KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science and Engineering

Second cycle, 30 credits

Comparative Analysis of Machine Learning Methods for Predicting Property Prices and Sale Velocities in the Real Estate Industry

LUCAS EREN

Stockholm, Sweden, 2023

Comparative Analysis of Machine Learning Methods for Predicting Property Prices and Sale Velocities in the Real Estate Industry

LUCAS EREN

Master's Programme, Computer Science, 120 credits
Date: June 28, 2023

Supervisor: Kiarash Kazari
Examiner: György Dán
School of Electrical Engineering and Computer Science
Host company: Bonava AB
Swedish title: Jämförande analys av maskininlärningsmetoder för att förutsäga fastighetspriser och försäljningshastigheter inom fastighetsbranschen

Abstract

The real estate industry is one of the largest industries in the world and using data-driven decision-making has been shown to increase companies' profitability. A technique to apply data-driven decision-making is machine learning. Within the real estate industry, predicting property selling prices and sale velocities (the duration a property remains on the market) are crucial factors of interest. Knowing the selling price and the sale velocity can motivate businesses to alter their plans in an effort to increase their profitability. The research conducted in this thesis employs a comparative approach to evaluate the performance of various machine learning methods in predicting both the selling price and the sale velocity of properties. The machine learning methods this study investigated are random forest, decision tree, K-nearest neighbor, support vector regression, and multilayer perceptron. After pre-processing, the data set used comprises 560,000 distinct data points from the Swedish housing market. The data set has a wide geographic scope, covering almost the entire country of Sweden. The data set was subjected to both normalization and standardization techniques in order to determine how they affected the machine learning methods. The results demonstrate that random forest oEutperforms the other machine learning methods in predicting property selling prices. However, the assessed machine learning methods encountered difficulties in predicting the sale velocity. The best-performing machine learning method for sale velocity is random forest. Notably, SVR demonstrates a lower MAE for sale velocity, but performs worse in the R^2 metric.

Keywords

Machine Learning, Real Estate Industry, Selling Prices, Sale Velocities, Comparative Analysis

Sammanfattning

Fastighetsbranschen är en av de största industrierna i världen och att använda datadrivet beslutsfattande har visat sig öka företags lönsamhet. En teknik för att tillämpa datadrivet beslutsfattande är maskininlärning. Inom fastighetsbranschen är förutsägelser av fastigheters försäljningspriser och försäljningshastigheter viktiga faktorer av intresse. Kunskap om försäljningspriset och försäljningshastigheten kan motivera företag att ändra sina planer i syfte att öka lönsamheten. I den forskning som bedrivs i denna avhandling används en jämförande metod för att utvärdera olika maskininlärningsmetoders prestanda när det gäller att förutsäga både försäljningspriset och försäljningshastigheten för fastigheter. De metoder för maskininlärning som undersökts i denna studie är random forest, decision tree, K-nearest neighbor, support vector regression och multilayer perceptron. Efter förbehandling består den använda datamängden av 560 000 distinkta datapunkter från den svenska bostadsmarknaden. Datamängden har en stor geografisk räckvidd och täcker nästan hela Sverige. Datamängden utsattes för både normaliserings- och standardiseringstekniker för att avgöra hur de påverkade maskininlärningsmetoderna. Resultaten visar att random forest överträffar de andra maskininlärningsmetoderna när det gäller att förutsäga försäljningspriser på fastigheter. De utvärderade maskininlärningsmetoderna stötte dock på svårigheter när det gällde att förutsäga försäljningshastigheten. Den bäst presterande maskininlärningsmetoden för försäljningshastighet är random forest. I synnerhet visar SVR en lägre MAE för försäljningshastighet, men presterar sämre i R^2 måttet.

Nyckelord

Maskininlärning, Fastighetsbranschen, Försäljningspriser, Försäljningshastighet, Jämförande Analys

Acknowledgments

I want to express my gratitude to Bonava for giving me the opportunity to write my thesis in collaboration with them. I am immensely thankful to Booli for allowing me to use their data for this master's thesis. Without access to their valuable data set, the completion of this research would not have been possible. I would like to thank my supervisor at Bonava, Helena Sjöberg, for her continuous support, informative discussions, and valuable guidance throughout this project. Additionally, I would like to express my gratitude to my supervisor at KTH, Kiarash Kazari, for his support, insightful feedback, and valuable guidance that contributed to the success of this work. Furthermore, I would like to thank my examiner György Dán, for his academic guidance and insightful feedback. I am also grateful to my family and friends for their support throughout this journey. Last and most important, I would like to express my utmost gratitude to my Lord and Savior, Jesus Christ, for everything he has given me on this journey.

Stockholm, June 2023
Lucas Eren

Contents

1 Introduction 1

1.1 Background . . . 1

1.2 Problem . . . 3

1.3 Purpose . . . 3

1.4 Research Question . . . 4

1.5 Goals . . . 4

1.6 Research Methodology . . . 4

1.7 Delimitations . . . 5

1.8 Structure of the thesis . . . 6

2 Background 7

2.1 Machine Learning . . . 7

2.1.1 Decision Tree . . . 8

2.1.1.1 Random Forest . . . 9

2.1.2 Support vector machine . . . 10

2.1.2.1 Support vector regression . . . 11

2.1.3 K-nearest neighbors classifier . . . 12

2.1.3.1 K-nearest neighbors regression . . . 13

2.1.4 Multilayer perceptron . . . 14

2.2 Methods and techniques related to machine learning . . . 16

2.2.1 K-fold cross-validation . . . 16

2.2.2 Standardization and Normalization . . . 17

2.2.3 Pearson correlation coefficient . . . 17

2.2.4 Hyperparameter . . . 18

2.2.5 One hot encoding . . . 19

2.3 Related work . . . 19

2.3.1 Finding low-priced houses with machine learning . . . 20

2.3.2 Price estimation of real estate in France . . . 21

2.3.3 Predicting real estate prices in Taiwan . . . 23

2.3.4 Sales forecasting in e-commerce 24

3 Method 25

3.1 Research Process 25

3.2 Data Collection 26

3.3 Data Set 27

3.3.1 Pre-processing of the Data Set 27

3.4 Tools and Environment 29

3.5 Machine learning methods 30

3.6 Hyperparameter tuning 30

3.7 Machine Learning Metrics 32

4 Results and Analysis 34

4.1 Feature Correlation 34

4.2 Sold Price 35

4.2.1 Default hyperparameter 35

4.2.2 Default hyperparameter with StandardScaler 36

4.2.3 Default hyperparameter with MinMaxScaler 37

4.2.4 Hyperparameter Optimization with Random Search Sold Price 37

4.3 Sale Velocity 39

4.3.1 Default hyperparameter 40

4.3.2 Default hyperparameter with StandardScaler 40

4.3.3 Default hyperparameter with MinMaxScaler 41

4.3.4 Hyperparameter Optimization with Random Search Sale Velocity 41

5 Discussion 44

5.1 Limitations 44

5.1.1 Data set 44

5.1.2 Pre-processing 45

5.1.3 Normalization and Standardization 45

5.1.4 Hyperparameter 45

5.1.5 Tools and environment 46

5.1.6 Convergence 46

5.2 Ethics and Sustainability 46

5.3 Practical use of the findings 47

6 Conclusion 48

6.1 Future work 49

References **49**

Chapter 1

Introduction

1.1 Background

The real estate industry market capitalization was reported to be worth 3.69 trillion dollars in the year 2021 [1]. This fact highlights the real estate industry as one of the largest in the world, and as such, an important and interesting sector to research. As in almost all industries, profitability is one of the most important goals for companies, and the larger the profitability is, the better. Meanwhile, reports have shown that companies that characterize themselves as data-driven when making decisions were on average 6% more profitable and 5% more productive than their competitors [2, 3]. One tool to utilize when implementing data-driven decision-making is machine learning. In general terms, machine learning uses computational algorithms to learn and improve its own analysis regarding a subject. In order to generate accurate analysis, machine learning algorithms require a large data set as input so they can learn as much as possible about a field. Some analysis that machine learning algorithms can make is that they can recognize patterns and make predictions. Some machine learning algorithms perform better than others depending on the use. To know how well a machine learning algorithm performs, the accuracy of the outputted data is often compared to real data that the algorithm has not used while learning [4]. The use of machine learning in the real estate industry is therefore interesting and can give companies an advantage compared to their competitors.

As the real estate industry is a large field, there are plenty of ways to increase productivity and profit. One field that is quite common in the real estate industry and where some research has been conducted is making price predictions on properties with machine learning [5]. Another field that is important but has received less attention is sales velocity (the duration a property remains on the market) in the real estate industry. Sales velocity in the real estate industry is important because it gives companies cash flow. According to a study, cash flow is the most important factor that affects profitability in the construction business. For larger construction businesses it is even more important because those companies handle several projects simultaneously [6]. Cash flow is needed so construction companies can proceed with new developments. If a company sells its properties at a faster pace, it will be able to proceed and create new properties that it can sell, leading to a potential increase in overall profits. This concept of financing a construction project is called retained earnings. In addition to retained earnings, the other two main financing choices are debt capital and equity capital. Debt capital can be obtained through private bank loans, those loans are paid back with principal and interest. Equity capital is when companies raise funds in exchange for a portion of the company in the form of shares. While equity capital does not require interest payments, it comes with two drawbacks, the profit will be shared with a larger portion and the shareholders will dilute the company's ownership control [7]. In Sweden, the past decade's interest rate for debt capital through bank loans has been quite low. This has enabled businesses to take out large loans with low interest rates, facilitating the start and completion of projects. With interest rates for debt capital increasing, it has become imperative for corporations to manage their cash flow prudently. With higher interest rates, corporations stand to lose a larger portion of their profit the longer it takes to get cash inflow that repays the loan. As a result, sales velocity has become increasingly important in the real estate industry.

Therefore, this project will investigate how well machine learning methods perform in regard to the sold price and sale velocity in the Swedish real estate industry. This will be done with the help of five different machine learning methods and various features in the data set to gain a better understanding. The aim is to investigate if a machine learning method can give practical insights and recommendations to companies, ultimately leading to improved profitability. Some of the various features are location, living area, number of rooms, and year of construction. Some machine learning methods of interest that will be used in this project are support vector regression, random forest, and multilayer perceptron.

The project is in cooperation with Bonava, Bonava is a residential development company in Northern Europe. It is interesting for Bonava to know how well machine learning methods can predict the selling price and the sale velocity for different properties in different locations in Sweden. Bonava may be able to use this information to plan more profit-oriented.

1.2 Problem

With higher interest rates for debt capital, corporations need to handle their cash flow more strategically to avoid any adverse impact on their profits. Residential development companies and construction projects often use debt capital to initiate and proceed with their projects. With higher interest rates, the time it takes from the start of the project to the selling of the properties is important. The longer time it takes the less profit will the corporation make. An essential component of the process is the time it takes to sell a property and receive cash inflow. Managing cash flow effectively entails comprehending the dynamics of the real estate market, knowing what a property will sell for, and how long it will take to sell the property.

Sales velocity is important for the real estate industry, but it is of course also important for other industries. The significance of sales velocity has not been adequately addressed in the existing research pertaining to machine learning. Additionally, there has not been much research about what machine learning method to use in regard to the sold price in a large geographical area and the Swedish housing market. Furthermore, among the many methods available, it is hard to know what machine learning method to use in certain areas. Therefore, it needs to be evaluated how well different machine learning methods work in regard to sold price and sale velocity in Sweden.

1.3 Purpose

The purpose of this thesis is to compare and investigate the performance of five different machine learning methods as predictive measures for sales velocity and sold price of properties in Sweden. The findings from this research will contribute to the field of real estate by providing insights into the optimal machine learning methods for enhancing the understanding and prediction of selling prices and sale velocities in the real estate market.

The thesis will mainly benefit the real estate industry by providing insight into predicting property prices and sales velocities with machine learning. However, the prediction of prices and sales velocities could be of interest to multiple industries. This paper can serve as a basis for similar research in other industries where price and sale velocity are of importance.

This project is done in collaboration with Bonava. The research will help Bonava understand how well machine learning methods can work in the real estate industry. Bonava is interested in knowing how well machine learning methods can predict both the selling price of a property and how long it takes for a property to sell (sale velocity). Having a clear understanding of which machine learning method performs best can subsequently be applied in their work to strategically plan their projects, knowing both what properties might sell for and for how long it would take for them to sell.

1.4 Research Question

The research questions that will be investigated in this thesis are:

1. How well do machine learning methods perform in regard to the sold price of properties in Sweden, and which method outperforms the others among the evaluated approaches?
2. How well do machine learning methods perform in regard to the sale velocity of properties in Sweden, and which method outperforms the others among the evaluated approaches?

1.5 Goals

The goal of this project is to determine the most accurate machine learning method regarding the area of real estate with a focus on price and sales velocity. This has been divided into the following three sub-goals:

1. Fetch, clean and prepare the data set.
2. Find the machine learning method that is most effective in regards to price prediction.
3. Find the machine learning method that is most effective in regards to sales velocity prediction.

1.6 Research Methodology

The data set for this thesis will be gathered from Booli. Booli is ”Sweden’s largest combined range of homes for sale”. They have a large and informative data set which is essential when using different machine learning methods. The large data set will increase the reliability of the results.

The library scikit-learn will provide the thesis with the machine learning methods. Consequently, the programming language Python will be used throughout the project. One method used in this thesis is that each machine learning method will utilize the technique of five-fold cross-validation. This is used to decrease the variance of the prediction of the machine learning methods. Furthermore, standardization and normalization will be tested with the machine learning methods to see if the performance of the machine learning methods is affected. Thereafter different hyperparameters will be used to see how well the machine learning methods perform in regard to sold price and sale velocity. There will be 4 different metrics that will evaluate each machine learning method.

The different machine learning methods that will be used in this thesis are:

- Random Forest Regression
- Decision Tree Regression
- Support Vector Regression
- K-Nearest Neighbors Regression
- Multi Layer Perceptron Regression

These different machine learning methods are both interesting and have performed well in other reports.

1.7 Delimitations

- Bonava is active in multiple countries in northern Europe, for example, Sweden, Germany, and Finland. The thesis will only cover the Swedish real estate industry. The result and method could however be applied to other markets as well. Thus one limitation is that the thesis will only handle data from the Swedish real estate industry.
- One limitation of the thesis is that there will only be a limited amount of machine learning methods that will be compared. There are a lot of machine learning methods and all of them can not be tested.
- Only apartments, villas, and terraced houses will be in the data set.

1.8 Structure of the thesis

Chapter 2 presents the relevant machine learning methods that will be utilized in the thesis. In addition, the chapter explains different methods and techniques that can be used within the area of machine learning. Finally, the chapter presents related works in the field of machine learning and sales predictions. Chapter 3 starts by explaining the research process of the thesis and then proceeds with explaining how the data set was collected and what pre-processing steps were performed on the data. Furthermore, the chapter describes the tools and environment used in the thesis, the applied machine learning methods, and the hyperparameters involved. Lastly, the chapter explains what metrics the machine learning methods would be evaluated on. Chapter 4, starts with presenting the Pearson correlation matrix to give some insight into the specific features. Thereafter the chapter presents the results of how well each machine learning method performed, starting with the default values and ending with different hyperparameters. Chapter 5 is regarding the discussion and what limitations that has affected the result. Additionally, the chapter also presents practical use of the findings and future work. Lastly, chapter 6 presents the conclusion of the thesis.

Chapter 2

Background

This chapter provides background information about machine learning and different machine learning methods. Additionally, the chapter describes some techniques and methods that could be used in machine learning. Lastly, this chapter presents related work about machine learning in the real estate industry.

2.1 Machine Learning

Machine learning is computational methods that use fundamental concepts in computer science in combination with statistics, probability, and optimization to make predictions and improve performance for a given task. The machine learning methods use experience in the form of a data set, S, to train the methods to make the predictions [8]. The content of a data set typically consists of observations [9]. Each observation is often made up of a variety of features, V, that describe the observation. These individual observations are commonly referred to as data points. The data points in the data set are represented as $(x_1, x_2, ..., x_N)$, where N is the number of data points, and each data point is a vector with a dimension of $|V|$. A data set can be in many forms, but the size is crucial for good performance in machine learning [8].

There exist different types of machine learning, two of them being supervised and unsupervised machine learning. Supervised machine learning is when the machine learning method receives a data set where the data is labeled. Supervised machine learning is the most common regarding classification and regression problems. Unsupervised machine learning on the other hand is when the data is not labeled. With no labeled data, it is harder to evaluate how well the machine learning model performs because there is no direct measure of correctness or accuracy [8].

The goal when using machine learning is to create efficient and accurate predictions. Two common objectives in machine learning are classification and regression. Classification in machine learning is when the prediction is regarding assigning a category, o, to each data point, from the set of categories, O. Whereas regression is about predicting a continues value for each data point. The penalty for an incorrect value is calculated differently for the different objectives, in regression the penalty depends on the difference between the predicted, $f(x)$, and true value, y, whereas for classification a prediction is either right or wrong [8]. The focus of this thesis is on the objective regression.

2.1.1 Decision Tree

Decision tree is a supervised machine learning method that is used for regression and classification problems. The decision tree is a hierarchical top-down method that uses iterative splits for decision-making. Working as a hierarchical top-down method, the first node is at the top of the graph and is called the root. Thereafter, the root is divided into two decision nodes depending on a characteristic from the given data set [10]. This procedure continues until a stopping criterion is met, it can for example be a max depth or a minimum number of data points for each leaf node [11].

The decision tree utilizes discrete splitting functions that are split on the value of one single feature, v. Consequently, the decision tree will search for the best feature to split on. There are different metrics that can be used when searching for the best feature to split on. One metric is information gain which uses entropy from information theory as the impurity metric [11]. Entropy values can be in the range of 0 to 1, the smaller the value, the more predictable the system is. If all data in a data set belongs to one class the entropy for that data is zero, whereas if the data is split evenly between two classes the entropy is one. Information gain shows the difference in entropy before and after a split is made on a given feature. The feature that has the highest information gain will be the best one to split on as it will classify the data best. The set of values the feature v can take is denoted by D_v. The formula for entropy and information gain is as follows [12]:

$$I(S) = -\sum_{o \in O} p(o) log_2 p(o) \tag{2.1}$$

$$IG(S, v) = I(S) - \sum_{i \in D_v} \frac{|S_i|}{|S|} I(S_i) \tag{2.2}$$

Formula for Entropy and Information Gain

- p - The empirical probability of a specific outcome or class
- $\frac{|S_i|}{|S|}$ - Represents the proportion of data points in the data set that has the specific value i for a given feature v.

A dilemma occurs when creating a decision tree. If hard stopping criteria are used it tends to create an under-fitted decision tree, in contrast, if the stopping criteria are loose the decision tree is over-fitted to the training data set. To solve this issue, the technique of pruning is employed. Pruning is accomplished through a two steps process, firstly a loosely stopping criterion is used to let the decision tree overfit the training data set. Secondly, the decision tree is cut into a smaller tree by removing sub-branches that do not contribute much to the generalization accuracy. Studies have shown that pruning can improve the performance of a decision tree, particularly when the data set is noisy [11].

2.1.1.1 Random Forest

Random forest is in simple words a cluster of decision trees where each decision tree is constructed with a degree of randomness. Every decision tree in the random forest uses the technique of bagging, meaning that a random subset of the data set is chosen, where specific data points might be used more than once [13]. Bagging reduces the variance of the data set and will only increase the bias slightly [11]. Additionally, in contrast to a single decision tree, which evaluates all the features at each node to find the one with the highest information gain to split on, in a random forest, the decision trees are randomly selecting a subset of all the features at each node. This improves the prediction power because it reduces the correlation between the decision trees. Furthermore, the technique of pruning is not used in the random forest, instead, the trees are grown to their largest extent [13]. Finally, when using the random forest, each decision tree will have a “vote” on what the data output should be. The average or majority of all votes will be the finalized output for the random forest [11]. The random forest is most of the time performing better than a single decision tree and two of its advantages are that it overcomes the problem with overfitting and it is less sensitive to outlier data in the training set [13].

2.1.2 Support vector machine

Support vector machine, SVM, uses a hyperplane to classify the data set, the hyperplane can both be linear and nonlinear. The hyperplane is used to determine how unseen data points should be classified. The optimal hyperplane is the one that can divide the data points into their respective classes with the maximum margin [14]. The weight vector, w, and bias, b, are used to find the optimal decision boundary. One metric that is often used in the measurement of distance in SVM is the Euclidean distance, $||w||^2$ [15].

$$min\frac{||w||^2}{2}, \text{ subject to: } o(w * x_i + b) \geq 1 \tag{2.3}$$

Formula for the maximum margin hyperplane

The margin is defined as the distance from the separating hyperplane to the nearest data point. Using the hyperplane that has the maximum margin maximizes the ability to predict the class of unseen data. A problem that could occur is that the data set has some noisy data which will affect the SVM negatively. To deal with noisy data, a technique called soft margin is used [16]. Soft margin relies on slack variables that describe each data point, the slack variables are often represented by ξ, where ξ can be a value from 0 to 1. Where ξ = 0 means that the data point is correctly classified. Furthermore, a penalty factor C is also introduced, its application is to give a penalty on the slack variables to measure the trade-off between hyperplane complexity and training errors, while it also reduces the risk of overfitting [14]. Kernel functions are used when the data set cannot be separated with a linear hyperplane. The kernel functions maps data to a higher dimensional space to be able to separate the data into classes [16].

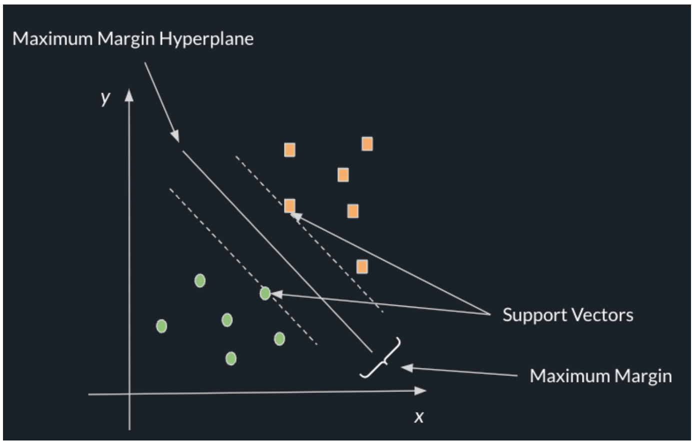

Figure 2.1: Illustration of an SVM

2.1.2.1 Support vector regression

Support vector regression, SVR, is an extension of SVM. The main difference between the SVM and SVR is that SVR handles regression problems that allow the method to output estimations, whereas SVM produces class labels as output. SVR has some advantages in comparison to other regression methods by utilizing kernels, which makes it possible to handle nonlinear regression problems [17].

$$L(y, f(x)) = \begin{cases} 0 & \text{if } |y - f(x)| \leq \varepsilon \\ (|y - f(x)| - \varepsilon)^2 & \text{otherwise} \end{cases} \tag{2.4}$$

Formula for the linear ε-insensitive loss functions

In difference to SVM where the main goal is to find a hyperplane that can divide the data set into classes, SVR uses an ε-insensitive loss function to determine a hyperplane that can predict the outputs with at most an ε deviation to the actual value. The optimal version is to minimize the ε-insensitive as much as possible while still containing most of the data points. The ε-insensitive loss function penalizes a prediction if it is farther away from the desired output than ε. As in SVM and its use of a soft margin, SVR is using slack variables to guard against noisy data. The slack variables determine the number of data points that are tolerated to be outside of the ε-insensitive tube. Furthermore, a parameter in SVR is the regularization parameter, C, which determines the trade-off between the prediction errors and the flatness of SVR. If the value of C is large, SVR will try to minimize the training error, resulting in a smaller margin and a more precise fit to the data set. However, the increased precision can come at the cost of potential overfitting the data set. Conversely, if the value of C is small, SVR will try to minimize the flatness of the function, consequently creating a larger margin where more data points can be within the margin. This makes the SVR function less complex but it can also result in underfitting [17].

2.1.3 K-nearest neighbors classifier

K-nearest neighbors, KNN, builds upon the idea that the nearest data points to a target data point provide the most useful information about it in the feature space. Consequently, the target data point will be assigned to the class that is most frequently represented among its k-nearest neighbors. For a data point x' and the data set of its nearest neighbor, $N_K(x')$, it will use the k nearest neighbors y label to predict the label of x'. If the objective is regarding binary classification and the label set is $Y = \{1, -1\}$, the K-Nearest Neighbors is defined as [18]:

$$f_{\text{KNN}}(x') = \begin{cases} 1 & \text{if } \sum_{i \in N_K(x')} y_i \geq 0 \\ -1 & \text{if } \sum_{i \in N_K(x')} y_i < 0 \end{cases} \tag{2.5}$$

Formula for binary K-Nearest Neighbor classification

A metric that is widely used in R^q to calculate the distance between the data points is the Minkowski metric [18]:

$$||x' - x_j||^p = (\sum_{i=1}^{q} |(x_i)' - (x_i)_j|^p)^{\frac{1}{p}} \tag{2.6}$$

Formula for Minkowski metric

The value of k in KNN is of importance and decides how a data point should be labeled. If k is equal to 1, it will create a lot of small regions in a noisy data set, where larger patterns will be unnoticed. If k is a larger value, for example, 20, smaller patterns in some locations will be ignored to instead focus on a larger area. Thus, a smaller value for k tends to overfit the data whereas a larger value for k will generalize the data and can underfit it [18].

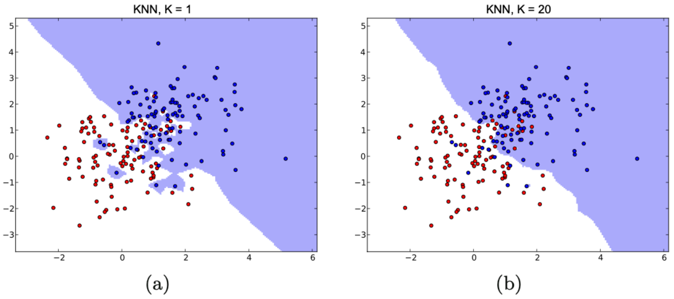

Figure 2.2: Illustrates how KNN classification works for k=1 and k=20 [18]

2.1.3.1 K-nearest neighbors regression

KNN regression is quite similar to KNN classifier. Whereas KNN classifier predicts what class a data point should be assigned to depending on its nearest neighbors, KNN regression computes the mean of the function values of the k-nearest data points [18]:

$$f_{\text{KNN}}(x') = \frac{1}{K} \sum_{i \in N_K(x')} y_i \tag{2.7}$$

Formula that predicts the value for x'

Similar to the classification, if k is a small value the regression will overfit the data and if k is a large value it will underfit the data [18].

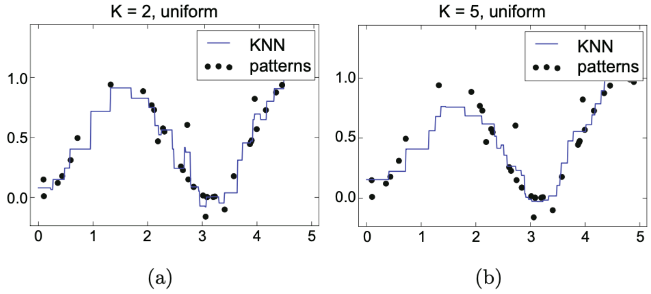

Figure 2.3: Illustrates how KNN regression works for k=2 and k=5 [18]

A problem that occurs in KNN is that it induces locally constant outputs, which can lead to a loss of accuracy in the predictions. A technique that solves this issue is the use of weight in the algorithms. The idea is that the data set of the k-nearest neighbors should contribute differently to the output value. The closer the data points in $N_K(x')$ is to the target data point the more should it contribute to the prediction of the target value. This results in the algorithm interpolating between data points in contrast to the algorithm that does not use weight [18].

$$f_{\text{wKNN}}(x') = \sum_{i \in N_K(x')} \frac{\Delta(x', x_i)}{\sum_{j \in N_K(x')} \Delta(x', x_j)} y_i \tag{2.8}$$

Formula for KNN with weight

$$\Delta(x', x_i) = \frac{1}{||x' - x_i||^2} \tag{2.9}$$

Formula for distance with weight.

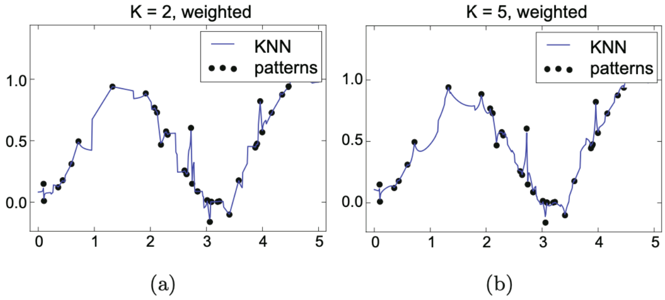

Figure 2.4: KNN regression with weight [18]

2.1.4 Multilayer perceptron

Artificial neural networks, ANN, is a machine learning method built upon the idea of imitating the human brain. As the human brain, ANN, has neurons that send signals to each other to come to a conclusion about a given task. ANN can handle both linear and non-linear problems by receiving a detailed data set where the algorithm can learn how the features affect each other [19]. To understand the structure of ANN it is simplest to divide the algorithm into three parts, the input layer, the hidden layer, and the output layer. The input layer's main task is to handle the input data from the data set, where the data is often normalized before using the ANN. The hidden layer is the part that makes most of the algorithm's calculations and is built on multiple neurons. The hidden layer can be a single hidden layer or multiple layers where each layer works as a separate processing stage. In a single hidden layer, each neuron receives input data from the input layer and produces an output that works as an input for the output layer. In the case of multiple hidden layers, each neuron receives input data from the layer before and is producing the input data for the next layer until reaching the output layer, see figure 2.5. The output layer can also be a set of neurons and the main responsibility is presenting the final output [20]. As each neuron receives multiple inputs, t, from the layer before, a weighted sum, z, is calculated to later be used for the activation function that calculates the output for each neuron. For each connection between the neurons a weight, w_i, is applied which determines the impact of each input on the activation function. Additionally, a bias term, b, is included in the weighted sum to further adjust the neuron's response [21].

$$z = \sum_{i=1}^{n} w_i * t_i + b \tag{2.10}$$

Calculation of the weighted sum for a neuron

One type of ANN is the multilayer perceptron (MLP). In MLP the weight is calculated and adjusted using a gradient descent backpropagation algorithm. MLP begins the training process with a set of initial weights that are chosen at random. The training continues until the predefined error threshold is met, and the error is calculated by comparing the difference between the output value and the target value. Furthermore, the most used activation function is the sigmoid function and is as follows [22]:

$$s(z) = \frac{1}{1 + e^{-z}} \tag{2.11}$$

Formula for the activation function sigmoid

Other activation functions are no-op activation (identity), hyperbolic tan function (tanh), and the rectified linear unit function (relu) [23].

$$i(z) = z \qquad t(z) = tanh(z) \qquad r(z) = max(0, z) \tag{2.12}$$

Formula for the activation functions identity, tanh, and relu.

Furthermore, it is possible to use learning rates in MLP regressor. The learning rates are affecting the weight updates in the machine learning method. A higher learning rate will update the weights more aggressively during the training phase, whereas a lower learning rate will update the weights slower, and more cautiously. Additionally, it is also possible to use different solvers for weight optimization. LBFGS, SGD, and ADAM are three different solvers in MLP, where LBFGS is based on quasi-Newton, SGD is stochastic gradient descent and ADAM is a stochastic gradient-based optimizer that uses optimizer pointers [23].

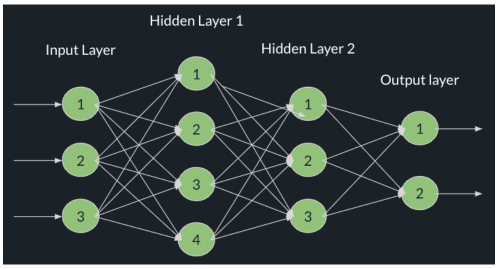

Figure 2.5: Illustration of Artificial Neural Network

2.2 Methods and techniques related to machine learning

2.2.1 K-fold cross-validation

Cross-validation is a technique widely used to decrease the variance of the computed prediction error of the tested machine learning models. Cross-validation prevents over- and underfitting that could occur when training the machine learning models. Cross-validation belongs to the Monte Carlo methods. K-fold cross-validation is a technique where the machine learning method is trained and tested K times. The data set is divided into K subsets, where $K-1$ subsets are used for training and 1 subset is used for validation. The procedure is repeated until each subset has served as a validation set. The performance of each iteration is summed up to finally be divided by K to get the average performance of the machine learning method [24].

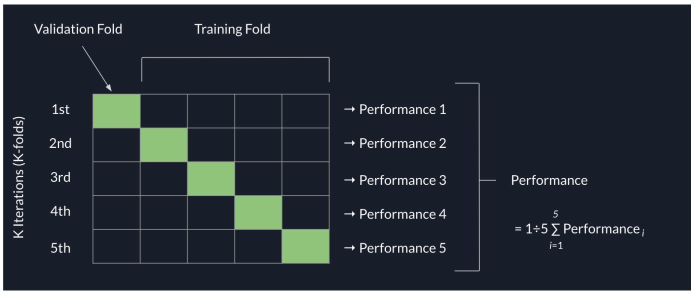

Figure 2.6: Five-fold cross-validation

2.2.2 Standardization and Normalization

Standardization and normalization are two standard machine learning techniques used to re-scale the data set to make the processing easier for the machine learning methods [25, 26]. Using normalization or standardization can boost the performance of machine learning methods. According to [25], the utilization of the techniques boosts the performance by 5-10% in general.

Normalization is the process of casting each feature, v, in the data to a specific range, often, 0 to 1 [26]. One normalization technique that is widely known is the MinMaxScaler. MinMaxScaler sets the lowest value for a feature to 0 and the largest value for the same feature to 1. The rest of the values will be a decimal number in between those values [25].

$$v_{\text{scaled}} = \frac{v - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} \tag{2.13}$$

MinMaxScaler

Standardization is when the mean, μ, of the feature will be set to 0, and all the feature values will lay within one standard deviation, σ [26]. One standardization technique is the StandardScaler [25].

$$Z_{\text{scaled}} = \frac{x_i - \mu}{\sigma} \tag{2.14}$$

StandardScaler

2.2.3 Pearson correlation coefficient

Pearson correlation coefficient is a statistical method that measures the linear relationship between two continuous variables. There exists multiple different formulas to express the correlation coefficients and the most commonly used is the covariance. For two features, with their data set A and B, the formula is [27]:

$$p_n = \frac{E(AB) - E(A)E(B)}{\sigma_A \sigma_B}, \tag{2.15}$$

Pearson correlation coefficient with covariance.

where E(AB) represents the average value of the product of A and B. E(A) and E(B) represent the average value for the specific feature. σ_A and σ_B represent the standard deviation of the features [27].

The value for the Pearson correlation coefficient can vary between -1 and 1. A positive correlation between the features is implied by a value greater than 0, and the closer the value is to 1, the stronger the correlation is. Correspondingly, if the value is lower than 0 it implies a negative correlation and the closer the value is to -1, the more negatively correlated are the features [27].

2.2.4 Hyperparameter

Each machine learning method has multiple hyperparameters that can be changed which directly impacts the model's performance. Some hyperparameters are for example the C parameter in SVR, the K in KNN and the number of decision trees in a random forest. There are various techniques to conclude how well a machine learning method performs because there is a large number of different combinations of the hyperparameters. Manual testing of different hyperparameter values is a traditional way of tuning machine learning methods. However, manual testing can be inefficient because there are a large number of combinations that can be created with the hyperparameters. Additionally, this manual approach can be time-consuming, rendering it impossible to execute all possible combinations [28].

In addition to manual testing, there are established techniques where the idea is to define a hyperparameter search space to then conclude the best combination of hyperparameters in the specific search space. One example is Grid Search, where it is required to pre-define a finite set of hyperparameter values. The Grid Search iterates over all pre-defined values to evaluate which combination performs best for a specific machine learning method. Two drawbacks of Grid Search are that it is both time-consuming and that it is difficult for it to detect global optimum values because of the pre-defined values. Furthermore, another technique is Random Search, Random Search is quite similar to Grid Search, but instead of having pre-defined values, Random Search uses pre-defined ranges. The algorithm selects randomly a value in the given range. Random Search will continue to search for the optimal value until a given budget, which is pre-defined, is used up. Theoretically, if the ranges are large enough the global optimum for the specific machine learning method can be found, or at least the approximation of the optimal value. In comparison to Grid Search, Random Search can explore a greater search space while still operating on a small budget [28].

2.2.5 One hot encoding

Using categorical features in string format is not suitable for the majority of machine learning methods. To use such a feature an encoding process needs to be done on the feature. One of the most common techniques to encode categorical features is one hot encoding. One hot encoding converts each category in the feature into a separate feature, creating q new features for q distinct categories. For each data point, one of the features will have the value "1" and the rest will have the value "0" [29]. This indicates that a feature is either "true" or "false". If a feature has the values {x,y,z}, one hot encoding will transform the feature into three new features, a figure to visualize it:

x	y	z
0	1	0
1	0	0
0	0	1

Example of one hot encoding

There exist different approaches to encode features, for example, binary encoding and feature hashing, but previous work has shown that one hot encoding has the best performance out of these three [29].

2.3 Related work

The larger part of research regarding machine learning and the real estate industry is about the price of a property and how to predict the value of a property. Thus, naturally, the majority of the related work is regarding price prediction in the real estate industry.

2.3.1 Finding low-priced houses with machine learning

The first related work is regarding price prediction and how machine learning can help to find investment opportunities in the real estate industry [30]. The paper is about finding houses that are for sale for a lower price than the market price. The data used in the article refer to housing in the Salamanca district of Madrid, Spain and the data was fetched from a famous Spanish site where they list home sales. The authors use a date range of six months, resulting in a collection of approximately 2250 unique sales. The data is a set of binary, categorical and continuous data. To be able to use some machine learning methods they changed all data to become a set of binary numbers. The article shows how the features in hand are correlated to each other with the help of the Pearson correlation coefficient. From the correlation matrix, it was possible to see that the variable that affected the price mostly, is related to the size of the house. Further, the authors point out that the construction year does not affect the price either positively or negatively.

The authors in [30] predict the price of a property using multivariate regression. The article compares four different machine learning methods, support vector regression, k-nearest neighbors, extremely randomized trees and multi-layer perceptron. The programming language that is used in the article is python with the package scikit-learn. To lower the bias they are using the method of 5-fold cross-validation. The result of the machine learning methods is presented using five different measurements:

- Explained variance regression score
- Mean absolute error
- Median absolute error
- Mean squared error
- Coefficient of determination (R^2)

The best configuration for each of the machine learning methods based on [30], is as follow:

- Support vector regression: Radial basis function kernel.
- K-nearest neighbors: 50 neighbors, Minkowski distance and a metric inverse to distance.

  - The error and variance for median absolute error grow when the number of neighbors increases. Regarding the weight, the result shows that it is better to use inversely proportional to the distance of the target than uniform weights. Furthermore, the Minkowski method is a better choice than the cosine method regarding the calculation of distance.

- Extremely randomized tree: 50 estimators with bootstrapping.

  - The number of trees that are used in the method does not have a big impact on how it performed, however, the median absolute error and the variance decrease a bit when the number of trees increases. The use of bootstrapping had a clearly negative effect on the model.

- Multi-layer perceptron: two layers, 256 and 128 units.

  - The mean absolute error, the median absolute error and the variance decrease when more layers with fewer units are used.

The result shows that the extremely randomized tree performs better than the rest of the methods in regard to the mean absolute error and median absolute error. Furthermore, regarding the median average error, the extremely randomized tree performed best, followed by k-nearest neighbors, support vector regression and multi-layer perceptron. The authors point out that the variance for the multi-layer perceptron is enormous and that it is mostly when they are using 1024 units, one of their hypotheses is due to the low amount of data they have. They also point out that the support vector regression has only been tested with one configuration.

2.3.2 Price estimation of real estate in France

Authors in [5] estimated the price of real estate across major cities in France. The study evaluates the performance of seven machine learning techniques to each other. The data set that the article uses comes from an open-source data set that is provided by the French government and is about property transactions in France. The range that the data set has is from 2015 to 2019 and contains 43 different features of which 9 were used in the study. Regarding cleaning the data, they removed transactions that were in the outliners to only focus on transactions that were normal to the general population of France. Further, they removed all data with missing details regarding, postal code, price, living area and the number of rooms. Finally, they adjusted the data set to a one-hot encoding so the machine learning techniques would work better. To measure the performance of the machine learning techniques these metrics were used:

- Mean absolute error
- Median absolute error
- Root mean square error
- Mean squared logarithmic error

They used 75% of the data to train the machine learning model to then use the remaining 25% for testing the model. They used five-fold cross-validation and some configuration for each machine learning model. In total, they trained 14,400 machine learning models, they split the data depending on the city and if geocoding was used (i.e if the data had the features latitude and longitude). The different machine learning techniques that were used:

- Neural Networks (MLP)
- Random Forest
- Adaboost
- Gradient Boosting
- K-Nearest Neighbors
- Support Vector Regression
- Linear Regression

The result shows that when geocoding is not in use, the best machine learning model is neural networks, followed by random forest and k-nearest neighbors. The configuration for the neural network was:

- 2 layers with 150 neurons in the first layer and 50 neurons in the second layer. ReLU was used as the activation function with Adam solver, 1000 max iterations and a learning rate of 0.1

However, upon incorporating geocoding data into the data set, the study identifies random forest, gradient boosting, and adaBoost as the top three performing machine learning models. The authors highlight that including geocoding data significantly improves the accuracy of price prediction across all evaluated criteria. More precisely, on average, on all cities and metrics, the improvement is 36%, in some cases, the improvement is above 50%. The configuration for the models:

- Random forest: It used bootstrap, with a max depth of 32 and the number of trees was 2500.
- AdaBoost: For base estimators, they used decision tree regression, with a max depth of 32, 2500 trees and a learning rate of 0.05
- Gradient Boosting: They used 2500 estimators, with a max depth of 32 and a learning rate of 0.1.

2.3.3 Predicting real estate prices in Taiwan

Authors in [31] evaluated four different machine learning methods to see how well they perform. The different machine learning methods are least squares support vector regression, classification and regression tree, general regression neural networks, and backpropagation neural networks. The data set includes 32215 sales transactions and is from the Taiwanese real estate market from 2016 to 2019. The data set was originally larger but because of insufficient information, some data were left out of the data set. The addresses of the properties were transformed into geographical coordinates. The article uses two versions of the data set, one where all 23 features were present and one where only 11 were present. The 11 that were present in the data remained after the features with absolute values greater than 0.1 were eliminated using Pearson correlation coefficients. However, for both versions, 80% of the data set was used for training the machine learning methods while the remaining 20% was used for evaluating the performance. Five-fold cross-validation was also used for training the machine learning methods. To evaluate the performance of the machine learning methods, two metrics were used:

- Mean absolute percentage error
- Normalized mean absolute error

The results show that when using chosen features in the data set, all four of the machine learning method performed better. Least squares support vector regression was the best-performing machine learning method with a mean absolute percentage error of 1.63% when all 23 features were used and 0.228% when the 11 chosen features were used. Shortly behind in performance were the method classification and regression tree. The other two methods did not perform as well and shifted between third and fourth place depending on the version of the data set.

2.3.4 Sales forecasting in e-commerce

Article [32] examines the use of machine learning for sales prediction in the E-commerce industry. Three different machine learning methods are evaluated to see the performance, the models are the generalized linear model, decision tree and gradient boost tree. The data set consisted of 85,000 distinct sales from the years 2015-2017. To measure the performance some metrics were used:

- Accuracy rate
- Error rate
- Precision
- Recall
- Cohen’s kappa score

The result shows that the best machine learning method is the gradient boost algorithm which had an accuracy of 98%. It is followed by the decision tree model with 71% accuracy and the generalized linear model with an accuracy of 64%.

Chapter 3

Method

This chapter outlines the research methodology employed in the project, the data collection process, and the pre-processing steps applied to the data set. Additionally, detailed information is provided about the tools and environments used, as well as the specific machine learning methods employed and their corresponding hyperparameters. Lastly, this chapter presents the metrics used to evaluate the machine learning methods.

3.1 Research Process

An overview of the research process can be seen in Figure 3.1. The project started with getting access to Boolis API so it would be possible to fetch the data set needed. As access was granted through a private key from Booli, a Python script was created making it possible to fetch the data needed with the API. The data received from Booli was in JSON format, making it difficult to overview and therefore another script was created, changing the data set from JSON to CSV format. Thereafter different procedures were conducted on the data set to enhance its quality and suitability for the rest of the project. The next process in the thesis was to create the code for the machine learning methods. In the first step, the code for the machine learning methods did only consist of the default hyperparameters and without any normalization or standardization techniques. Thereafter, the code for the machine learning methods was changed to first test the performance with a standardization technique and thereafter test the performance with a normalization technique. Lastly, Random Search was implemented on the code to test different hyperparameters on the machine learning methods. Additionally to the last phase, normalization or standardization was used depending on the result from the previous step.

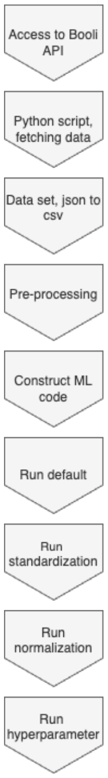

Figure 3.1: Overview of the research process

3.2 Data Collection

To collect the data that was used in this master thesis the website Booli was used. Booli describes itself as "Sweden's largest collection of homes for sale". Booli is owned by the Swedish bank SBAB and was created in 2007. Booli has an API that can be used, if access is granted, to collect their data. SBAB is, through Booli, making the housing market more transparent and safer for everyone, regardless of whether you are a seller, buyer, or owner of a property [33, 34]. The Booli API offers a feature that allows users to submit two coordinates, which define an area, and retrieves all the sold properties within that specific area. As Sweden is not a fully rectangular geographical country, determining the two coordinates necessary to fetch all the sold properties in Sweden was not a straightforward task. In common sense, more data will be fetched if you increase the distance between the coordinates. A drawback with the API is that it did not manage coordinates that were a bit outside the Swedish borders. This led to the utilization of trial and error to find the two coordinates that fetched as much data as possible. This resulted in that not all the sold properties from the whole of Sweden were fetched. The coordinates that were used: (56.0; 9.0) and (66.0; 19.0). The script to fetch the data points was done in the programming language Python.

3.3 Data Set

The fetched data set contained approximately 800,000 data points with 27 different features. The data set ranges from the year 2012 to 2023. To see all features, see Appendix A. Some of the features were directly related to Booli's own system, for example, the URL, the Booli id, and the source id of the specific property. Those features are irrelevant to the thesis and were therefore removed. The feature "apartment number" was removed because it contained many empty cells and apartment numbers do not typically contain information directly related to a property's physical or structural characteristics. Another feature that had many empty cells is the feature "city" and because the data set contained geographical coordinates (longitude and latitude features) for each data point, the city feature was removed. Two additional features related to the location of the properties were removed from the data set, ”street address” and "named area". The decision to exclude those features was based on their large number of distinct string values, each containing approximately 20,000 different values. Using one hot encoding to represent those features would have resulted in approximately 40,000 different features which are both computationally expensive and potentially prone to overfitting. Some features of the data set were regarding the brokers, those features were removed because they do not give any information about the property itself.

3.3.1 Pre-processing of the Data Set

After some cleaning of the data set, it contained 560,000 different data points, where the average value for sold price is 2,965,213 kr and the average value for sale velocity is 27.6 days. The majority of data points removed were because of empty cells. Almost 200,000 data points were removed because of the lack of data for construction year and the lack of floor number for the object type apartment. To see more detailed information of the cleaning process, see Appendix B. Furthermore, some categories in the feature "object type" were combined into one category:

- Firstly, the categories house and villa were combined into the category villa. Villa and house in the Swedish language are synonyms for each other and having two separate categories for the same property type is unnecessary. Additionally, the number of data points associated with the ”house” category was low, which further supports the merge of the two categories.

- Secondly, the categories semi-detached house (parhus), terraced house (radhus), and terraced house (kedjehus) were combined into one category. Because of their similarities, radhus and kedjehus have the same property type when translated from Swedish into English. The “semi-detached house” is also similar to “terraced house” and therefore these three categories were combined into one category - “terraced house”.

Additionally, more pre-processing were done to the data set:

- The ”published” feature was changed from timestamp to three separate features, published year, published month and published day.

- All empty cells in the feature “additional area” were set to 0.

- A new feature was created: “Sales velocity”. “Sales velocity”, is the difference in days between the “sold date” and “published”. With the new feature, some data points were removed. All negative values for the feature ”sales velocity” were removed because they were not logical since the sold date cannot be earlier than the published date. Approximately 2000 data points were therefore removed.

- The feature “floor” had some values with a decimal number, for example, floor “1.5”. Those values were rounded to the nearest integer. This was a procedure to simplify the data set.

- One hot encoding was used on the feature “floor”. This was mostly because villas and terraced houses did not have a specified floor in the data set. Adding a value as for example “0” to villas and terrace houses could mess up the data set and the machine learning method. Because living on floor “0” in an apartment is quite different from living on floor “0” in a villa. Thus one hot encoding was used.

- One hot encoding was also used on the three features: “object type”, “municipality name” and “country name”.

- All empty cells for “plot area” that belong to the object type “apartment” were changed to 0. This is because most apartments do not have plot area and therefore it was logical to add 0 to those data points for that feature.
- For two features, ”plot area” and ”Rent”, KNNImputer was used to fill in the empty cells. KNNImputer replaces the empty values with the technique of KNN as described in 2.1.3. According to [35], KNNImputer outperforms seven other imputation methods, such as linear regression, random sample and Bayesian linear regression. The decision to use KNNImputer instead of removing those data points was because a large amount of data would be lost, particularly to the object types ”villas” and ”terraced houses”. Further, both features are of interest and it was decided inappropriate to remove them, consequently the empty cells were filled in with the use of KNNImputer
- The feature ”list price” was removed because it has a high correlation to ”sold price” and would affect the machine learning methods.
- The feature ”sold date” was removed because having both ”sold date” and published date would maybe affect the machine learning methods in regards to ”sale velocity”.

3.4 Tools and Environment

All code written in this master thesis is written in the programming language Python. Python is one of the most popular programming languages for data science and has many useful add-on libraries [36]. One of the most commonly used libraries in machine learning is the open source library scikit-learn. The library makes it possible to easy and fast integrate machine learning methods into Python code. The library has a large and wide range of different machine learning methods and techniques that can be used when working with machine learning, a few examples are classification machine learning models, regression machine learning models, and data pre-processing [37]. All machine learning methods in this thesis will be from the library scikit-learn.

All of the machine learning models have been trained on the online platform Kaggle. Kaggle lets its users create notebooks where they can run their codes on Kaggle servers. One advantage of Kaggle is that it lets each account use up to 30 hours of GPU per week, where one can either choose to use GPU P100 or GPU T4 x2. Additionally, one can also choose to use TPU VM v3-8 where each account has 20 hours of use per week.

3.5 Machine learning methods

The thesis will utilize five supervised machine learning methods: Random Forest Regression, Decision Tree Regression, Support Vector Regression, K-nearest Neighbours Regression, and Multi-layer Perceptron Regression. These particular techniques were chosen because they are well-known, have demonstrated promising outcomes in prior studies, and provide a variety of machine learning methodologies.

The library scikit-learn has different support vector machine models. The first one, SVR, is implemented in terms of libsvm and involves different kernel types, for example, "RBF". Whereas the other one, linearSVR is a linear support vector machine that is implemented in terms of liblinear. Scikit-learn is warning its users that the time complexity of the model SVR is more than quadratic with the number of data points, making it difficult to use data sets that are more than a couple 10,000 [38]. Also, the authors in [39] point out that choosing what support vector machine one uses is important when working with large data sets. Their result shows that linear SVR has a similar predicting performance as πSVM with the kernel RBF for large data sets, but linear SVR has a significantly lower running time. For both of those reasons, this thesis will be using the model linearSVR and not SVR from the library scikit-learn.

3.6 Hyperparameter tuning

Random Search was used to fine-tune the machine learning methods. Each machine learning method was executed 10 times to try to find the optimal value. With Kaggles limited servers and the time-consuming machine learning methods, it was not possible to run all 10 iterations directly for all the machine learning methods. Instead, the budget for Random Search was put to 1 and the code was restarted 10 times. The different pre-defined ranges for the hyperparameters were:

- Random Forest:
  - 'n_estimators': randint(25, 300),
  - 'max_depth': randint(1, 100),

  - ’min_samples_split’: randint(2, 20),
  - ’min_samples_leaf’: randint(1, 20)

- Decision Tree:
  - ”max_depth’: randint(1, 100),
  - ’min_samples_split’: randint(2, 20),
  - ’min_samples_leaf’: randint(1, 20)
- Support Vector Regression
  - ’C’: uniform(0, 10),
  - ’epsilon’: uniform(0, 1),
  - ’max_iter’: randint(500, 3000)
  - ’loss’: [’epsilon_insensitive’, ’squared_epsilon_insensitive’]
- K-nearest Neighbors
  - ’n_neighbors’: randint(1, 100),
  - ’weights’: [’uniform’, ’distance’],
  - ’algorithm’: [’auto’, ’ball_tree’, ’kd_tree’, ’brute’],
  - ’p’: [1, 2]
- Multilayer Perceptron
  - ’hidden_layer_sizes’: randint(50, 201), where it could be 1-3 hidden layers.
  - ’activation’: [’identity’, ’logistic’, ’tanh’, ’relu’],
  - ’solver’: [’lbfgs’, ’sgd’, ’adam’],
  - ’alpha’: [0.0001, 0.001, 0.01, 0.01, 0.1],
  - ’learning_rate’: [’constant’, ’invscaling’, ’adaptive’],
  - ’max_iter’: randint(100, 700)

3.7 Machine Learning Metrics

There exist multiple ways to measure the performance of a machine learning method. This thesis will use four different metrics to gain a comprehensive understanding of the models performance: R^2, RMSE, MAE and MAPE. In the following metrics, $\bar{y}$ is the average value for the true values and $\hat{y}_i$ is the true value for the i:th data point.

- Coefficient of Determination (R^2)

The coefficient of determination (R^2) is a statistical measure that indicates how well the predicted values match with the actual values. It is used to assess the performance of a regression model and helps to determine how accurately the model can predict future samples. The maximum value for R^2 is 1.0, indicating that the model can predict all target values precisely for all data points. If R^2 is equal to 0 it indicates that the model predicts the average value for all targets. R^2 can be negative and it indicates that the model predicts worse than the average value [30]. The formula for R^2 is:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n}(y_i - \bar{y})^2} \tag{3.1}$$

Coefficient of Determination

- Mean Squared Error (MSE)

MSE is an error metric. For each evaluated data point, it calculates the difference between the output values and true values, to then square the difference. The difference (error) is then summed up to later calculate the mean square error (MSE). Squaring the error makes the metric more sensitive to outliners because it punishes larger errors more severely than smaller ones [40]. The optimal value for MSE is 0 as it indicates that there is no error in the prediction.

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n}|y_i - \hat{y}_i|^2 \tag{3.2}$$

Mean Squared Error

- Root Mean Squared Error (RMSE)

RMSE is the square root of MSE and is commonly used in the field. A benefit of RMSE is that it generates smaller values than MSE which makes it easier to compare the results [40].

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n} |y_i - \hat{y}_i|^2} \tag{3.3}$$

Root Mean Squared Error

- Mean Absolute Error (MAE)

MAE is very similar to MSE, the difference between them is that MAE does not square each error. The metric calculates the mean absolute error for the model, where 0 is the most optimal result [40].

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n} |y_i - \hat{y}_i| \tag{3.4}$$

Mean Absolute Error

- Mean Absolute Percentage Error (MAPE)

MAPE is an error metric that calculates the absolute error for each data point divided by the true value of the data point. This gives the percentage error for each data point, summarizing the percentage errors and dividing it by the total number of data points gives the mean absolute percentage error (MAPE) for the model [41]. As for the other error metrics, the best value for MAPE is 0.

$$MAPE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n} |\frac{\hat{y}_i - y_i}{\hat{y}_i}| \tag{3.5}$$

Mean Absolute Percentage Error

Chapter 4

Results and Analysis

All machine learning methods that are presented have used the technique of five-fold cross-validation. The result is divided into two large sections, Sold Price and Sale Velocity. Both sections start with presenting how the machine learning methods perform with their default values and how the methods react when StandardScaler or MinMaxScaler is used on the data set. The values regarding sold price are in the Swedish currency SEK.

4.1 Feature Correlation

Figure 4.1 presents the Pearson correlation matrix for the features in the data set. The features “floor”, “municipality”, and “county” were excluded from the matrix because of the one-hot encoding that was utilized on them. The most interesting part of the matrix is regarding the features sold price and sale velocity. Sold price had the highest positive correlation to the living area and the number of rooms, which indicates that those factors are of great importance. Additionally, there was a positive correlation between rent and sold price, which could be due to larger properties commanding higher rents and selling for a higher price. It is also possible to see that the published year had a positive correlation to sold price, one possible explanation for this could be the general increase in property prices over the years. Interestingly, the construction year had a slightly negative correlation with sold price, suggesting that old properties sell for a higher price than newer ones. This might be because Sweden has a lot of older buildings in the city centers which often can cost a bit more because of their location.

As for sale velocity, most features had a neutral correlation with it. The feature “villa” had the highest positive correlation with sale velocity, while “apartments” had the lowest negative correlation, indicating that there is a tendency that villas are selling a bit slower than apartments. However, because of the weak correlation between the features, it is not possible to draw definite conclusions.

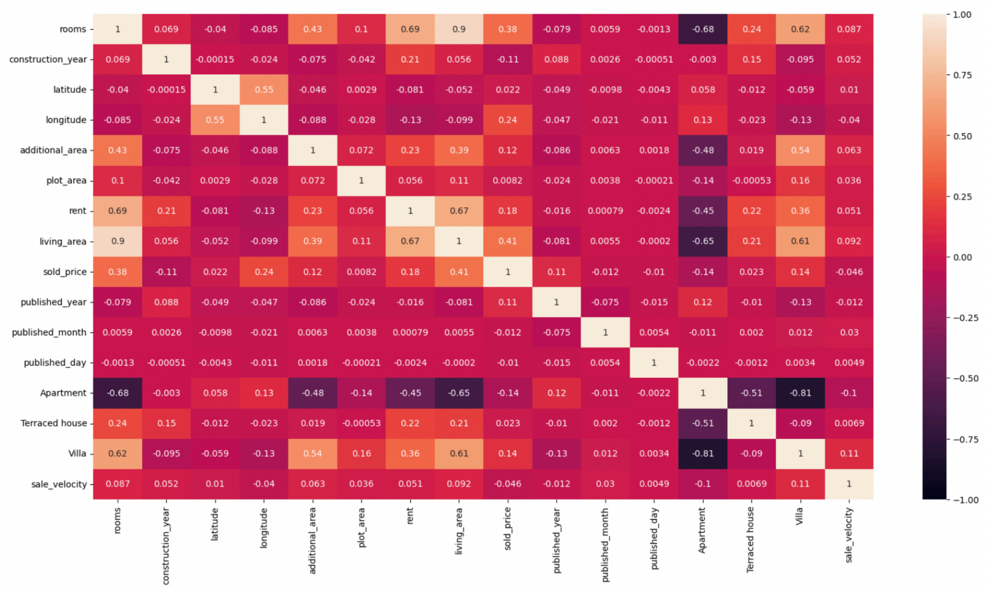

Figure 4.1: Pearson Correlation Matrix

4.2 Sold Price

Overall, random forest was the machine learning method that performed best in regard to sold price. It had the highest R^2 score (0.92) and the lowest MAE score (336,415.46 kr). It did not have a high variance in regards to different hyperparameters and the overall algorithm performed well on sold price. SVR performed worst of the machine learning methods and had a high variance regarding the different hyperparameters used. A point that should be considered when evaluating the result is that most of the training for SVR and MLP did not fully converge because of stopping criteria and its time complexity.

4.2.1 Default hyperparameter

As can be seen in table 4.1, random forest outperformed SVR, KNN, and MLP significantly on all metrics. The performance of the decision tree was relatively close to that of random forest. With default hyperparameters, random forest had a mean absolute error of 337,474.99 kr and a MAPE of 0.15. Comparing the result to [5], where the average property prices were similar to this study, [5] got an MAE of 44,000 Euros. It is worth noting that [5] data set focused on large cities and not a large geographical area. Furthermore, inspecting the R^2 metric gives the information that the random forest learns quite well from the data set. In fact, this thesis result outperforms [5] which got an R^2 score of 0.74. It is also evident that the data set contained outliers as the Root Mean Square Error (RMSE) differed significantly from the mean absolute error (MAE). Thus, the model performed well even with a data set that included outliers and variations in locations, floors, and living areas across the majority of Sweden. During the execution of SVR and MLP a convergence warning occurred, indicating that the models did not converge to their desired solutions due to the maximum number of iterations.

Model	$\boldsymbol{R^2}$	**RMSE**	**MAE**	**MAPE**
Random Forest	0.92	663,448.85	337,474.99	0.15
Decision Tree	0.84	949,170.11	483,695.99	0.20
SVR	0.20	2,090,367.00	1,281,781.31	0.76
KNN	0.47	1,701,529.45	1,082,004.17	0.64
MLP	0.65	1,391,816.60	876,719.32	0.47

Table 4.1: Hyperparameter default values

4.2.2 Default hyperparameter with StandardScaler

StandardScaler did not affect random forest, decision tree or MLP much. StandardScaler had a negative influence on SVR which got a prediction performance of -1.14 in the metric R^2. In contrast, the StandardScaler had a significant influence on KNN, rising from an R^2 score of 0.47 to 0.80.

Model	$\boldsymbol{R^2}$	**RMSE**	**MAE**	**MAPE**
Random Forest	0.92	663,799.44	337,028.15	0.15
Decision Tree	0.83	956,281.07	484,249.79	0.20
SVR	-1.14	3,430,939.61	2,534,488.81	0.78
KNN	0.80	1,038,189.79	606,569.74	0.25
MLP	0.66	1,360,460.81	776,335.78	0.33

Table 4.2: Hyperparameter default values with StandardScaler

4.2.3 Default hyperparameter with MinMaxScaler

Utilizing MinMaxScaler on the data set resulted in worse performance for KNN and MLP in comparison to StandardScaler, nevertheless, KNN still performed better with MinMaxScaler than without any normalization or standardization of the data set. For SVR, MinMaxScaler had a better influence on it than StandardScaler, but SVR performed better without any normalization or standardization technique on the data set.

Model	R^2	**RMSE**	**MAE**	**MAPE**
Random Forest	0.92	661,437.05	337,144.92	0.15
Decision Tree	0.83	969,075.39	485,636.37	0.20
SVR	-0.42	2,794,535.73	1,766,978.85	0.60
KNN	0.64	1,402,247.01	844,393.80	0.35
MLP	0.49	1,674,233.28	1,000,317.84	0.44

Table 4.3: Hyperparameter default values with MinMaxScaler

4.2.4 Hyperparameter Optimization with Random Search Sold Price

With the result from sections 4.2.1 to 4.2.3, it was decided to use StandardScaler on KNN and MLP, and no standardization or normalization for random forest, decision tree, and SVR. The use of Random Search to find better performance for the machine learning methods worked for almost all of them. Table 4.4 presents the best-performing model for each machine learning method and its hyperparameters. Random forest did not show improved performance with the change of hyperparameters in comparison to its default hyperparameters. Decision tree did an improvement with the use of different hyperparameters and was not far from how well random forest performed. This is interesting because a decision tree is less computationally and time expensive in comparison to random forest. Furthermore, while KNN's performance slightly improved, the change was not significant. SVR, on the other hand, performed better with the new hyperparameters but was still the worst performing machine learning method. MLP performed better with the new hyperparameters, increasing the R^2 metric with 0.14 and lowering MAE by almost 124,000 kr. As written before, MLP and SVR did not fully converge, their results could perhaps be improved further if their stopping criteria were increased.

Model	R^2	RMSE	MAE	MAPE	Hyperparameter
Random Forest	0.92	660,331.76	336,415.46	0.15	max_depth: 58 min_samples_leaf: 1 min_samples_split: 2 n_estimators: 177
Decision Tree	0.88	827,908.46	432,829.87	0.19	max_depth: 38 min_samples_leaf: 14 min_samples_split: 10
SVR	0.61	1,472,620.53	905,528.99	0.47	C: 0.412 epsilon: 0.042 loss: squared_epsilon_insensitive max_iter: 2554
KNN	0.81	1,017,373.70	592,289.71	0.25	algorithm: brute n_neighbors: 9 p: 2 weights: distance
MLP	0.80	1,059,455.80	652,784.91	0.29	activation: relu alpha: 0.01 hidden_layer_sizes: (175, 92, 138) learning_rate: invscaling max_iter: 295 solver: lbfgs

Table 4.4: Hyperparameter Optimization

Figures 4.2 and 4.3 show the combined results for all iterations with Random Search in the metrics R^2 and MAE, respectively. The figures show that random forest performed best and had the lowest variance, indicating that the structure of the algorithm worked well with a range of hyperparameters. Decision tree and KNN did also have a low variance with the different hyperparameters. SVR and MLP demonstrated larger variances in their performance, suggesting that hyperparameters matter more for those machine learning methods. SVR and MLP did not converge for the majority of the runs which also could be a factor for the large variance.

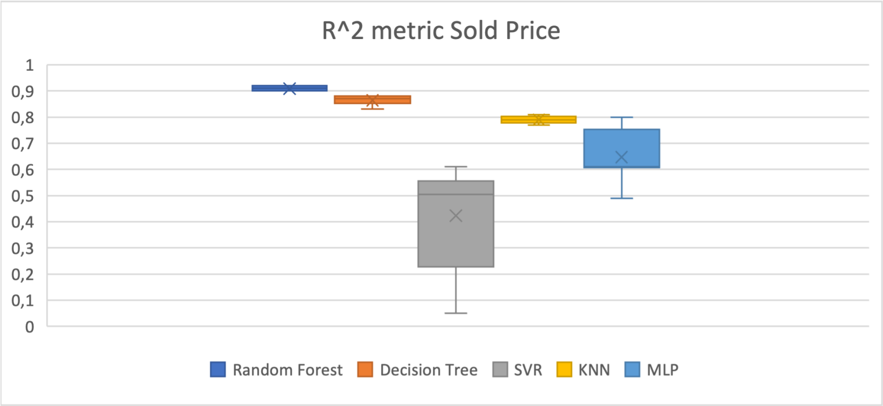

Figure 4.2: R^2 Sold price variance

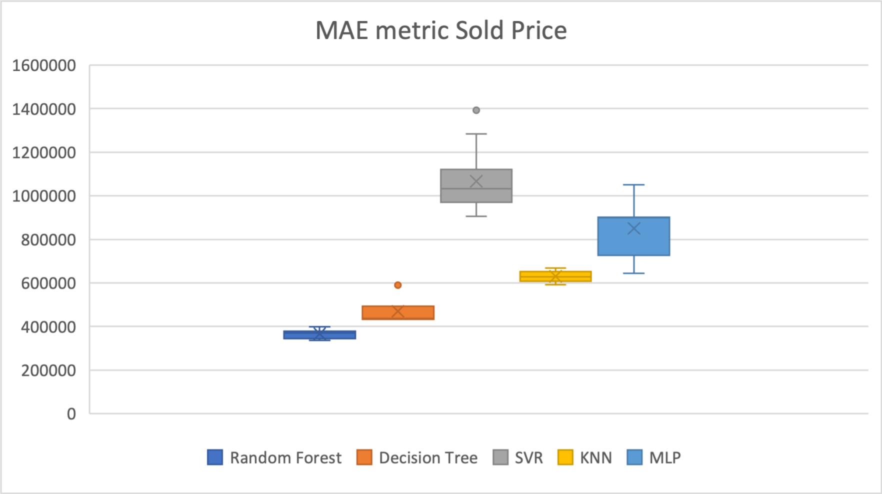

Figure 4.3: MAE Sold price variance

4.3 Sale Velocity

According to the evaluations, predicting the feature sale velocity has shown to be a hard task for the five different machine learning methods. Random forest performed best of the machine learning methods in regards to the R^2 metric (0.19), but the machine learning method SVR performed better in regards to MAE (15.91). Random forest did not have a large variance in regard to different hyperparameters, and neither did KNN and MLP. Conversely, did

SVR have a larger variance depending on the hyperparameter, showing that the machine learning method performed better with different hyperparameters. Similar to sold price, MLP and SVR did not converge for most of the training and evaluation, and it might have affected their performance. Furthermore, the data set had properties that were sold on the same day as they were published, resulting in a sale velocity of 0 for those properties. Using MAPE, which uses the actual value, in this case, 0, as the denominator in a division makes the metric useless as you can not divide by 0. Consequently, MAPE can not be used as a metric in the sense of sale velocity and is therefore not presented.

4.3.1 Default hyperparameter

Table 4.5 reveals that all machine learning methods encounter difficulties in predicting sale velocity. Random forest performed best of the machine learning methods in all metrics and was the only one with a positive R^2 score. MLP, on the other hand, was the worst-performing machine learning method and performed significantly worse on all metrics in comparison with the other machine learning methods. The metric RMSE shows that the data set had outliers that were large in comparison to the rest of the data set. The distinction becomes apparent when comparing MAE to RMSE. The RMSE metric will penalize outliers more than MAE because of its mathematical construction.

Model	R^2	RMSE	MAE
Random Forest	0.17	42.17	18.75
Decision Tree	-0.69	60.10	24.77
SVR	-0.29	52.45	26.68
KNN	-0.11	48.75	21.33
MLP	-7.15	126.23	82.47

Table 4.5: Hyperparameter default values

4.3.2 Default hyperparameter with StandardScaler

Using StandardScaler on the data set had a positive impact on SVR and KNN, all metrics lowered for both of them. Interestingly, SVR performed better than random forest when compared to the metric MAE. However, random forest still outperformed every machine learning method regarding the metric R^2. MLP obtained a mixed result from the use of StandardScaler; its R^2 score increased heavily but its MAE decreased simultaneously.

Model	R^2	RMSE	MAE
Random Forest	0.17	42.20	18.75
Decision Tree	-0.67	59.62	24.74
SVR	-0.03	46.89	15.91
KNN	-0.03	46.95	20.20
MLP	-174.79	574.08	23.41

Table 4.6: Hyperparameter default values with StandardScaler

4.3.3 Default hyperparameter with MinMaxScaler

MinMaxScaler worked better than StandardScaler for MLP and made MLP have a positive R^2 score, while it also decreased the MAE more in comparison to StandardScaler. In regards to the other machine learning methods, MinMaxScaler and StandardScaler performed quite similarly and there was no large difference between them.

Model	R^2	RMSE	MAE
Random Forest	0.17	42.14	18.77
Decision Tree	-0.64	59.22	24.67
SVR	-0.03	46.93	15.91
KNN	-0.04	47.22	20.38
MLP	0.07	44.52	19.56

Table 4.7: Hyperparameter default values with MinMaxScaler

4.3.4 Hyperparameter Optimization with Random Search Sale Velocity

After analyzing tables 4.5 to 4.7, it was decided to use MinMaxScaler for MLP, StandardScaler for KNN and SVR, and no normalization or standardization for random forest and decision tree. Table 4.8 presents the best performance from each machine learning method with the use of Random Search and its corresponding hyperparameters. The hyperparameter tuning showed an improvement in four machine learning methods, random forest, decision tree, KNN and MLP. It was a slight improvement for the best-performing machine learning method, random forest, the R^2 metric was improved with 0.02. KNN improved a bit more, increasing its R^2 score from -0.03 to 0.09. For the decision tree, on the other hand, there was a large improvement, from -0.64 to 0.13 for the R^2 metric and lowering the MAE with 6 days. There was no improvement for SVR, it performed similarly to the default hyperparameters with either standardization (table 4.6) or normalization (table 4.7). MLP performed quite similarly to the default hyperparameter with MinMaxScaler as can be seen in table 4.7, but there is a small improvement if compared with the metric MAE, a difference of 0.48 days.

Model	R^2	**RMSE**	**MAE**	**Hyperparameter**
Random Forest	0.19	41.57	18.01	max_depth: 45 min_samples_leaf: 8 min_samples_split: 12 n_estimators: 244
Decision Tree	0.13	43.05	18.67	max_depth: 10 min_samples_leaf: 19 min_samples_split: 6
SVR	-0.03	46.87	15.93	C: 3.377 loss: epsilon_insensitive max_iter: 2520
KNN	0.09	44.05	18.75	algorithm: brute n_neighbors: 72 p: 2 weights: distance
MLP	0.07	44.50	19.08	activation: tanh alpha: 0.0001 hidden_layer_sizes: (94, 87, 90) learning_rate: adaptive max_iter: 376 solver: sgd

Table 4.8: Hyperparameter Optimization

Figures 4.4 and 4.5 show the combined results from the 10 iterations with Random Search for each machine learning method. As for sold price, the variance for random forest was quite low and showed that the machine learning method worked best in regards to the R^2 metric. Unlike the performance decision tree had for sold price, the variance was quite large when it comes to sales velocity, the box plot showed that the mean value was 0.013 and ranges from -0.22 to 0.09. SVR was the machine learning method that had the largest variance when using different hyperparameters. SVR performed worse than the other machine learning methods when it comes to the R^2 metric, but when compared to the MAE metric, it achieved the highest performance. This showed that it might be possible to boost SVR performance further with other hyperparameters. For KNN and MLP, both machine learning methods had a low variance where KNN performed slightly better than MLP on both metrics.

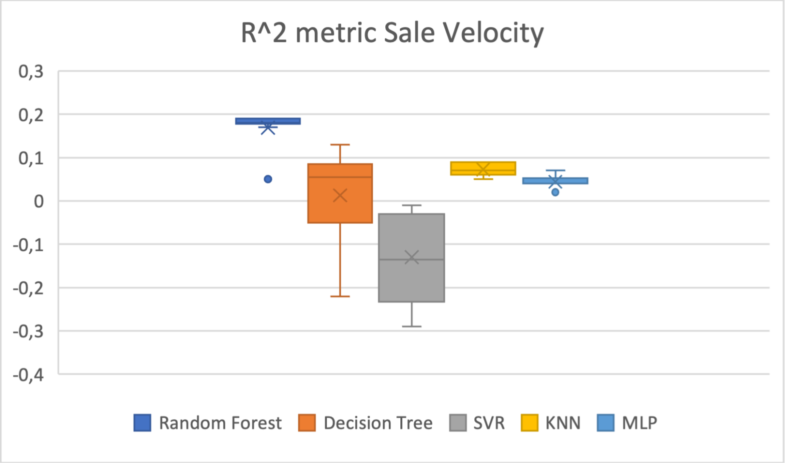

Figure 4.4: R^2 Sale Velocity variance

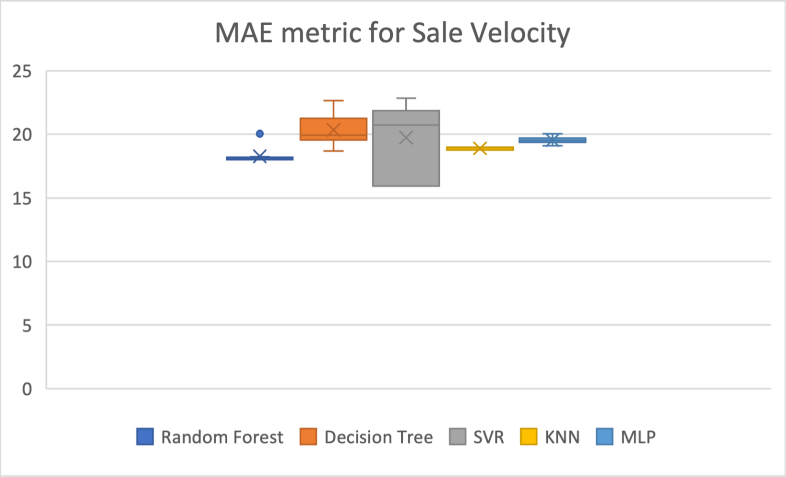

Figure 4.5: MAE Sale Velocity variance

Chapter 5

Discussion

In this chapter, the thesis's limitations are discussed, including constraints related to the data set, the pre-processing, and the selection of hyperparameters. Additionally, the chapter explores the practical applications of the findings, as well as the ethical and sustainability considerations of the thesis.

5.1 Limitations

When considering the study's findings, it is important to acknowledge the presence of several limitations. The following section aims to explore potential constraints that could have influenced the results of this thesis.

5.1.1 Data set

One limitation of the thesis was that all the data used came from Booli.se. It would have been interesting to gather data from other websites as well because there is a risk that some brokers do not use Booli. Furthermore, two coordinates were used to gather all the data, and as explained in section 3.2, the coordinates created a rectangle where all the sold properties in that area were fetched. Consequently, this resulted in not all sold properties being fetched from the Boolis data set, but the large majority were. Gathering all the data would have been complicated and time-consuming.

The data set gathered had a limited amount of features that described each property, and with more features, the machine learning methods might have performed better.

5.1.2 Pre-processing

In the pre-processing of the data set various decisions could have affected the result. Firstly, the removal of approximately 240,000 data points could have made the data set biased. The choice to remove those data points was because of their lack of data in features. Using a method such as KNNImputer to fill in all empty cells in the data set was an option, but there was a risk of making the data set generalized and removing patterns in the data. Therefore, it was decided to remove those data points and not use KNNImputer. Secondly, rounding decimal numbers to the nearest integer for the feature floor may have affected the result. Thirdly, for two features, a value of 0 was inserted as opposed to empty cells, which might have had an impact on the outcome.

5.1.3 Normalization and Standardization

It exists multiple normalization and standardization techniques but this thesis used one normalization and one standardization technique. The result points out that the different techniques matters and trying other techniques might have improved the machine learning methods. Because of the time limitation of the thesis, it was not possible to try out multiple techniques.

5.1.4 Hyperparameter

A limitation in the thesis was the choice of hyperparameter values. Having a larger search space when trying to find the optimal performance for a machine learning method is better. A problem arises from the fact that there exists a huge amount of combinations for each machine learning method, trying all of them to find the optimal one is time-consuming and could be impossible depending on the software that is used. Furthermore, there are other techniques available that may provide better results than Random Search in finding the optimal value within the search space. One example is Bayesian Optimization, which iteratively learns from its previous hyperparameters to find the optimal value. Consequently, Bayesian Optimization does not make similar calculations twice, resulting in a more efficient search. One drawback with Bayesian Optimization and why it could not be used in this thesis is that it has a larger time complexity than Random Search [28].

5.1.5 Tools and environment

A significant limitation in the thesis was the tool and environment that were used. While Kaggle proved to be a great and free online platform, it took a large amount of time to complete some of the machine learning models. This became a problem as the Kaggle servers had a maximum runtime restriction of 12 hours in a single session. As a result, some of the machine learning methods executed in this thesis could not be completed. Additionally, this was one of the factors that led to the selection of Random Search over alternative methods for hyperparameter optimization, the limitation of the search space for hyperparameters, and the 10-time training and evaluation of each machine learning model.

5.1.6 Convergence

Most MLP and SVR models did not converge. Some factors of why this happened are the large data set that was used, the tool and environment, and the stopping criteria for the models.

5.2 Ethics and Sustainability

The data used in the thesis came from Booli.se, which is open for the public to view. Booli is making the housing market more transparent and safer for all stakeholders in the real estate industry. No personal data of the buyer or the seller was used in the thesis. The thesis is open to the public so all people interested in the thesis can read it.

From an economic and social sustainability perspective, the thesis could help construction companies to build properties in locations that are popular and thus might be used by generations. It is also possible that if more properties are built in popular areas it might cause the overall price to go down as supply increases. Consequently, making it more affordable for a larger group of society. However, it could be a negative aspect for the companies as they may not succeed in getting the profit they had intended on the properties. Furthermore, from an ecological sustainability perspective, the machine learning methods that are used could have a lower carbon footprint than if the analyses would have been done by humans. It is known that the use of machine learning models requires a large amount of energy [42], which is not good for the environment. However, to reduce the carbon footprint, it is possible to use servers that are carbon neutral, for example, servers run by

Google [43].

5.3 Practical use of the findings

The findings of the thesis can be broadly used by different stakeholders. For instance, construction companies can leverage this information to develop their own machine learning models to forecast the selling price of their planned properties. This makes it possible for construction companies to plan their projects depending on the potential profitability. The result of the thesis also shows that random forest works in a large-scale environment, where there are properties in large cities and in the countryside. Previous studies have concentrated on specific areas or regions [5, 30], whereas this thesis shows that with a large enough data set, it was possible to predict the selling price of a property with an R^2 score of 0.92 and an MAE of 330,000 kr in a large and varied environment. Furthermore, the findings can also be used by brokers, investors, and banks to evaluate diverse properties and investment opportunities.

For sale velocity, the findings show that it is hard to estimate how fast a property will sell. The highest score was 0.19 in the R^2 metric, it is not a good enough performance to start fully using it to predict how fast a property will sell. SVR, which performed best in the metric MAE shows that it can predict with a mean absolute error of 15.91 days. It is moderately encouraging knowing that a sale velocity of a property may fall within the range of 16 days; however, it is not a large achievement because the mean of the sale velocity is not far from it either (27.6 days).

However, using both the prediction of sold price and sale velocity can make a great tool for companies in the real estate industry that want to evaluate the selling price and the duration it may take to sell a property. Such insights can enhance profitability, providing greater clarity on the expected sale price and the sale velocity for a given property.

Chapter 6

Conclusion

In conclusion, this thesis employed a comparative method to evaluate five different machine learning methods in regard to sold price and sale velocity in the real estate industry. The five different machine learning methods evaluated are random forest, decision tree, K-nearest neighbor, support vector regression, and multilayer perceptron. The data set that was fetched had 800,000 data points, but after the pre-processing, the data set used consisted of 560,000 distinct data points with various properties throughout a large geographical area.

The thesis examined the impact of normalization and standardization techniques on the performance of the machine learning methods. The purpose was to determine whether these techniques had a positive impact on the methods' performance. If a technique increased the performance, it was used into the respective machine learning method.

The first research question aimed to answer *How well do machine learning methods perform in regard to the sold price of properties in Sweden, and which method outperforms the others among the evaluated approaches?*. The result indicated that four out of the five machine learning methods performed with a high R^2 score, indicating that the methods perform well in regard to sold price. The best-performing machine learning method was random forest with a R^2 score of 0.92 and an MAE of 336,415.46 kr.

The second research question aimed to answer *How well do machine learning methods perform in regard to the sale velocity of properties in Sweden, and which method outperforms the others among the evaluated approaches?*. The result indicated that the machine learning methods performed ineffectively with a low R^2 score. Random forest performed best when evaluating the five machine learning methods to the R^2 metric, with a score of 0.19. However, when comparing the machine learning methods to MAE, SVR performed best with an MAE of 15.93 days. Consequently, it was hard to determine the best-performing machine learning method, further research is needed for a conclusive evaluation. However, considering that random forest performed better in the R^2 metric indicating a better fit to the data set compared to SVR, it can loosely be concluded that random forest performed best of the evaluated approaches.

6.1 Future work

The machine learning methods did not perform as well for sale velocity as they did for sold price, future work can investigate sale velocity further. More and different features might need to be taken into consideration to predict sale velocity better. Interesting features could be the interest rates from the central bank, if the economy is in a boom or bust, and any recent changes in government regulations. This could also be applied to sold price to try to get even better results from the different machine learning methods. Additionally, the features used in this thesis are mostly focused on the structure of the property, extending the data set with for example the distance to the closest public transport, school, and major industries could be interesting factors and might affect the result for both sold price and sale velocity.

For future work, other hyperparameter values and machine learning methods could complement this thesis. Additionally, having better tools and environments could make it possible to test larger search spaces for the hyperparameters. One example is that other tools and environments might be able to run longer and faster. This capability could permit the utilization of larger maximum iterations for MLP and SVR models, consequently leading to their convergence and potentially improving the predictive accuracy.

An interesting aspect, but one that could be more difficult to implement, involves expanding the geographical area of the data set and examining the performance of the machine learning methods. How would the machine learning methods react to a data set with properties from all countries in northern Europe? The difficulty here lies in managing a significantly larger data set, which will cause runtime problems if not suitable tools and environments are used.

References

[1] G. V. Research. Real estate market size, share & trends analysis report by property (residential, commercial, industrial, land), by type (sales, rental, lease), by region, and segment forecasts, 2022 - 2030. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/real-estate-market [Page 1.]

[2] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, “Big data: the management revolution,” *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012. [Page 1.]

[3] E. Brynjolfsson, L. M. Hitt, and H. H. Kim, “Strength in numbers: How does data-driven decisionmaking affect firm performance?” *US Census Bureau Center for Economic Studies Paper No. CES-WP-16-28*, 2011. [Page 1.]

[4] J. M. Helm, A. M. Swiergosz, H. S. Haeberle, J. M. Karnuta, J. L. Schaffer, V. E. Krebs, A. I. Spitzer, and P. N. Ramkumar, “Machine learning and artificial intelligence: definitions, applications, and future directions,” *Current reviews in musculoskeletal medicine*, vol. 13, pp. 69–76, 2020. [Page 1.]

[5] D. Tchuente and S. Nyawa, “Real estate price estimation in french cities using geocoding and machine learning,” *Annals of Operations Research*, pp. 1–38, 2022. [Pages 1, 21, 36, and 47.]

[6] K. C. Iyer and R. Kumar, “Impact of delay and escalation on cash flow and profitability in a real estate project,” *Procedia Engineering*, vol. 145, pp. 388–395, 2016. [Page 2.]

[7] C. Team, “Sources of funding,” Mar 2023. [Online]. Available: https://corporatefinanceinstitute.com/resources/accounting/sources-of-funding/ [Page 2.]

[8] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018. [Pages 7 and 8.]

[9] A. H. Renear, S. Sacchi, and K. M. Wickett, "Definitions of dataset in the scientific and technical literature," *Proceedings of the American Society for Information Science and Technology*, vol. 47, no. 1, pp. 1–4, 2010. [Page 7.]

[10] A. Munde and N. Mishra, "Corporate performance: Smes performance prediction using the decision tree and random forest models," *Corporate Ownership & Control*, vol. 20, no. 1, pp. 103–113, 2022. [Page 8.]

[11] L. Rokach and O. Maimon, "Decision trees," *Data mining and knowledge discovery handbook*, pp. 165–192, 2005. [Pages 8, 9, and 10.]

[12] "What is a decision tree," *IBM*. [Online]. Available: https://www.ibm.com/topics/decision-trees [Page 8.]

[13] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012. [Pages 9 and 10.]

[14] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Elsevier, 2020, pp. 101–121. [Page 10.]

[15] J. Ye and T. Xiong, "Svm versus least squares svm," in *Artificial intelligence and statistics*. PMLR, 2007, pp. 644–651. [Page 10.]

[16] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006. [Page 10.]

[17] F. Zhang and L. J. O'Donnell, "Support vector regression," in *Machine learning*. Elsevier, 2020, pp. 123–140. [Pages 11 and 12.]

[18] O. Kramer, *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, vol. 51. [Pages 12, 13, and 14.]

[19] B. Abdirahman Hussein and E. Samimi, "Prediction of flue gas properties using artificial intelligence: Application of supervised machine learning by utilization of near-infrared spectroscopy on solid biofuels," 2022. [Page 14.]

[20] I. N. Da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, S. F. dos Reis Alves, I. N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, *Artificial neural network architectures and training processes*. Springer, 2017. [Page 15.]

[21] S. Shanmuganathan, *Artificial neural network modelling: An introduction*. Springer, 2016. [Page 15.]

[22] I. Yilmaz and O. Kaynar, "Multiple regression, ann (rbf, mlp) and anfis models for prediction of swell potential of clayey soils," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5958–5966, 2011. [Page 15.]

[23] H. R. Patel, A. M. Patel, H. A. Patel, and S. M. Parikh, "Hyperparameter tune for neural network to improve accuracy of stock market prediction," in *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 1*. Springer, 2022, pp. 65–76. [Pages 15 and 16.]

[24] D. Berrar, "Cross-validation." 2019. [Page 16.]

[25] V. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the influence of normalization/transformation process on the accuracy of supervised classification," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2020, pp. 729–735. [Page 17.]

[26] P. J. M. Ali, R. H. Faraj, E. Koya, P. J. M. Ali, and R. H. Faraj, "Data normalization and standardization: a technical report," *Mach Learn Tech Rep*, vol. 1, no. 1, pp. 1–6, 2014. [Page 17.]

[27] X. Zhi, S. Yuexin, M. Jin, Z. Lujie, and D. Zijian, "Research on the pearson correlation coefficient evaluation method of analog signal in the process of unit peak load regulation," in *IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 2017. doi: 10.1109/ICEMI.2017.8265997 pp. 522–527. [Page 18.]

[28] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020. [Pages 18, 19, and 45.]

[29] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing," 2018. [Page 19.]

[30] A. Baldominos, I. Blanco, A. J. Moreno, R. Iturrarte, Ó. Bernárdez, and C. Afonso, "Identifying real estate opportunities using machine learning," *Applied sciences*, vol. 8, no. 11, p. 2321, 2018. [Pages 20, 32, and 47.]

[31] P.-F. Pai and W.-C. Wang, "Using machine learning models and actual transaction data for predicting real estate prices," *Applied Sciences*, vol. 10, no. 17, p. 5832, 2020. [Page 23.]

[32] S. Cheriyan, S. Ibrahim, S. Mohanan, and S. Treesa, "Intelligent sales prediction using machine learning techniques," in *International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2018, pp. 53–58. [Page 24.]

[33] Booli, "Om oss - booli." [Online]. Available: https://www.booli.se/p/om-booli [Page 26.]

[34] ——, "Booli api." [Online]. Available: https://www.booli.se/p/api [Page 26.]

[35] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of performance of data imputation methods for numeric dataset," *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, 2019. [Page 29.]

[36] S. Raschka, *Python machine learning*. Packt publishing ltd, 2015. [Page 29.]

[37] O. Kramer and O. Kramer, "Scikit-learn," *Machine learning for evolution strategies*, pp. 45–53, 2016. [Page 29.]

[38] S. learn developers, "sklearn.svm.SVR." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html [Page 30.]

[39] J. Alvarsson, S. Lampa, W. Schaal, C. Andersson, J. E. Wikberg, and O. Spjuth, "Large-scale ligand-based predictive modelling using support vector machines," *Journal of Cheminformatics*, vol. 8, no. 1, pp. 1–9, 2016. [Page 30.]

[40] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *Journal of Property Research*, vol. 38, no. 2, pp. 99–129, 2021. [Pages 32 and 33.]

[41] U. Khair, H. Fahmi, S. Al Hakim, and R. Rahim, “Forecasting error calculation with mean absolute deviation and mean absolute percentage error,” in *journal of physics: conference series*, vol. 930, no. 1. IOP Publishing, 2017, p. 012002. [Page 33.]

[42] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, “Estimation of energy consumption in machine learning,” *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, 2019. [Page 46.]

[43] Google, “Carbon commitments - Building a carbon-free future for all.” [Online]. Available: https://sustainability.google/commitments/carbon/ [Page 47.]

Appendix A

Table 1 shows the different features the data set had when first fetched.

Additional area
Apartment number
Booli id
City
Construction year
County name
Floor
Latitude
List price
Living area
Longitude
Municipality name
Named areas
Object type
Plot Area
Published
Rent
Rooms
Sold date
Sold price
Sold price source
Source id
Source name
Source type
Source url
Street address
Url

Table 1: The different features in the data set from Booli

Appendix B

The order and amount of data points that were removed from the data set.

1. Data points that had cells empty for the feature construction year were deleted. A total of 114,519 data points were removed.

2. 3957 data points were removed because there was no data in the feature rooms

3. 2 data points were removed because no data in the feature ”municipality name”.

4. 1500 data points were removed because of empty cells in the feature “living area”.

5. 3665 data points were removed because of empty cells in the feature “published”.

6. Several categories from the ”object type” feature were removed since they were not relevant to this thesis. The categories removed were: ”Plot/Land,” ”Courtyard,” ”Holiday Cottage,” and ”Other”, a total of 20,930 data points.

7. Apartments that did not specify on what floor the apartment was on were removed, 77,593 data points were deleted.

8. Removed 16,872 data points that did not specify the listening price

9. 142 data points were removed due to duplicates.

TRITA-EECS-EX-2023:514

www.kth.se