible to use with linear regression models.

8.3.3 Stacking

Stacking, also called Stacking generalization, introduced by David Wolpert [Wolpert, 1992] works by deducing the biases of the generalizers. It uses the output of all level-0 models as input features for the level-1 model Figure 8.4, inspiration from the image is taken from [Chio, 2013].

The stacking technique can be used to combine different types of models with different inputs, because only the output is used to construct the level-1 model. The level-1 model is also trained on the outputs of the level-0 model training data with the desired results [Chio, 2013].
In Figure 8.4 it is not visible but it is also possible to create more levels. A collection of level-1 models can be used to create a level-2 model. This will decrease the bias even more but also needs a lot more computation time for creation, testing and resulting calculations, because of the increase in models.

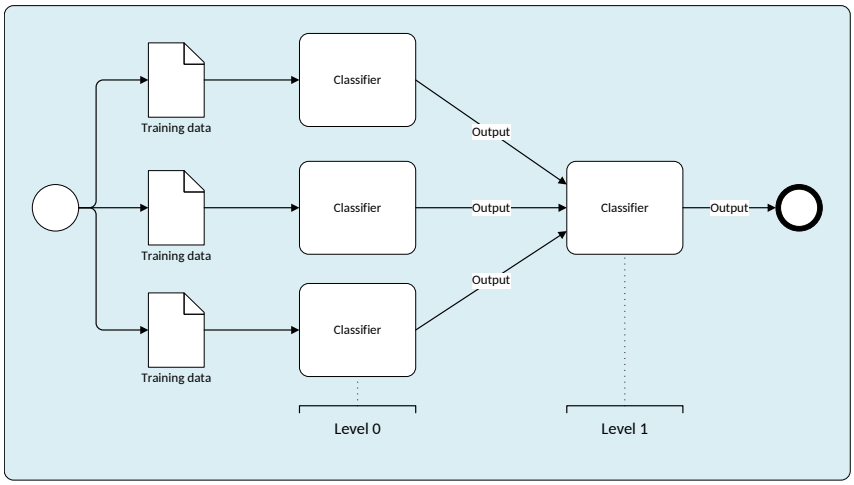


Figure 8.4: Global overview of the stacking algorithm.

8.4 Experiments

In this section the experiment is explained. First the goals are set in Subsection 8.4.1 and secondly the setup is explained in Subsection 8.4.2.

8.4.1 Goals of the experiment

The goal of this experiment is to have a multi model solution for the Netherlands that is more accurate than a single model. Therefore the following research goals are declared:

- What is the best combination of level-0 models to create a level-1 model.
- What are the best additional settings for a good model.
- What is the difference between the Cluster and K-near methods for prediction neighborhoods.

8.4.2 Experiment design

The experiment is split up into three types. First of all the neighborhood clusters are formed by the K-means algorithm. Secondly the models are constructed for these neighborhood clusters. The last step is testing and merging the models according to their own, municipality and country scores. Also a variability of extra thresholds and limits is used, in order to prefect the scores of the resulting models.

These settings consist of using only Extended models instead of using Extended models and Standard models. Using general models, these are models that are trained with data from all property types. There is a threshold on the amount of data that is necessary to train a model, where there is with the k-near neighborhood algorithm also a limit on the amount. The last threshold that is introduced is on the accuracy of the stacked model. This threshold is adjustable and can be used in order to make sure all resulting models have a descent test accuracy. If this threshold is not satisfied the model is not used and the test data will be tested on the next level model.

Neighborhood clusters

In Subsection 8.2.1 is listed how the neighborhood clusters are constructed. However the k need to be given as an input. Because neighborhoods normally exist under municipalities the goal is to

use the same amount of neighborhood clusters as municipalities as a starting point, which is 400. However this result gives 9 clusters with only one neighborhood.

Now two problems need to be taken care of. The first one is that all the neighborhoods need at least one neighbor. The second one is that outliers need to be in a separate cluster. Constructing a ton of neighborhood clusters increases the complexity in the next phase. Therefore is chosen to start with $k = 100$ to $k = 600$ with a range of 50, resulting in 11 different sets of neighborhood clusters and in total 3850 neighborhood clusters.

Model construction

The neighborhood cluster model construction first combines the data from the selected neighborhoods. The next step is to construct a model and this is done in two ways. The Default and the extended data sets are both used to construct this model. If there is enough data available the data is also split into different property types. Enough data can be declared as at least 50 testing data points which relates to a set of 500 entries, using 405 for each training model with $k = 10$.

This produces an amount of different models, there exist 5 different property types so that counts for 6 different models. Combine that with the two data sets and 12 is the result for a certain region. This region can be The Netherlands, a municipality or a neighborhood cluster. In total there are about 400 municipalities and 3850 neighborhood clusters in the Netherlands so therefore at most 51012 models are constructed. For the k-nearest it is also possible to construct models before merging, however the settings can change the cluster. Therefore is chosen to construct these models on the go.

Ensemble models

Stacking generalization (Subsection 8.3.3) is used with at most ten level-0 models to construct a level-1 model for that region. These ten models consist of the Netherlands model trained on one property type, the municipality model trained on one property type, municipality for all property types and the cluster with type and without, where all the models are doubled one standard and one extended.

If there is not enough data to construct all ten models a strategy is used to create a stacked model with less models. If the data for a neighborhood is not large enough or there is no near or cluster model for the neighborhood. The neighborhood will be calculated by the municipality model. The same technique is used for the municipality, however the data of the municipality is then tested on the Netherlands type model only.

8.4.3 Experiment validation

The initial validation of single models is done by the K-fold cross validation algorithm Section 7.2 for the individual and combined models. For each resulting configuration the level-1 model measurements are summed up to form a global measurement of the Netherlands.

Merging of the measurements is done by taking all of the predicted outputs and actual outputs, and create a new measurement with these values. It is also possible to get a weighted average, on large scale this will yield to the same result.

8.5 Results

The results in this section are only a small portion of all the experiments conducted, in total 98 different configurations are tested and a full overview is listed in Table C.1. The model with a

training size of at least 10 was the best in each configuration. In Table 8.1 different configurations are listed as well as the differences between clustering of near neighborhoods.

Table 8.1: Different model combination results, Standard means the use of models trained by the Default data set, General indicates the use of models that are trained on more than a single property type, Neighborhood indicates the neighborhood model that is used and Size is the training size threshold. The results are MAA, Standard deviation (SD) and Average error (AVG).

Standard	General	Neighborhoods	Size	Limit	MAA	SD	AVG
No	No	No	10	No	83.9524	42985.8517	38166.9043
No	Yes	No	10	No	84.1509	**42848.3141**	37894.4038
Yes	Yes	No	10	No	**84.4202**	43137.4871	**37408.6900**
No	No	Clusters	10	No	85.2896	42595.0938	35679.3006
Yes	No	Clusters	10	No	85.4613	43019.5542	**35431.6170**
No	Yes	Clusters	10	No	85.2940	**42358.8488**	35650.7027
Yes	Yes	Clusters	10	No	**85.4799**	43820.3669	35573.7599
No	No	Near	10	200	**85.6596**	**40495.5795**	**34674.9503**
Yes	No	Near	10	200	85.4883	40988.9831	35083.1138
No	Yes	Near	10	200	85.5973	40684.3877	34888.8916
Yes	Yes	Near	10	200	85.5742	40884.0557	34904.4516

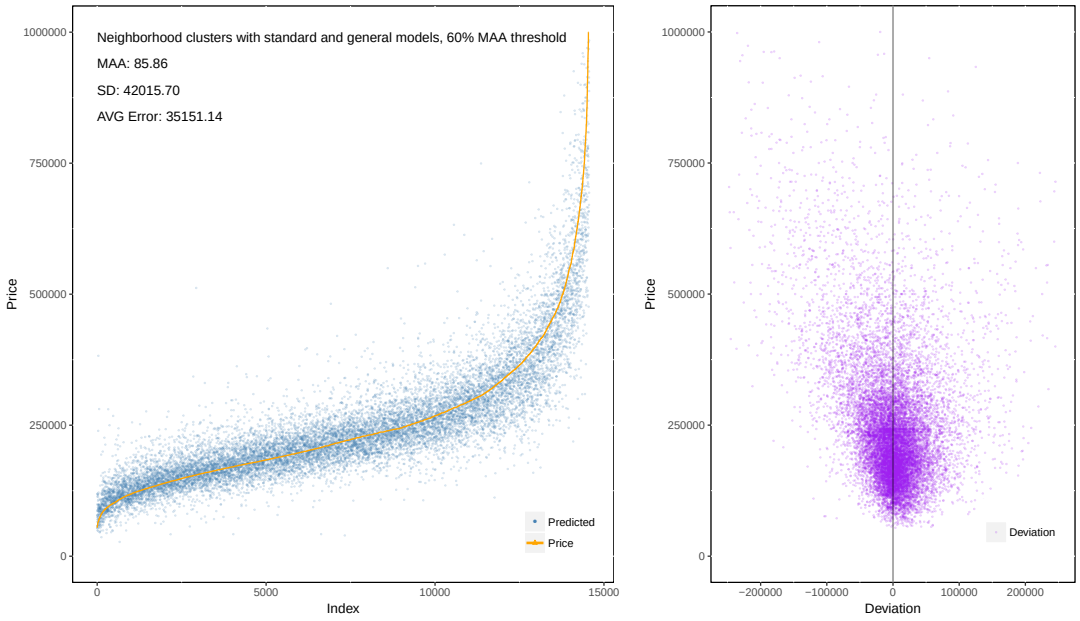
In Table 8.1 is visible that for the Cluster solution the configurations with standard models behaves quite well, therefore extra experiments are done with this setup and a threshold on the MAA of the resulting test model Table 8.2

Table 8.2: Different model combination results with cluster models where a threshold is set on the test measurement of the regional based stacked model, column Threshold. The General column indicates the use of models that are trained on more then a single property type. The results are MAA, Standard deviation (SD) and Average error (AVG).

General	Threshold	MAA	SD	AVG
No	50	85.6499	42597.6392	35497.3258
Yes	50	85.6837	42540.6339	35541.3933
No	60	85.7189	41752.9240	35306.3561
Yes	60	**85.8641**	42015.7016	35151.1389
No	70	85.5618	39715.2392	34663.6030
Yes	70	85.7819	39761.3969	34625.2615
No	75	85.5828	**38847.9478**	34251.7806
Yes	75	85.6727	39068.0736	34308.1806
No	80	85.4895	39767.7458	34452.1210
Yes	80	85.6077	39869.5534	**34248.9103**
No	85	85.2825	40945.3049	34939.9470
Yes	85	85.3089	41662.5891	34980.0963

The cluster results indicate that the model with a combination of general models and standard models and with a threshold MAA of 60% has the best resulting MAA. In Figure 8.5 the results of this configuration are visible.

In Table 8.1 is visible that for the Near solution the configuration without standard and general models behaves the best. Second best is the same configuration with general models. Therefore extra experiments are done without standard models and a threshold on the MAA of the resulting test model. The refinements are listed in Table 8.2.



(a) Prediction graph, output vs. predicted output.

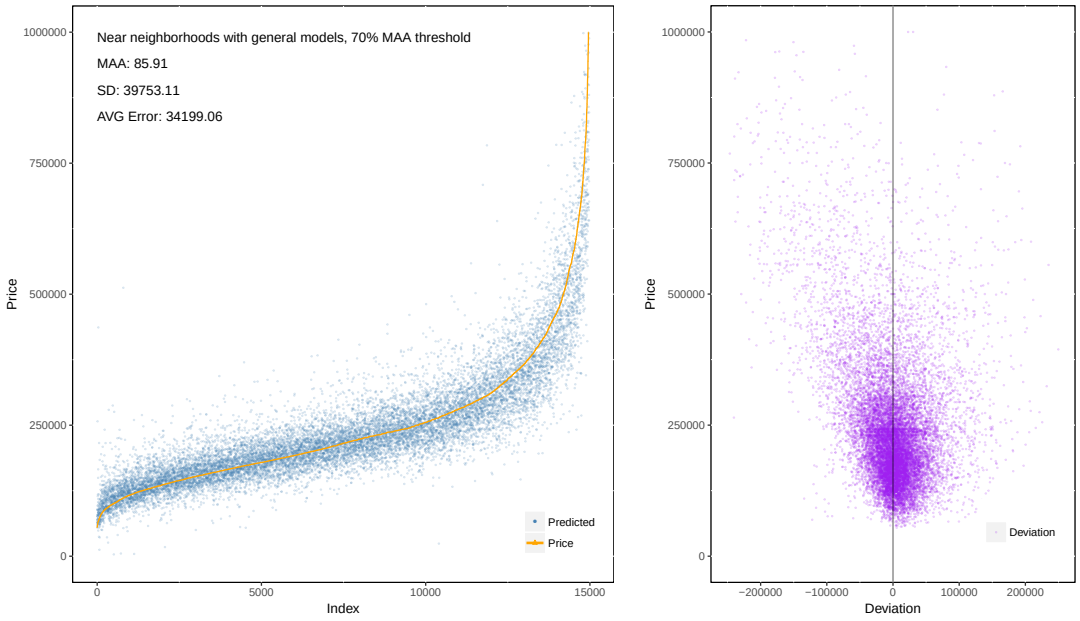
(b) Deviation graph, error vs. the output.

Figure 8.5: The model that has the highest MAA with the use of clusters, general, standard models and with ensemble models of training size $x > 10$.

In Table 8.3 the refined results for the Near neighbor models indicate that the total model that uses general models with a MAA threshold of 70% gives a MAA result of 85.9082%, therefore this results is also listed in Figure 8.6.

Table 8.3: Different model combination results with near neighbor models where a MAA threshold is set on the test measurement of the regional based stacked model, column Threshold . The General column indicates the use of models that are trained on all property types. The results are MAA, Standard deviation (SD) and Average error (AVG).

General	Threshold	MAA	SD	AVG
No	60	85.8415	39843.4523	34458.5599
Yes	60	85.7246	39662.6776	34482.9729
No	70	85.6949	39583.1368	34240.0753
Yes	70	**85.9082**	39753.1068	34199.0552
No	75	85.5342	39365.9674	34127.9709
Yes	75	85.5230	**39280.6239**	34146.2178
No	80	85.4706	39359.3603	33938.7032
Yes	80	85.4648	39417.0383	**33925.6242**
No	85	85.1894	41447.9324	35155.2872
Yes	85	85.3374	41415.9439	34853.5826



(a) Prediction graph, output vs. predicted output.

(b) Deviation graph, error vs. the output.

Figure 8.6: The model that has the highest MAA with the use of near and general models, with near and ensemble models of training size $10 < x < 200$.

8.6 Conclusions

In this chapter neighborhood comparison is used to create clusters as well as near neighbors in order to have more data for model creation. The use of these models as well as standard and general models have a positive influence on the stacked result. The ensemble models without standard, general or specific neighborhood models have a MAA of 83.95, SD of 42,985.85 and AVG error 38,166.90 which is significantly better than the 83.72 MAA, SD of 48,581.22 and AVG error of $40,750.39$ in the previous section.

All of the ensemble models have a configuration that uses a minimal amount of 10 training size. The reason that this gives a better accuracy and less error comes from the case when x the size of training set is small, around the value. In this case only a small specific portion of data is used as training and one or two items for testing. The changes that there are outliers in the small specific set is not that high. When the amount of training size is lower more specific models will be generated. Therefore the total accuracy increases while the error decreases.

The approach with cluster models performs at best when ensembled with standard and general models. While the near models perform better when ensembled with only general models. The reason for this is not quite clear, it could be due to the fact that the accuracy of standard models is averagely lower than for extended models but then it should also be the case with the cluster approach.

All of the experiments do not lead to one single best solution, there is no configuration that has the best accuracy with the lowest standard deviation and error. However the near neighborhood approach tend to do slightly better as the cluster approach. The reason for this is that the near neighborhood approach is limited on training data so it will not use all of the ten near neighborhoods when training a model. For the near approach the model configuration that uses general models in combination with the near models preforms best with a MAA threshold between

70 and 80, where most of the difference comes from the standard deviation.

Chapter 9

Conclusions

The goal of this project was: Given an address in the Netherlands with additional properties, estimate the price of the real estate on that address. In more detail we would like to have the real estate agent approach and a regression approach in order to validate which one is better.

One of the goals was to create a Price Index that was reliable and usable on small regions and different time intervals. Experiments with the repeated sales method have shown that this method is a good solution to the problem. From the data set that is used it performed actually better than the Sale Price Appraisal Ratio.

The first approach for price prediction was to predict in the same way as a real estate agent without visiting the house. Therefore the K-Nearest is used to find the nearest transaction in the neighborhood. In addition to the default algorithm, weighting is applied to the output. However, with or without weighting the accuracy of this prediction is not great. At 10000 entries the accuracy was below zero and the average error and standard deviation around 60.000 and 70.000 respectively. The main reasons for this are the maintenance of the house and the equality in weighting of the variables. The maintenance will be taken into account by the real estate agent because he can give a value to it when visiting the property. The real estate agent also gives a different weight to each variable, the regression approach solves this in our case.

The second approach was to use regression for prediction the house prices. The results from the prediction which has models for different property types and municipalities resulted in an MAA of 83.72, with average error of 40.000 and standard deviation of 48.000. With ensembling models can be combined into one resulting model, each municipality model combined with the Netherlands model of the same property type resulted in an accuracy of 83.95 which at first looks not much of an improvement. However by looking at the average error and standard deviation, 38.000 and 43.000 respectively a large improvement is visible.

By using two different neighborhood clustering techniques the total accuracy of the model improved to 85.90 with an average error and standard deviation of 34.000 and 39.750 respectively. Relating this to the business side an error of 34,000 is bad on a apartment of 150,000 but good on a 500,000 detached house.

9.1 Future work

Next some future work and improvements are listed which will improve the usage of the results for business use. The first improvement is adding data that has information about the state of a house at the time of the transaction. The transaction object and extension of Funda data do not contain this information. However from the Funda website also a description of the real estate

agent is collected. This description mostly contains some indication about the state of the house. Therefore an addition would be to extend the parser of for Funda objects with a scoring mechanism on the description of the house. A good starting point will be text parsing and sentiment analysis. This improvement will improve the price indices as well as the two prediction approaches.

Secondly, split the models into three segments according to the price. This will possibly reduce the amount of outliers in the prediction. However this technique has some downsides. If the amount of data in a neighborhood for a certain type of property has to be split into 3 parts the minimal limit of neighborhood data will be about 150 transactions.

Next improvement is to add safety, geographical and historical information to the neighborhood data. All of these additions are stand alone improvements, but in general all will improve the neighborhood comparison as well as the total prediction. Safety means adding information about the safety of a neighborhood such as crime rates. With the geographical data it would be possible to do a real near neighbor on the neighborhoods, what can be seen as a another neighborhood merging method, also used in [Ng and Deisenroth, 2015]. The historical addition is to measure the neighborhood data on different times. The downside of this is that neighborhoods can be split up and merged so this is a real challenge as well as gathering all the data.

A domain related improvement is to validate the results together with a real estate agent. To serve as a total solution for real estate valuation the results and findings should be presented to various real estate agents to find out discrepancies and solve error related issues.

Bibliography

[Bailey et al., 1963] Bailey, M. J., Muth, R. F., and Nourse, H. O. (1963). A regression method for real estate price index construction. *Journal of the American Statistical Association*, 58(304):933–942. 15

[Batista and Monard, 2003] Batista, G. E. and Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533. 21, 22

[Benesty et al., 2009] Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer. 23

[Bhageloe-Datadin, 2016] Bhageloe-Datadin, R. (2016). Recovery at last. 5

[Bourassa et al., 2010] Bourassa, S., Cantoni, E., and Hoesli, M. (2010). Predicting house prices with spatial dependence: A comparison of alternative methods. *Journal of Real Estate Research*. 27

[Bourassa et al., 2006] Bourassa, S. C., Hoesli, M., and Sun, J. (2006). A simple alternative house price index method. *Journal of Housing Economics*, 15(1):80–97. 14

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140. 41

[Calhoun, 1996] Calhoun, C. A. (1996). Ofheo house price indexes: Hpi technical description. *Office of Federal Housing Enterprise Oversight*, pages 1–15. 15

[Capital Value, 2016] Capital Value (2016). An analyses of the dutch residential (investment) market 2016. 5

[Case and Shiller, 2006] Case and Shiller (2006). S&p case-shiller metro area home price indices. 15

[Chio, 2013] Chio, E. (2013). Stacking, blending and stacked generalization. `http://www.chioka.in/stacking-blending-and-stacked-generalization/`. 42

[de Haan et al., 2009] de Haan, J., van der Wal, E., and de Vries, P. (2009). The measurement of house prices: A review of the sale price appraisal ratio method. *Journal of Economic and Social Measurement*, 34(2, 3):51–86. 14

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38. 22

[Dietterich, 2000a] Dietterich, T. G. (2000a). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer. 41

[Dietterich, 2000b] Dietterich, T. G. (2000b). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157. 41, 42

[Drucker and Cortes, 1995] Drucker, H. and Cortes, C. (1995). Boosting decision trees. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, pages 479–485. MIT Press. 42

[Fan et al., 2006] Fan, G.-Z., Ong, S. E., and Koh, H. C. (2006). Determinants of house price: A decision tree approach. *Urban Studies*, 43(12):2301–2315. 23, 24

[Finland, 2017] Finland, H. S. (2017). Price index for old detached houses, 2nd quarter 2016. 13

[Forgy, 1965] Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769. 40

[Freund et al., 1999] Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612. 42

[Griliches, 2013] Griliches, Z. (2013). *Price Indexes and Quality Change: Studies in New Methods of Measurement*. Harvard University Press. 33

[Grzymala-Busse and Hu, 2000] Grzymala-Busse, J. W. and Hu, M. (2000). A comparison of several approaches to missing attribute values in data mining. In *International Conference on Rough Sets and Current Trends in Computing*, pages 378–385. Springer. 22

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182. 24

[John et al., 1994] John, G. H., Kohavi, R., Pfleger, K., et al. (1994). Irrelevant features and the subset selection problem. In *Machine learning: proceedings of the eleventh international conference*, pages 121–129. 23

[Kira and Rendell, 1992] Kira, K. and Rendell, L. A. (1992). A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256. 23, 24

[Limsombunchao, 2004] Limsombunchao, V. (2004). House price prediction: hedonic price model vs. artificial neural network. 33

[Liu and Motoda, 2007] Liu, H. and Motoda, H. (2007). *Computational methods of feature selection*. CRC Press. 23, 24

[Ng and Deisenroth, 2015] Ng, A. and Deisenroth, M. (2015). Machine learning for a london housing price prediction mobile application. 40, 50

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106. 24

[Refaeilzadeh et al., 2009] Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer. 34

[Salzberg, 1994] Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240. 24

[Schekkerman, 2004] Schekkerman, C. (2004). Nauwkeurigheid in taxaties. Master's thesis, Amsterdam School of Real Estate. 1

[Selim, 2009] Selim, H. (2009). Determinants of house prices in turkey: Hedonic regression versus artificial neural network. *Expert Systems with Applications*, 36(2):2843–2852. 23

[Silverstein et al., 2014] Silverstein, J. M. et al. (2014). House price indexes: Methodology and revisions. *Research Rap Special Report*, (Jun). 15

[Sirmans et al., 2005] Sirmans, S., Macpherson, D., and Zietz, E. (2005). The composition of hedonic pricing models. *Journal of real estate literature*, 13(1):1–44. 21

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288. 24

[Tikhonov, 1966] Tikhonov, A. N. (1966). On the stability of the functional optimization problem. *USSR Computational Mathematics and Mathematical Physics*, 6(4):28–33. 33

[Tsymbal et al., 2005] Tsymbal, A., Pechenizkiy, M., and Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information fusion*, 6(1):83–98. 24

[van der Wal and Wiebe, 2008] van der Wal, E. and Wiebe, T. (2008). Waarom de gemiddelde koopsom geen huizenrpijsindicator is. 13, 20

[Vrieselaar et al., 2017] Vrieselaar, N., Bhageloe-Datadin, R., Groenewegen, J., Hoving, L., and Oevering, F. (2017). Scarcity expected to slow down growth of dutch housing market. 5

[Vrijdag, 2016] Vrijdag, K. (2016). Auction Price Prediction. Master's thesis, Eindhoven University of Technology. 1, 5, 6

[Wolpert, 1992] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259. 42

[Zhou et al., 2010] Zhou, Y., Jin, R., and Hoi, S. (2010). Exclusive lasso for multi-task feature selection. In *International conference on artificial intelligence and statistics*, pages 988–995. 24

Appendix A

Funda data

Table A.1: Funda properties with amount and percentage for apartments and houses.

Property	Amount present	Amount present %	Apartment amount	Apartment amount %	House amount	House amount %
Id	237369	100.00	72582	100.00	164787	100.00
Type	237369	100.00	72582	100.00	164787	100.00
Link	237369	100.00	72582	100.00	164787	100.00
ImageLink	237369	100.00	72582	100.00	164787	100.00
Address	237369	100.00	72582	100.00	164787	100.00
Afmelddatum	237369	100.00	72582	100.00	164787	100.00
StartVerkoop	237369	100.00	72582	100.00	164787	100.00
Badkamers Badkamer	164713	69.39	50283	69.28	114430	69.44
Badkamers Toilet	164713	69.39	50283	69.28	114430	69.44
Kamers Kamers	237367	100.00	72582	100.00	164785	100.00
Kamers Slaapkamers	237367	100.00	72582	100.00	164785	100.00
Woonlagen	237367	100.00	72582	100.00	164785	100.00
Achtertuin Totaal	133864	56.39	7177	9.89	126687	76.88
Achtertuin Diep	133864	56.39	7177	9.89	126687	76.88
Achtertuin Breed	133864	56.39	7177	9.89	126687	76.88
OtherAfmetingen Totaal	2	0.00	0	0.00	2	0.00
OtherAfmetingen Diep	2	0.00	0	0.00	2	0.00
OtherAfmetingen Breed	2	0.00	0	0.00	2	0.00
Badkamervoorzieningen	164411	69.26	50142	69.08	114269	69.34
Balkon	75330	31.74	47732	65.76	27598	16.75
Bedrijfsruimte Oppervlakte	318	0.13	6	0.01	312	0.19
Bedrijfsruimte inpandig	318	0.13	6	0.01	312	0.19
Bedrijfsruimte Mogelijk	318	0.13	6	0.01	312	0.19
VVE	49400	20.81	49400	68.06	0	0.00
Bouwjaar	237366	100.00	72582	100.00	164784	100.00
Bouwnorm	237369	100.00	72582	100.00	164787	100.00
Ketel Jaar	149075	62.80	36642	50.48	112433	68.23
Ketel Merk	149075	62.80	36642	50.48	112433	68.23
Ketel Eigendom	149075	62.80	36642	50.48	112433	68.23
Ketel Aardwarmte	149075	62.80	36642	50.48	112433	68.23
Ketel Combi	149075	62.80	36642	50.48	112433	68.23
Ketel Gas	149075	62.80	36642	50.48	112433	68.23
CapaciteitVoertuigen	2	0.00	0	0.00	2	0.00
Eigendomssituatie Einddatum	7288	3.07	5700	7.85	1588	0.96

Eigendomssituatie Bedrag	7288	3.07	5700	7.85	1588	0.96
Eigendomssituatie Afkopen	7288	3.07	5700	7.85	1588	0.96
Eigendomssituatie Afgekocht	7288	3.07	5700	7.85	1588	0.96
Eigendomssituatie Eigendom	7288	3.07	5700	7.85	1588	0.96
Energielabel	13134	5.53	7077	9.75	6057	3.68
RuimteExtern	156886	66.09	46420	63.96	110466	67.04
Garage	100060	42.15	17974	24.76	82086	49.81
RuimteGebouwGebonden	93052	39.20	52943	72.94	40109	24.34
GelgenOp	68352	28.80	68352	94.17	0	0.00
Inhoud	237292	99.97	72576	99.99	164716	99.96
Isolatie	201890	85.05	59799	82.39	142091	86.23
Kantoor Oppervlakte	329	0.14	9	0.01	320	0.19
Kantoor inpandig	329	0.14	9	0.01	320	0.19
Kantoor Mogelijk	329	0.14	9	0.01	320	0.19
Keurmerken	15596	6.57	5368	7.40	10228	6.21
VraagprijsLaatste Price	236644	99.69	72451	99.82	164193	99.64
VraagprijsLaatste KostenKoper	236644	99.69	72451	99.82	164193	99.64
Lasten Einddatum	6730	2.84	5129	7.07	1601	0.97
Lasten Bedrag	6730	2.84	5129	7.07	1601	0.97
Lasten Afkopen	6730	2.84	5129	7.07	1601	0.97
Lasten Afgekocht	6730	2.84	5129	7.07	1601	0.97
Lasten Eigendom	6730	2.84	5129	7.07	1601	0.97
Ligging	208663	87.91	64215	88.47	144448	87.66
LiggingTuin	137022	57.73	10720	14.77	126302	76.65
Looptijd	237369	100.00	72582	100.00	164787	100.00
Huurprijs Price	38	0.02	17	0.02	21	0.01
Huurprijs ServiceKosten	38	0.02	17	0.02	21	0.01
VraagprijsStart Price	51063	21.51	13583	18.71	37480	22.74
VraagprijsStart KostenKoper	51063	21.51	13583	18.71	37480	22.74
Oppervlakte Deelperceel	159301	67.11	360	0.50	158941	96.45
RuimteOverigInpandig	84769	35.71	13271	18.28	71498	43.39
PatioAtrium Totaal	1379	0.58	278	0.38	1101	0.67
PatioAtrium Diep	1379	0.58	278	0.38	1101	0.67
PatioAtrium Breed	1379	0.58	278	0.38	1101	0.67
Perceeloppervlakte	161117	67.88	0	0.00	161117	97.77
Plaats Totaal	975	0.41	207	0.29	768	0.47
Plaats Diep	975	0.41	207	0.29	768	0.47
Plaats Breed	975	0.41	207	0.29	768	0.47
Praktijkruimte Oppervlakte	886	0.37	16	0.02	870	0.53
Praktijkruimte inpandig	886	0.37	16	0.02	870	0.53
Praktijkruimte Mogelijk	886	0.37	16	0.02	870	0.53
Prijs Aanvraag	828	0.35	320	0.44	508	0.31
Schuur	164176	69.16	53099	73.16	111077	67.41
Soort Appartement	72582	30.58	72582	100.00	0	0.00
Soort Huis	164785	69.42	0	0.00	164785	100.00
Dak DakSoort	155375	65.46	37222	51.28	118153	71.70
Dak DakBedekking	155375	65.46	37222	51.28	118153	71.70
Parkeergelegenheid	131032	55.20	45060	62.08	85972	52.17
Specifiek	28278	11.91	10162	14.00	18116	10.99
Status	237369	100.00	72582	100.00	164787	100.00
Tuin	171675	72.32	13978	19.26	157697	95.70
Veilingdatum	0	0.00	0	0.00	0	0.00
Verkoopdatum	237369	100.00	72582	100.00	164787	100.00
Verwarming	224460	94.56	68809	94.80	155651	94.46

Voortuin Totaal	2698	1.14	470	0.65	2228	1.35
Voortuin Diep	2698	1.14	470	0.65	2228	1.35
Voortuin Breed	2698	1.14	470	0.65	2228	1.35
Voorzieningen	189837	79.98	56921	78.42	132916	80.66
WarmWater	213710	90.03	65760	90.60	147950	89.78
Winkel Oppervlakte	69	0.03	4	0.01	65	0.04
Winkel inpandig	69	0.03	4	0.01	65	0.04
Winkel Mogelijk	69	0.03	4	0.01	65	0.04
Woonoppervlakte	237287	99.97	72573	99.99	164714	99.96
Zijtuin Totaal	2617	1.10	98	0.14	2519	1.53
Zijtuin Diep	2617	1.10	98	0.14	2519	1.53
Zijtuin Breed	2617	1.10	98	0.14	2519	1.53
Zonneterras Totaal	5142	2.17	3925	5.41	1217	0.74
Zonneterras Diep	5142	2.17	3925	5.41	1217	0.74
Zonneterras Breed	5142	2.17	3925	5.41	1217	0.74

Appendix B

Funda property translations

1. Backyard and Sideyard: The back and side yard are two different variables, the translation combines these two into one called Yard Total. Where total stands for the total size, while depth and width of both are not used. Frontyard total space is still used except for the depth and width.

2. Bathroom amenities: Indicates the facilities inside the bathroom one or more of the following: Shower, Bath, Toilet, Sauna, Steam room, Jacuzzi. Where shower and bath are mostly present and in most cases also a Toilet. Luxury items like Sauna, Steam room or Jacuzzi occur not that often. Therefore four variables are created for: Bath, Shower, Toilet and Luxury.

3. Balcony: Indicates a Balcony, Roof terrace or French Balcony. For the whole dataset 40% contains a balcony, 10% contains a roof terrace and only 1% a French balcony. Two boolean fields are constructed, one for balcony and one for roof terrace.

4. Company Room, Office room, Store room these features indicate the amount of room, if it is internal or if it is possible to make it. Because of the sparsity of the data these fields are merged into one field containing the surface of the room.

5. Building year: The year the house is build obviously should have some impact on the price. The first year in the data set is 1993 therefore every entry is subtracted by 1990.

6. Garage: For the garage there are a lot of possibilities like: Standalone garage, Attached garage, Internal garage, Carport, Parking space or parking basement. Three fields are constructed for these fields, Garage (containing all types), Carport and Park space (containing parking space and parking basement).

7. Certificate: Indicates a certificate for safety or certificate. This is not something every house or apartment possesses. Therefore this indication will be added to a variable indicating an exception.

8. Location of the real estate: The location is presumed to have a large impact on the price. In the data set there are 14 different types where one feature can have multiple. Because only 10% of the data contains no information and the top combinations cover less than 50%, 14 variables are used one for each entry.

9. Position of the garden: The position of the garden is important because of the sun. The best position is a garden facing south. Because the relation of this variable to price is unknown, 8 variables are created for all the directions (N, NE, E, SE, S, SW, W, NW).

10. Heating and Hot water: The data set contains one variable for Heating and one for Hot water. In the case of Heating in 80% of the cases this is a Central Heating unit, in 14% other

types of heating including city or multiple property heating, the last 6% is unknown. For Hot water is 74% of the cases a Central Heating unit, 16% other types likes: Geyser, Boiler or Sun based, the last 10% is unknown. In both cases the majority is a central heating unit however the other types are different. Therefore these variable will be separate but both receive the similar solution. One variable for a central heating unit and one for other.

11. Isolation: There is a value for the isolation of a house present, this can contain many different texts: Full Isolation, Double glass, No Isolation, Partial double glass or Roof isolation. Because of the large amount of Full and No isolation, three variables are introduced: Full isolation, Partial isolation and No Isolation. Partial Isolation will contain every thing except the latter.

12. Parking facilities: The parking facilities variable indicates if payment or a permit is necessary to park in front of the real estate. Multiple text entries like: Public parking, parking permit, payed parking or garage. Therefore this is split up into three variables namely, Parking free, Parking permit and Parking paid.

13. Storage or Shed: This indicates a storage unit that is used for storage and not for a car. It can be attached or detached and build from stone or wood. The fact that the unit is present is likely to be enough therefore a variable is introduced called: Storage Unit.

14. Specific: The specific feature indicates that something extra is present at the property. It can be an indication that there is furniture or existing floors, a monumental status or double occupancy. Therefore the specific feature is split up into three variables indicating each item.

15. Specific property type: There are 226 different specific types of properties in text format like Type (size related) followed by Type (surroundings). Examples: "Eengezinswoning, tussenwoning", "Villa, vrijstaande woning" or "Bungalow, geschakelde woning". This variable is translated into all size related types and surrounding related types in total 15.

16. Type of Apartment: There are 87 different types of apartment. It also has a format with a size and location related type, Examples: "Bovenwoning (appartement)", "Portiekwoning (appartement met open portiek)". The location type of the apartment is not that interesting, we also have the floor the apartment is located on in another variable. Therefore this variable is translated into 10 different boolean variables indicating each status.

Appendix C

Prediction results

Table C.1: All model combination results with their settings and results, MAA, Standard deviation (SD) and Average error (AVG).

STD	General	Clusters	Trainingsize	Limit	Threshold	MAA	SD	AVG
No	No	No	10	None	None	83.9524	42985.8517	38166.9043
No	No	No	20	None	None	83.8672	43405.7511	38495.8395
No	No	No	30	None	None	83.8586	43503.1455	38675.0934
No	No	No	40	None	None	83.8089	43855.2965	38965.5987
No	No	No	50	None	None	83.6835	43565.7725	39195.0521
No	No	No	60	None	None	83.6879	43696.9318	39309.4661
No	No	No	70	None	None	83.6074	43883.1208	39504.1489
No	Yes	No	10	None	None	84.1509	42848.3141	37894.4038
No	Yes	No	20	None	None	84.0621	43386.3145	38229.7019
No	Yes	No	30	None	None	84.0416	43531.7875	38422.9892
No	Yes	No	40	None	None	83.9783	43866.1770	38723.0769
No	Yes	No	50	None	None	83.8441	43526.8241	38941.6252
No	Yes	No	60	None	None	83.8228	43725.3128	39109.4774
No	Yes	No	70	None	None	83.7318	43863.4003	39289.8179
Yes	Yes	No	10	None	None	84.4202	43137.4871	37408.6900
Yes	Yes	No	20	None	None	84.3160	43382.4754	37746.4660
Yes	Yes	No	30	None	None	84.2872	43345.3615	37940.7038
Yes	Yes	No	40	None	None	84.2147	43618.8012	38227.3336
Yes	Yes	No	50	None	None	84.0743	43277.2999	38445.5141
No	No	Clusters	10	None	None	85.2896	42595.0938	35679.3006
No	No	Clusters	20	None	None	84.9342	43139.7916	36669.3635
No	No	Clusters	30	None	None	84.6863	43255.5269	37234.7368
No	No	Clusters	40	None	None	84.4669	43513.9486	37796.4165
No	No	Clusters	50	None	None	84.2450	43225.1085	38113.5768
Yes	No	Clusters	10	None	None	85.4613	43019.5542	35431.6170
Yes	No	Clusters	20	None	None	85.1303	43210.4764	36323.0784
Yes	No	Clusters	30	None	None	84.8340	42957.9966	36956.7430
Yes	No	Clusters	40	None	None	84.6097	43245.0805	37504.3077
Yes	No	Clusters	50	None	None	84.3542	43086.9222	37907.2469
No	Yes	Clusters	10	None	None	85.2940	42358.8488	35650.7027
No	Yes	Clusters	20	None	None	85.1101	42835.0369	36412.3879
No	Yes	Clusters	30	None	None	84.8666	43211.7362	37009.1729

No	Yes	Clusters	40	None	None	84.6427	43469.9110	37539.4549
No	Yes	Clusters	50	None	None	84.4022	43184.1418	37890.7226
Yes	Yes	Clusters	10	None	None	85.4799	43820.3669	35573.7599
Yes	Yes	Clusters	20	None	None	85.2718	43239.8663	36198.5117
Yes	Yes	Clusters	30	None	None	85.0988	42919.6853	36529.3318
Yes	Yes	Clusters	40	None	None	84.8549	43229.5733	37062.4576
Yes	Yes	Clusters	50	None	None	84.6028	42925.6868	37445.2367
Yes	No	Clusters	10	None	50	85.6499	42597.6392	35497.3258
Yes	Yes	Clusters	10	None	50	85.6837	42540.6339	35541.3933
Yes	Yes	Clusters	20	None	50	85.4047	43108.6435	36062.3231
Yes	Yes	Clusters	30	None	50	85.1740	42919.5889	36456.4613
Yes	No	Clusters	10	None	60	85.7189	41752.9240	35306.3561
Yes	Yes	Clusters	10	None	60	85.8641	42015.7016	35151.1389
Yes	Yes	Clusters	20	None	60	85.5035	42279.7824	38882.8120
Yes	Yes	Clusters	30	None	60	85.1781	42776.9853	36525.3335
Yes	No	Clusters	10	None	70	85.5618	39715.2392	34663.6030
Yes	Yes	Clusters	10	None	70	85.7819	39761.3969	34625.2615
Yes	Yes	Clusters	20	None	70	85.4799	40804.8899	35464.7005
Yes	Yes	Clusters	30	None	70	85.3337	41633.7917	36150.4461
Yes	No	Clusters	10	None	75	85.5828	38847.9478	34251.7806
Yes	Yes	Clusters	10	None	75	85.6727	39068.0736	34308.1806
Yes	No	Clusters	10	None	80	85.4895	39767.7458	34452.1210
Yes	Yes	Clusters	10	None	80	85.6077	39869.5534	34248.9103
Yes	No	Clusters	10	None	85	85.2825	40945.3049	34939.9470
Yes	Yes	Clusters	10	None	85	85.3089	41662.5891	34980.0963
Yes	No	Clusters	10	None	90	84.6769	43243.7924	36319.3731
Yes	Yes	Clusters	10	None	90	84.6987	43381.1226	36358.2539
No	No	Near	10	50	None	85.3768	41099.6267	35272.0242
No	No	Near	20	50	None	84.9537	42001.0005	36488.4714
No	No	Near	10	100	None	85.4479	40733.9738	35102.6670
No	No	Near	10	200	None	85.6596	40495.5795	34674.9503
No	No	Near	10	300	None	85.6661	40513.3633	34650.1206
Yes	No	Near	10	50	None	85.2434	41315.4064	35404.3443
Yes	No	Near	20	50	None	84.8114	42205.0510	36804.0613
Yes	No	Near	10	100	None	85.4162	41061.9830	35239.7133
Yes	No	Near	10	200	None	85.4883	40988.9831	35083.1138
Yes	No	Near	10	300	None	85.4898	41000.4935	35083.8336
No	Yes	Near	10	50	None	85.4263	41165.2379	35262.4229
No	Yes	Near	20	50	None	85.0507	42081.6354	36397.6153
No	Yes	Near	10	100	None	85.5109	40745.4357	35061.1448
No	Yes	Near	10	200	None	85.5973	40684.3877	34888.8916
No	Yes	Near	10	300	None	85.5955	40694.5918	34877.3495
Yes	Yes	Near	10	50	None	85.4180	41249.5544	35241.4324
Yes	Yes	Near	20	50	None	84.9414	42100.9381	36495.2714
Yes	Yes	Near	10	100	None	85.4929	40949.1816	35063.9379
Yes	Yes	Near	10	200	None	85.5742	40884.0557	34904.4516
Yes	Yes	Near	10	300	None	85.5626	40890.5372	34918.2899
No	No	Near	10	200	50	85.7698	40109.2509	34549.0204

No	Yes	Near	10	200	50	85.8117	39975.6998	34498.2680
Yes	Yes	Near	10	200	50	85.8280	40376.1107	34542.7693
No	No	Near	10	200	60	85.8415	39843.4523	34458.5599
No	Yes	Near	10	200	60	85.7246	39662.6776	34482.9729
Yes	Yes	Near	10	200	60	85.8236	40372.1369	34598.2399
No	No	Near	10	200	70	85.6949	39583.1368	34240.0753
No	Yes	Near	10	200	70	85.9082	39753.1068	34199.0552
Yes	Yes	Near	10	200	70	85.7492	39664.4640	34216.6904
No	No	Near	10	200	75	85.5342	39365.9674	34127.9709
No	Yes	Near	10	200	75	85.5230	39280.6239	34146.2178
Yes	Yes	Near	10	200	75	85.6821	39506.2921	33962.9953
No	No	Near	10	200	80	85.4706	39359.3603	33938.7032
No	Yes	Near	10	200	80	85.4648	39417.0383	33925.6242
Yes	Yes	Near	10	200	80	85.3461	40021.3644	34255.4552
No	No	Near	10	200	85	85.1894	41447.9324	35155.2872
No	Yes	Near	10	200	85	85.3374	41415.9439	34853.5826
Yes	Yes	Near	10	200	85	85.3515	40764.6588	34494.1687