DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2019

Prediction of residential real estate selling prices using neural networks

PONTUS NILSSON

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Prediction of residential real estate selling prices using neural networks

PONTUS NILSSON

Master in Computer Science
Date: March 5, 2019
Supervisor: Erik Fransén
Examiner: Elena Troubitsyna
Principal: Virtusa
Swedish title: Uppskattning av bostadspriser med neurala nätverk
School of Electrical Engineering and Computer Science

Abstract

With the rising housing prices of the last 20 years, the appraisal of real estate has become more difficult. Underlined by the large differences between listing and selling prices, the valuation process brings a level of uncertainty. With the advances within the field of machine learning in recent years, attempts have been made to apply these techniques to the real estate market. This thesis investigates the potential of using neural networks to predict selling prices of apartments in Stockholm, based on apartment parameters. Networks are trained to either make an improved valuation, based on a listing price, or make a new valuation of an apartment. The results are promising, and in line with contemporary findings; however, the worst-case performance of the models could make them unsuitable for many purposes.

Sammanfattning

Med stigande bostadspriser under de senaste tjugo åren har fastighetsvärdering blivit en svårare uppgift. Värderingsprocesssen medför en grad av osäkerhet, märkbar på de stora skillnaderna mellan utgångspriser och försäljningspriser. Efter stora framsteg inom maskininlärning de senaste åren har försök gjorts att applicera dessa tekniker på bostadsmarknaden. Den här studien utforskar möjligheterna att använda neurala nätverk för att uppskatta försäljningspriser av lägenheter i Stockholm, baserat på lägenhetsparametrar. Nätverken tränas för att antingen göra en förbättrad värdering utifrån utgångspriset, eller för att göra en ny värdering av en lägenhet. Resultat visar på potential, och är i linje med liknande försök, men värstafallsprestandan kan göra modellen olämplig att använda för många syften.

Contents

Chapter 1

Introduction

For the last two decades, residential Stockholm real estate prices have risen steadily. According to Svensk Mäklarstatistik [20], 2008 has been the only recent year that average prices did not see an increase compared to the previous year. With this in mind, the appraisal of real estate can be a difficult task [7]. Further, the difference between a valuation and the selling price can be large. A 2017 study by *Mäklarhuset* [13] found that the difference between the listing price and the final selling price was substantial. As an example, the median difference for 1-bedroom apartments in Stockholm was more than 15% for 2010-2017, and almost 25% for 2016-2017. Whether the cause of this discrepancy between listing and selling prices is due to real estate brokers' valuation errors, or a conscious tactic to promote interest in a sale [9, 10], it makes it difficult to use the listing price as a reliable measure for the final selling price.

In recent years, with the advances that have been done in the field of machine learning, attempts have been made to apply this to the real estate market. Neural networks are one of the techniques used for this purpose [2, 8]. The neural networks are modelled from an idea to imitate the human brain, with the units in the network filling the same purpose as the brain's neurons [21]. The networks can be used by adapting to some data, for example historical sales data, to then predict an output value based on some input parameter(s). There are several factors that contribute to the final price of real estate, most of which can be grouped into certain categories. By established practice these categories consist of locational, structural and neighbourhood determinants [1]. Booli [5] is a company that "shows data on listed

and sold housing, and provides valuation indicators for houses and apartments". They provide access to historic data on housing sales in Sweden through their API [6]. The data from this API will be used in this thesis.

1.1 Problem statement

The aim of this thesis is to expand on the previous studies made on neural networks and real estate appraisal. Through this thesis, an attempt will be made to create models capable of appraisal of apartments and prediction of selling prices for apartments. To accomplish this, the thesis will attempt to investigate whether a neural network is able to accurately predict the selling prices of residential real estate.

1.2 Purpose

Forecasting of future outcomes is a key element in many fields, including financial areas. The predictions serve as support for decision-making, to aid in making optimal choices [4]. A successful model for predicting selling prices of apartments could have great potential, in real estate as well as other areas. It could be used as assistance for real estate agents, as well as for property owners to determine the value of their properties. Another area of use is real estate sales that have a low listing price to receive more attention from potential buyers (*"lockpriser"*). A predicting model could indicate sales of real estate with a low listing price as possibly employing *"lockpriser"*. Finally, a successful model would also reinforce the potential of uses of neural networks for related applications.

1.3 Scope and approach

To narrow down the scope of the thesis, it will encompass sales of *tenant-owned apartments* ("bostadsrätter") in parts of Stockholm. This was chosen due to the high concentration of sales data in the area. To ensure recent and relevant data, the period of time to investigate was decided to be the whole year of 2018. The data used in the study is gathered from Booli. After obtaining the data, part of it is used to

optimise hyperparameters by performing a gridsearch over possible values of the hyperparameters. When the optimised parameters have been found, the rest of the data is used for training and testing using a 10-fold cross validation technique. The architecture used for the models is the standard multi-layer perceptron, implemented with one and two layers.

1.4 Outline of thesis

Beginning, chapter 2 will present relevant background to the problem. It covers some theory on real estate appraisal, including the parameters that affect the appraisal, and common levels of accuracy of appraisals. The chapter continues with theory on machine learning and neural networks, up to the multi-layer perceptron used in this study. Finally, chapter 2 ends with some coverage of similar work that has been done in other studies.

Chapter 3 covers the methods used in this thesis. The data from Booli is presented and explained, the parameters used are shown, and the two sets of data are analysed. The way the model is built is also described in more detail. In chapter 4, the results from the thesis are presented, showing the plots and values for each model. Finally, in chapter 5, the results are analysed and commented, followed by a critical evaluation of the thesis work, some comments on the ethical aspects of the outcome, and some possible extensions to the study. The thesis is finished with the conclusions of chapter 6.

Chapter 2

Background & Theory

This chapter will introduce some relevant theory in the appraisal of real estate, which is important to consider when building the valuation model. Regarding this, we will look at what parameters affects the appraisal, and their categorisation. We will also look at the accuracy of real estate appraisal. Continuing, we will cover the theory on which the model will be built, neural networks. We will go on to explain how this relates to a specific neural network, the multi-layer perceptron. Finally, we will browse through some of the related work that has been done on the subject previously.

2.1 Real estate valuation and the market

Traditionally, the concept of appraising residential real estate is heavily reliant on the interpretation of human behaviour. This involves the prediction of human decision-making based on some factors, both subjective and objective [22, p. 4]. These factors form a complex union that influence the end value of the property. A 2016 study by Abidoye and Chan [1] found that, most commonly, the attributes that determine property value are categorised into three groups. The groups or classes contain factors that pertain to the *structural*, *locational*, and *neighbourhood* attributes. The *structural* attributes are the factors that are related to the housing itself. These can be values such as the floor area, or number of bedrooms and bathrooms, but can also consist of the age and state of the housing. *Locational* attributes concern factors about the location of the housing in relation to economic and social facilities, such as the availability of work, school, or public transport in prox-

imity to the housing. Finally, the *neighbourhood* attributes concern the quality and safety of the neighbourhood that the housing is located in. In the 2016 study [1], some important neighbourhood attributes were the availability of electricity as well as the availability of neighbourhood security. However, this study was focused on Lagos, Nigeria, and the factors does not necessarily have the same importance in other areas. The authors found that the set of attributes that determined the property values in Lagos was different than that of other parts of the world. They also found that, internationally, the structural attributes were prominently found as important determinants of property value.

One of the more common techniques used as an aid in valuation is *anchoring and adjustment*. With this technique, the valuer begins by using an *anchor* value as a guide. The anchor can be information such as previous sale prices, or prices of recently sold objects with similar traits. The valuer then *adjusts* the expected value according to the other factors [22, p. 6].

2.1.1 Uncertainty and accuracy

Due to the numerous factors affecting the value of a property, it can be difficult to determine its true value. What is instead used is "a professional estimate of what it is expected to sell for at the date of the valuation" [7]. Because the true value is unobservable, determining the accuracy of real estate valuation may not be possible in a strict sense [22, p. 46]. What is used instead of the true value is the market price of the real estate. Since the valuation is based on an expectation of the importance of certain parameters, it will have some variance and uncertainty. Some of the uncertainty comes from the estimation of importance of the factors done by the valuer. Another part comes from the difference in priority of factors that may have an effect in different markets and areas [7].

In a 2002 study, Pace, Sirmans, and Slawson [15] made a comparison between statistical pricing models and traditional hedonic models. They found that the traditional models, with a simple and complex version, produced estimations with a mean absolute error of 17% and 11% respectively. Meanwhile, the two statistically based models both had mean absolute errors of around 9%. The variance of appraisers' valuations can also have an effect on the errors. Findings by Brown, Matysiak, and Shepherd [7] imply that the chance for an appraiser of

making a valuation within a 5% range from the expected value was about 1 in 10. For a 10% range it had risen to about 1 in 5.

In 2017, a survey by Mäklarhuset [13] was done to investigate the differences between listing prices and selling prices of residential real estate in Sweden. The purpose of this was to look at the practice of valuers to intentionally undervalue the housing, setting a low listing price to gather a higher participation in the bidding *"lockpriser"*, with the intent of reaching a higher selling price. The study found a large difference between the listing price and the selling price for the investigated period, 2010-2017. The highest differences were seen for 1-bedroom apartments, which saw a deviation of about 12% in Sweden, and over 15% in Stockholm. For Stockholm, 2-bedroom apartments had a deviation of a little over 10%, while the value for 3- and 4-bedroom apartments was about 5%. When looking at only more recent data, 2016-2017, the deviations were even larger. For Stockholm-located 1-bedroom apartments, the deviation was more than 20%, it was 15% for 2-bedroom apartments, and almost 10% for 3- and 4-bedroom apartments. Almost half of the 1-bedroom apartments had a deviation of between 10 and 30%, while about a third of them had a deviation of 30% or more. The explanation to these discrepancies that is given in the study is not that the listing prices are intentionally low, but the difficulty in valuation in a volatile market, referring to the rising prices of the Stockholm housing market. In the period of 1997-2017, the average price per m^2 rose by more than 500%, and in the last 10 years the prices have doubled [20].

2.2 Artificial Neural Networks

The Neural Network (NN), or Artificial Neural Network (ANN), is a model inspired by the neural infrastructure of the brain. Neurons are brain cells that handle electrical and chemical signals. The neurons can form a network of connections, called synapses. The basis of the model comes from the *Rosenblatt perceptron*.

2.2.1 The perceptron

The perceptron was developed by Rosenblatt [16] in 1958. An example of the perceptron can be seen in figure 2.1. The first component of the perceptron is the sensor units, or input. They transmit signals through

connections to the activation component (also called projection area, association area, or activation function). Depending on the signal, the cells or nodes in the activation component may be activated, if the signals reach a certain threshold. Should they reach the threshold, they will pass along a signal to the response component that they have been activated. Based on the result of the responses, there is a possibility to send feedback from the response component to the activation component, promoting or discouraging activation in future cases. This allows for the perceptron the learn a behaviour based on the sensor input.

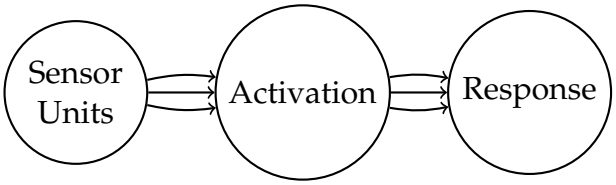

Figure 2.1: The Perceptron

In figure 2.2 is a more concrete example of a perceptron. The sensor component (input layer) features two input nodes, which are connected to the activation component. The activation component will, if triggered, send a signal to the single output node of the response component (output layer). The connections between the input nodes can be unaltered, or they can have different weights, that determine the importance of each connection.

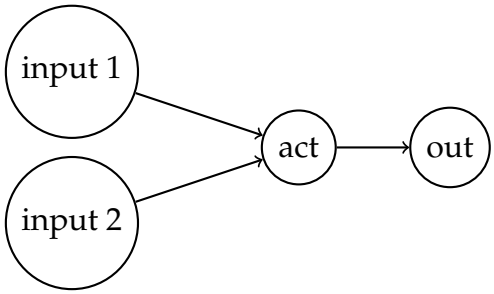

Figure 2.2: A perceptron example

2.2.2 Multi-layer perceptron

Since the input is directly mapped to the output through the activation function, there is no possibility for the network to create its own

knowledge representation. It produces similar output for similar input, which is favourable for certain applications, while it will have trouble with others. One such problem is the XOR-problem. The problem is equivalent to the classification or separation of the different classes in figure 2.3. A low x-value with a high y-value, and a high x-value with a low y-value should both result in the same classification, class 1, while similar x- and y-values, either high or low, should result in the other classification, class 0. This is in contrast to the "similar input, similar output" reasoning of the perceptron. The perceptron is only able to solve problems linearly. Graphically, it can be seen as drawing a straight line in figure 2.3 to separate the two classes. Since this is not possible, there is not a linear solution, and the problem is not *linearly separable*.

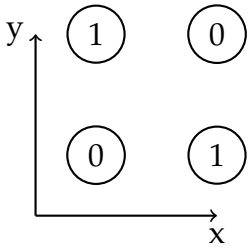

Figure 2.3: The XOR problem

This can be solved by adding another layer, as in figure 2.4. The additional layer, called a "hidden layer", is constructed between the input layer and the output layer. This allows the model to have an internal representation of the input pattern, which it can transfer to the output.

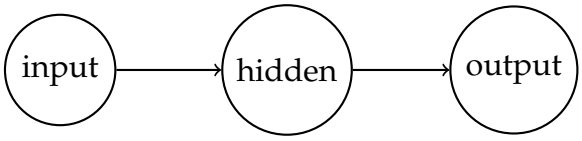

Figure 2.4: A perceptron with hidden layer

Rumelhart, Hinton, and Williams [17] authored a report which covered the issue of internal representation, and the potential for a *multilayer perceptron* to solve problems such as the XOR-problem. In figure 2.5 is a solution to the XOR problem using a perceptron with a hidden layer, containing three hidden neurons. The first and third hidden unit will be activated by the both inputs respectively, while

the middle neuron will activate only if both inputs are on. This middle neuron is the internal representation needed to solve the problem, as an activated middle neuron will not allow an activated output. The only way the output is activated is if it receives a signal from either the first or third hidden neuron, while not receiving both, which fulfils the requirements of the XOR-problem.

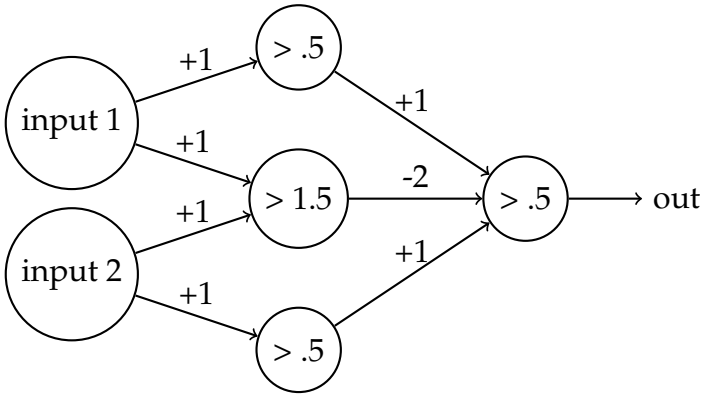

Figure 2.5: XOR solution

In their report, the authors also present a learning procedure for the network. The learning procedure consists of a supervised learning method, where a set of input patterns are injected into the network, and the corresponding output patterns are compared to the output of the network. For instances where the network's output corresponds to the output pattern, no updates are needed. If the network output is incorrect, an error signal is sent back through the network, prompting adjustments to the weights to decrease the error for future input.

2.3 Related work

Some previous attempts have been made to combine the area of valuation of real estate, with neural networks or other forms of machine learning. In this section an overview of that research will be provided.

In 2017, Abidoye and Chan wrote an article on the appraisal of real estate values in Lagos, Nigeria [2]. A neural network was used to predict the selling prices of residential real estate, using a dataset of 321 observed properties in high-income neighbourhoods of Lagos. The network consisted of a single hidden layer, in addition to the input and output layers. The architecture of the network was a 11-5-1

layout, of input, hidden, and output nodes, respectively. The resulting predictions provided by the model had a mean absolute error of 15.94%. The authors concluded that the results were satisfactory, and that the model was promising.

Another study was made in 2017 on residential real estate located in Naples, Italy [8]. This study uses a probabilistic approach, arguing that the appraisal is not a deterministic action, but instead subject to a probabilistic distribution. With a learning set of 35 housing units, the model predicted the price of another 30 with a mean absolute error of 6.61%. The study used a neural network with a single hidden layer, containing 30 hidden units.

In 2016, Wang et al. [23] conducted a study on the use of neural networks to predict the prices of housing in Singapore. This network uses time-series data of housing sales, as well as general economic variables, such as the population levels and the average monthly wages. The authors praised the results, claiming the model delivers an accurate forecast. This was based on a high measure of the coefficient of determination (r^2), and a low mean square error of the prediction, which suggests a good performing model. However, the authors do not provide any mean percentage errors, making comparisons to the other mentioned studies difficult.

A Chicago, USA, study [3] uses a recurrent neural network to predict house prices of 2014 and 2015, again using time-series data. The network used was a LSTM, *Long Short-term memory*, and allows for a certain recollection of earlier input, or memory, that some other types of networks do not have. In the study, the LSTM network is compared to two other modern regression methods used for house price prediction. The LSTM network predicted the house prices with a mean absolute percentage error of 24%, while the other two methods had errors of 30% and 31%, respectively.

Chapter 3

Methods

In this chapter, the methods will be explained that are used in the process of this thesis. The first section handles the data that the model is built on. This includes from where the data is obtained, and some description of the data. Continuing on, we will explain how the model is constructed. Finally, we will look at the methods used for the training and evaluation of the model.

3.1 General approach

The training was done in two major phases. In the first phase, the listing price was used as a basis for a predicted selling price. Using this approach, the listing price is one of the input parameters for training and testing. The target for this approach was to have the prediction be closer than the listing price to the final selling price. For the second phase, the listing price was omitted entirely. In this, the predictions made were intended to represent the appraisal of the real estate in question.

3.2 Data

The data used in this thesis comes from housing sales in Stockholm. The data was gathered from Booli [5], through their API [6]. The different parameters used for the data entries can be seen in table 3.1. Since some of the entries are incomplete, any entries with missing parameters were excluded from the dataset. Normalisation of the data

was considered, but since it is not strictly necessary [11], and was not used in similar studies [8, 2], it was decided to not use it. The *Street Address* parameter was only used for readable output and debugging. *Listing Price* is the original asking price for the apartment, while *Sold Price* is the *registered selling price*. The registered selling price can be from one of several sources. For this study the records come either from the broker directly, or they are based on the last registered bid on the apartment.

Parameter	***Unit***	**Parameter**	***Unit***
Listing Price	SEK	Rooms	no.
Sold Price	SEK	Floor	no.
Rent	SEK	Ocean Distance	m
Living Area	m^2	Street Address	String

Table 3.1: Booli API parameters

Based on the categories mentioned in section 2.1, the parameters can be divided in the way shown in table 3.2.

Structural	**Locational**	**Sale**
Living Area	Street Address	Listing Price
Rooms	Floor	Sold Price
Rent	Ocean Distance	

Table 3.2: Parameter groups

To further define the research area, this study would only look at sales of *tenant-owned apartments* ("bostadsrätter"). Two areas of differing sizes were chosen as the subject of the study. The larger area is "Stockholm inom tullarna", which encompasses the main parts of the Stockholm city centre, with a dense distribution of residential apartments. A subset of this area was also used as a dataset, the "Vasastan" district. These areas were chosen due to their high population density, as well as the large number of housing sales. The data consists of sales made during the whole year of 2018. The details of the sets of data can be seen in tables 3.3 and 3.4. The data was also checked for stationarity. First, the selling prices were investigated. The data can be seen in figures 3.1 and 3.2, and show a period of decrease in volume and lower selling prices (ca. Jul-Aug). Additionally, the price per living area was

observed, and can be seen in figures 3.3 and 3.4. The prices per living area did not reveal any additional trends, and suggest that the lower selling prices was due to a lack of sales of larger apartments during this time. From the above, it can be assumed that the data is stationary over this interval.

Inom tullarna				Size: 6726	
Parameter	**Mean**	**Median**	**Min**	**Max**	**Std**
Selling Price	5.15M	4.35M	1.825M	31M	2.73M
Listing Price	4.86M	3.995M	1.495M	28.8M	2.72M
Living Area	58.7	51.6	13	270	29.6
Rooms	2.25	2	1	7	1.01
Rent	2663	2434	0	12843	1236
Floor	2.97	3	0	34	2.13
Ocean Distance	1746	1745.5	36	4207	932
Price per m^2	90k	88k	30k	197k	15k

Table 3.3: Large dataset

Vasastan				Size: 1747	
Parameter	**Mean**	**Median**	**Min**	**Max**	**Std**
Selling Price	5.49M	4.88M	1.825M	26M	2.56M
Listing Price	5.15M	4.53M	1.89M	26M	2.49M
Living Area	60.2	56	13	197	28.1
Rooms	2.28	2	1	6	0.97
Rent	2531	2350	0	8532	1159
Floor	2.91	3	0	34	2.41
Ocean Distance	2201	2233	654	2926	389
Price per m^2	93k	92k	60k	167k	13k

Table 3.4: Small dataset

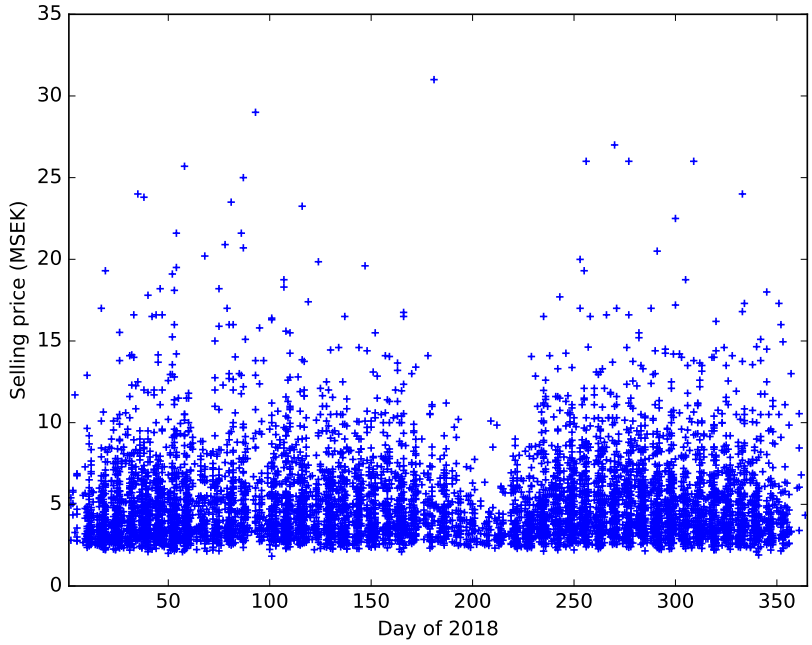

Figure 3.1: Selling prices 2018 (MSEK) - Large dataset

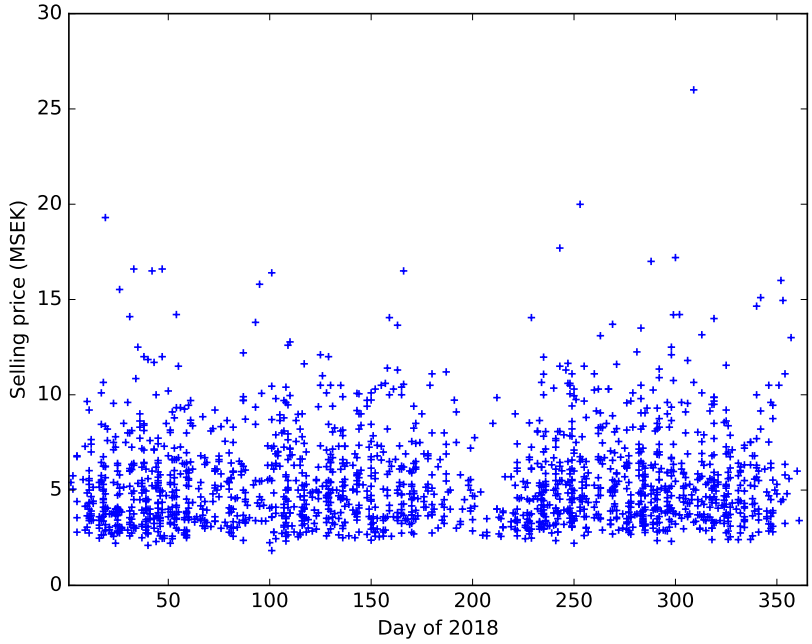

Figure 3.2: Selling prices 2018 (MSEK) - Small dataset

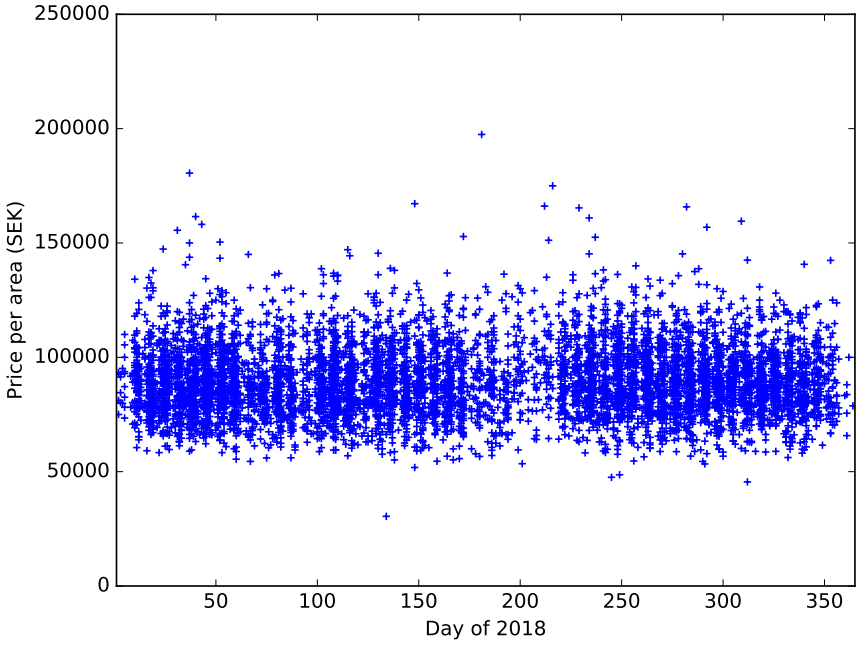

Figure 3.3: SEK per m^2 2018 - Large dataset

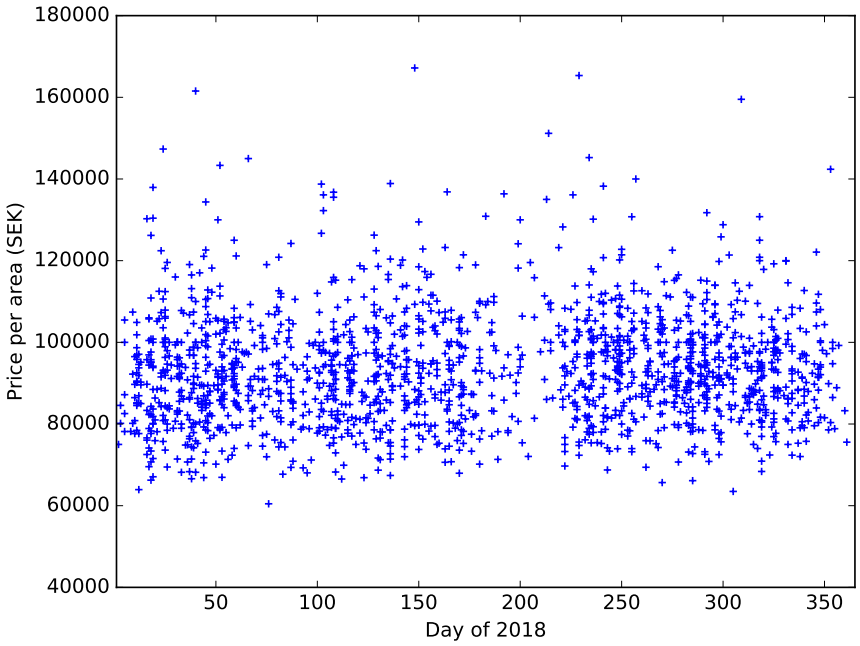

Figure 3.4: SEK per m^2 2018 - Small dataset

3.3 Model

Based on the results from the literature study, it was decided to use a multi-layer perceptron as the architecture for the thesis. The MLP was modelled using the Keras framework, on top of the TensorFlow library. Models using one or two hidden layers were used, and they were compiled with the *Adam* optimiser, a variation of the stochastic gradient descent [11]. The accuracy of the model is determined by the used cost function, which is *mean absolute percentage error*, MAPE. A gridsearch was performed to optimise the hyperparameters of the model. The hyperparameters that were optimised and the searched values can be seen in table 3.5. The number of input nodes depended on whether the listing price was used, with five input nodes for models without the listing price, and six input nodes for models with it. The number of hidden nodes was one of the hyperparameters determined by the gridsearch. Models with multiple hidden layers contained the same number of hidden nodes in each layer. There was a single output node, corresponding to the predicted value of the apartment. The data was split into training, validation and testing sets, as seen in figure 3.5. The 80-20 split used is a typical ratio for this purpose [11]. For regularisation, early stopping was used in all training phases, with a patience value of 10.

Parameter	**Value**
Learn rate	0.01, 0.005, 0.001
Activation function	ReLU, LeakyReLU
Batch size	10,25,50
Nodes per hidden layer	128, 256, 512,1024
Dropout	0.0, 0.1, 0.2

Table 3.5: Hyperparameters optimised in gridsearch

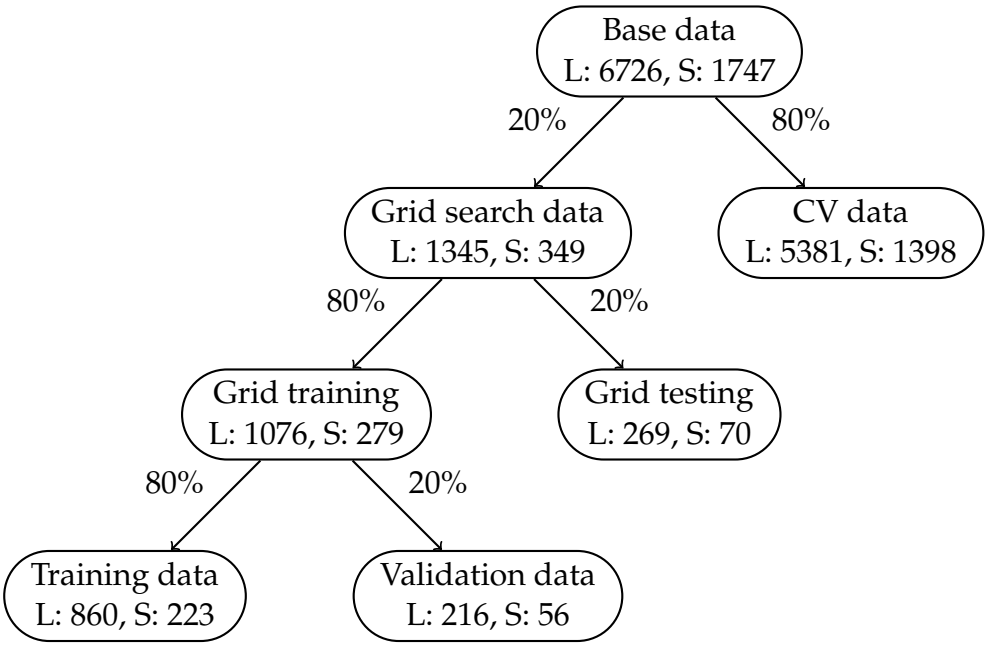

Figure 3.5: Dataset splits. L denotes the size of the large dataset, S denotes the size of the small dataset

3.4 Evaluation

To evaluate the performance, models was trained with the hyperparameters found in the gridsearch. A 10-fold cross validation was performed for each phase and dataset, as seen in figure 3.6. The mean and standard deviation was analysed, for each fold as well as for the folds as a whole.

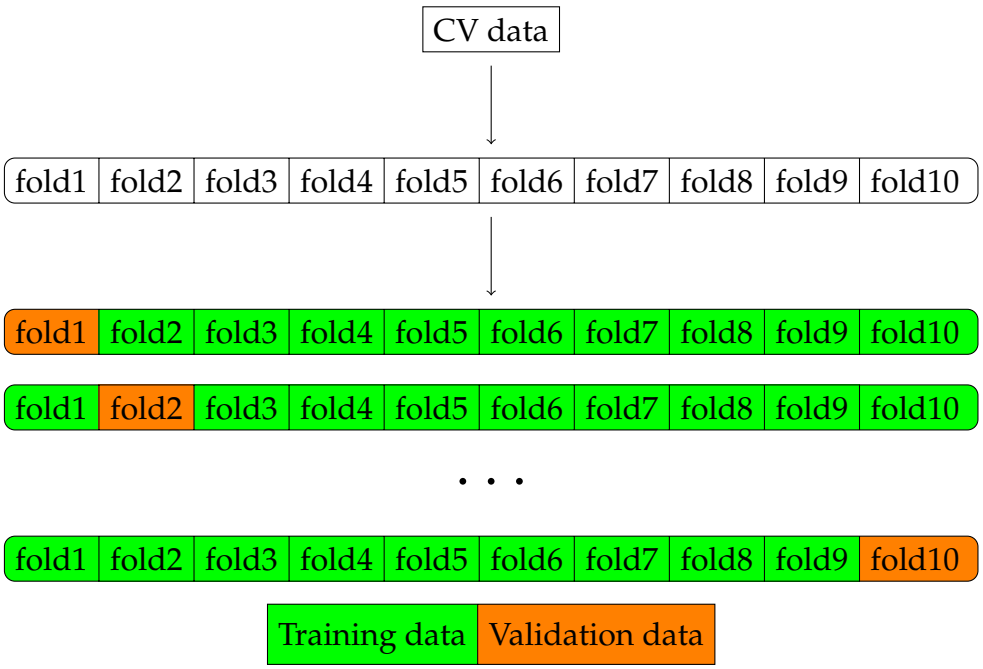

Figure 3.6: Cross validation data

Chapter 4

Result

In this chapter, the results of the two phases will be presented, beginning with the networks trained using the listing price as an input parameter, and ending with the networks unaware of the listing price. For each phase, the optimised hyperparameters obtained from the gridsearch are provided. Then, the results of the cross validation are presented. The loss used was the *mean absolute percentage error*, "MAPE". The main metrics provided in the results is the mean loss across the cross validation folds, and the corresponding standard deviation of the mean loss. Also provided is the mean of each fold's standard deviation, and the standard deviation of this mean. The boxes in the boxplots show the median value, along with the first and third quartiles. The whiskers (dashed lines) extend up to 1.5 times the interquartile range, while any outliers outside of that range are represented by a mark.

4.1 Disclosed listing price

4.1.1 Large dataset

The hyperparameters that were obtained from the gridsearch on the "Inom tullarna" dataset can be seen in table 4.1, for the one- and two-layer networks respectively. The results of the cross validation can be seen in figures 4.1 and 4.2. "Error" on the y-axis denotes the absolute percentage error of the prediction compared to the actual selling price, where 0% would be a perfect prediction.

One layer		Two layers	
Parameter	**Value**	**Parameter**	**Value**
Learn rate	0.005	Learn rate	0.01
Act. function	LeakyReLU	Act. function	LeakyReLU
Batch size	10	Batch size	10
Nodes per layer	256	Nodes per layer	1024
Dropout	0.0	Dropout	0.0

Table 4.1: Hyperparameters from grid search

The one-layer architecture had a mean loss of 6.34, while the two-layer network had a higher mean loss, at 7.49. The results can be seen in table 4.2.

One layer		Two layers	
Metric	**Value**	**Metric**	**Value**
Mean loss	6.34	Mean loss	7.49
Loss std	0.39	Loss std	1.59
Mean std	4.80	Mean std	5.31
Std of mean std	0.40	Std of mean std	0.69
Max loss	7.22	Max loss	11.57

Table 4.2: Cross validation results

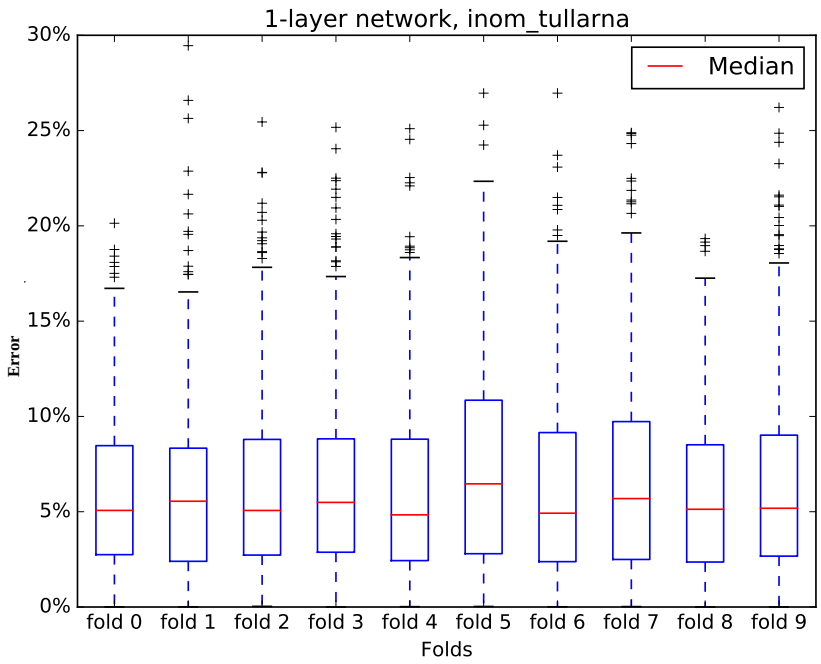

Figure 4.1: One hidden layer

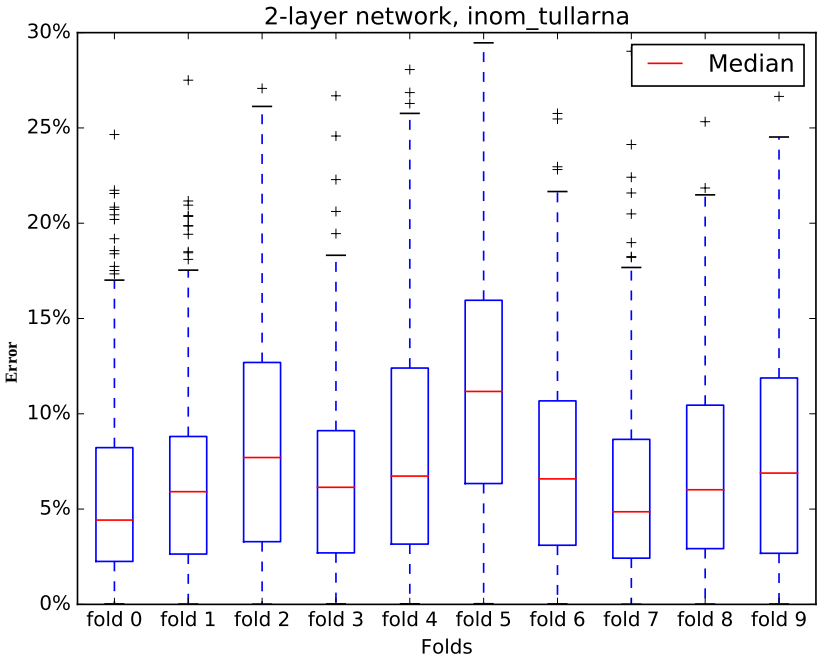

Figure 4.2: Two hidden layers

4.1.2 Small dataset

The hyperparameters that were obtained from the gridsearch on the smaller dataset, "Vasastan", can be seen in table 4.3, for the one- and two-layer networks respectively. The results of the cross validation can be seen in figures 4.3 and 4.4.

One layer		Two layers	
Parameter	**Value**	**Parameter**	**Value**
Learn rate	0.01	Learn rate	0.005
Act. function	ReLU	Act. function	LeakyReLU
Batch size	50	Batch size	10
Nodes per layer	1024	Nodes per layer	1024
Dropout	0.1	Dropout	0.1

Table 4.3: Hyperparameters from grid search

The one-layer architecture had a mean loss of 6.56, while the two-layer network again had a higher mean loss, at 7.67. The results can be seen in table 4.4.

One layer		Two layers	
Metric	**Value**	**Metric**	**Value**
Mean loss	6.56	Mean loss	7.67
Loss std	0.40	Loss std	1.24
Mean std	4.76	Mean std	5.30
Std of mean std	0.54	Std of mean std	0.74
Max loss	7.26	Max loss	9.43

Table 4.4: Cross validation results

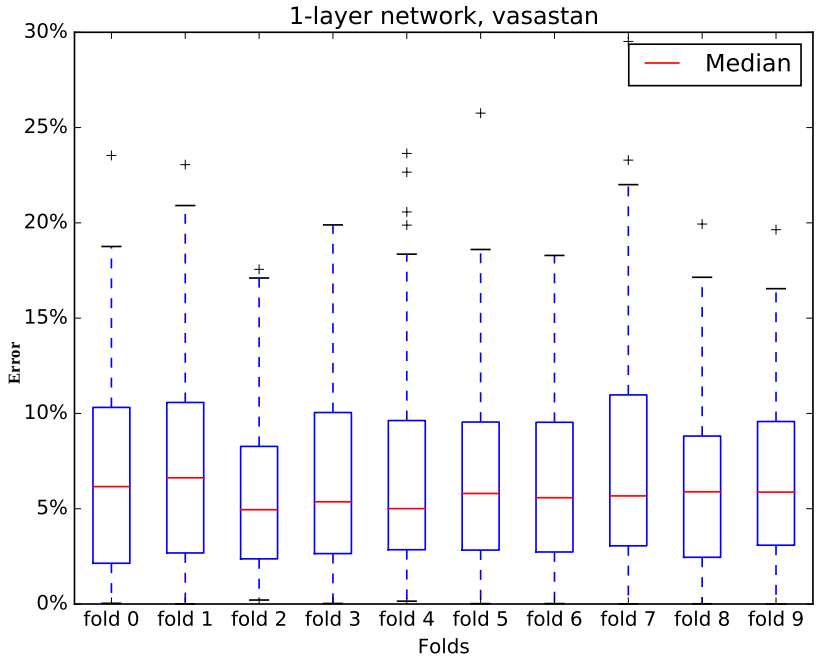

Figure 4.3: One hidden layer

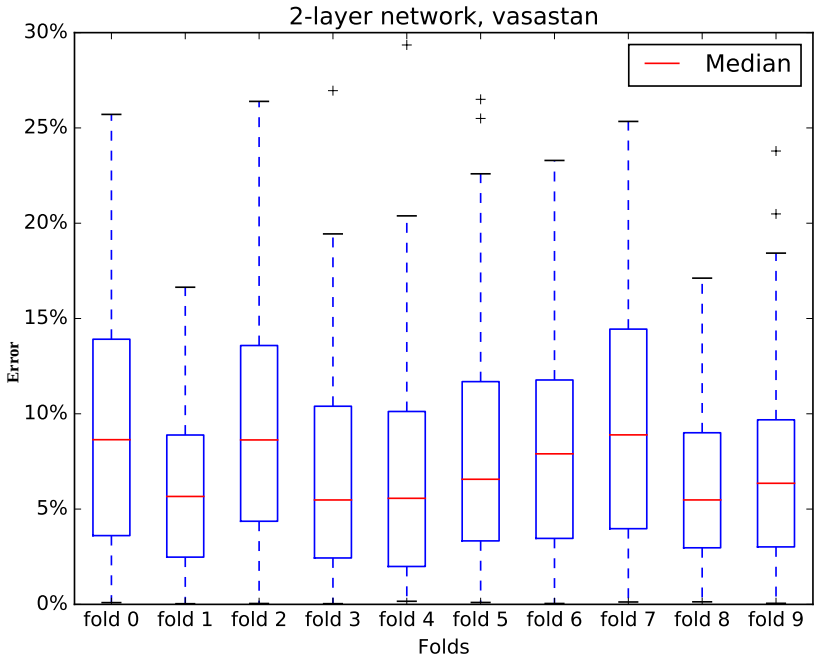

Figure 4.4: Two hidden layers

4.2 Hidden listing price

4.2.1 Large dataset

The hyperparameters that were obtained from the gridsearch can be seen in table 4.5, for the one- and two-layer networks respectively. The results of the cross validation can be seen in figures 4.5 and 4.6.

One layer		Two layers	
Parameter	**Value**	**Parameter**	**Value**
Learn rate	0.01	Learn rate	0.01
Act. function	LeakyReLU	Act. function	LeakyReLU
Batch size	25	Batch size	50
Nodes per layer	1024	Nodes per layer	512
Dropout	0.1	Dropout	0.0

Table 4.5: Hyperparameters from grid search

The one-layer architecture had a mean loss of 11.49, while the two-layer network had a mean loss of 11.67. The results can be seen in table 4.6.

One layer		Two layers	
Metric	**Value**	**Metric**	**Value**
Mean loss	11.49	Mean loss	11.67
Loss std	0.67	Loss std	0.56
Mean std	8.93	Mean std	9.06
Std of mean std	0.78	Std of mean std	0.86
Max loss	12.13	Max loss	12.55

Table 4.6: Cross validation results

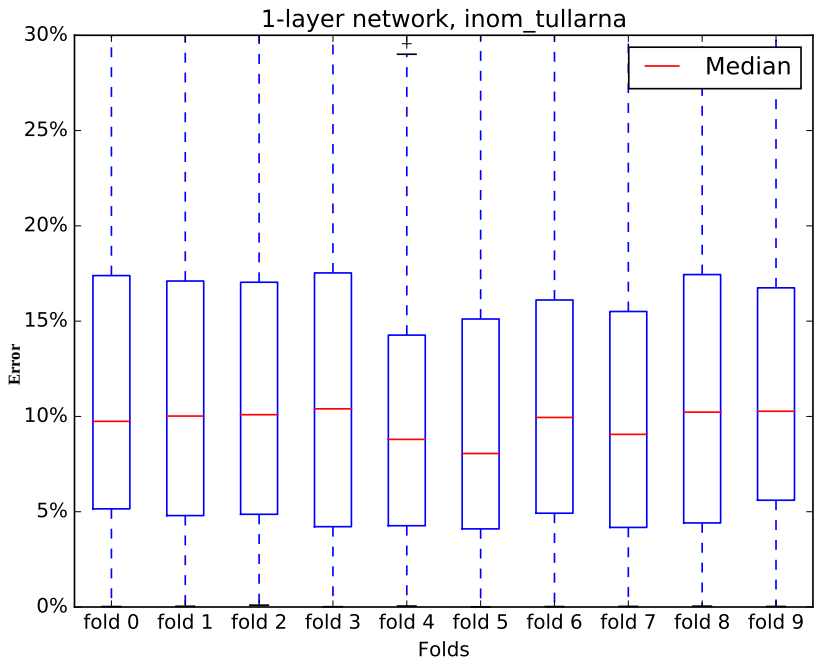

Figure 4.5: One hidden layer

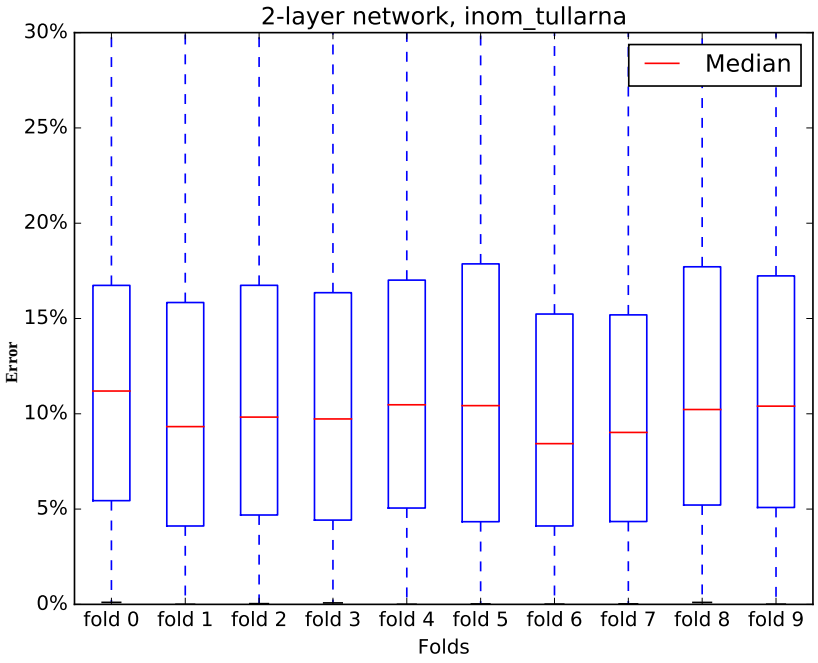

Figure 4.6: Two hidden layers

4.2.2 Small dataset

The hyperparameters obtained from the gridsearch can be seen in table 4.7, for the one- and two-layer networks respectively. The results of the cross validation can be seen in figures 4.7 and 4.8.

One layer		Two layers	
Parameter	**Value**	**Parameter**	**Value**
Learn rate	0.01	Learn rate	0.01
Act. function	ReLU	Act. function	LeakyReLU
Batch size	10	Batch size	25
Nodes per layer	1024	Nodes per layer	128
Dropout	0.1	Dropout	0.0

Table 4.7: Hyperparameters from grid search

The one-layer architecture had a mean loss of 9.06, while the two-layer network had a mean loss, at 9.29. The results can be seen in table 4.8.

One layer		Two layers	
Metric	**Value**	**Metric**	**Value**
Mean loss	9.06	Mean loss	9.29
Loss std	0.71	Loss std	0.62
Mean std	6.71	Mean std	6.93
Std of mean std	0.58	Std of mean std	0.57
Max loss	9.99	Max loss	10.31

Table 4.8: Cross validation results

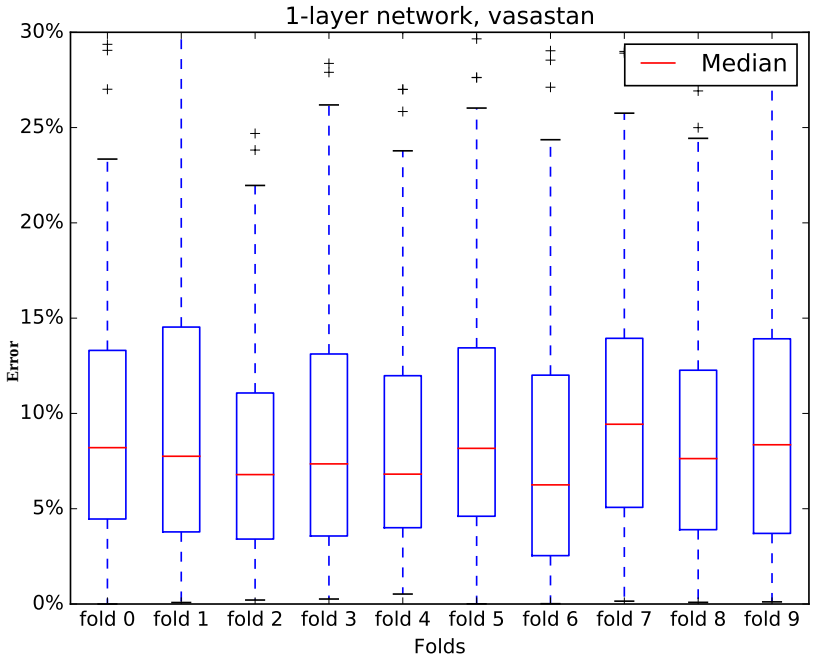

Figure 4.7: One hidden layer

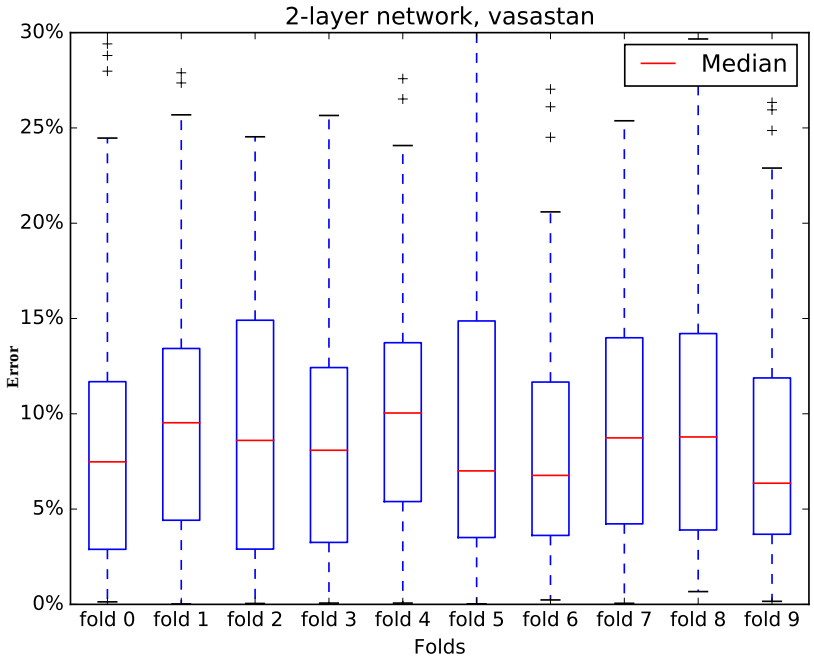

Figure 4.8: Two hidden layers

4.3 Listing price comparison

The data used for the cross validation was also analysed for the differences between the initial listing price and the final selling price for each dataset. The distribution of the differences can be seen in figure 4.9. The mean absolute differences were 8.62 percent for the "Inom tullarna" dataset, and 8.77 percent for the "Vasastan" dataset.

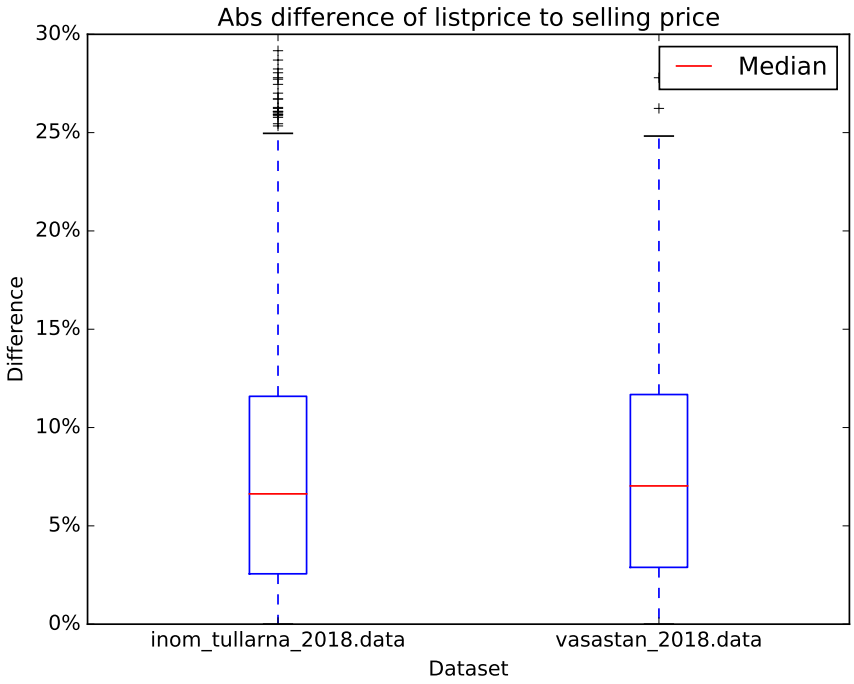

Figure 4.9: Error of listing price compared to selling price

Chapter 5

Discussion

In the first part of this chapter, the results of the study will be analysed, and then compared to the results of other similar studies. Then comes a critical evaluation of the study, followed by some comments on the ethical aspects of the outcome, as well as some possible extensions to the study.

5.1 Result analysis

The aim for this thesis was to build a model capable of prediction of selling prices for apartment sales. Models were examined in two phases, beginning with the models that use the listing price as one of its parameters, and finishing with the models that exclude the listing price. Two datasets were used, one covering most of the Stockholm city centre, and another covering one of the Stockholm city centre districts, *Vasastan*. For each phase and each dataset, two models were constructed, with one and two hidden layers respectively.

5.1.1 Disclosed listing price

Out of the models with access to the listing price, the 1-layer networks performed better than the 2-layer networks for both datasets. The results of the 1-layer networks are similar between the two datasets, with all statistics showing low differences between the models. Likewise, the 2-layer networks show the same similarities between themselves, with a similar offset compared to the 1-layer networks. With a mean loss of 1.15 and 1.11 percentage points higher than their 1-layer ver-

sions, for the large and small dataset respectively, the deeper networks show an obviously inferior performance. The large variance between the folds is the clearest difference between the 1-layer and 2-layer networks, with the standard deviation of the mean loss 3-4 times as high for the 2-layer networks compared to the the shallow ones.

5.1.2 Hidden listing price

As with the models working with the disclosed listing price, the 1-layer networks had a lower mean prediction loss than the 2-layer networks, although with a much smaller margin, at 0.18 and 0.23 percentage points for the large and small datasets respectively. The 2-layer networks have a slightly lower variance than the 1-layer networks, and the architecture with the lowest max loss is one of the 2-layer architectures. There is a notable difference in performance between the two datasets, with the loss of the large dataset networks almost 2.5 percentage points above their small dataset counterparts. This is reinforced by the means of the individual standard deviations, which are around 9 for the large dataset, but under 7 for the small dataset, indicating a lower spread of the predictions.

5.1.3 Comments

For all models using an included listing price, the mean absolute error was less than the difference between the listing price and the selling price, with a margin of around 2.2 and 1.1 percentage points, for the 1- and 2-layer networks. A prediction by the model is, on average, closer to the selling price than the listing price was, which was one of the goals with the thesis. Looking at the max loss, even the worst folds of the 1-layer networks are around 1.5 percentage points better estimations than the listing price.

From the results it is clear that the exclusion of the listing price significantly increases the error of the prediction. The best of the models without listing price predict with a mean error of almost 2.7 percentage points more than the best of the models with listing price. It is, however, only about 0.3 percentage points higher than the average difference between listing and selling prices, indicating a reasonable valuation for the best of the models.

When looking at the models' results compared to other studies, we

will primarily take the hidden listing price models into account, as the models from the mentioned studies generally do not have access to any listing prices. Compared to the results achieved by Pace, Sirmans, and Slawson [15], the models perform acceptable. Both of the small dataset networks beat the traditional models, and produce a similar result to the statistically based model at around 9 percent mean absolute error. The study by Abidoye and Chan [2] featured a single hidden layer neural network, that produced a mean absolute error rate of 16 percent. At that rate, it is beaten by all four networks in this thesis, although it could be due to a more difficult market for prediction. In the mentioned study, the standard deviation of the prices are more than 100 percent of the mean, compared to around 50 percent for the data in this thesis. The larger spread might be an indicator of an overall higher variance and uncertainty in the sales, which could make it more difficult to predict. Continuing, the models in this thesis do not match the performance of the model created by Del Giudice, De Paola, and Forte [8], which managed a mean absolute error of under 7 percent. Overall, the stability of the cross validation folds, underlined by the relatively low standard deviation, indicate a good performance.

The higher errors for the models without listing price is possibly, at least in part, due to the listing price being based on some additional parameters not available to the model, and it thus containing information not covered by the other parameters. Another possible factor behind the higher error rates could be the networks not being trained optimally. The fact that the 2-layer networks regularly perform worse than their 1-layer counterparts also indicate that the training part might have been sub-optimal. This could be due to bad hyperparameters, finding a local minimum during training, or stopping the training too early. The large variation in hyperparameters for the different architectures support the idea that they could have been better optimised.

For the models without listing price, the large dataset proved more difficult to predict. This could be due to the dataset containing apartments from several districts, with a wider range of price levels for the apartments. This can be seen in the analysis of the data done in chapter 3, where some statistical measures are given for the price per m^2. The mean, median and standard deviation is similar for the two datasets, but the minimum and maximum values show a large difference. The min-max spread is almost 170 kSEK for the large dataset, but

less than 110 kSEK for the smaller dataset, with a 30 kSEK difference for both the minimum values and the maximum values. The parameters provided to the model would not give enough information about the price level of the relevant district, causing an uncertainty that is not present for the small dataset models.

Some uncertainty about the result comes from the way Booli gathers the data on selling price. Some of the entries for selling prices are not reported officially by the brokers, but are instead the last registered bid for the apartment in question. If no bids are registered, the listing price is recorded as the selling price as well. As a consequence, some sales, such as instances with no disclosed bidding, would not have the correct selling price. It is not clear whether this would have had a positive or negative effect on the result, if at all.

5.2 Critical evaluation

As was previously mentioned, the fact that the dataset is partly unreliable with regard to a subset of the selling prices is undesirable. While it is unclear what effect this has on the results, it still causes some loss of credibility. However, for the available dataset, there was no way to distinguish the unofficial selling prices from the official records.

A possible issue is the number of hidden nodes in the network compared to the amount of data available for training. There are many rules of thumb concerning the ratio between the number of hidden nodes (and, as a consequence, the number of weights present in the network) and the size of the available dataset [18]. The examples include formulas based on the inputs and outputs of the network ("hidden units = $(n_{input} + n_{output}) * 2/3$") or the amount of data ("data size should be 10 times the number of hidden nodes"). While these rules of thumb have not been fully followed in this study, the optimal architecture also depends on the type of problem, and other techniques used for the model. While a too high number of hidden nodes can lead to overfitting, it is recommended to use a large number of them when using early stopping as regularisation [19]. A large number of hidden nodes can also make a network better at generalisation [12]. Taking into account the ratios used in other studies on real estate appraisal [2, 8], the ratios in this thesis should be regarded as reasonable.

While the results of this study were found to be reasonable on av-

erage, the results that were emphasised were the mean errors. Realistically, for real life usage it would probably be more interesting to look at the worst case prediction results. A large number of outliers would make the predictor unviable for many applications. Instead, it could be interesting to look at a minimisation of the maximum error. To expand on this, it would be more useful to give a probable price range prediction, rather than an expected selling price. The emphasis on the mean values in this thesis also translate to the plots of the results. The y-axis showing the error rate is capped at a 30% limit, despite some of the results exceeding this. This was a result of a trade-off between readability of the plots and an exhaustive representation of the results. While this was not intended to be deceptive, a comment on this was required.

It would also have been interesting to compare the results of the MLP with at least one other architecture, for an additional measure on the performance of the predictive model. It was, however, decided to focus solely on the MLP architecture, partly due to time frame constraints, and partly due to there being a lack of studies found during the pre-study that used other architectures. This has some drawbacks, as the results of this study must rely on comparisons to other studies combined with pure analysis of the result statistics. In hindsight, at least one additional architecture would have been preferable.

5.3 Ethics and sustainability

There are some ethical aspects to address, that would be possible in the case of an accurate selling price prediction model. The most obvious aspect is the possibility to prevent the use of "lockpriser". According to Fastighetsmäklarinspektionen, in 2018 there were 133 reports of real estate brokers using deceptive pricing information, so called "lockpriser" [9]. With a reliable selling price prediction, listing prices could be compared to the predicted selling prices, and listings with large deviations could be marked as suspicious. This could be done to promote a fairer and more open market with less deviation between the listing and selling prices. The downside is the possible lower demand of real estate brokers. If the predictive models prove accurate enough, they could be assigned the appraisal tasks usually done by the brokers.

An accurate model could also be beneficial with regard to the time and resources that is spent on the housing market. In a 2017 study [14], the cost to society from "lockpriser" is estimated as being around 1 billion SEK, due to the additional time spent by buyers while searching for a potential buy. With a selling price within a smaller margin, prospective buyers are able to focus on just the price ranges that are suitable for them.

5.4 Further research

To build on the differing results between the two datasets, it could be examined if this would be replicated for other districts as well. It could also be interesting to see how the results would fare compared to those for a dataset encompassing an even larger area. The better results on the smaller dataset in this study could indicate that there exists some difficulty for the model to accurately use a generalised prediction across the whole of the larger dataset. If instead a separate model was trained for each district, they could be combined in an ensemble predictor, possibly managing a better performance than a single model trained on the whole dataset.

Although this study only looked at data from a single year, 2018, there is several more years of data available through the Booli API. It could be interesting to investigate whether an increase in the size of the data outweighs the downside of dealing with older data. It could be expected that including data over a larger period of time would show the rising housing prices and other trends, and it would be interesting to see how they affect the predictions. As an addition, there are some parameters available through the API that were omitted in this study. The inclusion of those could give the models more information about the objects, and accordingly a better basis for prediction.

Chapter 6

Conclusions

This study investigated the potential of predicting selling prices using neural networks. The addition of the listing price was present in some of the models, and had a significantly positive effect on the error rate. For the other models, there were differences in performance depending on the size of the dataset, where the smaller dataset resulted in a lower error percentage. Overall, the models performed in line with was expected, and achieve similar results to related work done on the subject.

Bibliography

[1] Rotimi Boluwatife Abidoye and Albert P. C. Chan. "Critical determinants of residential property value: professionals' perspective". In: *Journal of Facilities Management* 14.3 (2016), pp. 283–300.

[2] Rotimi Boluwatife Abidoye and Albert P. C. Chan. "Modelling property values in Nigeria using artificial neural network". In: *Journal of Property Research* 34.1 (2017), pp. 36–53.

[3] Junchi Bin et al. "Regression model for appraisal of real estate using recurrent neural network and boosting tree". In: *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)* (2017).

[4] Michael Blackstaff. *Finance for IT Decision Makers - A Practical Handbook (3rd Edition)*. BCS The Chartered Institute for IT, 2012.

[5] Booli. URL: `http://www.booli.se/`.

[6] Booli API. URL: `http://www.booli.se/p/api/`.

[7] Gerald R. Brown, George A. Matysiak, and Mark Shepherd. "Valuation uncertainty and the Mallinson Report". In: *Journal of Property Research* 15.1 (1998), pp. 1–13.

[8] Vincenzo Del Giudice, Pierfrancesco De Paola, and Fabiana Forte. "Bayesian Neural Network Models in the Appraisal of Real Estate Properties". In: *Computational Science and Its Applications – ICCSA* (2017), pp. 478–489.

[9] Fastighetsmäklarinspektionen. *Fler mäklare, anmälningar och beslut om påföljder*. 2018.

[10] Fastighetsmäklarinspektionen. *Halvårsstatistik*. 2018.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2017.

[12] Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. "Lessons in Neural Network Training: Overfitting May be Harder than Expected." In: *Proceedings of the National Conference on Artificial Intelligence*. Jan. 1997, pp. 540–545.

[13] Mäklarhuset. *Bo-opinion, Juli 2017*. 2017.

[14] Anders Österling. *Lockpriser på bostadsmarknaden*. SNS Analys nr 45. Dec. 2017.

[15] R. Kelley Pace, C. F. Sirmans, and V. Carlos Slawson. "Are Appraisers Statisticians?" In: *Real Estate Valuation Theory Research Issues in Real Estate* (2002), pp. 31–43.

[16] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408.

[17] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning internal representations by error propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (1985).

[18] Warren S. Sarle. *How many hidden units should I use?* Neural-nets FAQ, Part 3 of 7: Generalization.

[19] Warren S. Sarle. "Stopped Training and Other Remedies for Overfitting". In: *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*. 1995, pp. 352–360.

[20] Svensk Mäklarstatistik. *Bostadspriser i Stockholms län*.

[21] Stéphane Tufféry. *Data Mining and Statistics for Decision Making*. John Wiley & Sons, Ltd, 2011. Chap. 8, pp. 217–233.

[22] Ko Wang and Marvin L. Wolverton. *Real estate valuation theory*. Kluwer Academic Publishers Group, 2002.

[23] Lipo Wang et al. "Predicting public housing prices using delayed neural networks". In: *2016 IEEE Region 10 Conference (TENCON)* (2016).

TRITA -EECS-EX-2019:70