Norwegian School of Economics
Bergen, Fall 2020

# Machine Learning for Property Valuation

*An empirical study of how property price predictions can improve property tax estimations in Norway*

**Martin Foldvik Buodd & Erlend Jørgensen Derås**

**Supervisor: Morten Sæthre**

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

# Acknowledgements

This thesis is written as part of the Master of Science in Economics and Business Administration, with a major in Business Analytics, at the Norwegian School of Economics (NHH). This thesis constitutes 30 ECTS in our master's degree.

We wish to express our gratitude towards our supervisor, Morten Sæthre, from the Department of Economics at the Norwegian School of Economics, for always showing great interest in our work and providing us with helpful insights throughout the semester. Your continuous support and feedback have been invaluable for keeping us on the right track.

We would also like to thank Eiendomsverdi for providing us with essential data material used and analyzed in our thesis.

Norwegian School of Economics

Bergen, December 2020

Martin Foldvik Buodd

Erlend Jørgensen Derås

# Abstract

This thesis investigates whether machine learning methods can improve property price predictions, leading to more accurate property tax estimations in Norway. This study is important to ensure fair and trustworthy taxation for Norwegian taxpayers. The current method for predicting property values is a hedonic pricing model, developed by Statistics Norway using multiple linear regression. This model shows that 25% of all predicted property prices deviate by more than 20% of their observed price. These predictions are further used to estimate property tax, and the deviation in the current model suggests there is potential for improvement.

The use of machine learning to improve property price predictions has yet to be explored by Statistics Norway. Consequently, this thesis investigates the predictive performance of more advanced machine learning methods on transacted properties, covering three districts in Oslo, from 2005 to 2020. These methodologies include decision trees, Random Forest, gradient boosting, and neural networks. All methodologies, except decision trees, performed better than multiple linear regression. Gradient boosting produced the best results, with an RMSE of 0.1140 compared to an RMSE of 0.2132 from the multiple linear regression. The total percentage of predictions deviating more than 20% of observed values were 6.4% using the gradient boosting approach, providing an improvement of 74% to the current method.

The main conclusion drawn from this research confirms the superiority of machine learning methods for property valuation, capable of improving the current methods for estimating property tax in Norway. Additionally, the use of Local Interpretable Model-agnostic Explanations (LIME) can make the results transparent and compliant with current GDPR legislation for automated decisions. This thesis recommends the implementation of gradient boosting as the new method for property valuation in Norway.

***Keywords*** – Property tax, machine learning, LIME, GDPR, gradient boosting

# Contents

# List of Figures

# List of Tables

# 1 Introduction

There has always been great interest in property price appraisal, and several discussions on how to best predict the «correct» price of properties have taken place. The advent of more sophisticated statistical learning methods in later years, such as machine learning, has enhanced the possibilities for modeling and understanding datasets complex in size, structure, and detail, thus potentially improving predictions (James et al., 2017).

While the possible applications for such predictions are numerous, this thesis is limited to how predicting a property's market value can improve property tax calculations on residential properties and holiday homes. Property tax is a tax that municipalities in Norway can opt to introduce for its inhabitants. The tax is calculated based on the property's predicted market value from the last wealth and income tax assessment (The Norwegian Tax Administration, 2020b). These predictions are based on valuations from Statistics Norway, in which factors such as structural and locational attributes are used to estimate a *hedonic pricing model* using multiple linear regression (MLR). Such models are often applied when predicting a quantitative response, such as sales prices or values.

A report written by Statistics Norway in 2020 looks at the ratio between the predicted and actual observed prices in the model (Statistics Norway, 2020a). It reveals that as many as 25% of predicted prices deviate by more than 20% of their observed prices. Research reveals that more advanced machine learning models can outperform the current use of multiple linear regression, suggesting it could be possible to reduce the discrepancy between the predicted and observed prices (Pai & Wang, 2020).

It is crucial to address the potential risks related to underestimation or overestimation for two reasons. First, if the model severely underestimates property values, taxpayers will be subject to less property tax than they should, because of the lower calculation basis. Second, if the model consequently overestimates property values, individuals may be subject to higher property taxes than they should, due to the higher calculation basis. Both of these scenarios are undesirable and serve as guidelines for the thesis' research and discussions.

We would like to mention that even though the terms *price* and *value* are used somewhat interchangeably throughout the thesis, they refer to the same concept, namely *market*

*value.*

## 1.1 Purpose and Research Question

Hedonic pricing models are commonly used for property valuations, and are often recognized for their simplicity and interpretable description of how the inputs affect the output. The main objective of the hedonic pricing models is to estimate the contribution of a property's attributes to its market price, and they are usually created using multiple linear regression (MLR).[1] MLR either assumes that "the regression function $E(Y \mid X)$ is linear, or that the linear model is a reasonable approximation" (Hastie et al., 2008, p. 44). If this relationship is far from linear, most conclusions drawn from the fit are suspect (James et al., 2017, p. 92). To account for such nonlinearity, nonlinear transformations, referred to as functional forms, can be applied to capture nonlinear relationships between the target variable and attributes in the model. Such transformations can be performed by taking the square root, logarithmic transformation, or square of one or more variables. The choice of this functional form is often not straightforward. Few theories offer sufficient guidance on choosing the proper functional form to capture nonlinearity (Roberts & Zhao, 2020; Halvorsen & Pollakowski, 1981). An incorrect choice of the functional form may result in inconsistent output estimates (Bloomquist & Worley, 1981; Goodman, 1978).

Usually, such pricing tools are utilized in combination with human expertise to predict a property's sales price when transacted. However, when estimating property tax, this would be inefficient as this prediction has to be updated often to account for changes in the property's market value. This challenge has resulted in the general prediction model created by Statistics Norway. Considering this, in combination with the inconsistencies in the current model, our defined research question is:

*How can machine learning methods improve property price predictions, leading to more accurate property tax estimations in Norway?*

Our research question is motivated by several factors. Statistics Norway still uses the same prediction model they introduced back in 2009, and they confirm that they have not explored the use of more advanced machine learning methods for similar prediction purposes

[1] For brevity, multiple linear regression will be referred to through the abbreviation MLR for the remainder of the thesis.

before (Statistics Norway, personal communication, November 23, 2020). Consequently, answering the research question will elevate this debate to a higher national level through new perspectives on the topic of advanced machine learning. Furthermore, the research question serves both a scientific and societal purpose. Scientific, by exploring how the application of several different machine learning methods can predict a property's market value, and societal, by ensuring that tax estimations are fair and transparent. As overestimations of property values directly affect the calculation basis for property taxes, it is in the public's interest that the predictions are accurate and reliable. The discrepancy between predicted and observed values in today's model will result in inaccurate property tax calculations; thus, we will explore how more accurate predictions improve these calculations' reliability. Further, we will discuss the importance of transparency of property tax calculations in light of current legislation on data protection and governance.

To answer the research question, the thesis is divided into eight chapters. This chapter has explained our motivation for the chosen topic and research question, and we will further introduce basic machine learning concepts in the coming two sections. In chapter two, we will present literature relevant to our research question. Chapter three contains important background information, covering current prediction practices in Norway, the concept of property tax, and GDPR. In chapter four, we describe the data used to estimate our models, along with preprocessing and treatment of the variables. Chapter five will explore machine learning terminologies before explaining the models used to estimate property values. Chapter six contains the empirical results with comparisons to the current results from Statistics Norway. Chapter seven will discuss the empirical results, elaborate on societal impacts, and offer suggestions to further research. Finally, in chapter eight, we present the concluding remarks and our answer to the research question.

## 1.2 Concepts of Machine Learning

In this section, a simple backdrop on machine learning is provided to facilitate the reader's understanding of the methods and concepts explored throughout the thesis. We emphasize that this section serves as a short introduction to machine learning and that the terminologies will be expanded upon further in chapter 5.

Machine learning involves learning hidden patterns within the data and subsequently using

the patterns to classify or predict an event related to the problem or research question. In essence, machine learning algorithms extract useful information from provided input data.

Generally, machine learning can be classified as supervised or unsupervised. In supervised learning, the machine learning algorithm learns from a set of independent variables that have an associated dependent response (outcome) variable. Based on this, we want the algorithm to predict the response variable of previously unseen data. The algorithm is trained over a series of provided data, and the idea is that, after enough training, it is able to predict the response variable of new observations to an arbitrary degree of precision. Whether the algorithm has found the correct answer is usually measured through a loss function. As we can use the loss function to define a precise measure of success, or lack thereof, it can be used to judge the machine learning algorithm's performance and compare the effectiveness of different types of algorithms over various situations (Hastie et al., 2008).

In unsupervised learning, there is no such clear measure of success. Unsupervised learning occurs when the machine learning algorithm is provided with data without a response variable, e.g., images or text, to find patterns based on common attributes in the data. This process is performed with minimal human intervention.

We will only explore supervised learning in this thesis, as our prediction task requires outputs of a response variable. To predict the response variable $Y$, we observe different values of the independent variable $X$, defined as the predictor. In machine learning, we assume that there is some relationship between $Y$ and $X$, which can simply be written as:

$$Y = f(x) + \epsilon \tag{1.1}$$

In equation 1.1, $f$ is a fixed function that is unknown, $\epsilon$ is a random error term, independent of $X$ and has mean zero. In this formula $f$ represents the systematic information that $X$ provides about $Y$ (James et al., 2017, p. 16). This $f$ function could either be used for inference or prediction purposes. We will focus on the latter throughout this thesis.

Another important aspect of machine learning is the tradeoff between bias and variance. The total prediction error for a given value $x_0$ can be decomposed into the sum of three

fundamental quantities shown in equation 1.2 below:

$$Err(x_0) = IrreducibleError + Bias^2 + Variance \tag{1.2}$$

where the irreducible error is the variance of the target around its true mean $f(x_0)$, and this cannot be avoided no matter how well we estimate our $f(x_0)$. $Bias^2$ is the amount by which our estimate's average differs from the true mean; the last term, variance, is the expected squared deviation of $\hat{f}(x_0)$ around its mean. As the irreducible error cannot be avoided, we can reduce the overall prediction error for a given value $x_0$ by reducing bias and variance, introducing the bias/variance tradeoff (Hastie et al., 2008, p. 37). It is impossible to develop a model with zero bias and zero variance. However, one would seek to optimize the tradeoff to reduce total prediction error when estimating models. The bias/variance tradeoff will be further elaborated in the methodology chapter.

## 1.3 Tradeoff Between Accuracy and Interpretability

An issue with more advanced machine learning methods is that they are not as interpretable as MLR. Some advanced methods contain internal properties so complex that they are uninterpretable to humans, and thus it can be challenging to know precisely why a model has made a particular prediction. This is important to keep in mind, as the purpose of property valuation in this thesis is to use the results to levy property tax, which demands some degree of interpretability to maintain transparency.

However, the field of interpretable machine learning is continuously developing, aiming to enhance the understanding of decisions made by more complex models. Ribeiro et al. (2016) suggest that using *model-agnostic approaches* could make such models more interpretable. Model-agnostic approaches in this context involve training an interpretable model on the predictions of a complex model, tweaking the inputs, and then see how the output of the complex model changes. This is considered a post-hoc explanation, meaning that rather than restricting the complexity of the model, thus keeping it interpretable, we extract explanations after the complex model is built. While MLR is recognized as highly interpretable, it may lack performance compared to other more complex methods and vice-versa; thus, there is a tradeoff between them.

With this in mind, Ribeiro et al. (2016) have developed a framework, Local Interpretable Model-agnostic Explanations (LIME), which is an explanation technique capable of interpreting any regression model. LIME's intuition is that every complicated machine learning method can be explained linearly on a local scale, meaning that LIME can be used to describe the logic behind each individual prediction of a complex model. Ribeiro et al. propose an implementation of local surrogate models trained to approximate predictions of the underlying complex model. Rather than training a global surrogate model, LIME focuses on training multiple local surrogates to explain each *individual* prediction (Molnar, 2020). These promising results indicate that the interpretation of more advanced machine learning may be more straightforward than first suggested. LIME's introduction enables us to reduce the tradeoff between accuracy and interpretability, thus making the use of more complicated models more attractive for predicting property prices. We will further elaborate on the application of LIME in section 5.5.

# 2 Literature Review

This section presents and reviews notable literature relevant to the research question, focusing on machine learning methods to predict housing prices. As stated, this area of research has not been explored in Norway before. Consequently, relevant key works, theories, and concepts from *other* countries useful for providing a backdrop to our research question will be defined and mapped out. The vast majority of this literature discusses alternative machine learning methods that have shown the potential to improve accuracy when predicting property values.

When predicting property values, there is no such thing as an estimation model capable of predicting the correct output result for every observation. However, authors in the field of prediction have suggested methodologies that yield superior predictions compared with traditional MLR. Tree-based models, in particular, have shown promising results for this purpose. Tree-based methods involve *stratifying* or *segmenting* the predictor space into several simple regions.

Consider two predictors of property value: age and size of living area. We can divide the predictor space into properties older or newer than 50 years. These two categories can then be separated into properties with a living area larger or smaller than $70m^2$, resulting in a predicted value for each property. To predict a given observation, the mean or mode of the training observations in the region to which it belongs is typically used. Because the set of splitting rules used to segment the predictor space can be summarized in a tree, these types of approaches are often described as *decision tree methods* (James et al., 2017, p. 303). Geurts et al. (2009) mention three key ingredients of decision trees' success. First, decision trees are recognized as highly interpretable, meaning that the model's decisions are transparent and understandable to the human eye. Second, decision trees carry inherent flexibility, making them applicable to a wide variety of problems. Finally, they emphasize their ease of use, making them accessible even to non-specialists.

Fan et al. (2006) explores the possibility of using decision trees to predict apartment prices in Singapore. Their paper applies decision trees to analyze the relationship between each apartment's attributes and their sales price, as input and output, respectively. They recognize several perks of using decision trees over MLR. The tree itself can analyze

both linear and nonlinear relationships between the input- and output variables – as opposed to MLR, where it can be challenging to choose the appropriate functional form. Second, they argue a decision tree is more interpretable than MLR, easily allowing users to determine the most influential attributes of the model. This is possible as the decision tree is produced by straightforward splitting rules that partition the dataset's observations into different interpretable regions.

However, Fan et al. (2006) also point out drawbacks of employing a decision tree for property valuation. While the algorithm is proficient at splitting continuous variables by choosing somewhere in the range of values, it might be difficult for this approach to analyze and predict a continuous variable's exact movement. Furthermore, James et al. (2017) assert that decision trees are inherently non-robust, implying a small change in the data might cause massive changes in the final predictions.

Addressing some of these concerns, Hong et al. (2020) have recently written a paper that builds on the work of Fan et al. (2006). In their research, Hong et al. compare the application of Random Forest with MLR for property valuation in Seoul, South Korea. Random Forest is a technique that ensembles *multiple* decision trees, producing an estimation based on averaging predictions made by the decision trees. Each of these decision trees is built independently after the following form: a subset of randomly chosen predictors are chosen to grow each tree on a sub-sample of the same dataset. After a desired number of trees are grown, predictions are averaged over the different trees.

Hong et al. (2020) found several advantages with Random Forest compared to MLR when predicting housing prices. First, the authors address the difficulty of choosing the appropriate functional form when using MRL. As Random Forest contains decision trees' properties, it can deal with both linear and nonlinear relationships without explicit user-specifications, meaning that Random Forest may be more appropriate for dealing with MLR's difficulty of choosing the appropriate functional form. Random Forest also handles and the unsteadiness of variable influence across different segments because it is built from multiple decision trees. Further, the authors emphasize that the Random Forest approach can manage categorical variables with several levels. With MLR, multiple qualitative variables lead to a larger number of estimated parameters, which often results

in overfitting.[2] In their research, only 0.3% of predictions deviated more than 50% of the observed price using Random Forest. The traditional MLR missed by more than 50% in almost 3.8% of all predictions, indicating that Random Forest has a lower spread in its predictions. Random Forest was also more accurate as 72% of predictions fell within 5% of the observed price, compared to 17% in MLR (Hong et al., 2020, p. 142).

Even though Random Forest has its perks compared to MLR and decision trees, Hong et al. (2020) express that this approach is more challenging to interpret, although not impossible. As the Random Forest grows multiple decision trees, the model's complexity increases accordingly, at the expense of reduced interpretability. In addition to this, as opposed to MLR, where the estimation could be clearly explained through *all* predictors, the Random Forest algorithm uses a *random* sample of predictors, which further complicates the explanation of the output.

Another way to assemble decision trees is through *gradient boosting.* This methodology involves the iterative fitting of multiple decision trees. The trees are grown sequentially, meaning that each new tree is grown based on previously grown trees. Gradient boosting is especially useful for predicting a continuous variable, such as property price, based on the input of many potentially interacting categorical and continuous variables (Gu & Xu, 2017). Researchers Kagie & van Wezel (2007) employed boosted decision trees in the Dutch housing market and compared the results with MLR. Their research discovered that employing boosted decision trees improved prediction accuracy by over 40% compared to MLR.[3] While this result is promising, gradient boosting suffers from some of the same limitations as Random Forest, in the sense that the model's interpretability is diminished by the large ensemble of trees grown. Further, Li & Bradic (2018) also underline that boosting can be sensitive to outliers because of its commitment to fixing errors from previously grown trees, leading the model to seek to perfect the prediction of outliers in the in-sample data.

While tree-based methods are recognized for their flexibility, while at the same time showing excellent performance for predicting housing prices in other countries, some researchers have looked into a more comprehensive approach known as *artificial neural*

[2]Overfitting refers to the situation where a model fits the training data too well, often resulting in inaccurate predictions.

[3]Accuracy in Kagie  van Wezel's (2007) paper is measured by Mean Relative Error.

*networks.* Artificial neural networks consist of three separate layers: the input layer (independent variables), one or more hidden layers, and the output layer (dependent variable). Guidotti et al. (2018) explain how the hidden layers are usually referred to as a black box, where the hidden layer(s) allow for very complicated prediction functions.[4] The logic behind the predictions is hidden behind these complicated functions, making them challenging to interpret reasonably.

The application of neural networks for predicting property prices has been explored by Limsombunchai et al. (2004), focusing on the housing market in Christchurch, New Zealand. The authors found that utilizing an artificial neural network offers several advantages over MLR for this purpose. First, the neural network's flexibility and nonlinear properties make them capable of learning any problem (Fortuner, 2017). Further, the user does not need to specify details about the structure or parametric form before estimating the regression equation. It lets the network determine the appropriate functional form, as opposed to MLR, where this has to be specified. Limsombunchai et al. (2004) found that, in some cases, the artificial neural network improved prediction accuracy by close to 50% compared to MLR.[5] Although artificial neural network applications have shown promising results, the authors further recognize the challenges of this approach due to the black-box nature of neural networks. Even though the model may return a desirable outcome, one can often not know precisely why the received results were produced, which may cause problems in cases where interpretability is a necessity, such as in housing price predictions and tax calculations.

Although the applications of artificial neural networks have shown promising results, the authors further recognize the challenges related to this approach due to the black-box nature of neural networks. Even though the model may return a desirable outcome, one can often not know precisely why the received results were produced, which may cause problems in cases where interpretability is a necessity, such as in housing price predictions and tax calculations.

Regarding what we will add to the existing literature, our thesis' novelty lies in investigating how similar machine learning approaches will perform in predicting property market values

[4]The term "black-box" is a common metaphor within machine learning terminology, and typically refers to a model where we can observe inputs and outputs, but find it harder to observe the internal workings.

[5]Accuracy in Limsombunchai et al. (2004) is measured by Root Mean Square Error.

in Norway. We will further see how increased accuracy often comes at the expense of interpretability, thus consider the tradeoff between them.

# 3 Background

This section serves to describe and establish fundamental knowledge relevant to our research question. In section 3.1, we present the current practice for property valuation in Norway. This will be discussed with reference to the prediction report written by Statistics Norway. The report is crucial for our thesis, as it provides a thorough explanation of the current model used to predict property market values, used by the Norwegian Tax Administration when calculating property tax. Section 3.2 will present the concept of property tax and how the tax is determined. Finally, in section 3.3, the importance of transparent predictions will be addressed in light of the recently introduced General Data Protection Regulation (GDPR).

## 3.1 Current Practice for Property Valuation

In Statistics Norway's pricing model, properties are defined in terms of attributes, each of which contributes to a property's predicted market value. Such attributes can be the number of bedrooms, size of the property, or the number of bathrooms. Statistics Norway and multiple independent actors rely on MLR to predict property prices due to its ease of use and interpretability. MLR is typically applied with a logarithmic form of the price because house prices tend to be log-normally distributed (Statistics Norway, 2020a). Statistics Norway explains that they use the logarithmic form because it limits the influence of extreme observations and that the log transformation gives better distributional properties compared with the data's original form.

In section 1.1, we introduced the prediction report written by Statistics Norway, which aims to provide detailed documentation on the applied regression model and an overview of results from current predictions of property prices in Norway. Furthermore, the weaknesses and possible improvements to their model are presented in the report. The report is published yearly, and it serves to inform individuals and public organizations on how these predictions are made.

**About the Model**

The model is based on residential property sales over the last ten years and estimates the average price per square meter for properties based on size, age, and geographical

location. Their model only includes five different attributes; the living area in square meters, location, age, yearly dummy indicators, and price zone.[6] Different areas of Norway are more expensive than others, thus such areas are divided into different price zones in which properties are categorized. The property's age is binned into four categories (<10 years, 10-19 years, 20-34 years, and >34 years). The relationship between the predicted value and these attributes is determined by a semi-logarithmic function, making the model linear in its parameters while introducing nonlinearity in the attributes. Statistics Norway justifies using only five variables with that it is easy for users to understand the model and that they are willing to sacrifice some of the model's accuracy to achieve this. Statistics Norway does, however, suggest that their model's accuracy can be improved upon by including more variables (Statistics Norway, 2020a).

**Uncertainty Related to the Model**

Considering all the potential drawbacks with MLR, the results should be further evaluated in terms of uncertainty. Table 3.1 below provides an overview of the percentage distribution of estimated prices over observed turnover prices.[7]

[6]Five main variables. Some of these are transformed into dummy-variables, which increases the total number to 30 predictors for the year 2019. See the report from Statistics Norway (2020a) for a full explanation.

[7]For a table covering the percentage distribution for apartments only, see appendix A1.

| **The percentage distribution of estimated prices over observed turnover prices** | **Percent** | **Cumulative percent** |
|---|---|---|
| **0 – 40** | 0.07 | 0.07 |
| **41 – 60** | 0.87 | 0.94 |
| **61 – 80** | 7.85 | 8.79 |
| **81 – 100** | 36.31 | 45.10 |
| **101 – 120** | 38.28 | 83.38 |
| **121 – 140** | 11.98 | 95.36 |
| **141 – 160** | 2.84 | 98.20 |
| **161 – 180** | 0.89 | 99.09 |
| **181 – 200** | 0.38 | 99.48 |
| **200 +** | 0.52 | 100 |

**Table 3.1:** Percentage distribution of estimated prices over observed turnover prices from Statistics Norway's model covering all housing types.

The results obtained from the model in Statistics Norway's report reveal that compared to their observed values, only 0.07% of properties are underestimated by more than 40%, while 4.63% of properties are overestimated by more than 40%. The table shows that while the model can miss by a lot, it only does so for under 5% of all observations. However, about 22% of estimations are outside the range of +/-20% of the observed values (Statistics Norway, 2020a, p. 14), implying that the model still has room for improvements in terms of accuracy.

It should further be addressed that the price achieved in the market may be affected by who is informed about the listing and who participates in the bidding process. These are factors causing fluctuations that are hard to account for due to their randomness.

## 3.2 Application of Property Tax

Versions of property tax can be dated back to 5,000 B.C. in Egypt, Persia, and throughout the ancient world. The primary focus of early property taxation was land and its production value, while throughout history, it has served different purposes (Carlson, 2018, p. 3).

Today, almost every country has some kind of annual tax on land and buildings, and there are different reasons why this tax is implemented.

Rosengard (2013) lists several reasons why property tax serves a societal and economic purpose. First, it is often the primary source of municipalities' discretionary revenue, and thus an essential component of fiscal decentralization that supports local autonomy and complements intergovernmental fiscal transfers. Second, some people view property tax as socially equitable because it is roughly progressive, loosely correlated with local government benefits, and a way to enable the public sector to get a share of private sector windfall gains from appreciation of property values, mainly due to public investments in previously unserviced land. Further, municipalities can perceive property tax as economically efficient because it is difficult for individuals to avoid and easily enforceable if evaded.

On the other hand, Rosengard (2013) also criticizes the concept of property tax. One argument against property tax is that while the high number of statutory taxpayers create a broad tax base, it can be a political and administrative nightmare to organize this in practice. Another criticism is that while citizens might accept the tax in principle, there is still a widespread resentment in some countries to enforcement proceedings – sometimes seen as a threat to the home's sanctity. Rosengard argues that there is no direct link between tax liability and actual ability to pay the tax. Some taxpayers may be considered "asset rich, but cash poor." Worst case, if they do not have the liquidity to pay the tax, they may end up selling their property to finance their tax obligation.

In Norway, property tax is described as a municipal tax that each municipality can levy according to the Property Tax Act first introduced in 1975. This act states that, if introduced, the tax rate must be between 2‰ and 5‰ of the property tax basis.[8] Each municipality also has the opportunity to introduce an annual basis deduction, which is subtracted from the basis when the property tax is calculated (The Norwegian Tax Administration, 2020b). Some municipalities also require property tax from commercial real estate, power plants, wind turbines, and petroleum plants. However, due to our research question's scope, we will restrict our thesis to focus on property tax from residential properties.

[8]The property tax rate interval was 2‰ - 7‰ up until 2020, when the top rate was reduced to 5‰. As of 2021 the tax rate will be reduced further to an interval of 2‰ - 4‰.

Municipalities collecting property tax are required to implement the valuation method described in section 3.1 created by Statistics Norway. This method enables municipalities to use data from Statistics Norway instead of doing appraisals themselves to estimate the market value of properties.[9] The Property Tax Act requires a mandatory reduction factor of 30% to be deducted from the estimated market value to ensure that certain properties do not get an unreasonably high property tax basis (The Norwegian Tax Administration, 2020a).[10] This reduction factor makes sure that the property tax basis is 70% of the property's actual market value. Municipalities are, however, allowed to set an additional basis deduction, complementing the mandatory reduction factor.

To illustrate how property tax is implemented in practice, Oslo municipality will serve as an example in table 3.2 below. The defined market values are assumed to be equal to the observed values, implying a perfect prediction. As of 2020, the additional annual basis deduction for Oslo properties is set to NOK 4,000,000, while the tax rate is set to 3‰ (Oslo Kommune, n.d.).

| **Market value** | **Basis after the obligatory reduction factor of 30%** | **Property tax basis after additional basis deduction (NOK 4,000,000)** | **Estimated annual property tax (3‰)** |
|---|---|---|---|
| 5,700,000 | 4,000,000 | 0 | No property tax |
| 6,000,000 | 4,200,000 | 200,000 | 600 |
| 10,000,000 | 7,000,000 | 3,000,000 | 9,000 |

**Table 3.2:** Illustration of property tax calculation for the municipality of Oslo. All numbers in NOK.

Table 3.2 only represents Oslo municipality, and other municipalities might operate without annual basis deductions and different property tax rates. As of 2020, a total of 319 Norwegian municipalities have implemented the property tax. From these 319 municipalities, 77 operate with an additional annual basis deduction when estimating annual property tax. The total property tax deriving directly from residential properties

[9] I.e., some form of market value (sales value).
[10] Described in §8 A-2(1).

and holiday homes was NOK 7.6 billion in 2019, which accounts for a total of 3.1% of total operating revenue for the municipalities (Statistics Norway, 2020b).

### 3.2.1 Amending the Tax Return

If taxpayers suspect that the predicted market value is too high relative to the actual market value, they can reduce the market value by amending their tax return within six weeks after receiving the property tax bill. To reduce the market value, certain requirements need to be met. For primary dwellings, the market value can be amended if the property's tax value is greater than 30% of the documented market value. The market value may be amended for secondary dwellings if the property's tax value is greater than the documented market value. These valuations must be documented through a professional appraiser and are valid for five years The Norwegian Tax Administration (n.d.a). The mandatory reduction factor of 30% is also applied to the new documented market value to ensure all taxpayers benefit from the reduction, even those whose properties are now valued correctly.

*Tax value* is a new term we have yet to define, and is not the same as market value or property tax basis. Tax value is determined differently for primary dwellings, the home you reside in, and secondary dwellings, such as holiday homes. For primary dwellings, the tax value is 25% of the market value, while for secondary dwellings, the tax value is 90% of the market value The Norwegian Tax Administration (n.d.b). This tax value is used to determine other forms of wealth tax but is not directly employed in determining property tax, which is a form of wealth tax. Therefore, tax value is somewhat counter-intuitively, not used in the calculation of property tax, but used to determine whether you have grounds for amending your property's market value, which in turn determines your property tax.

This is exemplified in table 3.3 below, examining four example cases where the predicted market value is higher than the market value documented through an appraisal.

| | **Predicted market value (NOK)** | **Documented market value (NOK)** | **Primary dwelling** $\frac{PMV*25\%}{DMV}$ | **Secondary dwelling** $\frac{PMV*90\%}{DMV}$ | **Overestimation of predicted market value** |
|---|---|---|---|---|---|
| 1 | 3,000,000 | 2,600,000 | 28.9% | 103.9% | 15.4% |
| 2 | 4,000,000 | 3,000,000 | 33.3% | 120% | 33.3% |
| 3 | 1,200,000 | 1,000,000 | 30% | 108% | 20% |
| 4 | 1,111,112 | 1,000,000 | 27.8% | 100% | 11.1% |

**Table 3.3:** Predicted market value is retrieved from Statistics Norway's model. Documented market value is determined through professional appraisal. PMV refers to predicted market value, and DMV refers to documented market value. Primary- and secondary dwellings calculate tax value and display whether the case is outside the threshold for amendment. As primary- and secondary dwellings have a separate tax value, both are included. Overestimation of predicted market value shows how much the prediction is overestimated in relation to the documented market value.

Case 1 can reduce the market value if the property is a secondary dwelling, as the tax value is greater than the documented market value but cannot reduce the market value if the property is a primary dwelling, as the tax value is not greater than 30% of the documented value. Case 2 can amend the market value regardless of if it is a primary or secondary dwelling as both values are above the specified threshold. Case 3 and 4 display the degree of overestimation required to amend the market value. For primary dwellings, one can amend the market value if the predicted property value is overestimated by more than 20%, and for secondary dwellings, one can amend the market value if the predicted property value is greater than 11.1% of the documented value. These thresholds are *important* to keep in mind, and they will be explored further in chapter 7.

The validity of these amendments is further subject to control by local tax authorities. Consequently, it is in both the tax authorities' and individuals' interest to obtain accurate predictions as the calculation process will be more effective by reducing the number of amendments and, subsequently, the number of controls.

## 3.3 General Data Protection Regulation

In 2016 the European Union introduced a new regulation called the General Data Protection Regulation (European Union, 2016b). The regulation consists of legislative

acts governing data protection and privacy within the European Union, and it affects how companies and government bodies can utilize or exploit the personal data of EU citizens.[11] EU citizens are provided certain rights such as the right to be forgotten, the right of access, the right to rectification, and the right not to be subject to a decision based solely on automated processing.[12] Using an automated machine learning algorithm to determine property prices would violate the right not to be subject to an automated decision. However, there are exceptions to this right. The relevant exception in our case is found in GDPR article 22(2)(b): "The decision is authorized by Union or Member State law to which the controller is subject, and which also lays down suitable measures to safeguard the data subject's rights and freedoms and legitimate interests" (European Union, 2016b, p. 46). The Norwegian Tax Administration is subject to Norwegian, and such EU laws, meaning that the Norwegian government can approve automated decisions. Norwegian law allows the use of valuation methods, and the Norwegian Tax Administration uses this law as a basis for their current calculations.[13]

The GDPR also specifies some noteworthy restrictions for automated machine learning models, such as a subject's right to ask *why* a particular automated decision was made. This involves a discussion on whether the subject can expect a full explanation of how the automated model works, or at the very least, the *logic* behind its decisions. Such restrictions are imposed to protect EU citizens' right to privacy and ensure they are not subject to discrimination based on automated decisions, which is further recognized in the EU Charter of Fundamental Rights under Article 21(1):

> Any discrimination based on any ground such as sex, race, color, ethnic or social origin, genetic features, language, religion or belief, political or any other opinion, membership of a national minority, property, birth, disability, age or sexual orientation shall be prohibited (European Union, 2012).

To comply with such restrictions, one is dependent of transparent decisions and results, which we intend to explore in subsection 7.2.2.

[11] EU regulations also apply to members of the European Economic Area, which Norway is a part of.

[12] For an in-depth explanation of the rights of the data subject, see GDPR Chapter 3: Rights of the data subject (European Union, 2016b)

[13] Property Tax Act of 1975 ((Eigedomsskattelova, 1975): §8 A-1

# 4 Data Processing

In this chapter, an overview of the dataset and an explanation of the necessary treatment to prepare the data by addressing missing values, outliers, and unreasonable observations is provided. The data used for analysis has been collected and provided to us by Eiendomsverdi, Norway's leading provider in automated valuation models for the residential property market. Eiendomsverdi cooperates with 90% of all Norwegian real estate brokerage firms, ensuring their database is updated in real-time with data from property sales performed by real estate agents. In addition to this, Eiendomsverdi cooperates with Norwegian housing cooperatives and real estate developers to further expand their database (Eiendomsverdi, n.d.). On the other hand, Statistics Norway states that they only use data from properties listed at the Norwegian marketplace Finn (https://www.finn.no). Consequently, our data may not be identical to theirs, but it should still carry similar characteristics.

The initial dataset consisted of 88,197 property sales in Oslo gathered from 2005 to 2020 and 33 variables.[14] The observations are relatively evenly split among Oslo districts Frogner, Gamle Oslo, and Grünerløkka, visualized in figure 4.1 below:

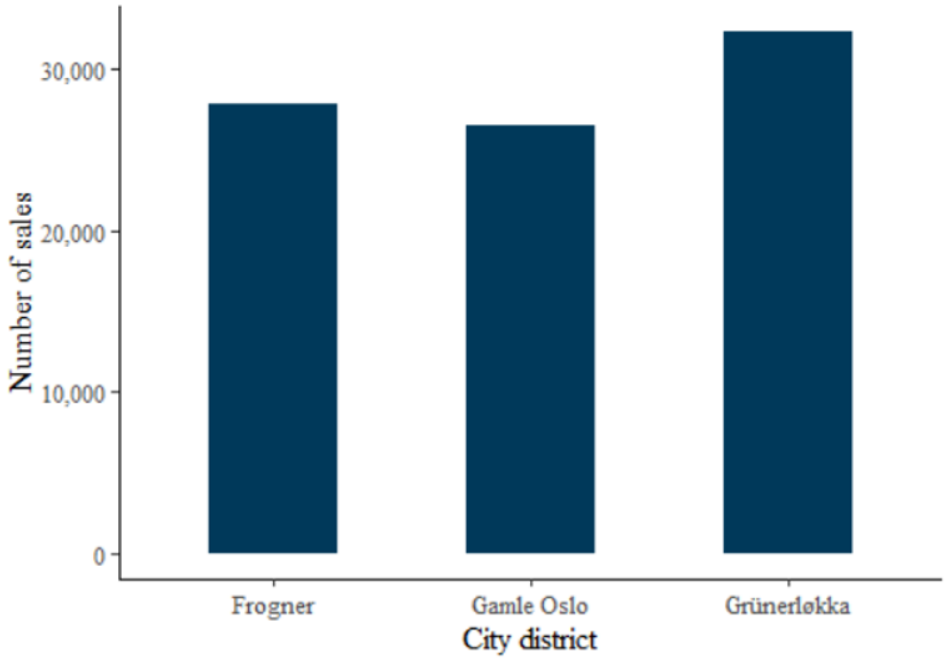


**Figure 4.1:** Distribution of sales on the three Oslo districts included the initial dataset.

[14] A full list of initial variables is found in appendix table A2, complemented with numbers of missing values in each variable.

## 4.1 Variable Treatment

The initial dataset included variables containing information on detached houses, row houses, and apartments. Frogner, Gamle Oslo, and Grünerløkka are three central parts of Oslo, and thus 99% of the observations in the data set consist of apartments. Based on this, we direct our analysis towards predicting the market value of apartments exclusively. The remaining housing types are removed from the initial data set. As we would like to compare our results with the results in Statistics Norway's report, we apply the same initial filters they use. Statistics Norway exclude apartments violating the following criteria:

- The total living area is between 12 and 350 square meters.
- The price per square meter is between NOK 8,000 NOK and NOK 200,000.

Statistics Norway specifies that these thresholds result in an insignificant dropout of extreme values where a total of 0.7% of their observations are removed. We found that these criteria excluded just two observations from our dataset.

**Cleaning Process**

Missing values are defined as values that are not available but would be meaningful if observed (Baijayanta, 2019). It is essential to locate missing values as they will pose problems for our analysis by reducing statistical power, thus potentially leaving us with a sample unrepresentative of the actual population. The presence of systematically missing values could further impose biased estimates between the predictors and the outcome variable (Groenwold et al., 2012). There can be several reasons why values are missing and multiple ways of dealing with them. Missing values can be treated through imputation methods or by merely removing observations containing missing values. If a particular variable contains a substantial number of missing values, we can discard the entire variable. A full overview of missing values present in the initial dataset can be found in Appendix A2.

Variables in the initial dataset related to whether an apartment has been sold previously or not are encoded with the most missing values. There is a logic behind this encoding; if an apartment is new or simply has not been sold previously between 2005 and 2020, these values are naturally not reported. As our data does not cover previous sales before 2005,

such potential transactions are not accounted for in the data. These missing observations are therefore accounted for through a dummy variable indicating missingness. Using *PreviousValue* as an example, the method works by replacing its empty values with 0 and then creating a new dummy variable that takes the value 1 if *PreviousValue* is 0 and zero otherwise. Intuitively, this means that the indicator variable will compensate for the missing data caused by an apartment not being previously sold while not providing any meaning to apartments with a record of its previous sale. Consequently, we can keep all observations and include them in the analysis to maintain statistical power (Groenwold et al., 2012, p. 1266).[15]

Variables related to the apartment's size, such as *PRom*, *BRA*, and *BTA*, required different treatment.[16] *PRom* has a total of 35 observations of value 0, which is impossible. We imputed missing values of *PRom* with corresponding present values of *BRA* and *BTA*. We chose this imputation method as *BRA* and *BTA* are highly positively correlated with *PRom*, with correlations of 0.9948 and 0.9851, respectively. Observations with missing values for all three variables are removed entirely, which accounted for 74 observations.

The presence of multicollinearity, which refers to the situation where two or more variables are closely related to each other, can make it difficult to separate the individual effects of these variables on the response. We computed the variance inflation factor (VIF) for MLR to remove redundant predictors, reducing potential multicollinearity. James et al. (2017, p. 101) explain VIF as "the ratio of the variance of $\hat{\beta}_j$ when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on its own." They also specify a rule of thumb, which states that "a VIF value exceeding 5 or 10 indicates a problematic amount of collinearity" (James et al., 2017, p. 101). In our dataset, the variables *BRA*, *BTA*, *SiteAreaShared*, *SiteAreaUndeveloped*, *Longitude*, *Latitude*, and *Altitude* all had VIF values above five and are removed.

We discovered a few unreasonable values in the dataset, which we illustrate through the variable *NumberOfBedrooms*. Some apartments were listed with an abnormally large number of bedrooms, so we decided to remove any observations which contained more than seven bedrooms, eliminating 20 observations.

[15] A more detailed explanation of the application of the method is provided in Appendix A3. For advantages and problems with the method, see Groenwold et al. (2012).

[16] *PRom* refers to the area of the primary room; *BRA* refers to the area of the dwelling's primary and secondary rooms in $m^2$; *BTA* refers to the area of the apartment, including outer walls.

### Data Transformation

As a rule of thumb, if the skewness is +/-1, the data is highly skewed (BPI Consulting, 2016). Thus, to prepare our data, we applied a log-transformation on numeric variables with an absolute skew greater than +/-1.[17] This transformation contributes to giving less weight on extreme observations. The reasoning behind this is to restore the data's symmetry, allowing us to obtain a bell-shaped distribution. This is illustrated with the transformation of our output variable, *TargetPrice*, below:

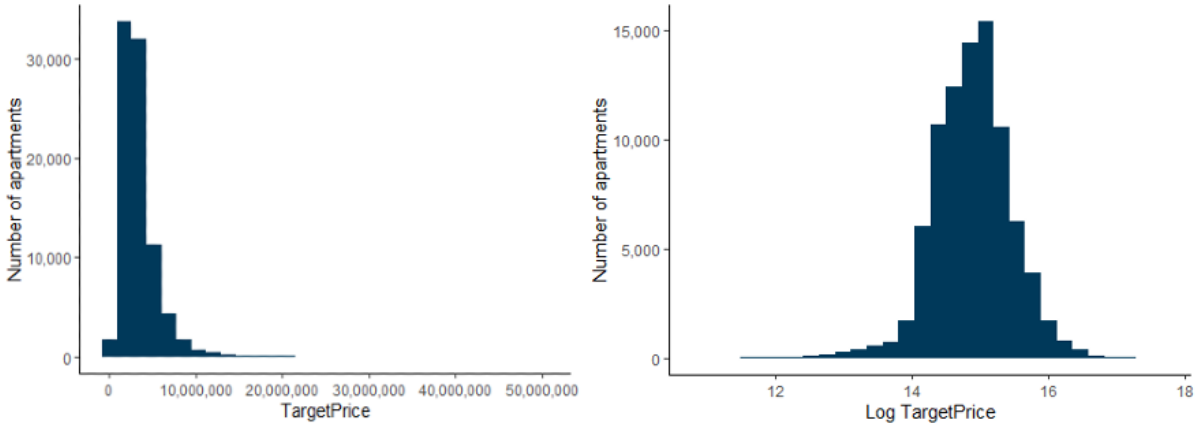


**Figure 4.2:** Log-transformation of the output variable *TargetPrice*. After transformation, the skewness is reduced from 2.6 to -0.28.

Figure 4.2 shows the distribution of our output variable *TargetPrice*. Before the log-transformation, most observations are gathered around the interval of 0 – 10,000,000, with a few outliers creating a very long tail. The plot tells us that its distribution is not gaussian and indicates a right-skew found to be 2.6, which is considered high. Skewness in the target variable will affect the prediction results by putting more weight on outliers. After transforming the variable, the skewness is reduced to -0.28.

The variable *YearSold* is further used to define the variable *Age*, which is constructed using the difference between *YearSold* and *YearBuilt*. Further, to account for time fixed effects in our data set, the variable *YearSold* is utilized as a yearly indicator, capturing macroeconomic effects, such as inflation, interest rates, changes in house price indexes, and GDP growth. This is useful as we do not have to adjust for such changes separately.

[17] We had to apply log of (x+2) to avoid taking the log of zero or negative numbers, predominantly in *Floor*, which has values of -1. This is factored in when converting results back to the data's original scale later.

The treatment process ultimately gave us a final data set with 22 variables containing 82,347 observations used for our analysis. The final dataset with variable names, corresponding definitions, units of measurement, as well as variable treatment is found in table 4.2 below:

| Variable | Definition | Unit | Treatment |
|---|---|---|---|
| ***TargetPrice*** | The sales price of the apartment | NOK | No treatment |
| ***TargetPrice Commondebt*** | Joint debt attached to theapartment | NOK | Missing values recoded as 0 |
| ***PreviousValue*** | Previous sales price | NOK | Missing values recoded as 0 |
| ***PreviousValue Commondebt*** | Joint debt attached at previous sale | NOK | Missing values recoded as 0 if PreviousValue is missing, 1 otherwise |
| ***PreviousPrice ValueCategory*** | Indicator variable, indicating whether an apartment has been previously sold in the time period of the dataset or not | 0/1 | Missing values recoded as 1 |
| ***YearSold*** | Transaction year | Year | No treatment |
| ***PreviousYearSold*** | Transaction year of the previous sale | Year | Missing values recoded as 0 |
| ***PRom*** | The primary living area of the apartment | $m^2$ | Missing values are imputed with values from BRA and BTA. Area lower than 12 and remaining missing values are removed |

| | | | |
|---|---|---|---|
| ***Floor*** | The floor level of the apartment | Numeric | Floor level >20 and missing values are removed |
| ***NumberOf Bedrooms*** | Number of bedrooms in the apartment | Numeric | Recoded missing values to 0 if PRom <50. Observations >7 and remaining missing values are removed |
| ***Balcony*** | Dummy variable for whether the apartment has a balcony attached or not | 0/1 | No treatment |
| ***Elevator*** | Dummy variable for whether the apartment has an elevator in the building or not | 0/1 | No treatment |
| ***Age*** | Age of the apartment at the time of sale | Numeric, measured in years | Negative observations recoded as 0. Missing values are removed |
| ***SiteArea*** | Area of lot | $m^2$ | Zeros and missing values are removed |
| ***CityDistrict*** | District in Oslo | - Frogner<br>- Gamle Oslo<br>- Grünerløkka | No treatment |

| | | | |
|---|---|---|---|
| ***CoastDistance*** | Distance to the nearest coast | Meters | Missing values are removed |
| ***CoastDirection*** | Direction to the nearest coast | - NE<br>- SE<br>- SW<br>- NW | Missing values are removed.<br>Binned from 360 degrees into the intercardinal directions |
| ***SiteSlope*** | Slope decline of lot | Numeric, measured in degrees | Missing values are removed |
| ***SiteSlopeDirection*** | Direction the slope of the lot is declining | - NE<br>- SE<br>- SW<br>- NW | Missing values are removed.<br>Binned from 360 degrees into the intercardinal directions |
| ***OwnershipType***[18] | Whether the apartment is freehold, stock, or part of a housing cooperative | - Freehold<br>- Cooperative<br>- Stock | No treatment |
| ***SiteOwnership Type*** | Whether the lot is freehold or leasehold | - Freehold<br>- Leasehold | No treatment |

[18]Freehold ownership is when an owner has the exclusive right to use the apartment/land for any purposes, but where the entire property is in joint ownership with other section owners; cooperative ownership is when the ownership is linked to a share in a housing cooperative; stock ownership is when the ownership of the apartment is organized as a limited company (or public limited company).

| | | | |
|---|---|---|---|
| ***SunsetHour*** | Time of day the sun sets at the property Measured at the same date for all observations | - Early<br>- Mid<br>- Late | Binned from format HH:MM<br>Early $<$ 20:00<br>20:00 $\leq$ Mid $<$ 21:00<br>Late $\geq$ 21:00 |

**Table 4.1:** Overview of the final data set with variables, variable definitions, measurement, and treatment processes.

After cleaning the data, all categorical variables were one-hot encoded, creating a dummy variable for each level in every categorical variable. As an example, the categories in *SunsetHour* were recoded into new dummies where *SunsetHour.Early*, *SunsetHour.Mid*, and *SunsetHour.Late* would take the value one if the original category corresponded to the dummy, and 0 otherwise. One-hot encoding makes the categorical variables numeric and is necessary as linear regression and the neural network cannot handle categorical data. As one-hot encoding categorical variables involve expanding the number of variables, the 22 variables we were left with after cleaning were expanded to a total of 71 variables.

# 5 Methodology

This chapter is motivated by the research discussed in the literature review. Evidence implied that more advanced machine learning methods, such as tree-based methods and artificial neural networks, have outperformed MLR in other countries for similar prediction tasks. As stated, the assumption of a straight-line relationship between predictors and the output in the MLR could pose problems, as we need to choose some functional form to capture the (potential) nonlinear relationship between property values and property attributes. Roberts & Zhao (2020) and Halvorsen & Pollakowski (1981) all pointed out that few theories offer sufficient guidance on choosing the appropriate functional form. More advanced machine learning methods are not faced with this issue, given their natural ability to handle both linear and nonlinear relationships without user specifications. Before we dive into which methods we have used, important machine learning terminologies are explained to enhance the understanding of the choices made throughout the chapter.

## 5.1 Model Evaluation

### 5.1.1 Bias-Variance Tradeoff

Recalling equation 1.2 in section 1.2, the total prediction error for a given value $x_0$ can be decomposed into the sum of three fundamental quantities; irreducible error, bias, and variance. The irreducible error cannot be reduced no matter how well we estimate our $f(x_0)$. Thus, we are left with bias and variance to reduce the total prediction error.

Ideally, we would develop a model with low variance and low bias to improve predictions and reduce the total test error rate. However, in reality, there is often a tradeoff between the two. In machine learning, variance refers to how much our estimates, $\hat{f}$, would vary across new training samples obtained under similar conditions. If the chosen methodology has high variance, then small changes in the training set could result in large changes in $\hat{f}$.

Bias, on the other hand, refers to the error introduced by approximating a real-life problem, which may be overly complicated, by a much simpler model. For example, MLR assumes a linear relationship between $Y$ and $X1, X2, \ldots, Xp$. As stated in section 1.1, it is unlikely that any real-life problem has such a simple linear relationship, so performing MLR will

undoubtedly result in some bias estimating *f*. As a general rule, introducing more flexible (complex) models will reduce bias and increase variance (James et al., 2017, p. 35).[19]

Figure 5.1 below shows changes in prediction error when increasing model complexity to further illustrate the tradeoff between bias and variance.

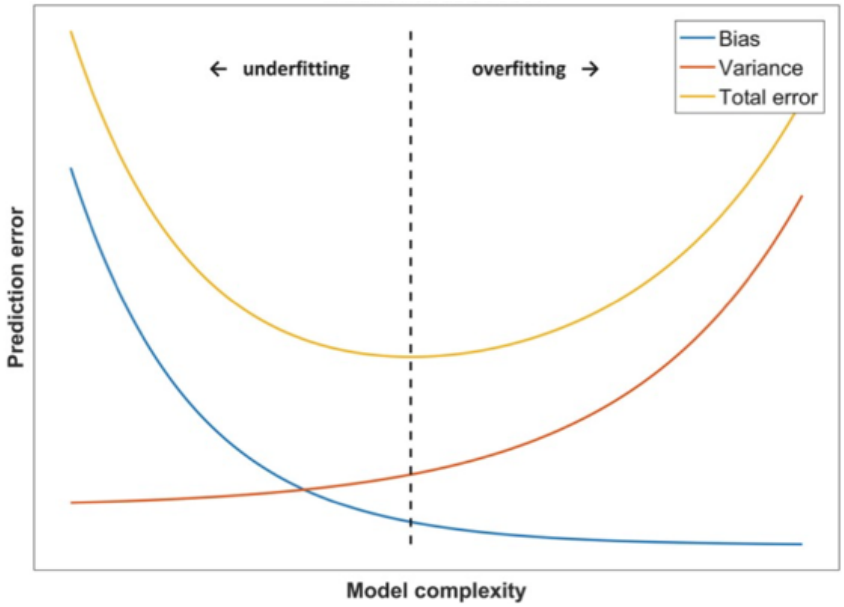


**Figure 5.1:** Total prediction error is minimized when the tradeoff between bias and variance is optimal. More complex models often result in higher variance and reduced variance and vice-versa.

Two central concepts, underfitting and overfitting, are introduced in figure 5.1. Underfitting occurs when the model is overly simplistic, or in other words, when the model cannot capture the underlying patterns in the data set. This situation could lead to inaccurate predictions suffering from high bias. Overfitting often occurs due to the temptation of adding complexity to the model to improve prediction accuracy. More complex models often include a large number of parameters. While this could lead to accurate results for in-sample predictions, as the estimation of the model is specific to the training data, but would yield less accurate out-of-sample predictions (new data), thus increasing variance (Theobald, 2017). This is elaborated upon in subsection 5.1.2.

To estimate a model with an optimal tradeoff between bias and variance, we need to modify the different hyperparameters accordingly. Hyperparameters are often described as parameters whose value is to control the learning process of the model. The modification of hyperparameters (also called tuning parameters) is essential to estimate models with

[19]For mathematical reasoning behind the bias-variance tradeoff, see Hastie et al. (2008, pp. 223-224)

appropriate complexity to find the optimal tradeoff between bias and variance. This modification is usually performed using either *grid search* or *random search*, but it can also be performed manually. The grid search approach can be described as an exhaustive search for finding the optimal model. Here, *every* possible combination of hyperparameters is used to train a model. This is costly both in terms of time and computation power. Random search, on the other hand, selects random combinations of hyperparameters used to train models. Here, the user controls the number of combinations that should be considered. Random search has shown to be nearly as effective as the grid search to find the optimal hyperparameters, while at the same time requiring less computing time (Bergstra & Bengio, 2012). With this in mind, a random search was applied to identify most hyperparameters in the coming methods. Each methodology requires different hyperparameter modification, and this process will be explained thoroughly in the corresponding sections on the various methods.

### 5.1.2 Resampling

A common approach in machine learning is to divide the dataset into separate parts to assess estimations from different models – also called resampling methods. This is usually done by splitting the data into a training-, validation-, and test set. This is important, as the only way we know how well our model will generalize on new data is to test it on unseen data. The model is trained using the training set, the optimal combination of hyperparameters is found using the validation set, and, finally, the model's performance on new data is assessed using the test set.

There are several ways to resample the data. In our thesis, we use k-fold cross-validation. The k-fold cross-validation approach involves randomly splitting the data into $k$ folds of approximately equal size. Here, the first fold is used as the validation set, while the method is fit on the remaining $k - 1$ folds. The mean squared error rate is computed on the observations which are held out. This procedure is repeated $k$ times, each time using a different subset of observations as the validation set. This process results in $k$ different error rates, which are averaged. Models utilizing k-fold cross-validation are fitted on substantially more splits, consequently yielding lower bias. Furthermore, it is shown empirically that setting $k = 5$ or $k = 10$ produces test error rate estimates suffering from neither high variance nor high bias (James et al., 2017, p. 183). An issue with the k-folds

approach is that each fold may not contain a proper representation of the total population for smaller datasets. However, as our datasets includes 82,347 observations, we do not consider this an issue. Consequently, we set $k = 5$ when estimating our models.

Alternatives considered to use over k-fold cross-validation were the validation set approach and leave-one-out cross-validation (LOOCV). The validation set approach is a more straightforward process that involves randomly splitting the data set into two parts, a training set, and a validation set. Here, the model fits the training set, giving a fitted model to predict responses in the validation set. The resulting validation set error provides an estimate of the test error rate. There are mainly two drawbacks to using this approach compared to k-fold cross-validation. Building a model using the validation set approach usually gives highly variable results, depending on which observations are included in the training set and the validation set. Further, only the observations in the training set are used to fit the model. Usually, statistical models tend to perform worse when trained on fewer observations, which indicates that the validation set error rate may overestimate the test error rate for the model fit on the entire data set (James et al., 2017, p. 178).

The LOOCV approach also involves splitting the dataset into two parts, but instead of creating two subsets, only *one* observation is used for the validation set, while the remaining observations are used as the training set. The model fits the $n - 1$ training observations, and a prediction is made for the excluded observation. This procedure is repeated throughout the whole dataset. This approach has some advantages over the validation set approach. First, it has far less bias, as we repeatedly fit the model using training sets that contain $n - 1$ observations. Consequently, the LOOCV approach does not overestimate the test error rate as much as the validation set approach. Second, the validation set approach will yield different results when applied repeatedly due to randomness in the training/validation set splits. The LOOCV, on the other hand, will always yield the same results: there is no randomness in the training/test splits (James et al., 2017, p. 180).

However, applying the LOOCV approach, fitting the model on $n - 1$ training observations, is often computational demanding compared with k-fold cross-validation. Further, since the mean of many highly correlated quantities, which can occur when training the model using almost identical observations, has higher variance than the mean of many

quantitates that are not as highly correlated – the test error rate estimates yielded from the LOOCV tends to have higher variance than the estimates from the k-fold cross-validation. Considering this, the k-fold cross-validation is used to train the model and finding the optimal combination of hyperparameters (James et al., 2017, p. 183).

### 5.1.3 Model Assessment

In the real world, we will rarely or never have the necessary inputs to sufficiently capture all factors responsible for determining the outcome. This is especially true in economic and social data, where the outcome is subject to various factors, some of which we cannot observe at all.[20] Without the necessary input data, the applied method will always be insufficient to capture the correct output for new observations, regardless of its complexity and flexibility. While it would be nice to make perfect out-of-sample predictions, the data's discrepancy forces us to search for the best predictions possible instead. To determine what we see as «best», we seek to minimize our predictions' average deviation to the observed values. We specify this through a mean squared error (MSE) loss function. Minimizing the MSE means that we are trying to approximate the conditional expectation function (CEF), which we define as

$$E(Y \mid X_1, X_2, ..., X_k) \tag{5.1}$$

where $X_k$ is a stochastic observable variable. Unfortunately, due to the limitations in our data, we are unlikely to define the CEF perfectly. The conditional expectation is an unknown nonparametric function and can thus be highly nonlinear. Such a function is difficult to approximate with simple linear regression methods, but Random Forest, gradient boosting, and neural networks are more optimal for approximating it. In theory, we could have also aimed to minimize other loss functions, such as the mean absolute error (MAE), but MSE is the most popular regression loss function (Gupta, 2020), and the CEF is seen as the best predictor for minimizing MSE (Angrist & Pischke, 2008) , making it an ideal loss function for our methods. Because of the convenience and established practice of using MSE, we will refrain from exploring alternative loss functions in this thesis. We do, however, emphasize that MSE is only one of many possible error metrics.

[20] This can or example be the bidding process affecting the final price of a property.

When *assessing* the performance of the different methods, we will use Root Mean Square Error (RMSE) to compare the different predictions validated on the test set. RMSE is useful over MSE as it takes the same unit of measurement as the output variable, making the results from the loss function more intuitive and interpretable. RMSE takes the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2}. \tag{5.2}$$

where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ gives for the $i$th observation. We wish to estimate a model with as small RMSE as possible and find the method with the smallest RMSE for our predictions.

## 5.2 Multiple Linear Regression

MLR is used as the benchmark method since it serves as the current practice for property price prediction in Norway. The MLR includes multiple coefficients or predictors to estimate the dependent variable and takes the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon, \tag{5.3}$$

Where $X_p$ represents the $p$th predictor and $\beta_p$ quantifies the association between that given variable and the dependent variable $Y$. We interpret $\beta_p$ as the average increase on $Y$ associated with a one-unit increase in $X_p$, holding all other predictors fixed (James et al., 2017, p. 72). MLR assumes that the regression function is linear in its parameters or that the particular, specified model is a reasonable approximation. The predictors can come from different sources such as quantitative inputs, dummy variables, and transformations of quantitative inputs (such as log, square-root, or squaring). Using MLR, we pick the coefficients $\beta$, which minimize the sum of squared residuals (Hastie et al., 2008, p. 44).

MLR is applied to our data set and further compared to the results obtained in Statistics Norway's report. Considering that our dataset includes more predictors than Statistics Norway's model, it is essential to uncover *if*, *how*, and *where* our results deviate from their results. This comparison would help us indicate if the more advanced machine learning methods could improve current practices' results.

## 5.3 Tree-Based Methods

In this section, we will describe how tree-based methods are used for prediction purposes. The motivation behind including tree-based methods stems from the promising research presented in the literature review. These methods involve stratifying the predictor space into a number of simple regions using a set of simple splitting rules. To predict a given observation, we typically use the mean of the training observations in the region to which it belongs. Because the set of splitting rules used to segment the predictor space can be summarized in a tree, these methods are known as *decision trees methods*. Simple tree-based methods, such as decision trees, benefit from being easy to interpret, even for non-experts. However, their simplicity makes them less competitive with the best machine learning methods to reduce prediction error. Addressing this issue, ensembles of decision trees, such as Random Forest and gradient boosting, can be employed. The central hypothesis behind ensembling multiple decision trees is to train each decision tree separately to solve the same problem and then combine them into one single prediction, thus improve accuracy through a more robust model than one would obtain training one single decision tree (Rocca, 2019). Ensembling multiple decision trees will often reduce prediction error by providing a more optimal tradeoff between bias and variance at the expense of some loss in interpretation due to the increased complexity (James et al., 2017, p. 303).

### 5.3.1 Decision Trees

Decision trees start with a *root node*, serving as the starting point. This is followed by splits that produce *branches* connected with decision points, called *nodes*. A final prediction, called a *terminal node*, is made when a node does not generate new branches. The step-by-step process behind a decision allows us to visualize each split, providing us with valuable intuition behind each prediction (Theobald, 2017). Further, we expect to see improved results compared to MLR, considering the findings from Fan et al. (2006). They emphasize that decision trees are powerful for analyzing both linear and nonlinear relationships between the input-and output variables - as opposed to MLR, where it can be challenging to choose the appropriate functional form.

A visual representation of a decision tree is given in figure 5.2 below:

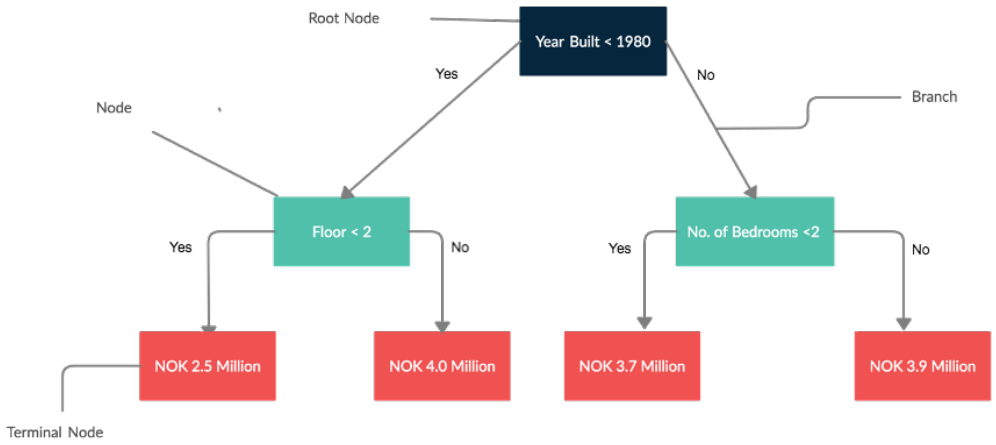


**Figure 5.2:** A visual representation of a decision tree predicting the price of a property. The process starts at the root node and moves left or right depending on the year the unit was built. Based on this splitting rule, the following node considers either floor level, or the number of bedrooms, before making a final prediction defined in the terminal node.

The algorithm behind a decision tree can be outlined through four steps, as described by (James et al., 2017) below:

**Algorithm 5.1:** *Building a decision tree. Reprinted from James et al. (2017), p. 309.*

1. Use recursive splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree to obtain a sequence of best subtrees, as a function of $\alpha$
3. Use K-fold cross-validation to choose $\alpha$. That is, divide the training observations into $K$ folds. For each $k = 1, \ldots, K$ :
   (a) Repeat Step 1 and 2 on all but the $k$th fold of the training data.
   (b) Evaluate the mean squared error on the data in the left-out $k$th fold, as a function of $\alpha$.
   Average the results for each value of $\alpha$, and pick $\alpha$ to minimize the average error.
4. Return the subtree from Step 2 that corresponds to the chosen value of $\alpha$

As described in algorithm 5.1, for each value of $\alpha$ there corresponds a subtree $T \subset T_0$ such that

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{Rm})^2 + \alpha | \, T \, | \tag{5.4}$$

is as small as possible. Here, $| \, T \, |$ indicates the number of terminal nodes of the tree $T$, $R_m$ is the rectangle (i.e., the subset of predictor space) corresponding to the $m$th terminal node, and $\hat{y}_{Rm})^2$ is the predicted response associated with $R_m$ – that is, the mean of the training observations in $R_m$. $\alpha$ is a tuning parameter that controls a tradeoff between the subtree's complexity and its fit to the training data (James et al., 2017, p. 309).

Implementing this algorithm, we found that including $\alpha$ did not improve estimations. Consequently, our $\alpha = 0$. The optimal sequence of subtrees is found to be 15 through k-fold cross-validation.

### 5.3.2 Random Forest

As an ensemble of multiple decision trees, Random Forest is less prone to overfitting compared to a single decision tree. The potential of overfitting is reduced as the model grows a large number of decision trees on bootstrapped training samples.[21] Predictions from Random Forests are based on averaging each tree's predictions in the ensemble, consequently reducing variance. All the trees in the ensemble are grown independently based on the following algorithm:

First, $P$ denotes the set of all possible predictors. Further, a *randomly* chosen subset of predictors is selected from $P$ as node candidates for each split. This subset is used to grow each tree on a bootstrapped sample of the training data. For each of these bootstrapped samples, an unpruned decision tree is grown. After a sufficiently large number of trees are grown, the predictions are averaged over the different trees (James et al., 2017, p. 320).

When estimating a Random Forest model, two hyperparameters need to be defined; 1) the number of trees to be grown (i.e., the number of samples to be selected), and 2) the total number of predictors considered at each node. Oshiro et al. (2012) have studied if there exists an optimal number of trees to be grown in a Random Forest model, i.e., a point

[21] A bootstrap sample is a sample of the same size as the original sample where each object is drawn with replacement from the original sample. For a deeper explanation, see (Hastie et al., 2008, p. 249)

where increasing the number of trees would not further improve performance. They found that growing more than 128 trees did not significantly improve the prediction error.[22] As this could depend on the specific problem, we also found, using the validation set, that the optimal number of trees in our case is 128. Increasing this number did not lead to further improvement in the estimated model's prediction error. Further, the subset of variables is found by $\frac{P}{3}$, giving us a total of 23 variables to be considered at each node, as we have 71 variables in our data set.[23]

There is a logical rationale behind only using 23 variables at each node. This forced split can be taught of *decorrelation* of the trees. Suppose all the predictors were to be considered at each split, and there is one predictor that has a more substantial impact than the other predictors in the dataset. Then, most or all trees would use this strongest predictor in the top split, consequently producing similar trees. Hence, the predictions from similar trees would be highly correlated. By forcing fewer variables to be considered at each node split, other predictors would also be used in the top split, thus reduce variance (James et al., 2017, p. 319).

### 5.3.3 Gradient Boosting

A different approach, which is also an ensemble of decision trees, is *gradient boosting*. This approach also grows multiple trees like Random Forest, except that the trees are grown *sequentially*: meaning that each new tree is grown based on previously grown trees. This is different from Random Forest, where each tree is trained on bootstrapped training samples. Each tree in gradient boosting is fit on a modified version of the original dataset. Considering that gradient boosting utilizes previously grown trees, the construction of each new tree depends heavily on the trees that have already been grown. The rationale behind this is to *learn slowly* and avoid overfitting, which can occur when fitting the data hard by e.g., using only one decision tree. Given the current model in boosting, we fit a decision tree to the current model's residuals. This means that we use the current residuals from the current model, rather than the outcome variable $Y$, as the response. Further, we add this new decision tree into the fitted function to update the residuals. By

[22] The authors compared 29 different datasets with a range of 64, 128, 256, 512, 1024, 2048, and 4096 grown trees. After growing more than 128 trees, the performance did not improve significantly.

[23] 70 when excluding our output variable. $70/3 \approx 23$ variables to be considered at each node.

repeating this process, $\hat{f}$ will slowly improve in areas where it does not perform well, thus reducing bias compared to a decision tree (James et al., 2017, p. 321). Additionally, as with Random Forest, combining multiple trees will reduce variance using the boosting approach.

The estimation of a boosted tree is further described through three different steps, as illustrated by (James et al., 2017) below:

**Algorithm 5.2:** *Boosting for Regression Trees. Reprinted from James et al. (2017), p. 323.*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all $i$ in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with $d$ splits ($d + 1$ terminal node) to the training data $(X, r)$.
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$
3. Output the boosted model,
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

Algorithm 5.2 includes three different hyperparameters that should be tuned. The first being the number of trees $B$ to be grown. Here, one should note that using too many trees could lead to overfitting. We found the optimal number of trees to be 2,516 by k-fold cross-validation. The second hyperparameter is $\lambda$, the shrinkage parameter. This controls the rate at which the model learns, allowing more and different shaped trees to attack the residuals from previously grown trees. We found our optimal $\lambda$ to be 0.1634 applying random search. The third and final hyperparameter is the number of $d$ splits in each tree, which controls the boosted ensemble's complexity. Often, $d = 1$ works well, implying that each tree consists of a single split. More generally, $d$ is the interaction depth, which controls the boosted model's interaction order, as d splits can involve at most $d$ splits (James et al., 2017, p. 323). We found our optimal number of splits to be 9 using random search.

## 5.4 Neural Networks

Neural networks are highly flexible models capable of estimating nonlinear relationships between the dependent and independent variables. They contain a property known as a universal approximation, meaning that in theory, given any continuous function, an artificial network is capable of approximating that function to an arbitrary degree of precision (Barron, 1993).[24] In short, this means that a neural network can learn any problem. This flexibility is possible because the network is not subject to restrictions in its parameters as opposed to, e.g., MLR, and this flexibility can make neural networks useful for predicting house prices.

Neural networks have a unique architecture compared to the tree-based methods. The networks themselves are built up of layers of nodes, in which nodes are the information-processing units operating the network (Haykin, 2009, p. 10).[25] Haykin illustrates the model of a node in figure 5.3.

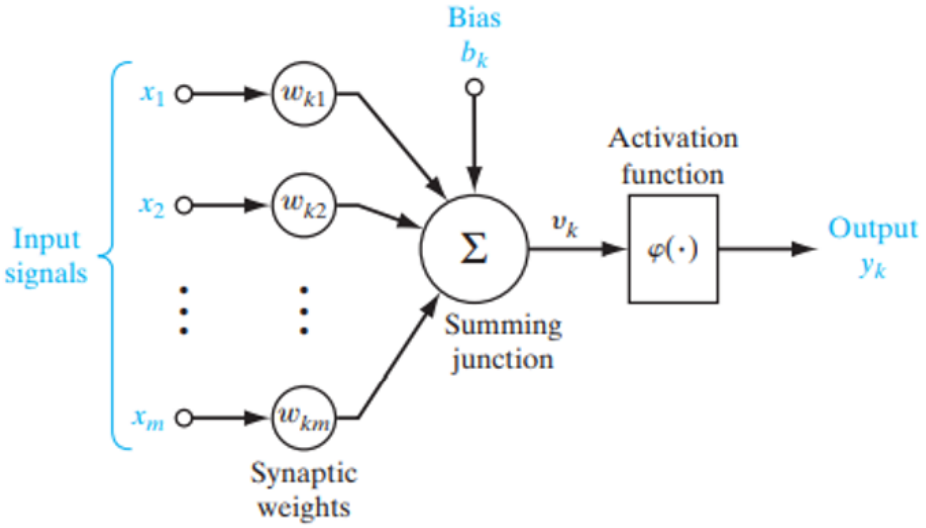


**Figure 5.3:** Haykin (2009) illustrates how a single node is built and how the input signals from the previous layer are transformed to become inputs for the next layer.

He describes three essential elements composing each node:

1. A set of links called *synapses* connecting the inputs to each node. These synapses are characterized by individual weights. The weights are multiplied with each signal

[24] For proof of the universal approximation theorem, see (Barron, 1993)
[25] Nodes are also called artificial neurons or perceptrons. In this thesis, they are referred to as nodes.

from the input layer, linking to the node. The network determines the weights themselves through backpropagation.[26]

2. A summing junction, responsible for summing the product of all inputs and their corresponding weights linearly. An external bias is applied to the summing junction, acting as a constant capability to lower or increase the input to the activation function, depending on whether the bias is positive or negative. This is done to help the network fit the given data better.

3. An *activation function* responsible for adding nonlinearity to the network. If we exclude the activation function, the network is essentially just linear regression. Ye (2019) complements this by explaining that "neural networks need nonlinearity to address nonlinear problems, and usually the only place where that nonlinearity can be injected is in the activation function." There are several types of activation functions, each suitable for different kinds of problems.

Keeping these three elements in mind, the mathematical formula for a node $k$ is

$$y_k = \varphi\left(\sum_{i=1}^{n} w_{ki}x_i + b_k\right) \tag{5.5}$$

where $\varphi$ is the activation function, $w_{ki}$ are the weights assigned to the inputs $x_i$, and $b_k$ is the externally applied bias. $y$ is the node's output and can either be the input of a node in the next hidden layer, or if it is the last node in the network, it will be the final output of the model.

[26] An optimization algorithm retroactively utilizes information from the loss function (MSE) to determine how to adjusts the synaptic weights to minimize the loss function further.

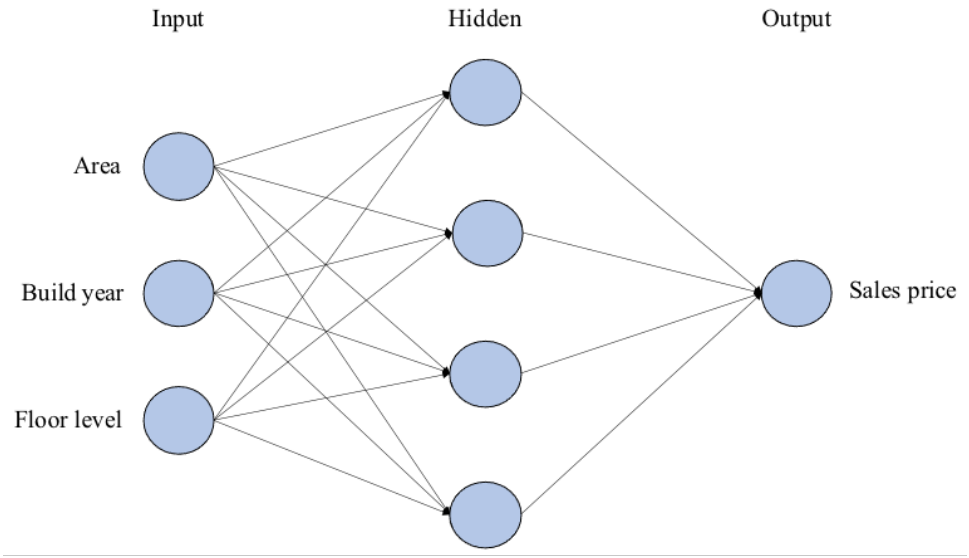


**Figure 5.4:** A simplified model of a neural network with three input nodes, one hidden layer of four nodes, and one output node.

Figure 5.4 displays a model of the architecture of a single-layer neural network. The independent variables make up the input layer, where each predictor is defined as its separate input node. These nodes have a different architecture than the nodes explained in equation 5.5 above. They have no associated activation function, and naturally, they are not weighted sums of prior nodes. The input nodes are passed on to the hidden layer's nodes, whose values are determined based on equation 5.5. The resulting activation from the nodes in the hidden layer is further passed forward to the output layer, which outputs the predicted value for the target variable. The node in the output layer does not require an activation function since we are predicting a continuous value; in our case, the log of *TargetPrice* (Chollet & Allaire, 2018). The intuition is that because the output layer is purely linear, the network is free to learn to predict values in any range, and we do not need to constrain the output.

We performed feature scaling on the network's input predictors to make the predictors more similar in their distribution. This is useful when having predictors that operate on different scales, such as *Age* and *PreviousValue*, in our case. A way of rescaling called min-max feature scaling consists of assigning the lowest value of all variables to 0 and the highest value to 1. Everything in between will take on a value between 0 and 1, proportional to their original value. Intuitively, this does not affect the relationship between the dependent and independent variables. An observation $i$ in variable $X$ in

the $90^{th}$ percentile before rescaling will remain in the $90^{th}$ percentile after rescaling. The general equation for min-max rescaling is shown below

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.6}$$

where $X$ is the original value, and $X'$ is the rescaled value. We can consider *Age*, which varies from 0 to 220, and *TargetPriceCommondebt*, ranging from 0 to 6 million. Rescaling them to the same interval of [0,1] helps the optimizer in the network converge faster during gradient descent, effectively reducing computation time (Levy, 2016).[27] After the predictions are finished, we revert the resulting predictions to their original range.

For the constant parameters in the network, we implemented the ReLU activation function in the hidden layer. It is defined mathematically as

$$y = max(0, x) \tag{5.7}$$

where $x$ and $y$ are the activation function's input and output, respectively. ReLU is suitable for regression problems and works by only activating a few nodes at a time, creating sparsity in the model. This sparsity, combined with the simple math of ReLU, makes the network efficient and able to run faster (Liu, 2017).

Neural networks often contain a vast number of parameters. Each synapse connecting every node accounts for one parameter. In addition to this, each bias term accounts for a parameter. Given a larger number of nodes in the network, the network's parameters become so high that we can essentially interpret the model as nonparametric. Lee et al. (2018) proved that when the width of a neural network approaches infinity, the network resembles a nonparametric model.[28] A disadvantage of this property is that having many parameters requires large amounts of input data to function sensibly. To ensure proper generalization, Haykin (2009) recommends a traditional rule of thumb, where the number of training observations is at least ten times the number of free parameters in the network.

[27] This statement is purposely simplified, as to not stray too far away from the ideas presented. More details on the power of normalization on gradient descent can be found in (Levy, 2016).

[28] They specifically refer to gaussian process regression, which finds the probability distribution over all possible functions $f(x)$ that fit the data. Gaussian processes will not be elaborated upon in this thesis, but for more information on gaussian process regression, see chapter 2 in Rasmussen & Williams (2006)

Depending on the number of predictors, hidden layers, and nodes, this number can quickly grow very large.

Another disadvantage of neural networks is that optimizing the model's hyperparameters can be challenging. The hyperparameter tuning is often done manually as automated cross-validation tuning, such as grid search, is computationally expensive. A drawback of manual tuning of the network is that the method is not very scientific, and it is unknown whether we have achieved the optimal hyperparameters. However, manual tuning is straightforward, and we can understand our decisions, unlike automated tuning, where decisions can be less intuitive.

The neural network has several hyperparameters which we can individually tune to improve performance. Through trial and error, we considered the following hyperparameters in table 5.1.

| **Hyperparameter** | **Description** | **Tuning result** |
|---|---|---|
| Hidden layers | Determines the depth of the network | 1 |
| Hidden nodes | The number of nodes present in the hidden layer<br>Composes the width of the network | 51 |
| Epochs | The number of times the data is passed forward and backward through the network | 100 |
| Optimization | Determines how the synaptic weights in the network are adjusted each epoch | Adam algorithm |
| Regularization | Constraints applied to prevent overfitting | No regularization is applied |

**Table 5.1:** The hyperparameters that were considered when tuning the neural network.

We constructed a neural network with one hidden layer, consisting of 51 nodes in the hidden layer. Increasing the number of hidden layers or nodes did not improve the accuracy of the network. The network was trained over 100 epochs, from which increasing the number of epochs did not improve the model's prediction error in 20 iterations. The Adam algorithm uses the output from the loss function to adjust the weights in the model. The

goal is to, through iterations (epochs), minimize the loss function. It is based on gradient descent and calculates which way to alter the weights so that the loss function can reach a global minimum.[29] Adam has become the most popular optimizer algorithm due to its performance and effectiveness compared to other algorithms (Bock et al., 2018; Ruder, 2016).

There are, in general, three ways of utilizing regularization to prevent overfitting. We can specify a dropout percentage, which tells the model to temporarily ignore a given number of nodes along with its incoming and outgoing connections during training (Srivastava et al., 2014, p. 1930). They found that training a neural network with dropout can significantly lower out-of-sample error than training with other regularization methods, such as L1 or L2.[30] However, after testing regularization variations, we found that the out-of-sample error was the lowest without any regularization.

## 5.5 Interpretation of Machine Learning Models

Understanding why a model makes a particular prediction can often be as crucial as its accuracy in many applications. However, due to the complexity of the more advanced machine learning models, the reasoning behind the achieved results is often challenging to explain, even for experts. This creates a trade-off between accuracy and interpretability, as introduced in section 1.3. In response, different methods have recently been proposed to assist users in the interpretation of predictions of complex models (Lundberg & Lee, 2017).

To interpret the contribution that each predictor has on the outcome variable, *TargetPrice*, we will make benefit of Local Interpretable Model-agnostic Explanations (LIME), introduced by Ribeiro et al. (2016). As described in section 1.3, LIME is used to explain individual predictions of black-box machine learning models.[31] We use a local interpretation because taxpayers subject to property tax should receive an explanation of *their* specific prediction – as opposed to global interpretations where model decisions are explained for the whole population.

[29]For an in-depth explanation of advantages and how the Adam algorithm works, see the introductory paper from the inventors of the algorithm Kingma & Ba (2017).

[30]An explanation of how and why dropout, L1, and L2 works can be found in (Oppermann, 2020).

[31]For a technical in-depth explanation of LIME, see (Molnar, 2020)

Behind the workings of LIME is the assumption that all complex models are linear on a local scale (Ferrando, 2018). Nguyen (2020) explains the intuition behind LIME with that for every input we pass to the model, we can perform a local sensitivity analysis to understand how each predictor affects the predicted output for this particular instance. LIME has been shown to handle both classification and regression tasks and work well with machine learning techniques such as tree-based methods and neural networks, making it applicable for our purpose. In practice, LIME modifies our specific data sample by slightly tweaking the feature values and collecting each feature change's resulting impact on the predicted output (Nguyen, 2020).

The intuition behind local surrogate models can be explained mathematically in the following equation:

$$explanation(x) = arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{5.8}$$

where the explanation model for instance $x$ is the model that minimizes the loss $L$ (MSE), measuring how close the explanation is to the prediction made from the original model $f$ (advanced model in our case), while keeping the model complexity $\Omega(g)$ low (e.g. prefer fewer features to be included). $G$ is the family of possible explanations, for example all possible linear regression models. The proximity measure $\pi_x$ defines how large the neighbourhood around instance $x$ is that we consider for the explanation. The user is responsible for determining the complexity of the surrogate, through e.g. specifying the maximum number of features in a multiple linear regression model (Molnar, 2020).

To further describe the workings behind LIME, an explanation of the generalized algorithm is provided below (Boehmke & Greenwell, 2020):

1. Given an observation, ***permute*** it to create replicated feature data with slight value modifications.
2. Compute ***similarity distance measure*** between the original observation and the permuted observation.[32]
3. Apply the selected machine learning model to ***predict outcomes*** of the permuted data.
4. ***Select m number of features*** to describe the ***predicted*** outcomes best.

[32]Euclidian distance

5. ***Fit a simple model*** to the permuted data, which explains the complex model outcome with $m$ features from the permuted data weighted by its similarity to the original observation.[33]
6. Use the resulting ***feature weights the explain local behavior***.

LIME is useful in situations where predictions must be explained more thoroughly, such as when predicted property values serve as the calculation basis for property tax. Although LIME will not be a fully complete attribution, it will go a long way even in scenarios where we might be legally required to explain individual predictions (Forecast, 2020). LIME will only be applied to the model yielding the lowest prediction error to exemplify how it can be implemented in a real-life situation.

We considered SHapley Additive exPlanations (SHAP) as an alternative surrogate model to LIME. SHAP derives from utilizing both game theory and local explanations to interpret a measure of variable importance from a machine learning model.[34] Compared to LIME, SHAP is substantially more computationally expensive. This is primarily because LIME perturbs data around an individual prediction, while SHAP computes all permutations globally and then finds the local accuracy (Molnar, 2020). Thus, LIME was preferred due to computational speed and ease of use.

[33]We fitted MLR on our permuted data.
[34]For in-depth research and explanation on SHAP, see Lundberg & Lee (2017)

# 6 Empirical Results

This chapter shows the results obtained after estimating the different models discussed in the methodology section. These are provided with their respective RMSE to assess prediction accuracy on the test set and compare results. We estimated a total of five different models. Section 6.1 presents the empirical results from MLR, serving as the benchmark, while the empirical results from the competing methods, covering the tree-based methods and neural network, are presented in section 6.2.

## 6.1 Benchmark Results

MLR serves as the benchmark, as it is comparable to the model used by Statistics Norway. We include more variables, but the methodological principles are the same. While the functional form of the variables may vary, both models are linear in their parameters. Therefore, results obtained from our model will be used as a benchmark for further comparisons. The estimated MLR returned an RMSE of 0.2132. As stated in subsection 5.1.3, the primary purpose of RMSE in our thesis is to compare the models, where we favor lower RMSE, indicating higher model accuracy.

Figure 6.1 below visualizes the percentage distribution of the errors between the predicted and observed values:

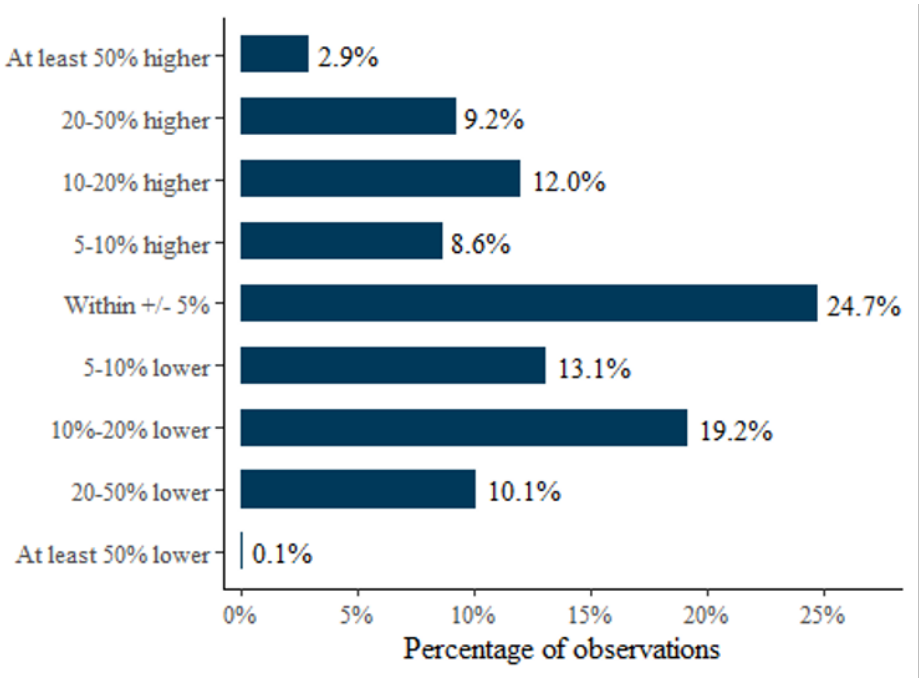


**Figure 6.1:** Percentage distribution of errors between the predicted market value and the observed value using in the period 2005-2020 for apartments in Oslo.

Figure 6.1 shows that 77.6% of predictions are within an interval of +/-20% of the observed values. The results are almost identical to Statistics Norway's model, where 78% of predictions are within +/-20%. Considering this similarity, we consider it reasonable to use the results from MLR as a benchmark for our other methods, and this should be comparable if applied by Statistics Norway. Additionally, compared to the observed values, 42.5% of predictions are underestimated by more than 5%, while a total of 32.7% of predictions are overestimated by more than 5%. This demonstrates that predictions from MLR show an overweight of underestimations.

## 6.2 Contending Methodologies

The decision tree will serve as a highly interpretable tree-based model, capable of replacing MLR without violating Statistics Norway's preference of having a simple and interpretable prediction model. We know that Random Forest and gradient boosting are not as interpretable as a single decision tree, but their potential to outperform the single tree makes us willing to sacrifice some interpretability for the sake of improved accuracy. The neural network can also be hard to interpret, but its flexibility gives it the potential to outperform the other methods.

### 6.2.1 Decision Tree Results

The optimal decision tree contained 15 subtrees and yielded an RMSE of 0.3275, thus producing less accurate results than MLR. The percentage distribution of the errors between the predicted and the observed values are visualized below in figure 6.2. The full decision tree can be found in Appendix A4.

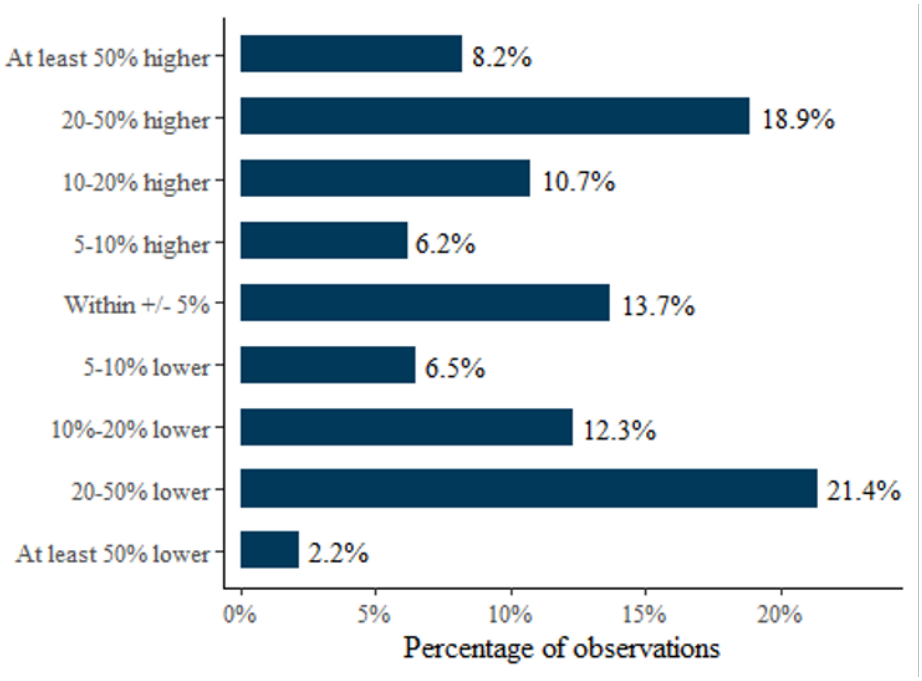


**Figure 6.2:** Percentage distribution of errors between the predicted market value and the observed value in the period 2005-2020 for apartments in Oslo.

Figure 6.2 shows that a total of predictions within +/- 5% of the observed values are only set to be 13.7% in total. The decision tree predicted 39.4% of observations within +/-20% of the observed values. Further, 42.4% of predictions are underestimated by more than 5%, while a total of 44% of predictions are overestimated by more than 5%. This implies a low prediction consistency, with a substantial number of both under-and overestimations.

### 6.2.2 Random Forest Results

The Random Forest model was estimated with 128 trees, and 23 variables considered for each subtree. This returned an RMSE of 0.1272, offering a substantial improvement compared with MLR. This improvement is prominent in the distribution of errors, visualized in figure 6.3.

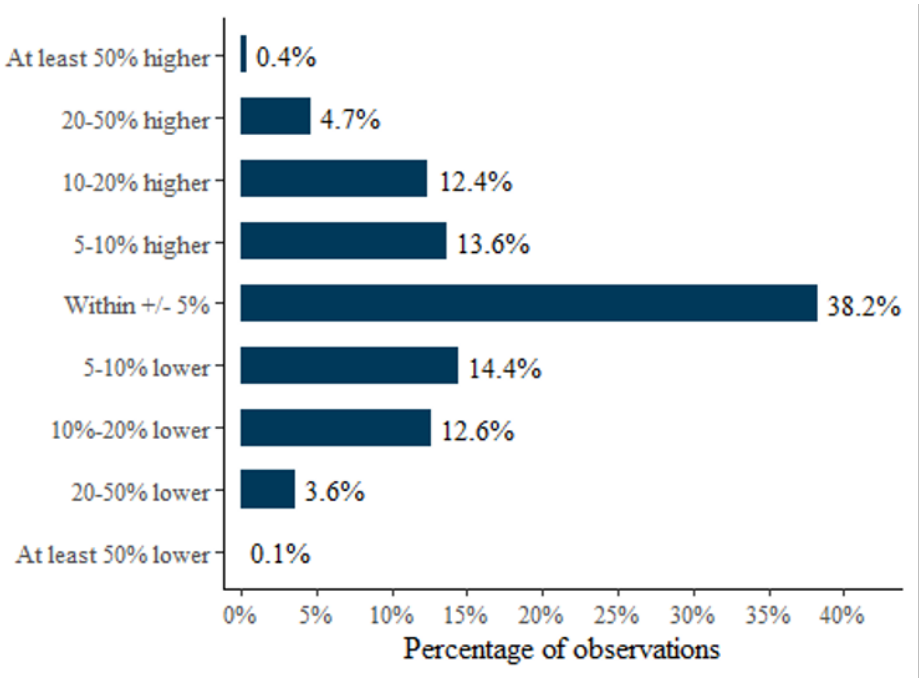


**Figure 6.3:** Percentage distribution of errors between the predicted market value and the observed value in the period 2005-2020 for apartments in Oslo.

Figure 6.3 shows that 38.2% of predictions fall within +/-5% of the observed values. A total of 90.3% of predictions are within the +/-20% interval, offering a substantial improvement compared to the MLR. Compared to the observed values, 30.7% of predictions are underestimated by more than 5%, while a total of 31.1% of predictions are overestimated by more than 5%. As these percentages are relatively similar, we see no indications of systematic over-or underestimation by the model.

### 6.2.3 Gradient Boosting Results

The boosted model was estimated with 2,516 trees, a shrinkage parameter of 0.1634, and an optimal number of 9 splits. This returned an RMSE of 0.1140, which is substantially lower than the RMSE of MLR. The percentage distribution of errors is displayed in figure 6.4.

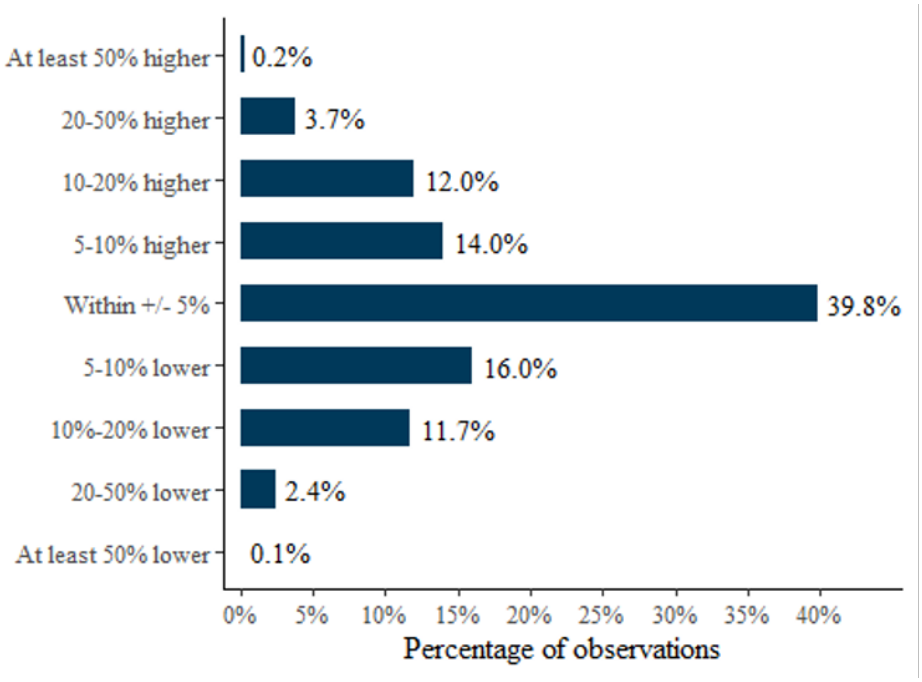


**Figure 6.4:** Percentage distribution of errors between the predicted market value and the observed value in the period 2005-2020 for apartments in Oslo.

Figure 6.4 shows that 39.8% of predictions fall within +/-5% of the observed values. The boosting model predicted 93.5% of observations within +/-20% of the observed values, showing a considerable improvement to MLR. Compared to the observed values, 30.2% of predictions are underestimated by more than 5%, while a total of 29.2% of predictions are overestimated by more than 5%. These percentages are also relatively similar, indicating no systematic over-or underestimation.

### 6.2.4 Neural Network Results

The neural network with a hidden layer of 51 nodes returned an RMSE of 0.1318 and substantially outperformed MLR. The percentage distribution of errors is displayed in figure 6.5.

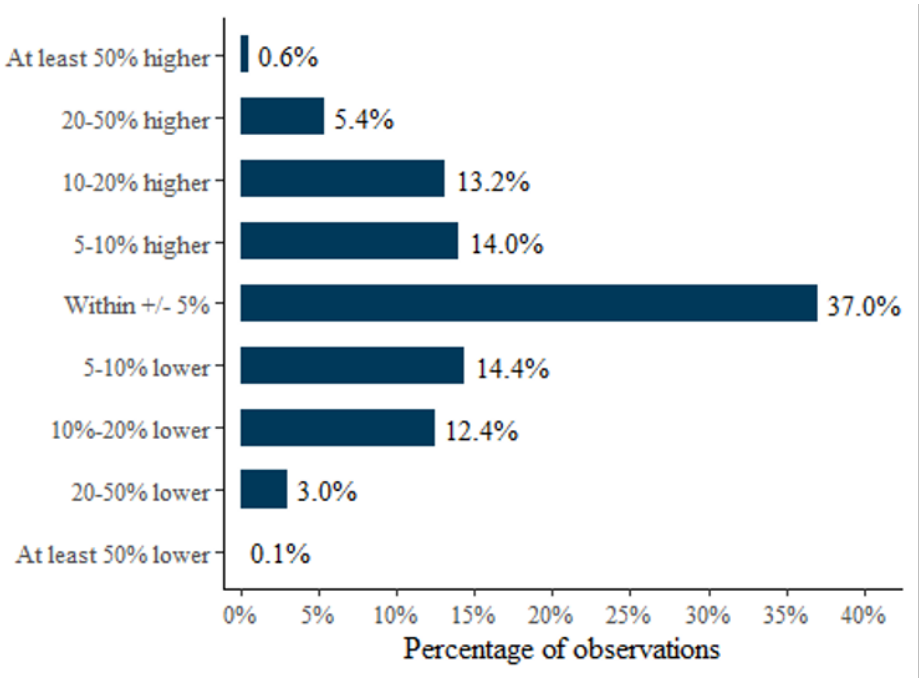


**Figure 6.5:** Percentage distribution of errors between the predicted market value and the observed value in the period 2005-2020 for apartments in Oslo.

Figure 6.5 shows that 37% of predictions fall within +/-5% of the observed values. We can see that the network predicted 91% of observations within +/-20% of the observed values, showing a considerable improvement to MLR. Compared to the observed values, 29.9% of predictions are underestimated by more than 5%, while a total of 33.6% of predictions are overestimated by more than 5%. These percentages are also relatively similar, indicating that the network has a slight tendency to overestimate its predictions.

# 7 Discussion

This chapter is divided into three subsections. In subsection 7.1, we discuss the empirical results from chapter 6 and compare them with the previously presented literature from chapter 2. Subsection 7.2 addresses the societal impact of our results. This includes exploring consequential effects on calculations of property tax, as well as implications posed by the GDPR regarding the transparency and interpretability of the applied prediction model. Finally, in section 7.3, we will consider our work's overall limitations and offer suggestions on how our results can be used to advance further research.

## 7.1 Discussion of Empirical Results

We initially suspected that MLR, Statistics Norway's current practice for predicting property market values, could be improved. This suspicion was established as their model shows that as many as 25% of the estimated values are outside the interval of +/-20% of the observed values. As a refresher, the results from each method along with their RMSE from chapter 6 are provided in table 7.1 below:

| **Method** | **RMSE** | **Percentage of predictions within +/-20% of observed values** | **Percentage of predictions within +/-5% of observed values** |
|---|---|---|---|
| Multiple Linear Regression (MLR) | 0.2132 | 77.6% | 24.7% |
| Decision tree | 0.3275 | 49.4% | 13.7% |
| Random Forest | 0.1272 | 90.3% | 38.2% |
| Gradient boosting | **0.1140** | **93.5%** | **39.8%** |
| Neural network | 0.1318 | 91% | 37% |

**Table 7.1:** Comparison of results from every applied method. Gradient boosting yields the best results, both in terms of RMSE and percentage of predictions closer to observed values.

Table 7.1 shows that gradient boosting is the superior method both in terms of RMSE and distribution of errors. Based on paper written by Kagie & van Wezel (2007) , we did expect the boosting approach to yield improved results compared to MLR, as they found boosting improved accuracy by over 40% compared to MLR in the Dutch housing market. Further, we believe that our results support the strengths of the boosting approach emphasized by Gu & Xu (2017). They suggest that gradient boosting is useful when predicting a continuous variable, based on the input of many potentially interacting categorical and numerical variables. They also argue that a gradient boosting model reduces bias by learning from previously grown trees. Thus, its ability to transform weak learners into strong learners contributes substantially to reducing the overall error. These arguments make the boosting approach great for our purpose, confirmed by our results. Still, the boosting model is not perfect, in part because of the limitations posed by the methodology, such as its potential for being sensitive to outliers due to its commitment to fixing errors from previously grown trees.

Random Forest and the neural network yielded slightly inferior results to gradient boosting, both regarding RMSE and distribution of errors. Still, both methodologies considerably outperformed MLR, and are, for this reason, strong potential candidates for improving the current practice. We believe that the improvement from Random Forest is due to its flexibility in dealing with continuous variables, emphasized by Hong et al. (2020), and further, its ability to reduce variance given the large number of ensembled trees. The neural network did seem to fit the data well, and while it was outperformed by gradient boosting and Random Forest, its performance is still superior to MLR. As presented in section 5.4, the network should theoretically have fit the data perfectly, supported by the universal approximation theorem (Barron, 1993), but is likely constrained by the available data.

The decision tree was the only method that did not outperform MLR, both when looking at RMSE and its distribution of errors. The disappointing result from the decision tree can be explained by its stated difficulties predicting a continuous variable and James et al. (2017) asserting that decision trees are inherently non-robust, implying that a small change in the data might cause massive changes in the final predictions.

## 7.2 Societal Impact of Results

### 7.2.1 Impact on Tax Calculations

Inaccurate predictions lead to imprecise tax calculations, and considering this, one would strive to get these predictions as precise as possible. Underestimations lead to lower calculation bases, which in turn reduces the total property tax paid. On the contrary, overestimations lead to higher calculation bases, potentially making individuals pay more property tax than they ought to. The reduction of both under- and overestimations will, in turn, result in fairer property tax calculations.

As described in section 3.2, the property tax is calculated based on the property's estimated market value after accounting for the mandatory reduction factor of 30%. This implies that properties overvalued by as many as 42.9% will not be subject to more tax than if the property value was estimated correctly and not factoring in the reduction factor.[35] From the government's perspective, this may justify why their model is still viable, even though it is not very accurate. In reality, the reduction factor may merely move the benchmark for what the property tax basis should be, rather than the basis being the actual market price. Whether this is fair is difficult to answer, but the reduction factor may simply shift the opinion of what is considered fair. For example, consider a property being overestimated by 40% compared with a similar property that is not overestimated. One can argue that this is unfair, even though the property owner is not subject to more tax than they would have been without the reduction factor, with a *correctly* estimated market value.

To confine such unfair estimations, taxpayers can amend their estimated property value. In subsection 3.2.1, we discovered that taxpayers are permitted to amend their tax return if their property's value is overestimated by more than 20% for primary dwellings and 11.1% for secondary dwellings. This amendment needs to be submitted within six weeks of receiving the property tax return. Amendments are further subject to control by local tax authorities, and individuals can be inquired to document their amendment through an appraisal performed by a professional agency. As amendments must be submitted within six weeks of receiving the tax return, this would potentially result in many cases for the

[35] Calculation is: $\frac{1-0.7}{0.7} * 100 = 42.9\%$

Norwegian Tax Administration to consider within a short period of time.

To illustrate the discrepancy in under- and overestimations further, a comparison between MLR and gradient boosting is provided in table 7.2 below. Considering that taxpayers can amend their estimated property value if overestimated by more than 20% for primary dwellings and 11.1% for secondary dwellings, the respective number of overestimations above these thresholds are included. Additionally, the total number of underestimations above 5% of the observed property value is included. This is to provide an overview of the percentage of underestimations leading to a notable reduction in collected property tax.

| | **MLR** | **Gradient boosting** | **Difference** |
|---|---|---|---|
| **Overestimations above 20% of the observed value** | 12.1% | 3.9% | 68% |
| **Overestimations above 11.1% of the observed value** | 22.3% | 13.7% | 39% |
| **Underestimations above 5% of observed value** | 42.5% | 30.2% | 29% |

**Table 7.2:** Over-and underestimations from MLR and gradient boosting. Difference between them is calculated by $(\frac{MLR - Gradient\ Boosting}{MLR})$

Table 7.2 shows that gradient boosting leads to a remarkable reduction of 68% in overestimations above 20%, which is relevant when considering potential amendments for primary dwellings. For secondary dwellings, the threshold of 11.1% is relevant when considering potential amendments, and gradient boosting reduces the percentage of overestimations by 39% compared to MLR. Recalling that our dataset only includes three districts in Oslo, we infer that implementing the boosting approach will substantially reduce the number of amendments the Norwegian Tax Administration would have to process nationally.

Additionally, table 7.2 shows that the boosting approach reduces underestimations above 5% by approximately 29% compared to MLR. This discrepancy suggests the possibility of reducing the number of people paying an insufficient level of property tax, leading to a fairer tax system, which is desirable for taxpayers and municipalities alike.

### 7.2.2 Legal Implications and Interpretability

As introduced in section 1.3, there is a tradeoff between accuracy and interpretability when estimating machine learning models. Statistics Norway state that they prefer MLR because of its simplicity and high interpretability. They further clarify that they are willing to sacrifice some accuracy to maintain the interpretability of the model. In spite of this, there are methods yielding superior results to MLR, as discovered from our results in chapter 6. From the methods we tested, gradient boosting returned the lowest RMSE while at the same time generating the most predictions within +/-5% of their observed values. However, gradient boosting's black-box nature implies that it offers less interpretability than MLR, which is essential to consider when adopting such a model.

Section 3.3 introduced the notion of a subject's right to question why a particular automated decision was made. The GDPR has no article explicitly stating a subject's right to explanation, but article 13(2)(f) describes a right to "meaningful information about the logic involved" in an automated decision (European Union, 2016b). Goodman & Flaxman (2016) are confident that article 13-15 *does* define an explicit right to explanation, and they find it reasonable to expect that the minimum requirement for an adequate explanation would be to provide an account of how the input variables relate to the predicted outcome.

In contrast to the ideas of Goodman & Flaxman, authors Wachter et al. (2017) argue that there does not currently exist such a right to explanation in the GDPR. They claim that the lack of precise language in the GDPR limits the regulation's impact on the topic of interpretability. The way the law is worded today, one could implement any machine learning method for automated decisions, regardless of its interpretability. In EU law, we can find recitals whose purpose is to set out the reasons for an act's provisions (Klimas & Vaiciukaite, 2008), and while the recitals are not legally binding themselves, they can be crucial in interpreting an ambiguous provision (Thomson Reuters, n.d.). Recital 71 of the GDPR states that the subject should have the right to obtain an explanation of the decision reached through an automated process (European Union, 2016a). With this recital in mind, Selbst & Powles (2017) agree with Goodman & Flaxman (2016) that there does exist a right to explanation. They suggest that the minimum threshold for the interpretability of the decision is that any subject, expert or not, should understand

if they are being discriminated against through the automatic process.[36] We do not find it entirely clear what Selbst & Powles (2017) mean by this specifically, but the idea indicates that the subject may not be entitled to a *full* explanation of a black-box model. Interpreting this from a property price perspective, one would expect some measure of which way each predictor affects the outcome variable.

By employing LIME, as described in section 5.5, gradient boosting should be compliant with current legislation. Recalling for every input we pass to the model, we can perform a local sensitivity analysis to understand how each predictor affects the predicted output for this particular instance, we can utilize a local interpretation model to explain the logic behind each individual prediction.

The algorithm in section 5.5 describes that we fit a simple model on permuted data, explaining the more complex model's outcome from the permuted data weighted by its similarity to the original observation.[37] The resulting feature weights are then used to explain local behavior. To illustrate how this works in practice, we will look at the results from the boosting approach, as these were superior compared to the other methods. We wish to emphasize that LIME is also applicable for the other methods discussed in chapter 5, and will produce similar outputs as in figure 7.1. Figure 7.1 demonstrates LIME's use with the five most influential variables to interpret a given prediction from the boosting approach. Here, we keep in mind that all variables but *CoastDirection.NE* are log-transformed. The illustration of an arbitrary prediction is shown below:

[36]Here, discrimination refers to Act 21(1) of the EU Charter of Fundamental Rights.
[37]Simple model is a standard MLR for this purpose.

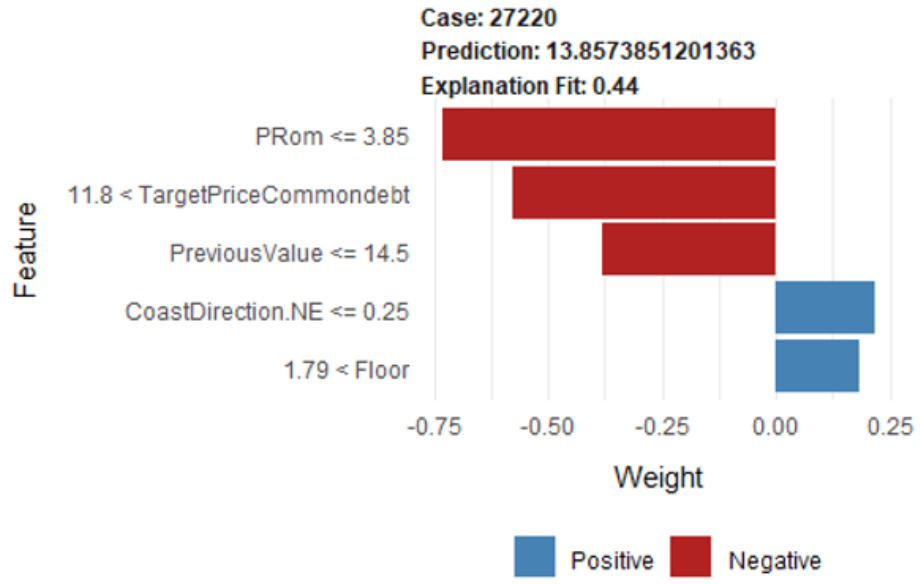


**Figure 7.1:** Local behavior for case number 27,220. Features are the top five most influential predictors for the outcome variable *TargetPrice*. Weight describes the influence of each predictor on the prediction outcome for this isolated case.

For this arbitrary case, *PRom*, *TargetPriceCommondebt*, and *PreviousValue* negatively affect the prediction, while *CoastDirection* and *Floor* positively affect the prediction.[38] The prediction of *TargetPrice* from the gradient boosting approach is 13.86 for this specific observation. The explanation fit in figure 7.1 refers to the $R^2$ of the simple model fitted locally to explain the variance in the neighborhood of the examined case and is set to 0.44.[39] It is important to remember that using LIME involves fitting a simpler model on top of the more complex model. Thus, we assume that the approximation of the simple model is faithful to the boosted model's behavior in the vicinity of the case we examine.

The fitting of the simpler model on the permuted data gives the following intercept and weights:

[38] Converted back to their original scale, 3.85 in *PRom* equals $45m^2$, 11.8 in *TargetPriceCommondebt* equals NOK 133,250, 14.5 in *PreviousValue* equals NOK 1,982,757, and 1.79 in *Floor* equals 4.

[39] $R^2$ represents the proportion of variance for the dependent variable *(TargetPrice)* that is explained by the independent variables.

| Coefficient | Weight |
|---|---|
| *Intercept* | 15.11 |
| *PRom* | -0.73 |
| *TargetPriceCommondebt* | -0.58 |
| *PreviousValue* | -0.38 |
| *CoastDirection.NE* | 0.22 |
| *Floor* | 0.18 |

**Table 7.3:** Coefficients with respective weights obtained for case 27,220. Summing the weights gives a local prediction of 13.82.

By summing the weights of the coefficients in table 7.3, we find that LIME's *local prediction* is 13.82. The deviation between the local prediction and the prediction obtained from gradient boosting is relatively small, and thus we can infer that the local model can be used as a reliable surrogate for the gradient boosting model.

Although we suggest that LIME can be used as a reliable surrogate for gradient boosting, it is ultimately up to legislators, data protection authorities, and courts to decide whether these interpretations adequately comply with the GDPR (Selbst & Powles, 2017). As the GDPR is relatively new, the issue has yet to come up for assessment, but it could disrupt the use of black-box models for automated decision making. Nevertheless, as long as laws are not specified to exclude the application of black-box methods, the current knowledge in the field of interpretable machine learning should be sufficient to comply with current legislation. The minimum requirements imposed by Goodman & Flaxman (2016) and Selbst & Powles (2017) is that you should be provided an account of the relationship between the output and the predictors and be able to determine whether you are subject to discrimination. With these requirements in mind, gradient boosting, our superior methodology, does seem to be compliant with the GDPR and could thus be used for property valuation to estimate property tax. However, it would be beneficial to assess the risks and clarify the laws further with legislators before implementing the new prediction model to avoid potential litigation.

## 7.3 Limitations of Thesis

Even though we consider the restrictions on interpretability imposed by the GDPR as a significant threat to the implementation of advanced machine learning methods, there are undoubtedly other threats present. The GDPR also restricts the recording, use, and storage of personal data. Hence, it is necessary to avoid data that could be considered personal to avoid legal complications. Our models' predictors do not contain personal information, but if we were to include variables such as property coordinates or other demographic data such as ethnicity, gender, and political views of the property owner, this would have to be addressed. However, as long as such data is held out from the estimated model, this issue should not pose a problem.

Limitations in our data can cause shortcomings for our analysis. The nature of economic and social data involves that as we will rarely or never have the necessary inputs required to capture all factors determining the outcome perfectly, the models will never produce perfect predictions. Further, as our data only covers central districts in Oslo, we cannot know how well the models will perform on data from other regions in Norway. Additionally, the models are trained solely on apartments, while Statistics Norway's current valuation method is applicable for detached houses, row houses, and apartments. Consequently, we may require even more robust measures to validate the trained models before implementing them and ensure that they can be used for all properties in Norway. If we compare the results from figure 6.1 on MLR, we can see that 77.6% of predicted apartment prices are within +/-20% of the observed prices. In their report, Statistics Norway presents that 78% of Oslo apartments are predicted within +/-20% of their observed prices. These values are relatively comparable, and keeping this in mind, we could argue that given a different dataset covering *all* property types, we would obtain similar results to Statistics Norway's model looking at all property types. Before concluding, this would naturally require testing and implementation of the more advanced models to ensure they can be used for all residential properties in Norway.

Another shortcoming derives from the use of LIME. Researchers suggest that by repeating the sampling process, the explanations that come out can be different. Unstable explanations mean that it may be hard to trust the results, and one should be critical when applying LIME (Molnar, 2020). However, as discussed in subsection 7.2.2, the

subject may not be entitled to a *full* explanation of a black-box model, but rather some measure of which way each predictor affects the outcome variable. Considering this, we argue that LIME is applicable for this purpose. It is worth mentioning that LIME is an innovative tool that is still in development, but the promising results combined with its applicability over different machine learning methods make it attractive for further exploration.

The large number of predictors included in our analysis could also pose implementation problems for real-world applications. Statistics Norway prefers few predictors in the valuation model to maintain simplicity and user-friendliness, and a large number of predictors may not be compatible with this motive. This also assumes that Statistics Norway is able to gain access to the required predictors. All predictors used in the models should, in theory, be attainable through an address search, but may involve collecting entries from multiple external databases, such as Eiendomsverdi, Finn, and the Norwegian Mapping Authority.

### 7.3.1 Suggestions for Further Research

As our thesis focuses on how advanced machine learning models can improve property value predictions and how these impact estimations of property tax, we recognize two distinct topics for further investigation. The first revolves around the more advanced machine learning methods' technical aspects, while the second considers the possibility of adjusting the reduction factor when calculating property tax.

Considering the technical aspects, we are proposing further research on the interpretability of the models using SHAP. As stated in section 5.5, we prefer LIME over SHAP due to less computational cost and ease of use. However, there are advantages to using SHAP to explain black-box predictions. One advantage is that SHAP provides us with both global and local interpretations, as opposed to LIME, where only local interpretations are provided. It could be valuable to examine how model decisions are explained for the *whole* population, made possible through global interpretations. Further, we would consider using SHAP for our thesis if the computational costs were less of an issue. Public bodies like Statistics Norway should have sufficient computer capacity making SHAP a more attractive alternative.

Another technical aspect relates to the tuning of the model's hyperparameters. As discussed in subsection 5.1.1, hyperparameter tuning is performed to optimize the tradeoff between bias and variance, reducing the total test error rate. In the case of our neural network, we tuned its hyperparameters manually through trial and error, meaning that we may have been unable to discover the optimal combination of hyperparameters. By employing a complete grid search, it will be possible to optimize the selection of hyperparameters further, potentially leading to better performance.

Based on our results, the government may seek to revise the mandatory reduction factor of 30%. The reason behind the reduction factor is to reduce the probability of overestimations, and our results indicate that this probability is further reduced using machine learning. As gradient boosting proved to be more accurate than MLR, one can argue that lowering the reduction factor is possible without increasing today's probability of overestimations. This would lead to increased tax revenues for municipalities, as the calculation basis for the property tax would be higher.

# 8 Conclusion

Initially, we defined our research's overall goal, which is to uncover whether machine learning methods can improve property value predictions, thus leading to more accurate property tax estimations in Norway. This concluding chapter will summarize the most important findings from our research and offer conclusions and recommendations based on this. Additionally, we will clarify our research's contribution to the current knowledge of property valuation in the context of property tax.

The literature review identified and discussed several machine learning methods applied for house price prediction in other countries. They all compared methods to MLR, serving as the most common approach. Their research motivated the choices made in our methodology section, resulting in four different methods to challenge the MLR. The results obtained from these methods show that all methodologies, except the decision tree, outperforms MLR sufficiently. Gradient boosting yields superior results, both in terms of lower RMSE and limiting both over-and underestimations of predictions. The RMSE from gradient boosting was 0.1140, compared to the RMSE of MLR, which was 0.2132. With the application of gradient boosting, the total number of overestimations above 20% and 11.1% is reduced by 68% and 39%, respectively, while the total of underestimations above 5% is reduced by 29% compared to MLR — all highly desirable outcomes. While the superior results from gradient boosting come at the compromise of interpretability, we were able to portray a degree of local interpretability through LIME, addressing the issue of low interpretability.

The main conclusion we draw from our research is that there are superior machine learning methods for property valuation and that these methods improve current property tax calculations in Norway. Our findings suggest that implementing gradient boosting as the new valuation method will result in a fairer tax system — for both taxpayers and the municipalities. Fewer taxpayers will be subject to a higher tax base than they should, and fewer taxpayers will be subject to a lower tax base than they should. Consequently, the number of yearly corrections of tax returns should be reduced, which is beneficial for both taxpayers and municipalities. Based on these findings, we recommend exploring gradient boosting as the new standard property valuation method in Norway for this purpose.

The literature review and the provided background chapter made it clear that applying machine learning in predicting property values has not been explored by Statistics Norway before. The empirical research in our thesis contributes to knowledge on this topic and serves as a framework for exploring new ways of predicting property values to further improve property tax calculations. Our findings and discussions help educate and challenge debates on the topic of machine learning by giving a more rounded and informed research picture. The results are very promising, and we hope our research can improve the current practice of property value predictions in Norway.

# References

Angrist, J. D., & Pischke, J.-S. (2008). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.

Baijayanta, R. (2019). All about missing data handling. *Towards Data Science*. Retrieved October 20, 2020, from https://towardsdatascience.com/all-about-missing-data-handling-b94b8b5d2184.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, *39*(3), 930–945. https://doi.org/10.1109/18.256500.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281–305. https://dl.acm.org/doi/10.5555/.

Bloomquist, G., & Worley, L. (1981). Hedonic prices, demands for urban housing amenities, and benefit estimates. *Journal of Urban Economics*, *9*(2), 212–221. https://doi.org/10.1016/0094-1190(81)90041-3.

Bock, S., Goppold, J., & Weiß, M. (2018). *An improvement of the convergence proof of the ADAM-Optimizer* [Conference paper]. OTH-Clusterkonferenz, Weiden, Germany. arXiv. Retrieved November 12, 2020, from https://arxiv.org/pdf/1804.10587.pdf.

Boehmke, B., & Greenwell, B. (2020). *Hands-On Machine Learning With R*. Retrieved December 2, 2020, from https://bradleyboehmke.github.io/HOML/.

BPI Consulting (2016). Are Skewness and Kurtosis Useful Statistics?. Retrieved October 1, 2020, from https://www.spcforexcel.com/Downloads/pdf/Are-Skewness-and-Kurtosis-Useful-Statistics.pdf.

Carlson, R. H. (2018). *A Brief History of Property Tax*. [Conference paper]. IAAO Conference on Assessment Administration, Boston, MA, United States. Retrieved November 4, 2020, from https://www.iaao.org/uploads/a_brief_history_of_property_tax.pdf,.

Chollet, F., & Allaire, J. J. (2018). *Deep Learning with R*. Manning Publications

Co. Retrieved from https://livebook.manning.com/book/deep-learning-with-r/about-this-book/.

Eiendomsverdi (n.d.). *About the Eiendomsverdi AVM*. Retrieved October 12, 2020, from https://eiendomsverdi.no/.

Eigedomsskattelova (1975). *Lov om eigedomsskatt til kommunane (LOV-1975-06-06-29)*. Retrieved November 3, 2020, from https://lovdata.no/dokument/NL/lov/1975-06-06-29?q=eigedomsskattelova.

European Union (2012). Charter of Fundamental Rights of the European Union, 2012/C 326/02. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:12012P/TXT.

European Union (2016a). GDPR Recitals. Retrieved from https://gdpr-info.eu/recitals/.

European Union (2016b). REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Retireved from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

Fan, G.-Z., Ong, S. E., & Koh, H. C. (2006). Determinants of house price: A decision tree approach. *Urban Studies*, *43*(12), 2301–2315. https://doi.org/10.1080/00420980600990928.

Ferrando, P. (2018). Understanding how lime explains predictions. *Towards Data Science*. Retrieved December 3, 2020, from https://towardsdatascience.com/understanding-how-lime-explains-predictions-d404e5d1829c.

Forecast (2020). Understanding the black box: Lime. Retrieved December 6, 2020, from https://forecast.global/insight/understanding-the-black-box-lime/.

Fortuner, B. (2017). Can neural networks solve any problem? *Towards Data Science*. Retrieved November 20, 2020, from https://towardsdatascience.com/can-neural-networks-really-learn-any-function-65e106617fc6.

Geurts, P., Irrthum, A., & Wehenkel, L. (2009). Supervised learning with decision tree-

based methods in computational and systems biology. *Molecular BioSystems*, *5*(12), 1593–1605. https://doi.org/10.1039/b907946g.

Goodman, A. C. (1978). Hedonic prices, price indices and housing markets. *Journal of Urban Economics*, *5*(4), 471–484. https://doi.org/10.1016/0094-1190(78)90004-9.

Goodman, B., & Flaxman, S. (2016). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, *38*(3), 50–57. https://doi.org/10.1609/aimag.v38i3.2741.

Groenwold, R. H. H., White, I. R., Donders, A. R. T., Carpenter, J. R., Altman, D. G., & Moons, K. G. M. (2012). Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. *Canadian Medical Association Journal*, *184*(11), 1265–1269. https://doi.org/10.1503/cmaj.110977.

Gu, G., & Xu, B. (2017). Housing market hedonic price study based on boosting regression trees. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *21*(6), 1040–1047. https://doi.org/10.20965/jaciii.2017.p1040.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Pedreschi, D., & Giannotti, F. (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, *51*(5). https://doi.org/10.1145/3236009.

Gupta, S. (2020). Most common loss functions in machine learning. *Towards Data Science*. Retrieved November 26 from https://towardsdatascience.com/most-common-loss-functions-in-machine-learning-c7212a99dae0.

Halvorsen, R., & Pollakowski, H. O. (1981). Choice of functional form for hedonic price equations. *Journal of Urban Economics*, *10*, 37–49. https://doi.org/10.1016/0094-1190(81)90021-8.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning* (2nd ed.). Springer.

Haykin, S. (2009). *Neural networks and Learning Machines* (3rd ed.). Pearson Education. Retrieved from http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf.

Hong, J., Choi, H., & Kim, W. (2020). A house price valuation based on the random

forest approach: the mass appraisal of residential property in south korea. *International Journal of Strategic Property Management*, *24*(3), 140–152. https://doi.org/10.3846/ijspm.2020.11544.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning with Applications in R*. Springer. Retrieved from https://doi.org/10.1007/978-1-4614-7138-7.

Kagie, M., & van Wezel, M. (2007). Hedonic price models and indices based on boosting applied to the dutch housing market. *Intelligent Systems in Accounting, Finance and Management.*, *15*(3-4), 85–106. https://doi.org/10.1002/isaf.287.

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* [Conference paper]. ICLR 2015, San Diego, CA, United States. arXiv. https://arxiv.org/pdf/1412.6980v9.pdf.

Klimas, T., & Vaiciukaite, J. (2008). The law of recitals in european community legislation. *ILSA Journal of International Comparative Law*, *15*, 62–93. https://ssrn.com/abstract=1159604.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., & Sohl-Dickstein, J. (2018). *Deep Neural Networks as Gaussian Processes* [Conference paper]. ICLR 2018, Vancouver, Canada. arXiv. Retrieved 16 December, 2020, from https://arxiv.org/pdf/1711.00165.pdf.

Levy, K. Y. (2016). *The Power of Normalization: Faster Evasion of Saddle Points*. arXiv. https://arxiv.org/abs/1611.04831.

Li, A. H., & Bradic, J. (2018). Boosting in the presence of outliers: Adaptive classification with nonconvex loss functions. *Journal of the American Statistical Association*, *113*(522), 660–674. https://doi.org/10.1080/01621459.2016.1273116.

Limsombunchai, V., Gan, C., & Lee, M. (2004). House price prediction: Hedonic price model vs. artificial neural network. *American Journal of Applied Sciences*, *1*(3), 193–201. https://doi.org/10.3844/ajassp.2004.193.201.

Liu, D. (2017). A Practical Guide to ReLU. *Medium*. Retrieved November 25, 2020, from https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7.

Lundberg, M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. arXiv. https://arxiv.org/pdf/1705.07874.pdf.

Molnar, C. (2020). *A Guide for Making Black Box Models Explainable*. Retrieved from https://christophm.github.io/interpretable-ml-book/.

Nguyen, D. (2020). Explain Your ML Model Predictions With Local Interpretable Model-Agnostic Explanations (LIME). *Medium*. Retrieved December 3, 2020, from https://bit.ly/3mytX8O.

Oppermann, A. (2020). Regularization in deep learning — l1, l2, and dropout. *Towards Data Science*. Retrived 29 November, 2020, from https://towardsdatascience.com/regularization-in-deep-learning-l1-l2-and-dropout-377e75acc036.

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). *How Many Trees in a Random Forest?*. Lecture Notes in Computer Science, 154–168. https://doi.org/10.1007/978-3-642-31537-4_13.

Oslo Kommune (n.d.). *Hvor mye skal du betale i eiendomsskatt?*. Retrieved October 21, 2020, from https://www.oslo.kommune.no/skatt-og-naring/skatt-og-avgift/eiendomsskatt/hvor-mye-skal-du-betale-i-eiendomsskatt/.

Pai, P.-F., & Wang, W.-C. (2020). Using machine learning models and actual transaction data for predicting real estate prices. *Applied Sciences*, *10*(1), 5832. https://doi.org/10.3390/app10175832.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *Model-Agnostic Interpretability of Machine Learning*. arXiv. https://arxiv.org/pdf/1606.05386.pdf.

Roberts, E., & Zhao, L. (2020). *A Bayesian Mixture Model for Changepoint Estimation Using Ordinal Predictors*. arXiv. https://arxiv.org/ftp/arxiv/papers/2008/2008.00300.pdf.

Rocca, J. (2019). Ensemble methods: bagging, boosting and stacking. *Towards data science.*. Retrieved December 13, 2020, from https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

Rosengard, J. K. (2013). The Tax Everyone Loves to Hate: Principles of Property Tax Reform. In McCluskey, W. J., Cornia, G. C. Walters, L. C. (Ed.), *A Primer on Property Tax.* (pp. 173-185). New Jersey, NY: Blackwell Publishing Ltd.

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. arXiv. https://arxiv.org/pdf/1609.04747.pdf.

Selbst, A. D., & Powles, J. (2017). Meaningful information and the right to explanation. *International Data Privacy Law*, *7*(4), 233–242. https://doi.org/10.1093/idpl/ipx022.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. https://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf.

Statistics Norway (2020a). *Modell for beregning av boligformue.* (Report 9/2020). Retrieved September 17, 2020, from https://www.ssb.no/priser-og-prisindekser/artikler-og-publikasjoner/_attachment/415132?_ts=170cda32ca0.

Statistics Norway (2020b). *Property tax*. Retrieved December 8, 2020, from https://www.ssb.no/en/offentlig-sektor/statistikker/eiendomsskatt.

The Norwegian Tax Administration (2020a). *How much tax must I pay?*. Retrieved October 12, 2020, from https://www.skatteetaten.no/en/person/taxes/get-the-taxes-right/property-and-belongings/houses-property-and-plots-of-land/property-tax/how-much-tax-must-i-pay/.

The Norwegian Tax Administration (2020b). *What is property tax?*. Retrieved October 8, 2020, from https://www.skatteetaten.no/en/person/taxes/get-the-taxes-right/property-and-belongings/houses-property-and-plots-of-land/property-tax/what-is-property-tax/.

The Norwegian Tax Administration (n.d.a). *Changing the basis for property tax calculation*. Retrieved October 19, 2020, from https://www.skatteetaten.no/en/person/taxes/get-the-taxes-right/property-and-belongings/houses-property-and-plots-of-land/property-tax/changing-the-basis-for-the-property-tax-calculation/.

The Norwegian Tax Administration (n.d.b). *Tax value of housing*. Retrieved October 28, 2020, from https://www.skatteetaten.no/en/rates/tax-value-of-housing/.

Theobald, O. (2017). *Machine Learning for Absolute Beginners* (2nd ed.). Scatterplot Press.

Thomson Reuters (n.d.). Recital (EU). Retrieved November 2, 2020, from https://uk.practicallaw.thomsonreuters.com/w-009-6368?transitionType=Default&contextData=(sc.Default)&firstPage=true.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law.*. https://doi.org/10.2139/ssrn.2903469.

Ye, A. (2019). Finally, an intuitive explanation of why ReLU works. *Towards Data Science.*. Retrieved November 24, 2020, from https://towardsdatascience.com/if-rectified-linear-units-are-linear-how-do-they-add-nonlinearity-40247d3e4792.

# Appendix

## A1 Statistics Norway: percentage distribution estimated/observed prices. Apartments only.

| Estimated/ Observed | Quantity | Percentage | Cumulative percentage |
|---|---|---|---|
| **0-40** | 93 | 0.02 | 0.02 |
| **41-60** | 2,786 | 0.60 | 0.62 |
| **61-80** | 31,985 | 6.87 | 7.49 |
| **81-100** | 175,441 | 37.70 | 45.19 |
| **101-120** | 189,055 | 40.63 | 85.82 |
| **121-140** | 51,966 | 11.17 | 96.98 |
| **141-160** | 9,633 | 2.07 | 99.05 |
| **161-180** | 2,446 | 0.53 | 99.58 |
| **181-200** | 883 | 0.19 | 99.77 |
| **200+** | 1,071 | 0.23 | 100.00 |

**Table A1.1:** Statistics Norway: percentage distribution estimated/observed prices. Apartments only.

# A2 Initial dataset

| **Numeric variables** | | |
|---|---|---|
| **Variable name** | **Description** | **Total number of missing values** |
| *UnitID* | An identifier for each apartment building. The same ID is applied for all apartments in the same building. | 0 |
| *TargetPrice* | The sales price of the apartment. | 0 |
| *TargetPrice Commondebt* | Joint debt attached to the apartment. | 23,293 |
| *PRom* | The primary living area in m2. | 264 |
| *BRA* | Sum of primary and secondary living area in m2. | 7,437 |
| *BTA* | Area of the whole unit, including outer walls, in m2. | 31,229 |
| *BuildYear* | Initial year the unit is built. | 142 |
| *Floor* | The floor level of the unit. | 3,045 |
| *NumberOfFloors* | Total number of floors in the building. | 4,478 |
| *NumberOf Bedrooms* | Number of bedrooms in the unit. | 43,472 |
| *SiteArea* | Area of lot. | 715 |
| *SiteArea Undeveloped* | Undeveloped area of lot. | 3,972 |
| *SiteAreaShared* | Area of lot shared with other units. | 37,598 |

| | | |
|---|---|---|
| *CoastDistance* | Distance to nearest coast in meters. | 98 |
| *Altitude* | Meters above sea level. | 98 |
| *SiteSlope* | Slope decline of lot. | 98 |
| *PreviousValue* | Previous sales price. | 53,415 |
| *PreviousValue Commondebt* | Joint debt attached at previous sale. | 64,727 |
| *PreviousPrice ValueCategory* | Indicator variable, indicating whether an apartment has been previously sold in the time period of the dataset or not. | 53,415 |
| *Latitude* | Latitude of unit's location. | 0 |
| *Longitude* | Longitude of unit's location. | 0 |
| **Categorial variables** | | |
| **Variable name** | **Definition and category** | **Total number of missing values** |
| *CityDistrict* | District in Oslo<br>- Grünerløkka<br>- Frogner<br>- Gamle Oslo | 0 |
| *TargetPrice MarketSaleDate* | Unit's date of sale, measured in DD.MM.YYYY<br>Binned into year of sale:<br>2005 – 2020. | 0 |
| *EstateType* | Unit category<br>- Detached house<br>- Row house<br>- Apartment<br>- Semi-detached house | 0 |

| | | |
|---|---|---|
| *OwnershipType* | Whether the apartment is freehold, stock, or part of a housing cooperative<br>- Freehold<br>- Cooperative<br>- Stock | 0 |
| *SiteOwnership Type* | Whether the lot is freehold or leasehold<br>- Freehold<br>- Leasehold | 0 |
| *Balcony* | Dummy variable for whether the apartment has a balcony attached or not<br>- True<br>- False | 0 |
| *Elevator* | Dummy variable for whether the unit has an elevator in the building or not<br>- True<br>- False | 0 |
| *CoastDirection* | Direction to the nearest coast. Binned from 360 degrees into the intercardinal directions<br>- NE<br>- SE<br>- SW<br>- NW | 99 |

| | | |
|---|---|---|
| *SunsetHour* | Time of day the<br>sun sets at the unit.<br>Binned from HH:MM<br>into categories:<br>- Early <20:00<br>- 20:00 Mid <21:00<br>- Late 21:00 | 98 |
| *SiteSlopeDirection* | Direction the slope<br>of the lot is declining.<br>Binned from 360 degrees<br>into the intercardinal directions<br>- NE<br>- SE<br>- SW<br>- NW | 98 |
| *PreviousValueDate* | Previous date sold,<br>measured in DD.MM.YYYY.<br>Binned into year of sale:<br>2000 – 2020. | 53,415 |

**Table A2.1:** Initial dataset.

# A3 Missing indicator method

To explain the missing indicator method we refer to the work of Groenwold et al. (2012) who discusses use of the method in clinical research:

> The missing-indicator method does not impute missing values. Instead, missing observations are set to a fixed value (usually zero, but other numbers will give the same results), and an extra indicator or dummy (1/0) variable is added to the multivariable model to indicate whether the value for that variable is missing (p. 1266).

If we consider the following multivariate model for *TargetPrice* of a dwelling $i$:

$$TargetPrice_i = \beta_0 + \beta_1 PRom_i + \beta_2 PreviousValue_i + \beta_3 Indicator \tag{.1}$$

where $\beta_0$ is a constant, $PRom_i$ is the living area of a dwelling, $PreviousValue_i$ is the previous sales price of the dwelling. In cases without missing values, the indicator is zero, and the model fitted to the data is:

$$TargetPrice_i = \beta_0 + \beta_1 PRom_i + \beta_2 PreviousValue_i \tag{.2}$$

For cases where a dwelling has not been previously sold, the indicator takes on the value 1, and the model fitted to the data is:

$$TargetPrice_i = \beta_0 + \beta_1 PRom_i + \beta_3 Indicator \tag{.3}$$

# A4 Visualization of Decision Tree

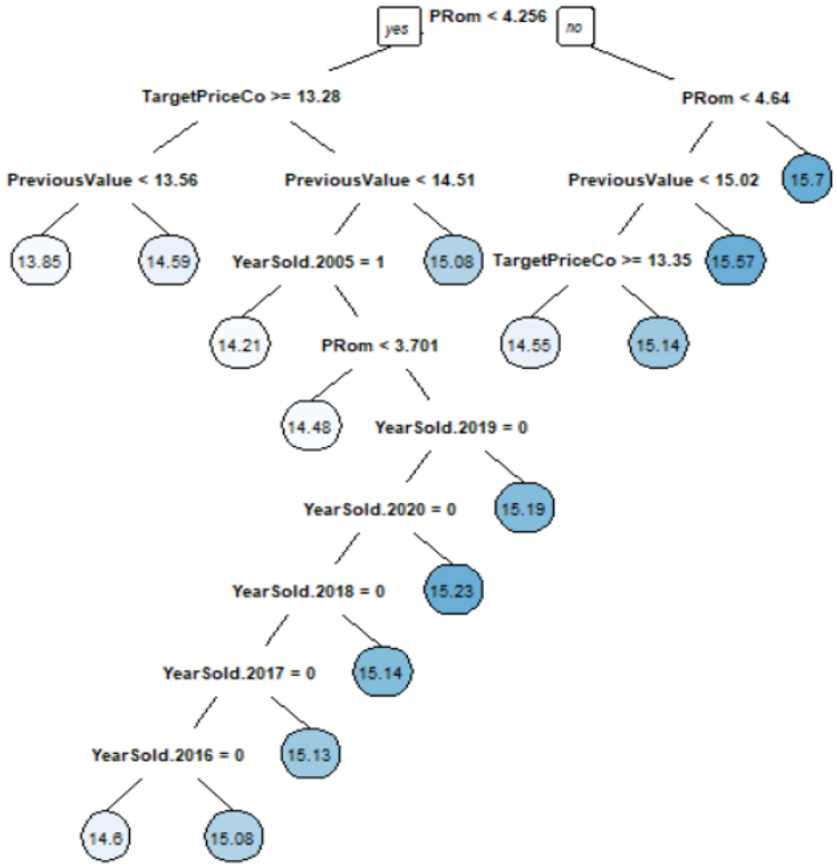


**Figure A4.1:** A reprint of the full decision tree along with the splitting rules and results in each terminal node.