Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in Industrial Engineering and Management

Property valuation with interpretable machine learning

Kaapro Hartikainen

Master's Thesis
2023

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Kaapro Hartikainen

Title of thesis Property valuation with interpretable machine learning

Programme Industrial Engineering and Management

Major Organisation Design and Leadership

Thesis supervisor Prof. Lauri Saarinen

Thesis advisor(s) PhD. Ruth Kaila

Collaborative partner Nordea

Date 24.02.2023 **Number of pages** 69 **Language** English

Abstract

Property valuation is an important task for various stakeholders, including banks, local authorities, property developers, and brokers. As a result of the characteristics of the real estate market, such as the infrequency of trades, limited supply, negotiated prices, and small submarkets with unique traits, there is no clear market value for properties. Traditionally property valuations are done by expert appraisers. Property valuation can also be done accurately with machine learning methods, but the lack of interpretability with accurate machine learning methods can limit the adoption of those methods. Interpretable machine learning methods could be a solution to this issue, but there are concerns related to the accuracy of these methods.

This thesis aims to evaluate the feasibility of interpretable machine learning methods in property valuation by comparing a promising interpretable method to a more complex machine learning method that has had good results in property valuation previously. The promising interpretable method and the well-performed machine learning method are chosen based on previous literature.

The two chosen methods, Extreme Gradient Boosting (XGB) and Explainable Boosting Machine (EBM) are compared in terms of prediction accuracy of properties in six big municipalities of Denmark. In addition to the accuracy comparison, the interpretability of the EBM is highlighted.

The accuracy of the XGB method is better, even though there are no big differences between the two methods in individual municipalities. The interpretability of the EBM is good, as it is possible to understand, how the model makes predictions in general, and how individual predictions are made.

Keywords Property valuation, machine learning, interpretable machine learning, XGBoost, XGB, Explainable Boosting machine, EBM

Aalto-yliopisto
Aalto-universitetet
Aalto University

Tekijä Kaapro Hartikainen

Työn nimi Kiinteistöjen arviointi ymmärrettävällä koneoppimisella

Koulutusohjelma Tuotantotalous

Pääaine Organisaatioiden suunnittelu ja johtaminen

Vastuuopettaja/valvoja Prof. Lauri Saarinen

Työn ohjaaja(t) TkT Ruth Kaila

Yhteistyötaho Nordea

Päivämäärä 24.02.2023 **Sivumäärä** 69 **Kieli** Englanti

Tiivistelmä

Kiinteistöjen arviointi on tärkeä tehtävä eri sidosryhmien, kuten pankkien, kuntien, kiinteistökehittäjien ja välittäjien kannalta. Kiinteistömarkkinoiden ominaisuudet, kuten harvoin tapahtuvat kaupat, rajoitettu tarjonta, neuvotellut hinnat ja paikalliset erot, vaikuttavat siihen, että kiinteistöillä ei ole selkeää markkina-arvoa. Perinteisesti kiinteistöjen arvioinnin tekevät asiantuntijat. Kiinteistöjen arviointi voidaan tehdä tarkasti myös koneoppimismenetelmillä, mutta tulkittavuuden puute tarkoilla koneoppimismenetelmillä voi rajoittaa näiden menetelmien käyttöönottoa. Tulkittavat koneoppimismenetelmät voisivat olla ratkaisu tähän ongelmaan, mutta näiden menetelmien tarkkuus ei välttämättä ole tarvittavalla tasolla.

Tämän työn tavoitteena on arvioida tulkittavien koneoppimismenetelmien toteutettavuutta kiinteistöjen arvioinnissa vertaamalla lupaavaa tulkittavissa olevaa menetelmää monimutkaisempaan koneoppimismenetelmään, jolla on aiemmin saatu hyviä tuloksia kiinteistöjen arvioinnissa. Lupaava tulkittava menetelmä ja hyvin tuloksia saanut koneoppimismenetelmä valitaan aikaisemman kirjallisuuden perusteella.

Valittuja menetelmiä, Extreme Gradient Boosting (XGB) ja Explainable Boosting Machine (EBM), verrataan kiinteistöjen ennustetarkkuuden suhteen kuudessa Tanskan suuressa kunnassa. Tarkkuusvertailun lisäksi EBM tulkittavuutta esitellään.

XGB-menetelmän tarkkuus on parempi kokonaisuudessaan, vaikkakin erot yksittäisissä kunnissa ovat pieniä. EBM tulkittavuus on hyvä, ja on mahdollista ymmärtää, miten malli tekee ennusteita yleisesti ja miten yksittäisiä ennusteita tehdään.

Avainsanat Kiinteistöjen arviointi, koneoppiminen, tulkittava koneoppiminen, XGBoost, XGB, Explainable Boosting Machine, EBM

Contents

Preface

I want to thank my supervisor Lauri Saarinen and my advisor Ruth Kaila for their guidance and support during this project.

I also want to thank Kristian Schaadt and other people at Nordea, who helped me throughout this project.

I also want to thank my partner for keeping me sane and alive.

Helsinki, 23 March 2023
Kaapro Hartikainen

Abbreviations

CNN	Convolutional neural network
DFSA	Danish Financial Supervisory Authority
DKK	Danish krone
EBM	Explainable Boosting Machine
EI	Expected Improvement
ETRS89	European Terrestrial Reference system
GAM	General additive model
GA2M	General additive model plus interactions
GDPR	General Data Protection Regulation of the European Union
KNN	K-nearest-neighbours
LIME	Local Interpretable Model-Agnostic Explanations
MARS	Multivariate adaptive regression splines
MDAPE	Median absolute prediction error
MLP	Multilayer Perceptron
MSE	Mean squared error
PDP	Partial Dependence Plot
RBF	Radial Basis Function
RMSE	Root mean squared error
SHAP	SHapley Additive exPlanations
SMBO	Sequential Model-Based Global Optimization
STAR	Structured additive regression
SVM	Support vector machine
TPE	Tree-structured Parzen Estimator
XGB	Extreme gradient boosting

1 Introduction

Property valuation is important for various stakeholders. Banks use it for collateral loans and mortgage release, local authorities use it for taxation, property developers use it for investment purposes, and brokers use it for transactions (Lee, 2022; Su et al., 2021). However, the real estate market has characteristics that make the market price unclear (Hilbers et al., 2001). Each property is unique, trades are infrequent, supply is limited, transaction costs are high, prices are negotiated and change over time, and the real estate market consists of smaller submarkets with unique traits.

As there is no clear market price, a method for evaluating properties is needed. Properties are valued by estimating what the price of a property would be if it was sold (Pagourtzi et al., 2003). Traditionally valuations have been done based on the experience and knowledge of expert appraisers. More recently, as more data has become obtainable and infrastructure for computing has developed, machine learning algorithms have been used to predict the price of properties with great success. For example, the neural network approach of Peng et al. (2021) outperformed valuations made by professional appraisers in most of the comparisons.

Many machine learning methods are so complex that the predictions that are made are not understandable. The lack of understandability can reduce the attractiveness of machine learning algorithms, as organizations can be reluctant to adopt models that are not understandable (Veale et al., 2018). In addition, subjects of automated decision-making, for example, individuals seeking a loan with a property as collateral, have a right to meaningful information about the logic involved under the General Data Protection Regulation (GDPR) of the European Union (Selbst & Powles, 2018).

Post-hoc interpretability methods could be a solution to this issue, as they can be used to explain any machine learning method. However, there are some issues related to the uncertainty of the explanations produced by post-hoc interpretability methods (Rudin, 2019). Another possibility is using machine learning methods that are constrained in a way, which makes them interpretable. Some authors think that there is a trade-off between accuracy and interpretability with intrinsically interpretable methods (Breiman, 2001; Lundberg & Lee, 2017). Rudin (2019) however claims that the trade-off between accuracy and interpretability does not necessarily exist.

1.1 Research problem and research questions

The main objective of this thesis is to estimate the usability of interpretable methods compared to more complex black-box methods in property valuation and highlight the benefits of interpretability in property valuation

machine learning tasks. In this thesis, the research problem is considered from the point of view of Nordea, for which an interpretable challenger model is provided in this thesis.

The objective of this thesis is to answer the following research questions:

1. What are the benefits of interpretability in property valuation?
2. Can interpretable machine learning methods predict property prices as accurately as complex state-of-the-art machine learning methods?
3. How can the used interpretable method be explained?

1.2 Research design and scope

First, interpretability in machine learning is described in detail. A review of previous literature is done to determine, what methods have been successfully used in estimating property prizes, and what interpretable methods could be used in this thesis. In addition, methods that were chosen based on the literature are described in detail.

After that, interpretable and black-box methods that were chosen based on the literature review, are compared to determine if there is a trade-off between the chosen methods. In addition, a model that was built with the chosen interpretable method is presented to highlight the interpretability of the chosen method.

There is extensive literature on machine learning methods that have been compared in property valuation. Based on the literature, the most promising black-box type methods are identified, and a single best method will be chosen for this thesis. As there is extensive literature on this topic, the considered machine learning methods are limited to methods that have been used in multiple studies.

For interpretable methods, all methods that have been used in property valuation are included, as the literature on interpretable machine learning in property valuation is scarce. The chosen interpretable method is determined based on the performance when compared to the previously identified promising methods.

The research questions are answered based on the data that has been provided by Nordea. The data is limited to transactions of owner-occupied properties in six, big municipalities of Denmark.

1.3 Structure of the thesis

This thesis has the following structure. Section 2 introduces interpretable machine learning as a topic and discusses the strengths and weaknesses of different ways of ensuring interpretability. Section 3 reviews the existing literature on machine learning methods that have been used in property valuation. The most promising black-box and interpretable methods are chosen based on the literature review. The used data, chosen methods, and

other decisions related to the modelling process are detailed in Section 4. Section 5 presents the results of different methods and highlights the interpretability of the chosen interpretable method. Lastly, Section 6 includes discussions of findings and avenues of future research.

2 Interpretability in machine learning

In this section, different definitions of interpretability are introduced and the reasons for interpretability are discussed. After that, characteristics of good explanations of machine learning systems, and strengths and weaknesses of different ways of providing explanations are presented.

There is no clear consensus on the definition of interpretability in the machine learning context (Doshi-Velez & Kim, 2017; Lipton, 2018; Molnar, 2020). According to Rudin (2019), the meaning of interpretability is domain specific.

Even though there seems to be an agreement that there is no clear definition of interpretability, some authors have provided useful definitions that are related to human understanding. Doshi-Velez and Kim (2017) define interpretability as the ability to present or present machine learning systems to humans. According to Molnar (2020), the interpretability of machine learning is related to the ease of understanding why certain predictions have been made. If it is easy to understand the reasoning for predictions made by a model, the interpretability of said model is high.

Interpretability can be divided into two categories, post-hoc interpretability and intrinsic interpretability (Molnar, 2020). Post-hoc interpretability refers to separate methods that can be applied to any machine learning model after training (Doshi-Velez & Kim, 2017). Post-hoc interpretability methods are used to explain black-box machine learning models that do not produce understandable explanations of predictions (Rudin, 2019).

According to Rudin (2019), intrinsically interpretable models are constrained in a way that conveys useful knowledge, while Molnar (2020) notes that intrinsic interpretability stems from the simple structure of machine learning models that are easily understandable.

2.1 Why interpretability?

According to Lipton (2018), the reasons for decisions are often important. In addition to making accurate predictions, understanding why predictions are made is also important. Lundberg and Lee (2017) state that understanding why a model makes certain predictions can be as important as prediction accuracy.

Adoption of machine learning

Lack of interpretability is limiting the adoption of complex machine learning methods in many industries. According to Veale et al. (2018), several practitioners noted that getting organizational buy-in with models that were

not interpretable was difficult. Lorenz et al. (2022) state that the adoption of machine learning may seem surprisingly slow in many industries, and the lack of model interpretability is limiting the acceptance and implementation of machine learning methods. According to Alvarez Melis and Jaakkola (2018), the lack of interpretability can hamper the adoption of machine learning in high-stakes decision-making especially. Lee (2022) brings up the low interpretability of often-used black-box type methods as a reason for the slow adoption of machine learning methods for property valuation.

Interpretability in machine learning is expected to catalyse the adoption of machine learning according to Molnar (2020). Similarly, Lorenz et al. (2022) note that understanding how predictions are made through interpretability will ease practical applications. Lee (2022) anticipates that interpretable machine learning will be a catalyst in the adoption of machine learning techniques in property valuation.

Mismatch of goals

The goals of developing machine learning models, e.g. maximizing out-of-sample accuracy, are not always aligned with real-world tasks that they are meant to solve (Ribeiro et al., 2016). Interpretable models might help with the issue of mismatching goals of machine learning models and the real-world (Lipton, 2018).

According to Doshi-Velez and Kim (2017), algorithms often optimize a proxy of function for the ultimate goal. By understanding how a model works, it is possible to ensure that the proxy goal and the ultimate goal align or that the gap between the goals is visible. Similarly, Molnar (2020) notes that interpretability helps with the gap that is caused by the imperfect goal specification of machine learning models.

Confirming important criteria

When machine learning systems are understood by humans, it is possible to confirm other important criteria of machine learning systems (Doshi-Velez & Kim, 2017). Doshi-Velez and Kim (2017) list five of these criteria, which are fairness, privacy, robustness, causality, and trust.

Rudin et al. (2018) bring up the possibility to evaluate and debate the fairness of a model as a benefit of interpretability. With explanations, it is possible to prove that the decision-making has been done fairly and ethically (Adadi & Berrada, 2018).

According to Lorenz et al. (2022), interpretability enhances the reliability of machine learning models. Ghorbani et al. (2019) note that it is possible to establish necessary trust by reliably explaining predictions of machine learning models. Trusting single predictions and trusting the model are

impacted by how well humans understand the model's behaviour, which is necessary for the model to be used (Ribeiro et al., 2016).

High stakes

Rudin (2019) states that interpretability is needed for high-stakes decision-making. According to Lipton (2018), the ability of humans to understand the reasoning of machine learning models is important in critical areas, such as medicine, criminal justice, and finance. Similarly, Molnar (2020) claims that as soon as a model has a significant financial or societal impact, understanding how and why predictions are made ensures trust and fairness.

Gaining knowledge

Another reason, why interpretability is important is related to gaining knowledge. According to Lorenz et al. (2022) understanding how a machine learning model arrived at its prediction will make it possible to gain new insights about the phenomenon that is studied. Doshi-Velez and Kim (2017) note that explanations of the inner workings of machine learning models can be converted to knowledge. According to Adadi and Berrada (2018), explainable artificial intelligence can help with gaining knowledge on the subject matter.

Troubleshooting and improving through iterations

According to Rudin (2019), one main benefit of interpretability is that it makes troubleshooting easier. Adadi and Berrada (2018) bring up the ability to quickly identify and correct mistakes as a reason for explainable artificial intelligence.

Adadi and Berrada (2018) also mention that explainable artificial intelligence makes it easier to improve models through iteration. According to Rudin (2019), interpretability can lead to more accuracy through a better understanding of the studied problem and iteration.

Legal requirements

GDPR adds to the need to address the interpretability of machine learning models. Goodman and Flaxman (2017) argue that the right to explanation exists, while it is not clear what is meant by that, while Wachter et al. (2017) disagree with Goodman as according to them there is no clear right to explanation outlined in GDPR. Selbst and Powles (2018) however argue that rights to "meaningful information of the logic involved" when individuals are subject to automated decision-making in articles 13-15 of GDPR essentially give individuals the right to an explanation. Adadi and Berrada (2018), Rudin

(2019), and Doshi-Velez and Kim (2017) also bring up compliance with the law as a reason for interpretability.

Why not?

Interpretability is not always necessary. If a model can be trusted because it has been used successfully over time, or if there are no significant consequences for unacceptable results, interpretability might not be needed (Doshi-Velez & Kim, 2017; Molnar, 2020). In addition, interpretability might allow manipulation of the system (Molnar, 2020).

Interpretability can also cause misplaced trust in machine learning. Poursabzi-Sangdeh et al. (2021) found that participants of their study were worse at detecting sizable mistakes made by an interpretable prediction model when compared to a black-box type prediction model.

2.2 How to evaluate interpretability?

Swartout and Moore (1993) list five general requirements for useful explanations of artificial intelligence systems. These requirements can also apply to the explanations of modern machine learning models, as there are many similarities in more recent machine learning interpretability literature. The five requirements are fidelity, understandability, sufficiency, low construction overhead, and efficiency.

Fidelity

The first general requirement by Swartout and Moore (1993) is fidelity, which refers that explanation must accurately represent the system that is being explained. In more recent literature, Ribeiro et al. (2016) bring up local fidelity as an essential criterion for explanations and Alvarez Melis and Jaakkola (2018) mention faithfulness as a desideratum of explanations. Important aspects of explanations that are related to the fidelity desideratum are the stability of explanations brought up by Alvarez Melis and Jaakkola (2018) and the robustness of explanations brought up by Ghorbani et al. (2019).

Understandability

The second requirement listed by Swartout (1993) is understandability: the explanations are only useful if they are understood by the users of the systems. Also, Ribeiro et al. (2016) bring up understandability as an important factor in explanations.

Lipton (2018) lists different notions of interpretability that are all related to understandability. These notions are simulatability, which refers to the

ability to replicate the decision-making process of a trained algorithm, decomposability, which refers to understanding parts of the model, such as the impact of an input variable, and algorithmic transparency, which refers to understanding how a learning algorithm converges.

Molnar (2020) presents different scopes of interpretability. In addition to algorithmic transparency, the difference between global and local interpretability is brought up. Global interpretability refers to understanding how a model makes predictions, while local interpretability refers to understanding why a single prediction was made.

Molnar (2020) mentions that holistic, global interpretability is often very hard to achieve, as completely understanding any model that has over a few parameters or weights is difficult. This is related to a measure of interpretability presented by Rudin (2019), which is sparsity, as the sparsity of models affects how easy they are to understand. For linear models, Molnar (2020) argues that humans cannot imagine feature spaces with more than 3 features, which limits the understandability of models with many features.

Molnar (2020) brings up another level of global interpretability, global interpretability on a modular level. It refers to understanding parts of the model at a time, such as the effects of single features or interactions between features, and is easier to achieve than holistic, global interpretability. Global interpretability on a modular level is similar to the decomposability notion introduced by Lipton (2018).

Sufficiency

The third requirement of explanations listed by Swartout and Moore (1993) is sufficiency, the explanations need to produce answers to all the questions that users of the system might have. Alvarez Melis and Jaakkola (2018) also brought up a similar desideratum of interpretability that should be satisfied, which is explicitness.

Low construction overhead

Low construction overhead is the fourth requirement of explanations listed by Swartout and Moore (1993). Producing explanations should not add too much additional workload for the designer of the system, as it might result in foregoing explanations altogether.

Efficiency

The last requirement of explanations presented by Swartout and Moore (1993) is efficiency, which focuses on the runtime efficiency of creating explanations. Similar to low construction overhead, creating explanations should not affect the runtime of a system too much.

2.3 Post-hoc interpretability

With simple models, the best explanation is the model itself. With more complex models that cannot produce understandable explanations of predictions, an explanation model that approximates the original model is needed for producing explanations (Lundberg & Lee, 2017). With the growing availability of big data, complex models are often more accurate than simpler ones (Lundberg & Lee, 2017). Modern machine learning techniques can automatically detect non-linearities, transformations, and high-order interactions, which results in great accuracy (Mayer et al., 2019).

According to Lipton (2018), one advantage of post-hoc interpretability comes from the ability to provide interpretability without losing accuracy. Model-agnostic characteristic of post-hoc interpretability methods makes it possible to explain the best current methods as well as all future methods (Ribeiro et al., 2016).

The main deficiency of post-hoc interpretability methods is related to fidelity. According to Rudin et al. (2022), post-hoc explanations are too often misleading or wrong. Lipton (2018) also mentioned that post-hoc explanations can be misleading. Alvarez Melis and Jaakkola (2018) claim that most post-hoc interpretability methods are not faithful to the original model. The untrustworthy explanations can create misplaced trust in the black-box models, which can be harmful (Rudin et al., 2022).

Dimanov et al. (2020) show that explanations produced by post-hoc interpretability methods, including Partial Dependence Plot (PDP), Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), can be misleading, as these methods do not necessarily identify significant features correctly.

Gosiewska and Biecek (2019) found problems related to inconsistency, uncertainty, and infidelity with commonly used post-hoc interpretability methods including SHAP, and LIME. According to Gosiewska and Biecek (2019), the root cause of these issues is the simplicity of explanation methods, when compared to complex black-box methods. Baniecki et al. (2021) fooled PDP with data poisoning algorithms. They found that the explanations of low-variance models are robust, while the robustness of the explanations of more complex models is not satisfactory.

Rudin (2019) argues that explanation methods might leave out so much information that the explanations make no sense. According to Rudin (2019), summary statistics of predictions would be a more accurate term instead of explanations, as the produced graphs show trends on how features are related to predictions instead of explaining how the models work. Also, Alvarez Melis and Jaakkola (2018) argue that explanations of most post-hoc interpretability methods are not explicit enough.

According to Rudin (2019), the additional explanation model with post-hoc interpretability creates an extra burden, when the explanations show something that is not intended, as the practitioners need to troubleshoot the explanation model as well as the underlying black-box model. Dimanov et al. (2020) suggest rigorous robustness checks to combat robustness issues of explanation methods. Similarly, Baniecki et al. (2021) suggest ensuring the reliability of explanation results by utilizing the data poisoning methods.

Rudin (2019) argues that the whole idea of post-hoc interpretability is problematic because if the explanation model would be completely faithful to the underlying model, the underlying model would not be needed, and the interpretable explanation model should be used. If the explanation model is not faithful to the underlying model, neither the explanation model nor the underlying model can be trusted.

2.4 Intrinsic interpretability

With intrinsically interpretable methods, the explanation is the model itself, which is the biggest strength of intrinsic interpretability. No additional methods are needed for producing explanations, and the explanations accurately represent the trained model. In addition, with interpretable models, it is easier to improve the model throughout the process, as the user has a clear understanding of what is happening. Because of this, Rudin (2019) argues that this could reverse the trade-off between accuracy and interpretability, as interpretability would lead to more accuracy.

The main weakness of interpretable models is connected to the effect that the simple, understandable structure that is needed for interpretability, can have on predictive performance. Rudin (2019) states that there is a widespread belief that there exists a trade-off between interpretability and accuracy in machine learning.

This trade-off between interpretability and accuracy in machine learning methods is also brought up by other authors. Breiman (2001) states that in general, accuracy requires complex models, while simple and interpretable models are not accurate. According to Lundberg and Lee (2017), big data has increased the attractiveness of complex black-box models, making the interpretability-accuracy trade-off relevant.

Lipton (2018) claims that using inherently interpretable models requires giving up predictive performance. Mayer et al. (2019) have a similar view of the trade-off between interpretability and accuracy as according to them complex black-box type machine learning methods can be more accurate than standard linear methods with the cost of losing interpretability. Similarly, Lorenz et al. (2022) state that machine learning researchers must decide between maximizing predictive power and limiting the model to understand basic mechanisms at work. According to Rudin et al. (2022) however, the accuracy of black-box models is not generally better than the

accuracy of well-designed interpretable models. Lack of accuracy can lead to untrustworthy explanations, as according to Breiman (2001) if there is a trade-off between interpretability and accuracy, the gained information is untrustworthy and generally not useful.

Rudin (2019) argues that the belief that there is necessarily a trade-off between accuracy and interpretability in machine learning, is a myth. When considering, how this myth might have been justified, Rudin (2019) brings up an unlikely example of comparing the predictive performances of a decision tree method developed in 1984 to a deep neural network developed in 2018 and concluding that the trade-off exists. However, in literature, where the trade-off has been discussed, simple and old interpretable models have been compared to more modern black-box models. For example, Breiman (2001) compared a decision tree method to a random forest method, Lipton (2018) compared linear regression models with deep neural network models and Mayer et al. (2019) compared modern machine learning techniques such as random forests, neural networks, and gradient boosting to linear regression when discussing the trade-off between interpretability and accuracy.

Rudin (2019) brings up several additional arguments for why the trade-off does not exist necessarily. One is based on the Rashomon effect introduced by Breiman (2001), which describes a phenomenon, where often a multitude of equations have similar error rates. For example, with linear regression with different variable selections or decision trees with slightly perturbed data, where the produced trees are different, the error rates across different models were very similar. Semenova et al. (2022) researched the effect of large Rashomon sets, where many different models have similar predictive performance, on the existence of interpretable and accurate models. One of their findings was that simple and interpretable yet accurate models are more likely to exist with datasets that have large Rashomon sets.

Being able to capture complex relationships is seen as a strength of black-box type models (Mayer et al., 2019), while Rudin (2019) argues that if a hidden pattern is important enough, also interpretable models could leverage it. According to Rudin (2019) the belief that the trade-off between accuracy and interpretability exists acts as a self-fulfilling prophecy, as it makes researchers ignore attempting to build interpretable and accurate models. This effect is also magnified by the fact that many researchers are trained to build deep models but not interpretable models (Rudin, 2019).

Rudin (2019) mentions that compared to unconstrained black-box models interpretable methods can be significantly harder to construct in terms of domain expertise and computation. Solving constrained optimization problems is generally harder than solving unconstrained ones. More complex methods learn transformations, non-linearities, and high-order interactions automatically (Mayer et al., 2019) while discovering and utilizing these patterns with interpretable models is difficult (Rudin, 2019).

One weakness in terms of the interpretability of linear models brought up by Lipton (2018) is that linear models often have good algorithmic transparency, while their simulatability and decomposability might be lower than other methods, due to heavy feature engineering, or a large number of features.

3 Property valuation

In this section, different property valuation approaches are presented. After that, the best complex and interpretable methods are identified.

3.1 Property valuation approaches

Gabrielli and French (2020) present three internationally recognized property valuation approaches. These approaches are the cost approach, the income approach, and the market approach. In the cost approach, a property is valued based on how much it would cost to reconstruct it. In the income approach, the value of a property is derived from possible future income that it would provide. In the market approach, price information of other properties is utilized to determine the value of a property. (Pagourtzi et al., 2003) In this thesis, the focus is on owner-occupied properties. According to Pagourtzi et al. (2003), the market approach is most suitable for valuing these kinds of properties.

In the market approach, properties can be valued with the comparable method, where properties are valued by selecting similar comparable properties and adjusting the prices of the comparable properties based on their similarity to the subject property (Pagourtzi et al., 2003). Alternatively, the value of properties can be also evaluated based on historical data with regression methods and machine learning methods (Gabrielli & French, 2020).

Algorithms that mimic the comparable method were utilized by Trawiński, Lasota, et al. (2017), Trawiński, Telec, et al. (2017), Lasota, Telec, et al. (2011), and Kempa et al. (2011). In Trawiński, Lasota, et al. (2017) and Trawiński, Telec, et al. (2017) methods called nearest similar transactions, latest transactions in the area and random similar transactions had worse accuracy than all the machine learning methods that they were compared to. Lasota, Telec, et al. (2011) and Kempa et al. (2011) utilized an algorithmic approach developed by professional valuators to simulate the routine work of property valuators. In Kempa et al. (2011) many of the machine learning methods had better performance than the expert algorithm, while in Lasota, Telec, et al. (2011) the expert algorithm had a good comparative performance.

3.2 Linear regression

Linear regression is categorized as a traditional valuation method by Pagourtzi et al. (2003). Linear regression has been popular in property valuation literature, where the accuracy of different methods has been

compared. It is often used as a baseline model that other more complex models are compared to.

In semi-log functional form, where the dependent variable is transformed with natural logarithm, linear regression has often been the weakest performing method out of the compared methods in property valuation literature (Gnat, 2021b; Hong et al., 2020; Hurley & Sweeney, 2022; Mayer et al., 2019; Selim, 2009). There are also some comparative studies (Alexandridis et al., 2019; Bogin & Shui, 2020), where linear regression with semi-log functional form is better than some of the compared methods, while still being considerably worse than the best methods.

The comparative performance of linear regression with linear functional form has been similar to the performance with semi-log functional form. In many comparative studies standard linear regression was the worst in accuracy out of the compared methods (Abidoye & Chan, 2018; Ghatnekar & Shanbhag, 2021; Guan et al., 2014; Liu et al., 2018; Reyes-Bueno et al., 2018; Štubňová et al., 2020; Valier, 2020a; Yilmazer & Kocaman, 2020). There are also many studies, where linear regression has been more accurate than some of the methods that it was compared to, while still being considerably weaker than the best methods (Antipov & Pokryshevskaya, 2012; Dimopoulos & Bakas, 2019; Jamil et al., 2020; Mrsic et al., 2020; Nejad et al., 2016; Tchuente & Nyawa, 2022; Trawiński, Telec, et al., 2017; Valier, 2020b; Yee et al., 2021). The reason for the low predictive performance of linear regression lies in its inability to model non-linearity and complex patterns (Hong et al., 2020; Hurley & Sweeney, 2022).

Due to its interpretable nature, linear regression is often used to determine the casual relationship between the price of the property and some characteristics of the property or external factors. The effect of adjacent public transportation infrastructure (Chen et al., 2019; Zhang & Shukla, 2023), Airbnb density in the post-code area (Thackway et al., 2022), proximity to former prison (Shehata et al., 2021), school quality (Rajapaksa et al., 2020), tourist volume (Liu et al., 2020), green certificates of buildings (Dell'Anna & Bottero, 2021), the proximity of urban villages or slums (Hussain et al., 2021), distance to nearest cell phone tower (Rajapaksa et al., 2018), the proximity of historical sites (Hicks & Queen, 2016), septic system and type of soil (Vedachalam et al., 2013), and drinking water quality (Des Rosiers et al., 1999) to value of a property has been studied with linear regression

In addition to standard linear regression, other methods for fitting linear functions have also been used in property valuation. These include lasso and ridge regression. Ridge and lasso regression both include a regularization term, which pushes the values of coefficients towards zero (Bogin & Shui, 2020; Doumpos et al., 2021; Gnat, 2021a). The difference between lasso and ridge is that lasso tends to set the coefficients to zero, while ridge just shrinks the coefficients, without setting them to zero (Tibshirani, 2011).

When included in studies with linear regression, neither ridge nor lasso regression offers considerable improvements in property valuation accuracy (Bogin & Shui, 2020; Doumpos et al., 2021; Jamil et al., 2020). When compared to more complex machine learning methods, lasso and ridge regression have often been less accurate than the more complex machine learning models (Bin et al., 2017; Bogin & Shui, 2020; Doumpos et al., 2021; Gnat, 2021a; Jamil et al., 2020)

The accuracy of linear regression is often considerably lower than the accuracy of the best methods it has been compared to, so even though linear regression has good interpretability, it will not be considered for the empirical part of this thesis.

3.3 Machine learning methods

3.3.1 Decision tree methods

Machine learning methods based on decision trees have been very popular in property valuation. Methods based on a single decision tree are often not as accurate as more complex machine learning methods. There are a lot of comparative studies, where other methods are considerably more accurate (Antipov & Pokryshevskaya, 2012; Hurley & Sweeney, 2022; Lasota et al., 2013; Lasota, Makos, et al., 2009; Lasota, Sachnowski, et al., 2009; Lasota et al., 2015; Masrom et al., 2022; Mrsic et al., 2020; Nejad et al., 2016; Nejad et al., 2017; Valier, 2020a, 2020b; Yee et al., 2021). In Jamil et al. (2020) however, the decision tree method was better than the methods it was compared to. While methods based on a single decision tree can be interpretable, they will not be included in the empirical part of this thesis due to their poor performance, when compared to other methods.

While the performance of a single decision tree has not been good, other methods based on decision trees have had a very good performance. Ensemble methods that are typically used with decision trees include bagging and boosting. In bagging, separate training sets, which do not involve all the training samples, are created for individual models, and the final prediction is the average prediction of the separate models (Breiman, 1996). In boosting, new instances are iteratively trained to minimize the difference between the previous prediction and the prediction target (Friedman, 2001).

Methods based on bagged decision trees, including random forest, are often among the best-performing methods (Antipov & Pokryshevskaya, 2012; Dimopoulos & Bakas, 2019; Gnat, 2021a; Ho et al., 2021; Hong et al., 2020; Lasota, Sachnowski, et al., 2009; Masrom et al., 2022; Nejad et al., 2016; Nejad et al., 2017; Talaga et al., 2019; Valier, 2020a, 2020b; Yee et al., 2021; Yilmazer & Kocaman, 2020). However, in Bańczyk et al. (2011), bagged decision stumps, which have only one split per tree, had the worst performance.

There are also studies, where some other methods have outperformed decision tree-bagging methods. In these studies, random forest is often outperformed by methods based on gradient boosting (Ghatnekar & Shanbhag, 2021; Jarosz et al., 2020; Mayer et al., 2019; Niu & Niu, 2019; Schulz & Wersing, 2021). In Niu and Niu (2019) random forest was also outperformed by multilayer perceptron (MLP), while advanced interpretable methods that will be presented later, have also outperformed random forest (Hurley & Sweeney, 2022; Mayer et al., 2019; Schulz & Wersing, 2021).

Methods based on boosted decision trees have had good performance in property valuation. These methods include gradient boosting machine, AdaBoost, and extreme gradient boosting (XGB). Gradient boosting machine is often the best method or has almost the same accuracy as the best method (Ho et al., 2021; Jarosz et al., 2020; Liu et al., 2018; Mayer et al., 2019; Mrsic et al., 2020; Tchuente & Nyawa, 2022). There are also some studies, where other methods are better than gradient boosting. In Niu and Niu (2019) MLP was better than gradient boosting, in Nejad et al. (2017) random forest was more accurate, and in Nejad et al. (2016) random forest and few neural network approaches were better. In Dimopoulos and Bakas (2019) the performance of the gradient boosting machine was very poor, but the gradient boosting model was overfitted in that study.

XGB is an improved implementation of gradient boosting, which has built-in regularization (Chen & Guestrin, 2016). It was the best method in Ghatnekar and Shanbhag (2021), Gnat (2021b), Mrsic et al. (2020), and Nejad et al. (2016).

Interestingly, when XGB has not been the absolute best method, the results have been very poor. Other decision tree ensemble methods outperformed XGB in Nejad et al. (2017), where the hyperparameter tuning for XGB was very limited. In Talaga et al. (2019) all other methods, including decision trees and linear regression, had better performance. In that study, the feature selection methods that increased the accuracy of other methods were not utilized for the XGB method. Also, the data was divided into very small subsets for model training, which might make XGB overfit. AdaBoost, where the decision trees are limited to one split into two nodes, has similar but slightly worse accuracy than XGB in Tchuente and Nyawa (2022) and Mrsic et al. (2020).

Another tree-based method that has seen a lot of use in property valuation literature is a model tree. In the model tree, there are separate multivariate linear regressions in each leaf node (Lasota et al., 2013). Model trees are often better than MLP (Graczyk et al., 2009; Lasota et al., 2015; Nejad et al., 2016; Trawiński, Lasota, et al., 2017; Trawiński, Telec, et al., 2017) and linear regression (Graczyk et al., 2009; Nejad et al., 2016; Trawiński, Telec, et al., 2017). The model tree performed well overall but was still worse than XGB and random forest in Nejad et al. (2016).

The model tree is often combined with bagging and boosting, and the comparative performance is similar to the standard model tree. Bagged model tree is often better than bagged MLP (Bańczyk et al., 2011; Graczyk et al., 2010; Lasota, Łuczak, et al., 2011; Trawiński, Lasota, et al., 2017), bagged RBF (Bańczyk et al., 2011; Lasota, Łuczak, et al., 2011) and linear regression (Graczyk et al., 2010; Lasota, Łuczak, et al., 2011), while being worse than bagged SVM (Graczyk et al., 2010; Lasota, Łuczak, et al., 2011). The results are similar with boosted model tree (Graczyk et al., 2010; Trawiński, Lasota, et al., 2017).

3.3.2 Neural network methods

Neural networks have received a lot of attention in property valuation literature. Convolutional neural networks (CNN) used in image classification can offer additional useful information for property valuation. Different CNN architectures have been used for valuing Google street view images of properties (Johnson et al., 2020) and satellite images around the properties (Lin et al., 2021). For this thesis, however, only machine learning approaches that use tabular data are considered.

Two neural network methods have mainly been used in previous property valuation literature. These are MLP and Radial Basis Function (RBF). The performance of the two methods has been compared in several studies. In some of the studies, MLP has outperformed RBF (García et al., 2008; Lasota, Makos, et al., 2009), while in others RBF had better performance than MLP (Antipov & Pokryshevskaya, 2012; Graczyk et al., 2010). In addition, Telec et al. (2013) found no differences in the performance of RBF and MLP.

Other neural network methods have also been used in property valuation. Lee and Park (2020) compared MLP to a Bayesian neural network with MLP outperforming the Bayesian neural network. Lasota, Makos, et al. (2009) also included a DMNeural method, which was worse than MLP and RBF in terms of accuracy.

When methods based on neural networks have been compared to other property valuation methods, the comparative performance of neural networks has varied a lot. There are many studies, where neural networks have been compared to linear regression (Abidoye & Chan, 2018; Guan et al., 2014; Mimis et al., 2013; Selim, 2009; Štubňová et al., 2020). In all of them, neural networks were more accurate than linear regression.

In many studies, MLPs have been the best or one of the best methods when compared to other machine learning approaches (Alexandridis et al., 2019; Lasota, Makos, et al., 2009; Lasota, Sachnowski, et al., 2009; Masrom et al., 2022). In addition, Talaga et al. (2019) used MLP with bagging in Talaga et al. (2019).

There are also many studies, where MLP and RBF have had considerably worse accuracy than other machine learning methods (Antipov &

Pokryshevskaya, 2012; Lam et al., 2009; Liu et al., 2018; Mayer et al., 2019; Nejad et al., 2016; Tajani et al., 2015; Tchuente & Nyawa, 2022; Telec et al., 2013; Trawiński, Telec, et al., 2017; Valier, 2020a). Also bagged MLPs have been outperformed by other methods in many studies (Bańczyk et al., 2011; Graczyk et al., 2010; Trawiński, Lasota, et al., 2017).

The reasoning for the performance of MLPs has been discussed in some of the papers, where its comparative performance has been subpar. Antipov and Pokryshevskaya (2012) explain the poor performance of MLPs to the insufficient tuning of the model, while Lam et al. (2009) mention the large number of tuneable hyperparameters, which make the model hard to tune.

3.3.3 Other methods

K-nearest-neighbours (KNN) method has also received a lot of attention in property valuation literature. It is often either one of the worst methods (Mrsic et al., 2020; Nejad et al., 2016; Valier, 2020b) or at least far from the best method (Antipov & Pokryshevskaya, 2012; Gnat, 2021b; Hurley & Sweeney, 2022; Lasota, Sachnowski, et al., 2009; Tchuente & Nyawa, 2022; Valier, 2020a). As an exception, the accuracy of KNN was very close to the most accurate method of random forest in Gnat (2021a). According to Tchuente and Nyawa (2022), KNN performs poorly with large datasets and a large number of important variables, which could explain the poor performance of KNN.

Support vector machine (SVM) is another machine learning method, which has been used in many property valuation studies. It consistently outperforms KNN, different neural network methods, and methods based on linear regression (Bin et al., 2017; Ghatnekar & Shanbhag, 2021; Graczyk et al., 2009; Graczyk et al., 2010; Lam et al., 2009; Lasota, Łuczak, et al., 2011; Lasota, Sachnowski, et al., 2009; Liu et al., 2018; Valier, 2020b). There are some exceptions, as in Tchuente (2022) SVM is worse than KNN, linear regression, and MLP, in Lasota et al. (2015) linear regression is more accurate than SVM, and in Nejad et al. (2016) best MLP approaches outperform best SVM approaches. When compared to the model tree, SVM was better in Graczyk et al. (2009) and worse in Lasota et al. (2015).

The performance of SVM is also consistent when compared to well-performing methods based on gradient-boosted trees or bagged trees. The performance of these methods is consistently better than SVM (Ghatnekar & Shanbhag, 2021; Ho et al., 2021; Liu et al., 2018; Nejad et al., 2016; Tchuente & Nyawa, 2022; Valier, 2020b). In Lasota, Sachnowski, et al. (2009) the performance of SVM is similar to the performance of random forest.

3.3.4 Conclusion of machine learning methods

Overall, random forest, XGB and MLP seem to be the most potential machine learning methods for property valuation. MLP seems to have some inconsistency issues due to being hard to tune, while random forest seems to have consistently good but not great results. XGB seems to be the most promising method overall, as most of the time it outperforms all other methods.

3.4 Advanced interpretable methods

While standard linear regression has not been very accurate compared to other methods overall, linear regression methods have had better comparative performance when spatial effects have been introduced in some way. Doumpos et al. (2021) used an approach, where only the closest properties are used for estimating the value of a property. They also tried using weights for properties used in model fitting, where the weights are inversely proportional to the distance between the target property and the property used in model fitting. With the weighted local approach, linear regression and lasso regression were more accurate than machine learning methods that were considerably better in the normal approach. Dimopoulos and Moulas (2016) used a similar geographically weighted regression approach, where different functions for linear regression are created locally. This approach outperformed linear regression. Similarly, Oust et al. (2020) used geographically weighted regression, which outperformed linear regression, regression kriging, and vicinity-based regression tuning.

Mayer et al. (2019) used regression with a semi-log functional form that had separate intercepts at state, regional, and municipal levels. This regression approach was only slightly less accurate than the best machine learning method. Mimis et al. (2013) used a linear regression approach, where in addition to the independent variables, the estimated price of a property was also affected by the price of nearby properties. This approach was less accurate than the MLP it was compared to. Schulz and Wersing (2021) used a similar spatial autoregressive model, which was considerably worse than the best methods in that study.

Dimopoulos and Bakas (2019) used a high-order regression approach, where combinations of independent variables up to third order were included and added in a greedy stepwise manner. This approach was a clear improvement over standard linear regression, while still not being as accurate as random forest, which had the best comparative performance.

Reyes-Bueno et al. (2018) used a multivariate adaptive regression splines algorithm (MARS), which outperformed model tree and linear regression. In MARS data is modelled with piecewise linear segments that are created by

iteratively adding knots that improve performance until an end condition is reached and removing knots that are not effective afterwards.

Mayer et al. (2022) created structured additive regression (STAR) models that were fitted with XGB and lightGBM and compared them to normal XGB and lightGBM approaches respectively. The STAR method is based on additive functions that are restricted to only a subgroup of features. Mayer et al. (2022) did two separate property valuation case studies. In the first one, a STAR model was fitted with XGB, and only locational variables were allowed to interact with each other, while separate functions were fitted for the rest of the features. In the second one, a STAR model that was fitted with lightGBM, and locational variables and transaction time were allowed to interact with each other. In both case studies, the STAR models had slightly worse accuracies than the complex unconstrained models.

A generalized additive model (GAM) is a special case of the STAR model, where the additive functions are limited to single features (Mayer et al., 2022). Generative Additive Model plus interactions (GA^2M) are similar to GAMs in structure, while also including functions that involve a pair of features (Lou et al., 2013). Hurley and Sweeney (2022) compared GAM and GA^2M models with different specifications to machine learning methods including KNN, decision tree, and random forest. In their approach, they had separate functions fitted with splines for four of the variables and one pairwise function that included latitude and longitude. The best GA^2M outperformed all the machine learning methods, including random forest.

Rajapaksa et al. (2018) compared the accuracy of GAM and GA^2M to the accuracy of linear regression in a quantitative study. In their different GA^2M models they predetermined the allowed pairwise functions. While there was no difference in terms of accuracy between GAM and GA^2M, both offer a significant improvement in terms of accuracy compared to standard linear regression.

Schulz and Wersing (2021) compared random forest, gradient boosting machine, polynomial model, penalised spline model and geo-spatial model in property valuation. The penalised spline model has a structure similar to GA^2M, where the features are modelled with either coefficients or functions limited to one or two features. Their penalised spline model was as accurate as the gradient boosting machine, and more accurate than other methods.

Explainable Boosting Machine (EBM) is an implementation of the GA^2M algorithm (Nori et al., 2019), where all separate functions for all the features are fitted first and functions for pairwise interactions that increase the accuracy the most, are fitted after. In Yang et al. (2021) EBM was compared to Multilayer Perceptron, random forest, generalized linear model, EBM without pairwise interactions, and their approach GAMI-Net in many different regression tasks. EBM outperformed all other methods in property valuation.

When it comes to comparative performance, when compared to the best black-box methods identified in the previous section, GA^{2}M, penalised spline, and STAR all seem to have the possibility of performing well with a minimal trade-off between interpretability and accuracy. In the STAR and the penalised spline methods, the allowed functions or feature interactions need to be predetermined, which would need knowledge of how different features typically affect the value of a property. In the EBM implementation of GA^{2}M, all features are included in the single feature fitting phase and two feature functions are selected automatically, which is why it is chosen for this thesis, as there is no accurate knowledge available of how different features affect the property prices with these type of models.

4 Methodology

In this section, dataset and accuracy metrics are presented. After that, the machine learning methods and hyperparameter optimisation method are described in detail.

4.1 Data

Data used in this thesis include property transactions from Denmark. The data includes all property transactions between the years 2009 and 2021. The quality of data is good enough that it could be used without any data cleaning. Still, some transactions and features are excluded for the following reasons.

As Danish Financial Supervisory Authority (DFSA) has limited the use of automated valuation models to big municipalities, the transaction data is limited to Copenhagen, Aarhus, Aalborg, Odense, Frederiksberg, and Roskilde. There are different types of properties included in the original data, but only residential properties are selected for this thesis, as the aim of this thesis is to valuate residential properties. The original dataset includes features related to public valuation and valuation for tax purposes, which are omitted to not make the model reliant on other property valuations. In addition, transactions over 7.5 million DKK are omitted, as DFSA does not allow the use of automated valuation models for properties over that amount. Also, the date of the transaction is changed to two different features, month, and year of the transaction. Both methods, EBM and XGB, can handle categorical features, such as the postal number of an area, without any data transformations, so categorical features are not encoded in any way. The features that were included in the final dataset are outlined in Table 1.

The data is split into training and testing data based on the time of the transaction. The latest transactions from the last quarter of 2021 are used for testing, while all other transactions, from 2009 to September 2021, are used for training.

No additional feature selection was done, as there are no prior studies that discuss the effect of different features with EBM in property valuation. Also, XGB should automatically detect the most important patterns in the data, so omitting features should not increase accuracy. The impact of a variety of factors on the value of a property has been studied in previous literature, but the findings of those studies are not relevant in feature selection for this thesis, because those studies used linear regression to study the causal relationships and both methods have the ability capture more complex relationships.

Table 1: Feature descriptions

Feature	Type	Explanation
kom nr	Categorical	municipality number
post nr	Categorical	postal number
zone kode	Categorical	Zoning of the area
ejendom type	Categorical	Property type
opfoerelse aar	Numeric	Built year
ombyg aar	Numeric, optional	Most recent remodelling year
areal vaegtet	Numeric	Floor area
areal grund	Numeric, optional	Land area
areal kaelder	Numeric, optional	Basement area
areal kaelder bolig	Numeric, optional	Residential basement area
areal tageetage saml	Numeric, optional	Attic area
areal tageetage udnyt	Numeric, optional	Residential attic area
areal garage indb	Numeric, optional	Built-in garage area
area udbus indb	Numeric, optional	Built-in shed area
areal udestue	Numeric, optional	Winter garden area
areal garage saml	Numeric, optional	Garage area
areal carport	Numeric, optional	Carport area
vaerelse antal	Numeric	Number of rooms
etager antal	Numeric	Number of floors
bad antal	Numeric	Number of bathrooms
toilet antal	Numeric	Number of flushing WCs
Coordinates:		
koor oest	Numeric	ETRS89 x-coordinate
koor nord	Numeric	ETRS89 y-coordinate
Distance to nearest:		
tog station	Numeric	Train station
bus stop	Numeric	bus stop
metro station	Numeric	Metro station
s tog station	Numeric	S-train station
motorvej	Numeric	Motorway
laege	Numeric	General practitioner
supermarked	Numeric	Supermarket
skole	Numeric	School
daginstitution	Numeric	Daycare centre
hospital	Numeric	Hospital
apotek	Numeric	Pharmacy
bibliotek	Numeric	Library
lugthavn	Numeric	Airport
kyst	Numeric	Coastline
skov	Numeric	Forest
soe	Numeric	Lake
idraetshal	Numeric	Sports centre
svoemmehal	Numeric	Football field
Transaction:		
tran year	Categorical	The month of sale agreement
tran month	Numeric	Year of sale agreement
Label:		
transaction amount	Numeric	Agreed purchase price

The final dataset includes transformations from Copenhagen, Aarhus, Aalborg, Odense, Frederiksberg, and Roskilde. The number of transactions used for training and testing models from each city is reported in Table 2. The number of transactions in testing data is untypically small when compared to the number of transactions in training data. This train-test split aims to mimic the intended use of the property valuation model in Nordea.

Table 2: Number of transactions used in training and testing in each municipality.

City	Training size	Testing size
Copenhagen	58494	209
Aarhus	34423	101
Aalborg	22571	87
Odense	18389	60
Frederiksberg	11701	39
Roskilde	9956	31
Total	155 534	527

For both methods, two different modelling approaches are used. In the global approach, all the data is used for training a single model for all municipalities. In the local approach, separate models are trained for all six municipalities included.

4.2 Machine learning pipeline

In this thesis, a datapoint represents single property transaction $i = (1, .., n)$, where n is the size of dataset, label y_i represents the transaction price, and feature vector $\boldsymbol{x}_i = (x_{i1}, \dots, x_{ij})$ represents features in Table 1. The objective of the machine learning algorithms is to model the relationship between features and labels:

$$y_i \approx F(\boldsymbol{x}_i). \quad (1)$$

Machine learning models predict labels based on feature vectors:

$$\hat{y}_i = F(\boldsymbol{x}_i). \quad (2)$$

Machine learning models aim to minimise the difference between predicted label $\hat{y}_i$ and the actual label y_i.

4.2.1 Accuracy metrics

Root mean squared error (RMSE) is used as a criterion that is optimised with hyperparameter optimization algorithms and for comparing finalised models. RMSE is derived from mean squared error (MSE), which is used as a criterion that is minimised in the training stage of both methods, so RMSE ranks compared models similarly to MSE. MSE is defined as:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}. \quad (3)$$

RMSE is used instead of MSE because it is more understandable according to Steurer et al. (2021), as its scale is similar to the prediction target. RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}. \quad (4)$$

In addition to RMSE, median absolute prediction error (MDAPE) is reported, when different methods are compared. According to Steurer et al. (2021), ratio-based measures can be more relevant than difference measures such as RMSE. MDAPE is defined with the following formula:

$$MDAPE = median\left|\frac{\hat{y}_i}{y_i} - 1\right|. \quad (5)$$

4.2.2 Gradient boosting

Both XGB and EBM are based on gradient-boosted decision trees, so gradient-boosted decision trees are introduced before XGB and EBM. Tree-structured predictors introduced in Breiman et al. (1984) are constructed by repeatedly splitting subsets of learning sample into two subsequent subsets or nodes. The predicted values for each instance are determined by terminal nodes, where the samples are classified based on the splitting criteria of each intermediate node. The predicted value in each terminal node is constant.

Breiman et al. (1984) described three main elements that are needed for a tree-structured predictor:

1. A way to select how splits are performed in each intermediate node
2. A rule to determine, when nodes are terminal
3. A rule to assign value to every terminal node.

Schapire (1990) introduced an algorithm that combines multiple weak learners to form a strong learner. The basic idea is that weak learners that are barely better than coinflip can achieve high accuracy by combining multiple weak learners. Friedman (2001) introduced the gradient boosting machine, which utilizes the same base idea, in which multiple weak learners are combined to form a strong learner.

$$F_{\mathrm{M}}(x) = \sum_{m=0}^{M} f_m(x), \quad (6)$$

where $f_0(x)$ is an initial guess, and $\{f_m(x)\}_1^M$ are individual functions or boosts that are trained iteratively and that are based on the predictions of the previous iterations.

The first step $f_0(x)$ is a constant value for all the samples i that minimizes the used loss function $l(y_i, \hat{y}_i)$. The loss function that is most used with

gradient boosting is squared-error loss. The constant value $\hat{y}_0$ of the first step is given by the following equation:

$$f_0(x) = \underset{\hat{y}_0}{\operatorname{argmin}} \sum_{i=1}^{N} l(y_i, \hat{y}_0) = \underset{\hat{y}_0}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2}(y_i - \hat{y}_0)^2, \tag{7}$$

which is solved by calculating the zero point of the derivatives with respect to the predicted value $\hat{y}_0$:

$$\frac{1}{2}\sum_{i=1}^{N} \frac{\partial}{\partial \hat{y}}(y_i - \hat{y}_0)^2 = -\sum_{i=1}^{N}(y_i - \hat{y}_0) = 0 \tag{8}$$

By solving this equation, the value of the initial step $f_0(x)$ becomes:

$$f_0(x) = \hat{y}_0 = \frac{1}{n}\sum_{i=1}^{N} y_i \tag{9}$$

so, the constant value of the initial step is the mean of the observed values $\{y_i\}_1^n$.

Each subsequent step $\{f_m(x)\}_1^M$ is a weak regression tree that is limited in size and created with the following procedure. New prediction targets $\tilde{y}_{im}$ are calculated for each sample i based on the negative gradient of previous predictions loss $l\big(y_i, F_{m-1}(x)\big)$ with respect to the predicted value $F_{m-1}(x)$:

$$\tilde{y}_{im} = -\left[\frac{\partial l\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1, N, \tag{10}$$

which is the residual error of the previous prediction when the squared error loss function is used:

$$\tilde{y}_{im} = y_i - F_{(m-1)}(x_i), i = 1, N. \tag{11}$$

With other loss functions, $\tilde{y}_{im}$ are close to the residual error, which is why they are called pseudo-residuals. Next, a weak regression tree is created to predict the pseudo-residuals of the previous step $\tilde{y}_{im}$.

The tree $f_m(x)$ splits space x into disjoint regions $R_{tm}, t = 1, T$. The split finding criterion is determined by the used loss function. The newly selected split has the lowest loss across all the possible split candidates. The number of terminal nodes T is determined by tuneable hyperparameter(s) that constrain the size of the three, such as the maximum number of terminal nodes or the maximum tree depth. The output of the tree $f_m(x)$ is given by:

$$f_m(x) = \sum_{t=1}^{T} \omega_{tm}\, 1(x \in R_{tm}), \tag{12}$$

where $1(\cdot)$ is an indicator with a value of 1 if it is true, and 0 otherwise. ω_{tm} is a coefficient of the terminal node R_{tm} and is given by:

$$\omega_{jm} = \underset{\omega}{\operatorname{argmin}} \sum_{x_i \in R_{tm}} l(y_i, F_{m-1}(x) + \omega), \tag{13}$$

which reduces to:

$$\omega_{jm} = \underset{\omega}{\operatorname{argmin}} \sum_{x_i \in R_{tm}} \frac{1}{2}(\tilde{y}_{im} - \omega)^2, \tag{14}$$

with the squared error loss function. In that case the coefficient ω_{tm} for each terminal node R_{tm} is the average of pseudo-residuals $\tilde{y}_{im}$ in that node.

Finally, the approximation is updated in the following way:

$$F_m(x) = F_{m-1}(x) + \eta f_m(x), \tag{15}$$

where η is a tuneable hyperparameter, also called the learning rate, which controls the size of each boosting step. Friedman (2001) found that small values, where $\eta < 0.1$, generate the most accurate results, and that with smaller values for η, the number of iterations M needs to be higher to achieve accurate predictions.

4.2.3 Extreme gradient boosting

XGB, introduced by Chen and Guestrin (2016), has several improvements over the gradient boosting algorithm that are related to regularization, speed, and scalability. Similar to the gradient boosting algorithm, XGB creates an ensemble of regression trees that predicts the target variable in an additive manner. Instead of trying to minimize some differentiable loss function $l(y, \hat{y})$, XGB maximizes the following regularized objective, which resembles loss reduction:

$$L(F) = -\sum_{i=1}^{n} l(y_i, \hat{y}_i) - \sum_{m=1}^{M} \Omega(f_m), \tag{16}$$

where $\Omega(f) = \gamma T + \lambda\omega^2$.

The regularization term Ω includes hyperparameters γ and λ, which will be introduced later. The objective of a single step m is:

$$L_m = -\sum_{i=1}^{n} l(y_i, \hat{y}_{i(m-1)} - f_m(x_i)) - \Omega(f_m) \tag{17}$$

Chen and Guestrin (2016) used a second-order approximation to optimize the objective:

$$L_m \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}_{i(m-1)}) - g_i f_m(x_i) - h_i {f_m}^2(x_i)] - \Omega(f_m), \tag{18}$$

where $g_i = \frac{\partial}{\partial \hat{y}_{i(m-1)}} l(y_i, \hat{y}_{i(m-1)})$ and $h_i = \frac{\partial^2}{\partial \hat{y}_{i(m-1)}} l(y_i, \hat{y}_{i(m-1)})$ are gradient and Hessian statistics of the loss function respectively. With the squared error loss function $l(y_i, \hat{y}_{i(m-1)}) = \frac{1}{2}(y_i - \hat{y}_{i(m-1)})^2$, the gradient g_i is the negative residual error of the previous step:

$$g_i = -(y_i - \hat{y}_{i(m-1)}) = -\tilde{y}_{im}, \tag{19}$$

and the Hessian h_i is:

$$h_i = -1 \tag{20}$$

By removing constant values simplified objective can be obtained:

$$\tilde{L}_m = -\sum_{i=1}^{n}[g_i f_m(x_i) + h_i {f_m}^2(x_i)] - \Omega(f_m), \tag{21}$$

Let I_t be the set of instances in a terminal node t with tree structure $q(x)$. By expanding equation 16 the maximized objective of a single leaf node becomes:

$$\tilde{L}_m = -\sum_{i=1}^{n}[g_i f_m(x_i) + h_i {f_m}^2(x_i)] - \gamma T - \lambda \sum_{t=1}^{T} {\omega_t}^2$$

$$\tilde{L}_m = \sum_{t=1}^{T} -2\left(\sum_{i \in I_t} g_i\right)\omega_t - \left(\sum_{i \in I_t} h_i + \lambda\right) {\omega_t}^2 - \gamma T. \tag{22}$$

For a fixed structure $q(x)$ the optimal weight ω_t^* for leaf t can be obtained with partial derivate with respect to weight ω_t:

$$\frac{\partial}{\partial \omega_t}\left(-2\left(\sum_{i \in I_t} g_i\right)\omega_t - \left(\sum_{i \in I_t} h_i + \lambda\right){\omega_t}^2 - \gamma T\right) = 0$$

$$-2\left(\sum_{i \in I_t} g_i\right) - 2\left(\sum_{i \in I_t} h_i + \lambda\right)\omega_t = 0$$

$$\omega_t^* = -\frac{\sum_{i \in I_t} g_i}{\sum_{i \in I_t} h_i + \lambda}. \tag{23}$$

Utilizing this equation, the simplified objective becomes:

$$\tilde{L}_m(q) = \sum_{t=1}^{T} \frac{\left(\sum_{i \in I_t} g_i\right)^2}{\sum_{i \in I_t} h_i + \lambda} - \gamma T. \tag{24}$$

This simplified objective is used as a criterion for split selection. Let I_L and I_R be the set of instances in the left and right nodes after a split. The best split is the split that maximizes loss reduction, which is calculated in the following way:

$$L_{split} = \left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{25}$$

The original paper by Chen and Guestrin (2016) introduced two regularisation hyperparameters, λ and γ. Additional regularisation parameter α, which has similar effect with lasso regularisation, is also included in XGB. With the α regularisation parameter the optimal weight ω_t^* becomes:

$$\omega_t^* = -\frac{\sum_{i \in I_t} g_i \pm \alpha}{\sum_{i \in I_t} h_i + \lambda}, \tag{26}$$

the simplified objective becomes:

$$\tilde{L}_m(q) = \sum_{t=1}^{T} \frac{\left(\sum_{i \in I_t} g_i \pm \alpha\right)^2}{\sum_{i \in I_t} h_i + \lambda} - \gamma T, \tag{27}$$

and equation 25 becomes:

$$L_{split} = \left[\frac{\left(\sum_{i \in I_L} g_i \pm \alpha\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i \pm \alpha\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i \pm \alpha)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{28}$$

With the squared error loss function, the optimal weight is ω_t^*:

$$\omega_t^* = \frac{\sum_{i \in I_t} \tilde{y}_{im} \pm \alpha}{\sum_{i \in I_t} 1 + \lambda}, \tag{29}$$

and the simplified objective is:

$$\tilde{L}_m(q) = \sum_{t=1}^{T} \frac{\left(\sum_{i \in I_t} -\tilde{y}_{im} \pm \alpha\right)^2}{\sum_{i \in I_t} 1 + \lambda} - \gamma T. \tag{30}$$

The impact of λ can be seen as the denominator of equations 29 and 30 is the number of instances in a node + λ. λ penalises calculated loss reduction and weight of nodes with a low number of instances. The numerator part of these equations consists of the sum of residuals or negative residuals and a regularisation term $\pm\alpha$. The regularisation term affects the weights ω_t^* and the corresponding loss reduction $\tilde{L}_m(q)$ values by shrinking or ignoring the values of the summation $\sum_{i \in I_t} g_i$. If $\sum_{i \in I_t} g_i$ is bigger than α, the sign is negative, if $\sum_{i \in I_t} g_i$ is smaller than negative α, the sign is positive, otherwise, the optimal weight and the corresponding loss reduction value for a given node are zero. So α-regularization pushes weights and loss reduction values towards zero and ignores values smaller than the threshold α altogether.

Overall, XGB works in a similar way to gradient boost. In each iteration, a tree is created by adding splits that maximise the split criterion L_{split} in equation 28. After a tree is created, it is pruned based on L_{split} criterion. If the L_{split} of a leaf branch is negative, the branch is pruned, until the L_{split} of all leaf branches are non-negative. The regularisation parameter γ creates a boundary that the first part of equation 28 needs to exceed to not be pruned.

Finally, weights ω_t^* are assigned to terminal nodes R_t. After each iteration the approximation is updated in the same way:

$$F_m(x) = F_{m-1}(x) + \eta f_m(x) = F_{m-1}(x) + \eta \sum_{t=1}^{T} \omega_{jm}^* 1(x \in R_{jm}) \tag{31}$$

In addition to learning rate η, XGB includes another technique that aims to prevent overfitting, which is called subsampling. In subsampling, only a subset of the original dataset is used for a certain part of the algorithm. Row subsampling can be done in each iteration m, while column subsampling can be done for each tree, each level of the tree, or each node. The column subsampling hyperparameters work in a multiplicative manner. For example, if subsample ratios for all three column subsampling values are 0.5,

the ratio of used columns becomes $0.5^3 = 0.125$. The subsampling techniques also speed up the computations.

XGB has two algorithms for finding split candidates. A basic exact greedy split finding algorithm enumerates all possible split points of all features and chooses the best split according to equation 28. With large datasets with continuous features, this becomes computationally expensive, as the number of possible split points becomes large. Chen and Guestrin (2016) also propose an alternative split-finding algorithm called the approximate algorithm, which proposes split points based on distributions of features. The approximate algorithm maps continuous features into buckets, where each bucket has a similar number of instances. The boundaries of these buckets then become candidate split points. This algorithm has two variants, global and local, where the global variant proposes all split points before building a tree, while the local variant re-proposes split points after each split.

In addition, XGB can handle missing values with sparsity aware split finding algorithm. In each candidate split with features that have missing values, the split criterion in equation 28 is calculated twice. Once with grouping missing values to the left node and once with grouping the missing values to the right node. The direction, which obtains higher loss reduction, is chosen as the default direction of missing values for the split candidate in question. The rest of the improvements over gradient boost introduced by Chen and Guestrin (2016) are related to the parallelizability of the algorithm.

4.2.4 Explainable boosting machine

EBM is a part of the InterpretML framework by Microsoft. EBM is an implementation of the GA²M algorithm introduced by Lou et al. (2013). GA²M is an improvement over GAM originally introduced by Hastie and Tibshirani (1986). GAM has the following form:

$$\hat{y} = g(E[Y]) = \beta_0 + \sum \phi_j(x_j), \tag{32}$$

where g is the link function that adapts the GAM to different situations such as regression and classification, and ϕ_j represents individual functions for each feature x_j. The identity link function is used for regression tasks.

Lou et al. (2012) compared GAMs to more complex models and found out that standard GAMs cannot always achieve the same accuracy as more complex models such as random forests. GA²M extends the GAM by adding pairwise interactions:

$$\hat{y} = g(E[Y]) = \beta_0 + \sum \phi_j(x_j) + \sum \phi_{j,k}(x_j, x_k), \tag{33}$$

where $\phi_{j,k}$ represent individual pairwise interaction functions for each involved pair of features x_j, x_k. By adding the pairwise interactions Luo et al. (2013) aimed to increase the accuracy while maintaining the inherent interpretability of standard GAMs.

GA²Ms are constructed in two stages. In the first stage the main effects $\sum \phi_j(x_j)$ for single features are fitted, and in the second phase, the most important pairwise interactions $\sum \phi_{jk}(x_j, x_k)$ are detected and fitted. Similar to gradient boosting the initial guess $F_0(x)$ is the average of observed values y_i:

$$F_0(x_i) = \hat{y}_0 = \frac{1}{n}\sum_{i=1}^{N} y_i, \tag{34}$$

which is also the intercept β_0.

In each subsequent step $\{F_m(x)\}_1^M$ weak individual trees $f_{jm}(x)$ are constructed for each feature j. First residual error $\tilde{y}_{ijm}$ for each instance of the previous prediction is calculated as the prediction target for the current step:

$$\tilde{y}_{ijm} = y_i - \left(F_{m-1}(x_i) + \sum_{k=1}^{j-1} \eta f_{km}(x_i) \right), \tag{35}$$

where $F_{m-1}(x_i)$ is the prediction of the previous round, and $\sum_{k=1}^{j-1} \eta f_{km}(x_i)$ is the sum of the predictions of trees created in the current round. Next a new tree $f_{jm}(x)$ that is limited to feature j, is constructed to predict the residual $\tilde{y}_{ijm}$. Similar to gradient boosting, the coefficients of the tree $f_{jm}(x)$, are the average residual errors $\tilde{y}_{ijm}$ in a terminal node. In each round, all features are cycled through, and after each round, the predictions are updated in the following way:

$$F_m(x) = F_{m-1}(x_i) + \sum_{j=1}^{J} \eta f_{jm}(x). \tag{36}$$

The learning rate η is typically considerably lower with EBM compared to gradient boosting or XGB. The low learning rate leaves room for the impact of each feature and makes the order that the features are cycled through irrelevant. These rounds are repeated until the maximum number of rounds is reached, or the stopping criterion is met. After that, individual functions $\phi_j(x_j)$ for each feature j, are created with trees $f_{jm}(x)$ that have been created throughout interactions:

$$\phi_j(x_j) = \sum_{m=1}^{M} \eta f_{jm}(x). \tag{37}$$

In the second stage functions for the pairwise interactions $\sum \phi_{jk}(x_j, x_k)$ are constructed. First, the most important pairwise interactions are detected with a FAST algorithm. Two sets S and Z, where S includes selected pairwise interactions and Z includes remaining pairwise interactions, are maintained. Small trees $f_{jk}(x)$ are created for each pairwise interaction in Z to predict residual error r of main effects:

$$r = y_i - F_M(x_i). \tag{38}$$

These small trees are limited to three splits, the first split is done with one of the features and both nodes created with the first split are split based on the other feature. For each of the four terminal nodes created, the weight is calculated as the average residual error in that node. The pairwise interaction that reduces residual error the most is selected and moved to set S. After, the residual error r is updated in the following manner:

$$r = y_i - F_M(x_i) - \sum_{(j,k)\in S} f_{j,k}(x_i), \tag{39}$$

where $\sum_{(j,k)\in S} f_{j,k}(x_i)$ is the output of the small trees of selected pairwise interactions. This procedure of creating trees for remaining pairwise interactions in Z, moving the best pairwise interaction to S, and updating the residual error r, is repeated until the size of S is equal to a hyperparameter that limits the number of pairwise interactions, or until none of the interactions in Z improves the accuracy of the prediction.

After detecting the most important pairwise interactions, the functions for the pairwise interactions $\sum \phi_{jk}(x_j, x_k)$ are constructed in a manner that is similar to the main effects $\sum \phi_j(x_j)$. The initial step $G_0(x_i)$ is the prediction of the model built so far:

$$G_0(x_i) = F_M(x_i). \tag{40}$$

In each round, the pairwise interactions in S are cycled through. For each pairwise interaction a tree $f_{(j,k)}$ limited to the two features is created to predict the residual error of the previous prediction. The weights for each terminal node are assigned as the average residual error of that node, and the predictions are updated. The update after one round is summarized by this:

$$G_m(x) = G_{m-1}(x_i) + \sum_{(j,k)\in S} \eta f_{(j,k)m}(x_i). \tag{41}$$

After M rounds, or after the stopping criterion is met, the predictive model is complete, and the individual functions for pairwise interactions are given by:

$$\phi_{j,k}(x_{j,k}) = \sum_{m=1}^{M} \eta f_{(j,k)m}(x). \tag{42}$$

EBM implementation of the GA2M algorithm includes additional techniques to increase accuracy or decrease the computational cost of the algorithm. To speed up the algorithm, similar to the approximate algorithm of XGB, EBM discretizes continuous features into bins based on quantiles, where each bin has a similar number of instances in them. These bins are used as candidate split points for finding splits when constructing trees f. This discretization is done separately for main effects and pairwise interactions. The maximum number of bins for detecting and fitting pairwise interactions $\phi_{j,k}(x_{j,k})$ is determined by the max interaction bins hyperparameter, while the maximum

number of bins for fitting main effects $\phi_j(x_j)$ is determined by the max bins hyperparameter.

To increase the accuracy of the algorithm, EBM allows bootstrap aggregating in two levels. On the outer level, data are subsampled without replacement to create samples, which involve only a part of the data. The left-out part of the dataset is used for the early stopping criterion. The number of subsamples created is equal to the outer bags hyperparameter. Each bag works independently with the expectation of pairwise interaction detection, where each bag ranks the pairwise interactions separately, but the final pairwise interactions are selected for all the bags based on the average rank of pairwise interactions across the bags. For the final shape functions of main effects and pairwise interactions, the outputs are the average outputs of shape functions for each bag. This outer bag procedure is parallelized across the number of cores in a machine, so it doesn't necessarily affect the computational cost of the algorithm.

In the inner level, datapoints are sampled with replacement for each tree f that is constructed. The average output across inner bags is the final output of each tree. The number of inner bags is controlled with hyperparameter inner bags. Bagging in the inner level increases computational cost.

4.3 Tree-structured Parzen Estimator

Due to the high number of tuneable hyperparameters in XGB and the slowness of EBM, the traditional grid search method for optimizing hyperparameters was determined to be too slow. Instead, an implementation introduced by Bergstra et al. (2015) of the Tree-structured Parzen Estimator (TPE) algorithm, introduced by Bergstra et al. (2011), is used.

Putatunda and Rama (2018) compared the TPE algorithm with grid search and random search for tuning hyperparameters of XGB and found that TPE is faster than grid search and more accurate than random search. TPE is one of the suitable hyperparameter optimization methods for large hyperparameter spaces mentioned by Yang and Shami (2020).

TPE algorithm belongs to the group of Sequential Model-Based Global Optimization (SMBO) algorithms. SBMO algorithms iterate between evaluating the true function F and approximating the surrogate function given observation history H. Observation history consists of pairs of loss $F(x^{(i)})$ of the true function and parameters $x^{(i)}$ that were used to obtain the loss.

The point x^* that maximizes the surrogate function is proposed for the next evaluation of the true function. SBMOs differ in the way, in which the surrogate function is modelled and the next candidate point x^* is obtained. The TPE algorithm, as well as other some other SBMO algorithms, obtain x^*

by optimising Expected Improvement (EI) criterion. EI is defined by the following formula:

$$EI_{y^*}(x) := \int_{-\infty}^{\infty} \max(y^* - y, 0) p_M(y|x) dy\,, \tag{43}$$

where $p_M(y|x)$ refers to the surrogate model of the true function, and y^* is some threshold, which's set up changes from one algorithm to another.

Configuration space, where the best values for hyperparameters are searched, is defined in Hyperopt with different distributions, such as uniform distribution between thresholds or gaussian distribution. Also, the representation of discrete and conditional variables is possible. The tree-structure of TPE refers to the ability to represent hyperparameters in a tree-like structure, in a way that some hyperparameters are only relevant when other hyperparameters have a certain value. TPE transforms these configuration spaces into some variations of truncated Gaussian distributions. The hyperparameters for the first 20 iterations are randomly sampled from the prior distributions, and the following iterations are done based on the suggestions of the TPE algorithm.

Instead of directly modelling $p(y|x)$ TPE models $p(x|y)$ and $p(y)$. The modelling of $p(x|y)$ is done by forming two densities $l(x)$ and $g(x)$:

$$p(x|y) = \begin{cases} l(x)\ if\ \ y < y^* \\ g(x)\ if\ \ y \geq y^* \end{cases}, \tag{44}$$

where $l(x)$ is created with observations $\{x^{(i)}\}$, where obtained loss $F(x^{(i)})$ was smaller than the threshold y^*. In a sense, density $l(x)$ represents good observations and density $g(x)$ represents bad observations. The TPE algorithm sets y^* to be some quantile γ of the observed loss values:

$$\gamma = p(y < y^*). \tag{45}$$

Bergstra et al. (2011) show that EI can be approximated in the following way:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy \propto \left(\gamma + (1-\gamma)\frac{g(x)}{l(x)}\right)^{-1}. \tag{46}$$

The approximation of EI can be maximized by maximizing $\frac{l(x)}{g(x)}$, which means choosing values x that have a high probability in $l(x)$ and a low probability in $g(x)$.

Separate densities are formed for each tuneable hyperparameter. With continuous search spaces, the formed densities are a mixture of prior and gaussian distributions centred at each point $x^{(i)}$ in search history, with a standard deviation equal to the distance to the greater of its distances to nearest neighbours in each direction. For search spaces with uniform priors over intervals (a, b), the gaussian mixture is limited between a and b. For discrete variables with N different possible values, the prior distribution is a vector of N probabilities p_i, and the posterior probabilities of each value are proportional to $Np_i + C_i$, where C_i represents the number of occurrences of value i in the search history.

The weight of observations is scaled in TPE, as the 25 most recent observations have weight 1. The observations before that have evenly spaced weights over an interval $(\frac{1}{n}, 1)$, where n is the number of iterations.

In each iteration, the TPE algorithm utilizes search history H to form densities for the good group $l(x)$ and bad group $g(x)$, and suggests new values for x, which maximize the density $\frac{l(x)}{g(x)}$. The suggested values are used to obtain new loss value $F(x^{(i)})$ and the search history H is updated. These iterations are repeated until some stop criterion, such as time or number of iterations is reached.

The approximation of EI is illustrated in Figure 1, which shows loss values of individual trials, which are represented with dots, on the right y-axis and densities of the different distributions on the left y-axis. This illustrative example has 10 searches in total, and the searches are grouped based on loss values to create two densities l(x) and g(x). Individual searches that are used to form l(x) are the most accurate ones, while the rest of the searches are used to form g(x). The value of the learning rate, which corresponds to the peak of the EI curve, would be the suggested value for the next search.

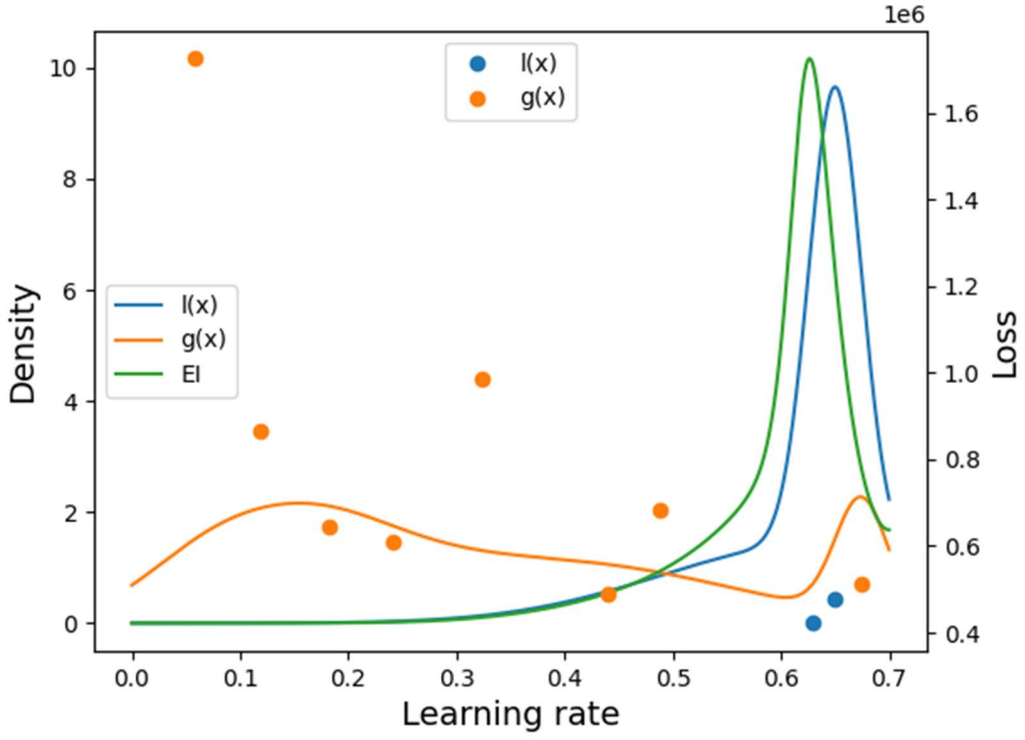

Figure 1: Approximation of EI in TPE

4.4 Hyperparameter optimisation decisions

Ghatnekar and Shanbhag (2021), and Mrsic et al. (2020) reported significant increases in property valuation accuracy when comparing XGB with optimized hyperparameter values to XGB with standard hyperparameter values. The tuned hyperparameters and their search spaces are reported in Table 3. For all XGB models, the TPE algorithm was performed for 300 rounds.

Table 3: XGB hyperparameter search spaces used in TPE

Hyperparameter	Search space
max depth	quniform(3,18,1)
gamma	uniform(1,1 000)
alpha	uniform(1,1 000 000)
lambda	uniform(1,1 000 000)
colsample by tree	uniform(0.5,1)
colsample by level	uniform(0.5,1)
colsample by node	uniform(0.5,1)
subsample	uniform(0.5,1)
min child weight	quniform(0,20,1)
number of estimators	quniform(100,1 000,1)
eta, learning rate	uniform(0.001,0.9)

The creators of EBM claim that the default hyperparameter values should work well in all situations and that there is no need for hyperparameter tuning with EBM. According to Nori et al. (2021), who utilized a slight variation of standard EBM, the default parameters work well on a variety of datasets. However, hyperparameter tuning was performed for EBM models. The tuned hyperparameters and their search spaces are reported in Table 4. For all EBM models, the TPE algorithm was run for 100 rounds. The number of rounds is lower compared to XGB because EBM is considerably slower, the number of tuned hyperparameters is lower, and the expected significance of hyperparameter tuning based on literature is smaller.

Table 4: EBM hyperparameter search spaces used in TPE

Hyperparameter	Search space
max bins	quniform(2,500,1)
number of interactions	quniform(1,50,1)
outer bags	quniform(1,20,1)
inner bags	quniform(0,20,1)
min samples per leaf	quniform(0,20,1)
learning rate	uniform(0.001,0.1)
max leaves	quniform(2,20,1)
max interaction bins	quniform(2,64,1)

Uniform prior was used for search spaces, as there is no clear information on what type of values each hyperparameter could be expected to have. Quniform refers to uniform search space where possible values are spread across some interval. For example, if the interval is 1, possible values are integers that are in the defined range. With most hyperparameters, if the best results were obtained with a hyperparameter value that was close to the edge of its initial search space, that search space was enlarged and the hyperparameter search was repeated. With the number of pairwise interactions hyperparameter in EBM the search space was not enlarged, because the least important pairwise interactions did not have a significant impact on the predictions and increasing the number of interactions would slow down the hyperparameter search too much.

For each model, a 10-fold cross-validation was used to calculate the loss value for TPE. Data were split ten times into a training part that is 9/10 of the training data and a validation part that includes 1/10 of the training data. The average RMSE across the folds is used as a criterion that is optimized with TPE. The hyperparameters that were used to obtain the lowest RMSE for a single model were then used as the final hyperparameters for that model.

5 Results

In this section, the results of chosen methods and approaches are compared, and the interpretability of an EBM model is highlighted. The graphs that are used to highlight the interpretability of EBM are produced with the InterpretML framework, where EBM is implemented.

5.1 Model comparison with the global approach

The accuracies of both methods with the global approach are shown in Table 5. The total accuracy refers to the accuracy of a model across all municipalities, which is the objective that is minimized in model training. In addition, the accuracies in each municipality are also reported to compare to local approach more easily and to identify, which municipalities work well with the global approach. RMSE for training data is also reported to show the level of overfitting in the models. The municipalities are sorted in descending order based on how many transactions from each municipality are included in the dataset.

Table 5: Accuracies of different models with the global approach

Method	XGB	XGB	XGB	EBM	EBM	EBM
Dataset	Train	Test	Test	Train	Test	Test
Metric	RMSE	RMSE	MDAPE	RMSE	RMSE	MDAPE
Copenhagen	274 184	433 661	7.0	417 672	523 902	9.1
Aarhus	267 311	386 528	6.5	458 607	444 351	9.1
Aalborg	232 561	398 069	10.6	411 962	506 057	14.0
Odense	256 405	638 692	12.1	438 311	758 965	15.7
Frederiksberg	291 882	487 860	7.4	442 800	574 156	10.1
Roskilde	293 817	635 932	13.9	486 195	746 694	14.0
Total	267 665	466 070	8.1	435 180	558 342	10.6

The results reported in Table 5 show that the trained XGB model predicts property prices more accurately than the trained EBM model with the global approach. MDAPE and RMSE are better in total and in all the municipalities except for Roskilde, where the MDAPE of both models are similar, while RMSE is still better with XGB.

The function of EBM that captures the impact of municipalities is shown in Figure 2. Municipality 1 refers to Copenhagen, where otherwise similar properties have the lowest valuations. In municipality 3, which refers to Aarhus, valuations are highest, while there are no sizable differences between the other municipalities. The low accuracy of EBM, when compared to XGB, could stem from its inability to model local differences across the municipalities. The differences across municipalities in EBM are mostly captured with the single values shown in Figure 2 that are added to the

additive function, while XGB can capture more complex relationships that could more accurately represent the uniqueness of individual municipalities.

Figure 2: Effect of the municipality feature in the EBM model

5.2 Model comparison with the local approach

The accuracies of both methods with the local approach are shown in Table 6. In addition to the individual accuracies of single municipalities, which is the target of learning algorithms, the total accuracy across all municipalities is reported to allow comparisons to the global approach.

Table 6: Accuracies of different models with the local approach

Method	XGB	XGB	XGB	EBM	EBM	EBM
Dataset	Train	Test	Test	Train	Test	Test
Metric	RMSE	RMSE	MDAPE	RMSE	RMSE	MDAPE
Copenhagen	225900	413791	6.9	359942	432480	7.4
Aarhus	201725	420565	6.0	382932	422635	7.2
Aalborg	184177	335179	8.5	322846	446716	16.0
Odense	180719	547984	9.2	336175	684175	8.7
Frederiksberg	237390	470150	6.8	320811	469094	6.6
Roskilde	230219	672576	14.7	387352	653793	12.6
Total	211221	444414	7.4	356389	485876	8.5

With the local approach, the accuracy of EBM increased significantly compared to the global approach, while the accuracy of XGB also improved slightly. The total accuracy of XGB is still better, while the difference in both RMSE and MDAPE is significantly smaller than in the global approach. When individual municipalities are compared, there is no clear difference between the two methods with the local approach. In some municipalities, the out-of-sample accuracy of EBM is better, while in others, the out-of-

sample accuracy of XGB is better. Except for Aalborg, the differences are not very big. Interestingly in Odense, the RMSE of XGB is significantly better, while the MDAPE of EBM is better.

While the test accuracies of the two methods are mostly similar, the training RMSE is significantly lower with XGB, which makes it seem that the XGB models were overfitted, even with the hyperparameter optimization and 10-fold cross-validation procedure that aims to reduce overfitting. The difference in terms of RMSE between the training set and the test set is smaller with EBM models, which makes EBM as a method seem more robust.

5.3 Global interpretability of EBM

The global interpretability of EBM is highlighted by presenting the built Copenhagen EBM model with the local approach. Figure 3 showcases feature importances that are calculated based on the average absolute effect that each feature has on the predictions. The Copenhagen EBM model has 43 features with individual functions and 50 functions that capture pairwise interactions. These feature importance measures in Figure 3 show, which of these functions has the biggest effect on the predictions on average, and it will be utilized in this thesis as only functions of some of the most important features will be presented. In the single feature function graphs, the grey areas represent uncertainty that comes from the difference in functions across different outer bags in the training phase of the algorithm.

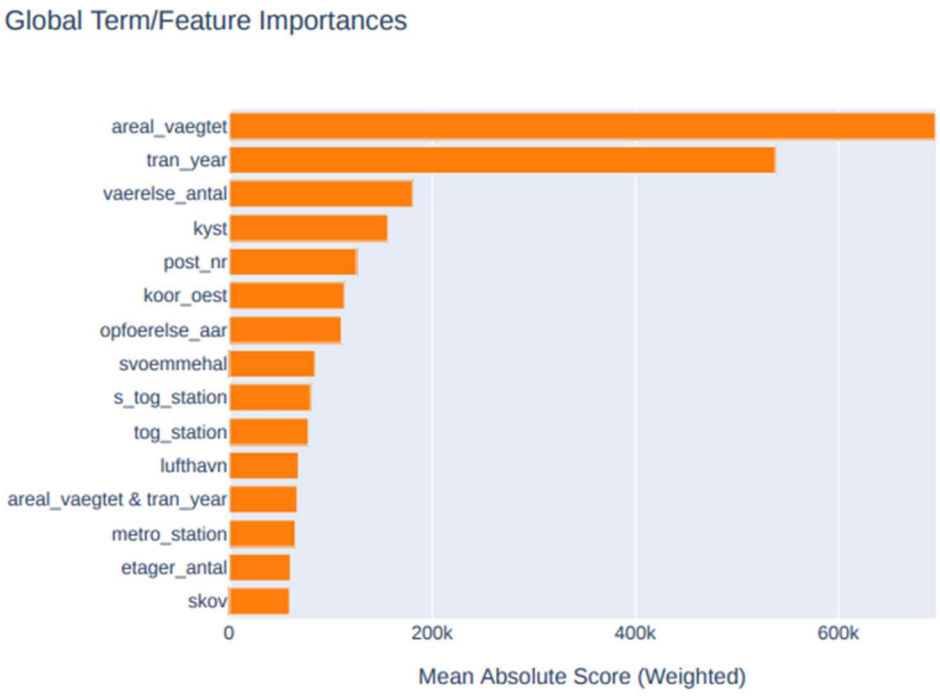

Figure 3: Feature importance of EMB Copenhagen model

In addition, distributions of features are presented under most of the functions to get an idea, of how the values of each feature are distributed.

Figure 4 shows the effect of the weighted floor area of a property. An increase in weighted floor area increases the predicted price and the increase from each added square meter gradually decreases. There are not a lot of properties, where the weighted floor area is more than 200 square meters, which could explain the inconsistency of the function with high values of the weighted area. There can be close to a 3 million DKK difference in predicted price just based on the weighted floor area.

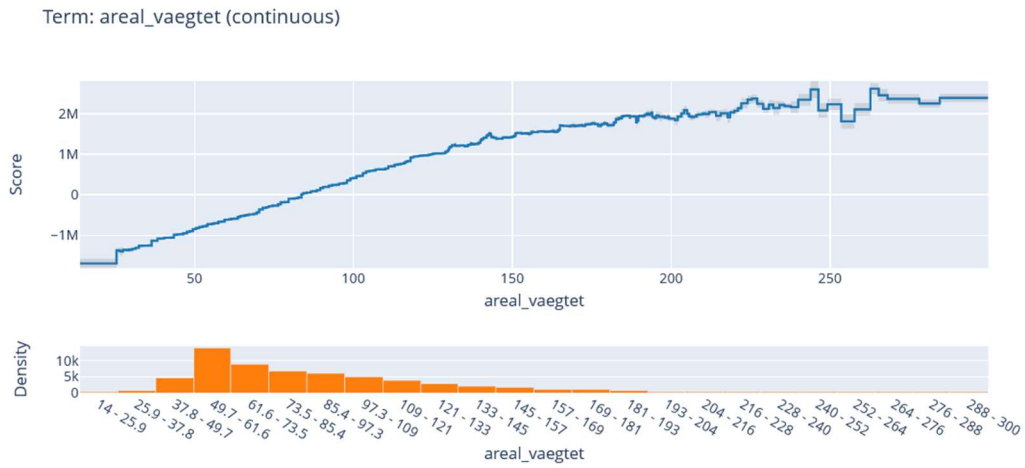

Figure 4: Single feature function of the weighted floor area

Figure 5 shows the function for the year of the transaction. The predicted price of properties grows each year, while the differences between years vary. According to this function, the valuation of a property would be 2 million DKK more in 2021, compared to 2010.

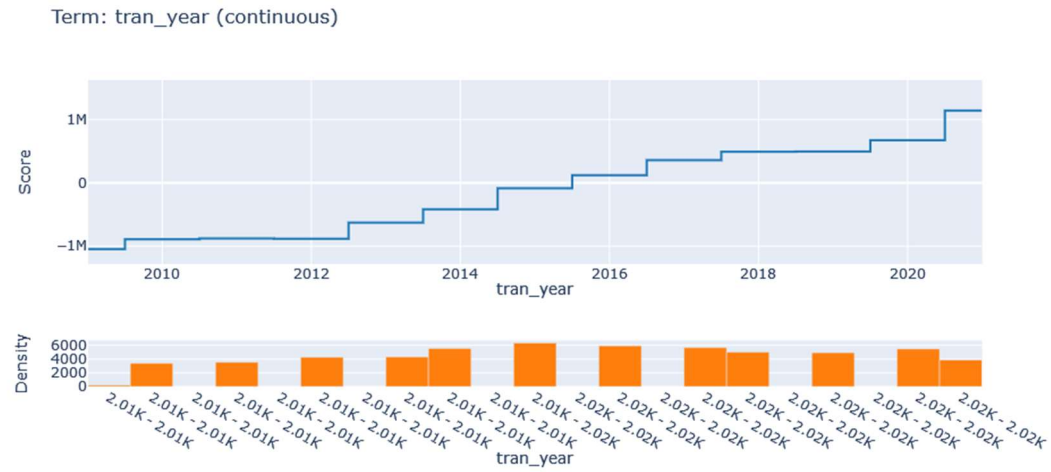

Figure 5: Single feature function of the year of the transaction

Figure 6 shows the function of the number of rooms, which shows that each additional room of up to 8 rooms increases the predicted price. The difference in predicted price between 1 and 8 rooms is close to 1 million DKK.

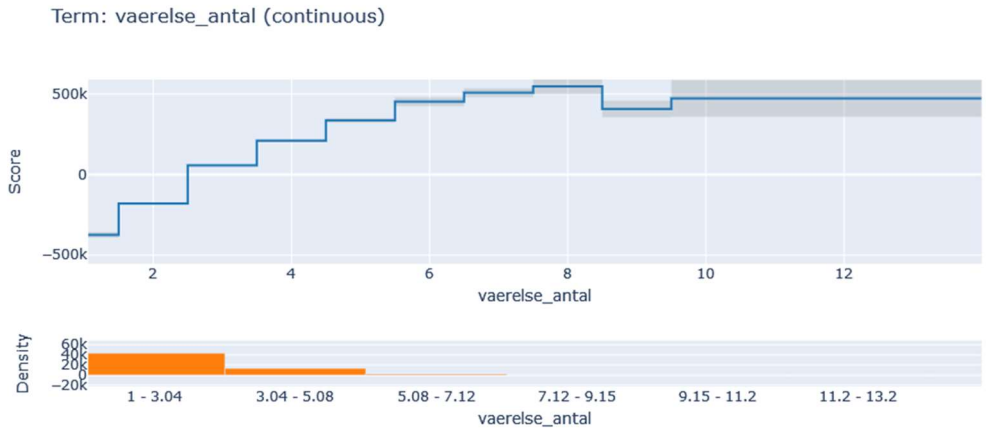

Figure 6: Single feature function of number of rooms

The function of feature distance to the nearest coast is shown in Figure 7. Properties near a coast have higher valuations, the effect of distance to the nearest coast diminishes fast as the distance increases, and after 2 kilometres the effect stays similar until 10 kilometres of distance. Being right next to the coast can increase the predicted price by close to 1 million DKK compared to distances, where the function plateaus.

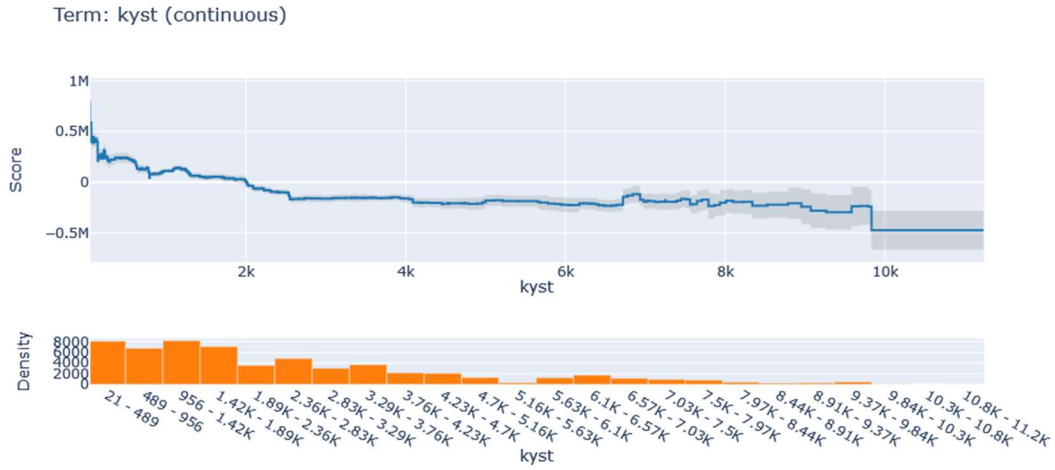

Figure 7: Single feature function of distance to the nearest coast

The function of the postal code is shown in Figure 8. There are differences up to 0.5 million DKK between different post number areas. The biggest effect is on areas, where there are not many property transactions. In areas, where most of the transactions happen, there is a smaller effect.

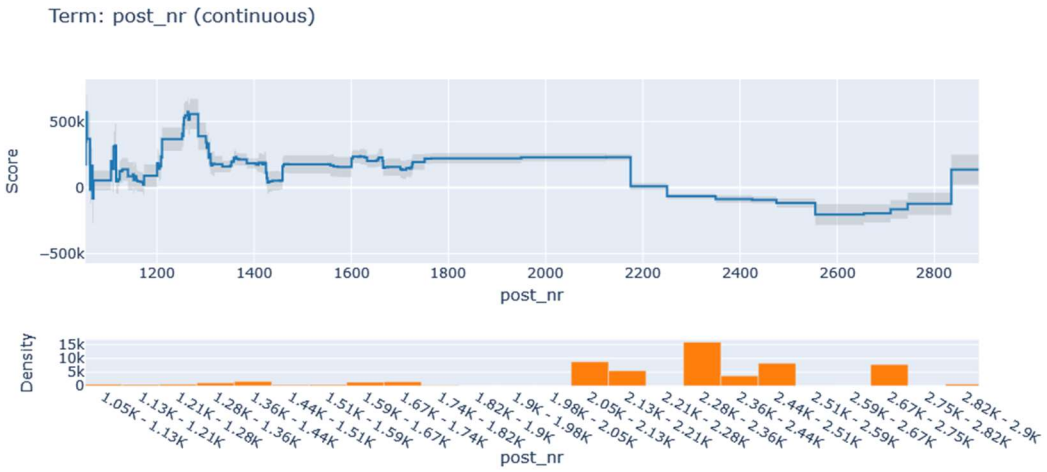

Figure 8: Single feature function of postal number in area

Figure 9 shows the function of the ERTS89 x-coordinate of the property. The function increases the predicted prices of properties in eastern parts of Copenhagen while decreasing the predicted prices of properties in western parts of Copenhagen.

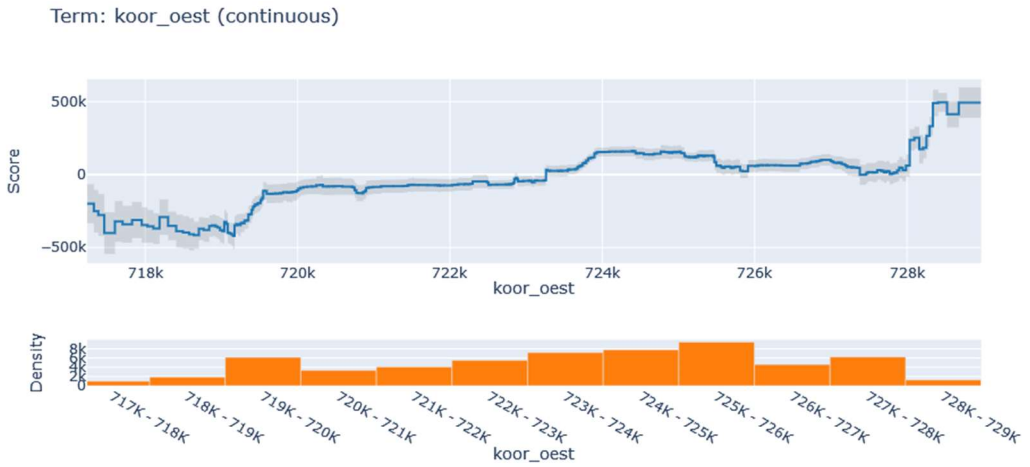

Figure 9: Single feature function of ERTS89 x-coordinate of property

The function of the built year is shown in Figure 10. Properties in newly constructed buildings have higher predicted prices. The function has a U-shape in the last 100 years, where the price of properties in buildings built between 1970 and 1980 is the lowest, and the predicted price is higher in older buildings.

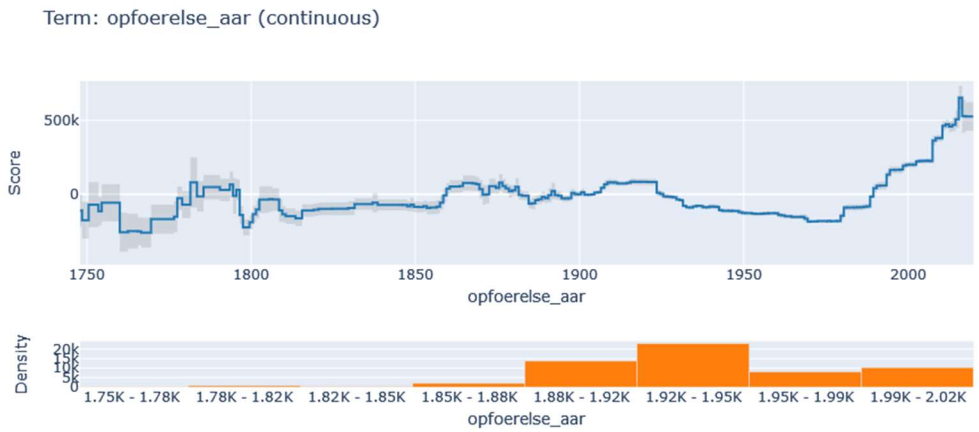

Figure 10: Single feature function of built year

The function of distance to the nearest swimming hall is presented in Figure 11. Interestingly, properties with bigger distances to the nearest swimming hall have a higher predicted price, compared to properties that are closer to swimming halls.

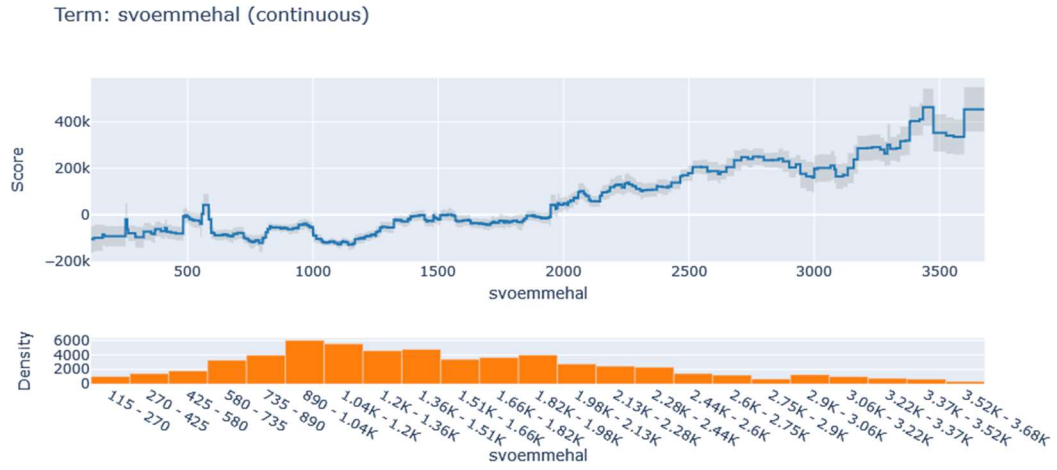

Figure 11: Single feature function of distance to the nearest swimming hall

Figure 12 presents the function for distance to the nearest train station. Low distances to train stations have higher predicted prices on the properties.

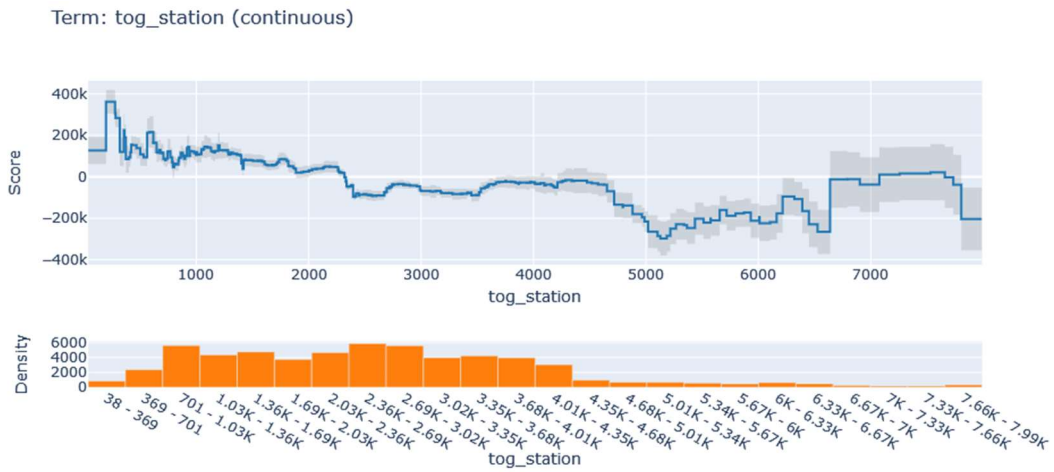

Figure 12: Single feature function of distance to the nearest train station

The function for distance to the nearest s-train station is shown in Figure 13. Very small distances seem to have a negative effect on predicted prices, while medium distances have close to zero effect on the predicted price, and longer distances decrease the predicted price.

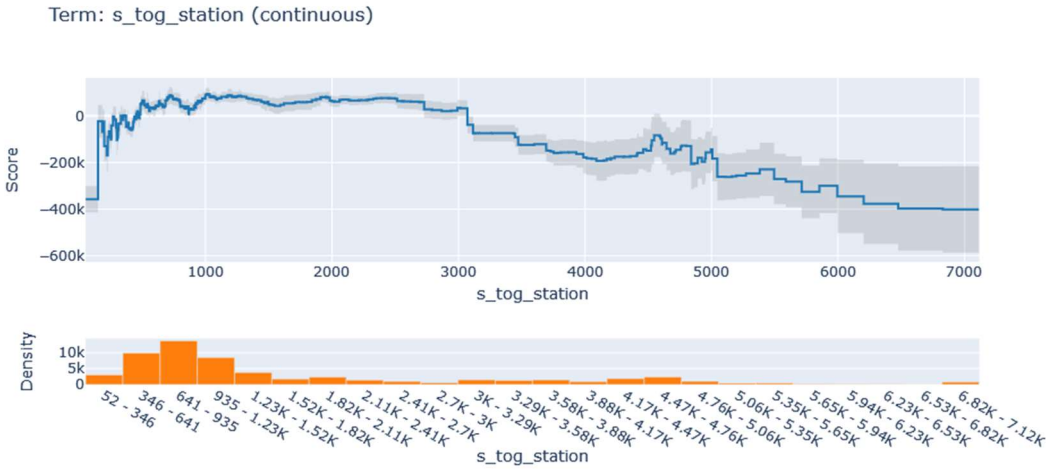

Figure 13: Single feature function of distance to the nearest s-train station

The function of distance to the nearest airport is shown in Figure 14. Property being too close to airports decreases the predicted price, and the biggest positive effects are between 10 and 18 kilometres in distance. Higher distances have low predicted prices.

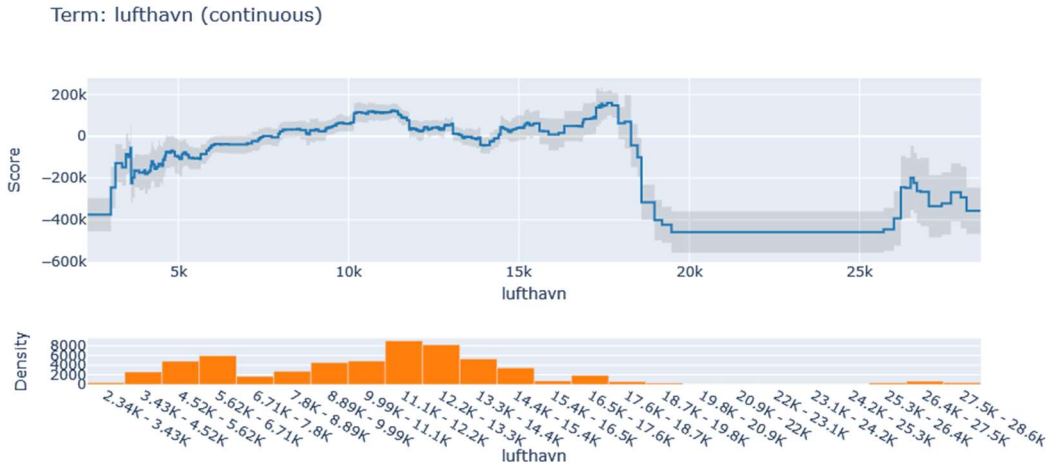

Figure 14: Single feature function of distance to the nearest airport

The most important function that includes two features is shown in Figure 15. The two features are the year of the transaction on the y-axis and the weighted floor area of a property in square meters on the x-axis. The function shows that in more recent years the predicted prices of properties with high weighted floor area are higher, while closer to 2010 the predicted prices of smaller properties are higher. When the effect of this function is considered together with the effect of transaction year function in Figure 5 it can be understood better. The average property has risen 2 million DKK in value, the value of bigger properties has grown nearly 2.5 million DKK, while the value of smaller properties has grown roughly 1.5 million in the same timeframe of 10 years.

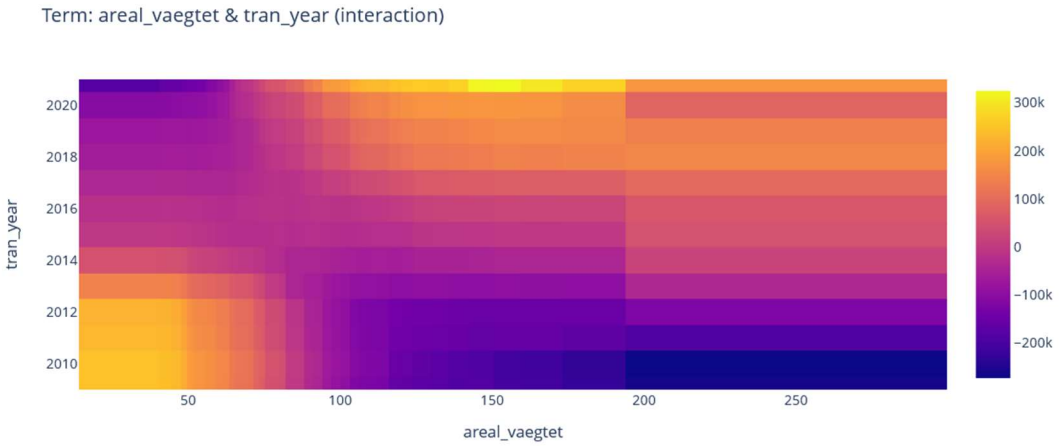

Figure 15: Two feature function with weighted floor area on the x-axis and transaction year on the y-axis

5.4 Local interpretability of EBM

In this part, the local interpretability of EBM is presented through individual predictions made by the local approach Copenhagen model with the test data. Four example transactions from different price ranges were chosen for presentation. The intercept in each prediction is the average transaction price in the training data. The rest of the explaining factors are based on the functions of individual features and pairwise interactions.

Figure 16 shows the local explanation of the first example transaction. The high prediction is a result of the relatively large weighted floor area and the novelty of the building. Also, the number of rooms and pairwise interactions of the previously mentioned features increase the predicted price significantly.

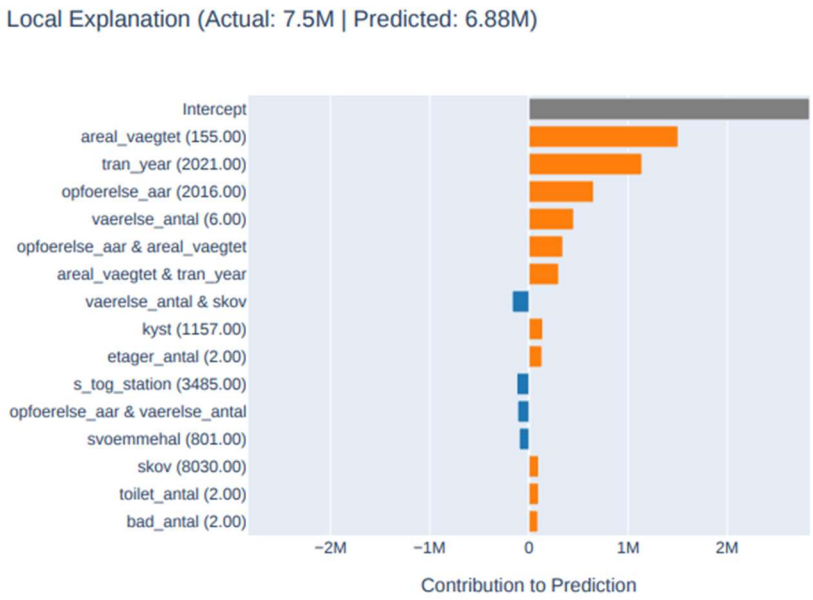

Figure 16: Explanation of the first example prediction

The local explanation of the second example transaction is shown in Figure 17. The predicted price stems mostly from the intercept and transaction year. The novelty of the building also increases the predicted price.

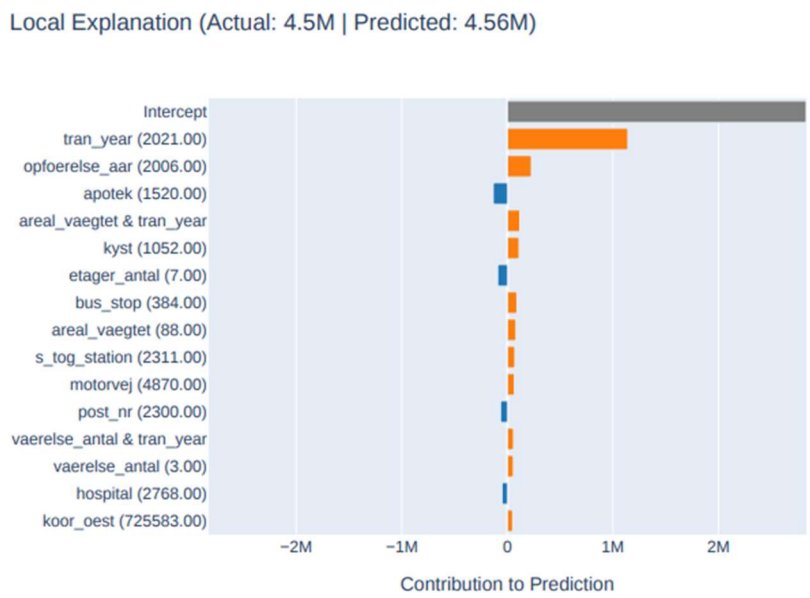

Figure 17: Explanation of the second example prediction

Figure 18 shows the third example transaction. The baseline of the predicted price, which comes from intercept and the year of the transaction, is decreased due to the relatively small weighted floor area.

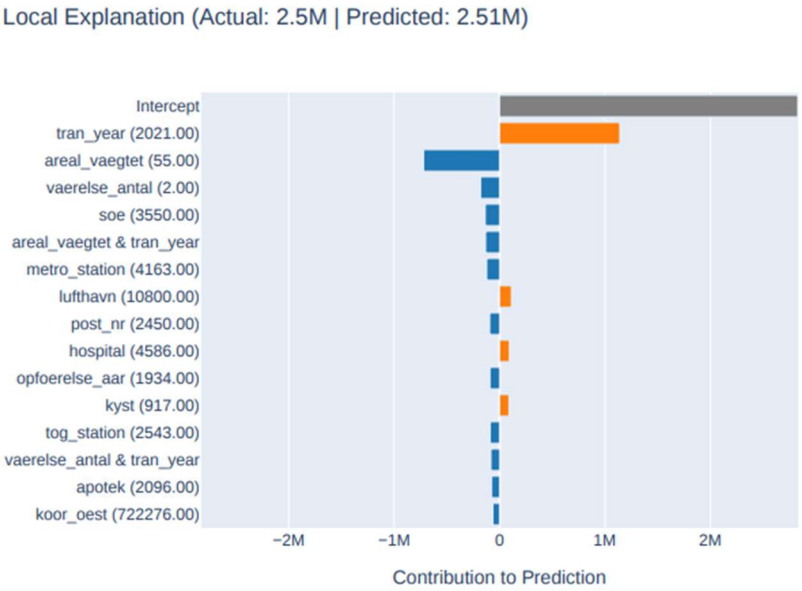

Figure 18: Explanation of the third example prediction

The final example transaction is presented in Figure 19. In the final example, the predicted price is mainly decreased due to the small size of the property, the low number of rooms, and the western location.

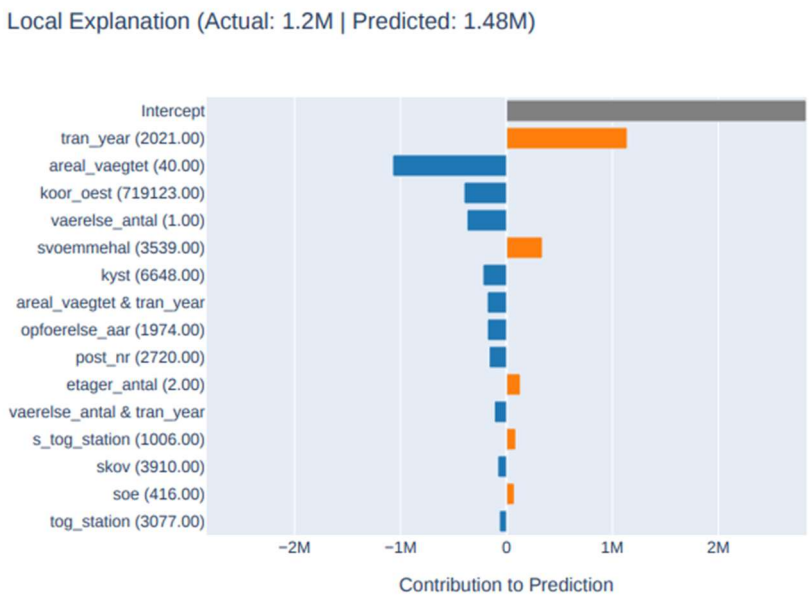

Figure 19: Explanation of the fourth example prediction

As the intercept and the year of the transaction are the same in each property, the biggest differences in predicted price across the properties stem from features that reflect the size of the property and the built year feature. The impact of the rest of the features is small in most of the examples, which aligns with the feature importance calculations in Figure 3.

6 Conclusion

6.1 Discussion of the results

This thesis aimed to analyse the feasibility of interpretable machine learning in property valuation and to produce an interpretable machine learning model that could be compared to the property valuation model used in Nordea. In the literature review, the benefits of interpretability in machine learning were discussed and the best-performing black-box method and promising interpretable method were identified. The chosen methods, XGB and EBM were compared in the empirical part of the study.

6.1.1 Benefits of interpretability in property valuation

The benefits that could be achieved with interpretable machine learning in property valuation were examined in the second section. As the EBM model is highly interpretable, it could make a difference, if the lack of interpretability has impeded the adoption of complex machine learning methods. Interpretability of EBM could make it seem more plausible than more complex machine learning methods because it is easier to verify and trust the model, as the impact of each feature is understandable. In addition, it is possible to comply with the requirements of GDPR, as providing meaningful information of the logic involved can be done, when there are individuals that would be affected by the predictions made by an EBM model.

There are additional benefits that interpretability might bring. Even though the feature functions of EBM do not imply causal relationships between the features and prediction price, it is possible to gain some information on the relationships between the features of properties and property prices. Also, interpretability can help the model-building progress. For example, in the early stages of this thesis, some inconsistencies in the data were found after looking at the feature functions of an EBM model. Those inconsistencies might have not been found with more complex machine learning models.

6.1.2 Accuracy-interpretability trade-off

The chosen methods were compared with two different modelling approaches. In the global approach, where data from all of the six included municipalities were used to train a single model for both methods, the XGB was considerably better overall. With the global modelling approach, there is a clear trade-off between accuracy and interpretability.

With the local modelling approach, where individual models were trained for all the included municipalities, the difference between the two methods

is considerably smaller. While XGB still had better accuracy overall, the interpretable EBM predicted property prices more accurately in some of the municipalities, and the differences in accuracy between the two methods were very small in most municipalities. With the local modelling approach, the trade-off between accuracy and interpretability is small overall, and non-existent in some of the municipalities.

6.1.3 Interpretability of EBM

The interpretability of the local approach Copenhagen EBM model was highlighted by presenting the most important functions of the model and some example predictions. With the functions, it is possible to infer, how the EBM model makes predictions.

In EBM, the value of a function that corresponds to the value of a feature or values of two features is just added to the final prediction. Because of this, the impact that single features have is easy to understand. Due to the high number of features and functions based on single features or pairs of features, global, holistic interpretability is impossible to achieve. However, global interpretability on a modular level is possible to achieve, as the impact of each individual function is understandable.

The functions that are trained in the EBM do not imply a causal relationship between the features and predicted price as many of the features can be correlated and the impact of one feature can be captured in the function of another feature. For example, the graph in Figure 6 shows that an increase in the number of rooms increases the predicted price, but it might be possible that when comparing two otherwise similar properties, the property with more rooms does not seem more valuable to a possible buyer. This might be a result of correlated features and EBM fitting all the single-feature functions simultaneously.

There are a lot of features included that in some way reflect the location of a property, and according to Mayer et al. (2022), there is typically collinearity among these types of features. For example, some post number areas that have highly valued properties, might not have a significant impact through the post number function, and the impact could be included through the distance to coast function or the distance to city centre function. Also, the U-shape of the built year could be explained with location, as some older buildings might have better locations than buildings that were built later.

The explanations of example predictions show, how individual predictions can be explained. The most important factors are easily identifiable, and it is easy to understand, what type of impact each function has. If the effect of a function does not make sense initially, it is possible to refer to the global explanation of that function to understand, why it impacts the prediction in that way. It is possible to achieve local interpretability with EBM.

6.2 Contributions

The main contributions of this thesis come from the comparing accuracies of the two methods. With the two chosen methods, there is a trade-off between interpretability and accuracy, as the interpretable EBM method does not produce as accurate predictions as the less interpretable XGB method. Even though the XGB method is more accurate, the difference in terms of accuracy is not very big, and the EBM method with the local approach is more accurate in some of the municipalities that were included in the study. This is one of the first studies that use EBM in property valuation, and the first one that compares XGB and EBM in property valuation.

Another contribution of this thesis comes from the comparison of accuracies between local and global approaches. The accuracy of the XGB method increases slightly when individual models are trained in each municipality, while with EBM, the accuracy increases significantly. The EBM method benefits more from localized model training, which might be a result of the relatively simple and additive structure of EBM.

This thesis contributes to interpretable property valuation literature by highlighting the interpretability of EBM. In this thesis, the interpretability of the EBM is highlighted by displaying the most important single feature and two feature functions, which shows how easy it is to interpret EBM models. In addition, the local interpretability of EBM is highlighted by showing example explanations of predictions. With these explanations of single predictions, it is easy to understand, why the prediction was made, which could be useful, when companies looking to adopt machine learning methods consider, how the requirements of GDPR can be fulfilled. In previous interpretable property valuation literature only Hurley and Sweeney (2022) have interpreted interpretable models that were used in property valuation, as they included three graphs that highlight how features affect the price per square meter in their GAM approach. In this thesis, the prediction target is the transaction price of a property, which can be even more interpretable, because it shows, how features affect the price of a property directly.

6.3 Limitations and future research

This thesis aimed to find out, if there is an accuracy-interpretability trade-off in property valuation, by comparing one proven and previously well-performing machine learning method to a candidate interpretable method. Several limitations need to be addressed and offer possibilities for future research.

One of the aims of the literature review was to identify the best possible machine learning method that would be compared to the interpretable method, and while earlier property valuation literature supports the XGB decision, other methods could be even more accurate, which could increase

the accuracy-interpretability trade-off. Also, the difference in accuracies between train RMSE and test RMSE indicates, that there might be room for improvement, with the optimization of XGB. Similarly, there might be some room for improvement when it comes to the choice or design of the interpretable method that could decrease or remove the accuracy-interpretability trade-off. There might be other methods that are more accurate, yet interpretable that have not been used in property valuation previously or have not been used optimally. In future research, including more interpretable methods and more benchmark machine learning methods, as well as implementing XGB better, could help to gain an even more accurate picture of the accuracy-interpretability trade-off in property valuation.

Another limitation of this thesis, which affects both XGB and EBM, is that there are no features that represent the condition of a property, which could be a key factor for possible buyers. The built year feature and the year of most recent remodelling are the only features that could be connected to the condition of a property, but they are only indicative when it comes to the condition of a property. Having additional data that would represent the condition of a property more accurately, could increase the accuracy of both methods. If it would be possible to obtain data that represents the condition of properties, determining how much more accurate these methods could be and what the differences in terms of accuracies would be with the additional data, could be an interesting future research topic.

This thesis has some limitations when it comes to generalisability. The data that was used was limited in different ways, which affect the generalisability of findings. The data only includes transactions from six big municipalities of Denmark, and the comparative accuracies of the chosen methods and the accuracy-interpretability trade-off could be different in more rural areas of Denmark or other countries.

The training data included transactions of owner-occupied properties, which also limits the generalisability of the findings, as the findings do not apply to other types of properties, where the value of properties stems from different factors. Also, when it comes to owner-occupied properties specifically, the upper limit on the price of properties to be included limits the generalisability of the findings, as the used methods might not be similarly capable of predicting prices of more expensive properties.

The generalisability of these findings over different periods can also be problematic. In this thesis, the transactions that were used in the test set were from the last quarter of 2021, and the train set used data between the years 2009-2021. It is possible that changes in external factors, such as interest rates, could affect transaction prices, and there might be differences in how well the chosen methods can capture these changes that happen over time.

These generalizability issues of this thesis provide ideas for future research. It would be interesting to see how the two methods compare with

data from different periods, other countries, more rural areas, properties from different price ranges, or properties with different ownership types and use.

The accuracy of the EBM model could be improved if the EBM feature fitting procedure was changed. For example, it might make more sense to exclude the x- and y-coordinates from the single feature fitting procedure and include them as a two-feature function instead. It is likely that possible buyers value the location of a property instead of its x- and y-coordinates separately. However, even if this was done, the impact of the function involving the two coordinates might be low, because all the two-feature functions are fitted after the single-feature functions in EBM. Also, as all the single feature functions are fitted simultaneously, it can be possible that a feature captures an impact that would be more accurately represented by another feature. For example, if the function for the number of rooms was fitted after the function for the weighted area, it could result in more accuracy and a different function for both of the features. The standard feature fitting procedure of EBM is strict in this sense as it only allows fitting single-feature functions first and either searches for two-feature functions or fits pre-determined two-feature functions after that. This might restrict the EBM models in terms of accuracy, and modifications that would allow more flexibility in terms of feature fitting order could result in more accuracy.

The interpretability of the built EBM model could also be improved. The previously mentioned exclusion of coordinates from the single-feature functions could also increase the interpretability of the model, as it could be more understandable to have one two-feature function for location instead of two single-feature functions. Also, potential problems that could arise, when all single feature functions are fitted simultaneously, might affect the interpretability of the model. It could make more sense if the impact of a certain factor, the size of the property, for example, would be captured in one function instead of several functions.

In addition, the vast number of features and feature interactions decrease the interpretability of EBM. It could be possible to exclude some of the most insignificant and feature interactions, without losing accuracy, which could improve the interpretability of EBM. These ideas for improvements of EBM offer a possibility for further research to determine if the interpretability-accuracy trade-off could be smaller and if the interpretability of EBM could be increased without losing accuracy.

References

Abidoye, R. B., & Chan, A. P. (2018). Improving property valuation accuracy: a comparison of hedonic pricing model and artificial neural network. *Pacific Rim Property Research Journal*, *24*(1), 71-83.

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access*, *6*, 52138-52160.

Alexandridis, A. K., Karlis, D., Papastamos, D., & Andritsos, D. (2019). Real estate valuation and forecasting in non-homogeneous markets: a case study in Greece during the financial crisis. *Journal of the Operational Research Society*, *70*(10), 1769-1783.

Alvarez Melis, D., & Jaakkola, T. (2018). Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, *31*, 7775-7784.

Antipov, E. A., & Pokryshevskaya, E. B. (2012). Mass appraisal of residential apartments: an application of random forest for valuation and a CART-based approach for model diagnostics. *Expert systems with Applications*, *39*(2), 1772-1778.

Bańczyk, K., Kempa, O., Lasota, T., & Trawiński, B. (2011). Empirical comparison of bagging ensembles created using weak learners for a regression problem. Asian conference on intelligent information and database systems, 312-322. Springer

Baniecki, H., Kretowicz, W., & Biecek, P. (2021). Fooling partial dependence via data poisoning. *arXiv preprint arXiv:2105.12837*.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, *24*, 2546-2554.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Bin, J., Tang, S., Liu, Y., Wang, G., Gardiner, B., Liu, Z., & Li, E. (2017). Regression model for appraisal of real estate using recurrent neural network and boosting tree. 2017 2nd IEEE international conference on computational intelligence and applications (ICCIA), 209-213. IEEE

Bogin, A. N., & Shui, J. (2020). Appraisal accuracy and automated valuation models in rural areas. *The Journal of Real Estate Finance and Economics*, *60*(1), 40-52.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123-140.

Breiman, L. (2001). Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Statistical science*, *16*(3), 199-231.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Taylor & Francis.

Chen, T., & Guestrin, C. (2016). Xgboost: a scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 785-794.

Chen, Y., Yazdani, M., Mojtahedi, M., & Newton, S. (2019). The impact on neighbourhood residential property valuations of a newly proposed public transport project: the Sydney northwest metro case study. *Transportation Research Interdisciplinary Perspectives*, *3*, 100070.

Dell'Anna, F., & Bottero, M. (2021). Green premium in buildings: evidence from the real estate market of Singapore. *Journal of Cleaner Production*, *286*, 125327.

Des Rosiers, F., Bolduc, A., & Thériault, M. (1999). Environment and value does drinking water quality affect house prices? *Journal of Property Investment & Finance*, *17*(5), 444-463.

Dimanov, B., Bhatt, U., Jamnik, M., & Weller, A. (2020). You shouldn't trust me: learning models which conceal unfairness from multiple explanation methods. SafeAI@ AAAI.

Dimopoulos, T., & Bakas, N. (2019). Sensitivity analysis of machine learning models for the mass appraisal of real estate. Case study of residential units in Nicosia, Cyprus. *Remote sensing*, *11*(24), 3047.

Dimopoulos, T., & Moulas, A. (2016). A proposal of a mass appraisal system in Greece with CAMA system: evaluating GWR and MRA techniques in Thessaloniki municipality. *Open geosciences*, *8*(1), 675-693.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Doumpos, M., Papastamos, D., Andritsos, D., & Zopounidis, C. (2021). Developing automated valuation models for estimating property values: a comparison of global and locally weighted approaches. *Annals of Operations Research*, *306*(1), 415-433.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Gabrielli, L., & French, N. (2020). Pricing to market: property valuation methods–a practical review. *Journal of Property Investment & Finance*, *39*(5), 464-480.

García, N., Gámez, M., & Alfaro, E. (2008). ANN+ GIS: an automated system for property valuation. *Neurocomputing*, *71*(4-6), 733-742.

Ghatnekar, A., & Shanbhag, A. D. (2021). Explainable, multi-region price prediction. 2021 International conference on electrical, computer and energy technologies (ICECET), 1-7. IEEE

Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of neural networks is fragile. *Proceedings of the AAAI conference on artificial intelligence*, *33*(01), 3681-3688.

Gnat, S. (2021a). Impact of categorical variables encoding on property mass valuation. *Procedia Computer Science*, *192*, 3542-3550.

Gnat, S. (2021b). Property mass valuation on small markets. *Land*, *10*(388).

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, *38*(3), 50-57.

Gosiewska, A., & Biecek, P. (2019). Do not trust additive explanations. *arXiv preprint arXiv:1903.11420*.

Graczyk, M., Lasota, T., & Trawiński, B. (2009). Comparative analysis of premises valuation models using KEEL, RapidMiner, and WEKA. International conference on computational collective intelligence, 800-812. Springer

Graczyk, M., Lasota, T., Trawiński, B., & Trawiński, K. (2010). Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. Asian conference on intelligent information and database systems, 340-350. Springer

Guan, J., Shi, D., Zurada, J. M., & Levitan, A. S. (2014). Analyzing massive data sets: an adaptive fuzzy neural approach for prediction, with a real estate illustration. *Journal of organizational computing and electronic commerce*, *24*(1), 94-112.

Hastie, T., & Tibshirani, R. (1986). Generalized additive models. *Statistical science*, *1*(3), 297-310. http://www.jstor.org/stable/2245459
Hicks, R. L., & Queen, B. M. (2016). Valuing historical and open space amenities with hedonic property valuation models. *Agricultural and Resource Economics Review*, *45*(1), 44-67.
Ho, W. K., Tang, B.-S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48-70.
Hong, J., Choi, H., & Kim, W.-s. (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.
Hurley, A. K., & Sweeney, J. (2022). Irish property price estimation using a flexible geo-spatial smoothing approach: what is the impact of an address? *The Journal of Real Estate Finance and Economics*, 1-39.
Hussain, T., Abbas, J., Wei, Z., Ahmad, S., Xuehao, B., & Gaoli, Z. (2021). Impact of urban village disamenity on neighboring residential properties: empirical evidence from Nanjing through hedonic pricing model appraisal. *Journal of Urban Planning and Development*, *147*(1), 04020055.
Jamil, S., Mohd, T., Masrom, S., & Ab Rahim, N. (2020). Machine learning price prediction on green building prices. 2020 IEEE symposium on industrial electronics & applications (ISIEA), 1-6. IEEE
Jarosz, M., Kutrzyński, M., Lasota, T., Piwowarczyk, M., Telec, Z., & Trawiński, B. (2020). Machine learning models for real estate appraisal constructed using spline trend functions. Asian conference on intelligent information and database systems, 636-648. Springer
Johnson, E. B., Tidwell, A., & Villupuram, S. V. (2020). Valuing curb appeal. *The Journal of Real Estate Finance and Economics*, *60*(1), 111-133.
Kempa, O., Lasota, T., Telec, Z., & Trawiński, B. (2011). Investigation of bagging ensembles of genetic neural networks and fuzzy systems for real estate appraisal. Asian conference on intelligent information and database systems, 323-332. Springer
Lam, K. C., Yu, C., & Lam, C. K. (2009). Support vector machine and entropy based decision support system for property valuation. *Journal of Property Research*, *26*(3), 213-233.
Lasota, T., Łuczak, T., Niemczyk, M., Olszewski, M., & Trawiński, B. (2013). Investigation of property valuation models based on decision tree ensembles built over noised data. International conference on computational collective intelligence, 417-426. Springer
Lasota, T., Łuczak, T., & Trawiński, B. (2011). Experimental comparison of resampling methods in a multi-agent system to assist with property valuation. KES international symposium on agent and multi-agent systems: technologies and applications, 342-352. Springer
Lasota, T., Makos, M., & Trawiński, B. (2009). Comparative analysis of neural network models for premises valuation using SAS enterprise miner. New challenges in computational collective intelligence, 337-348. Springer
Lasota, T., Sachnowski, P., & Trawiński, B. (2009). Comparative analysis of regression tree models for premises valuation using statistica data miner. International conference on computational collective intelligence, 776-787. Springer
Lasota, T., Sawiłow, E., Telec, Z., Trawiński, B., Roman, M., Matczuk, P., & Popowicz, P. (2015). Enhancing intelligent property valuation models by

merging similar cadastral regions of a municipality. International conference on computational collective intelligence, 566-577. Springer
Lasota, T., Telec, Z., Trawiński, G., & Trawiński, B. (2011). Empirical comparison of resampling methods using genetic fuzzy systems for a regression problem. International conference on intelligent data engineering and automated learning, 17-24. Springer
Lee, C. (2022). Training and interpreting machine learning models: application in property tax assessment. *Real Estate Management and Valuation*, *30*(1), 13-22.
Lee, C., & Park, K. K.-H. (2020). Representing uncertainty in property valuation through a bayesian deep learning approach. *Real Estate Management and Valuation*, *28*(4), 15-23.
Lin, R. F.-Y., Ou, C., Tseng, K.-K., Bowen, D., Yung, K., & Ip, W. (2021). The spatial neural network model with disruptive technology for property appraisal in real estate industry. *Technological Forecasting and Social Change*, *173*, 121067.
Lipton, Z. C. (2018). The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. *Queue*, *16*(3), 31-57.
Liu, Y., Yang, L., & Chau, K. W. (2020). Impacts of tourism demand on retail property prices in a shopping destination. *Sustainability*, *12*(4), 1361.
Liu, Z., Yan, S., Cao, J., Jin, T., Tang, J., Yang, J., & Wang, Q. (2018). A Bayesian approach to residential property valuation based on built environment and house characteristics. 2018 IEEE international conference on big data (big data), 1455-1464. IEEE
Lorenz, F., Willwersch, J., Cajias, M., & Fuerst, F. (2022). Interpretable machine learning for real estate market analysis. *Real Estate Economics*.
Lou, Y., Caruana, R., & Gehrke, J. (2012). Intelligible models for classification and regression. Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, 150-158.
Lou, Y., Caruana, R., Gehrke, J., & Hooker, G. (2013). Accurate intelligible models with pairwise interactions. Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, 623-631.
Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*, 4765-4774.
Masrom, S., Mohd, T., & Abd Rahman, A. S. (2022). Green building factor in machine learning based condominium price prediction. *IAES International Journal of Artificial Intelligence*, *11*(1), 291-299.
Mayer, M., Bourassa, S. C., Hoesli, M., & Scognamiglio, D. (2019). Estimation and updating methods for hedonic valuation. *Journal of European Real Estate Research*, *12*(1), 134-150.
Mayer, M., Bourassa, S. C., Hoesli, M., & Scognamiglio, D. (2022). Machine learning applications to land and structure valuation. *Journal of Risk and Financial Management*, *15*(5), 193.
Mimis, A., Rovolis, A., & Stamou, M. (2013). Property valuation with artificial neural network: the case of Athens. *Journal of Property Research*, *30*(2), 128-143.
Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.
Mrsic, L., Jerkovic, H., & Balkovic, M. (2020). Real estate market price prediction framework based on public data sources with case study from Croatia. Asian conference on intelligent information and database systems, 13-24. Springer

Nejad, M. Z., Lu, J., Asgari, P., & Behbood, V. (2016). The effect of google drive distance and duration in residential property in Sydney, Australia. Uncertainty modelling in knowledge engineering and decision making: proceedings of the 12th international FLINS conference, 646-655. World Scientific

Nejad, M. Z., Lu, J., & Behbood, V. (2017). Applying dynamic Bayesian tree in property sales price estimation. 2017 12th International conference on intelligent systems and knowledge engineering (ISKE), 1-6. IEEE

Niu, J., & Niu, P. (2019). An intelligent automatic valuation system for real estate based on machine learning. Proceedings of the international conference on artificial intelligence, information processing and cloud computing, 1-6.

Nori, H., Caruana, R., Bu, Z., Shen, J. H., & Kulkarni, J. (2021). Accuracy, interpretability, and differential privacy via explainable boosting. International conference on machine learning, 8227-8237. PMLR

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Oust, A., Hansen, S. N., & Pettrem, T. R. (2020). Combining property price predictions from repeat sales and spatially enhanced hedonic regressions. *The Journal of Real Estate Finance and Economics*, *61*(2), 183-207.

Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., & French, N. (2003). Real estate appraisal: a review of valuation methods. *Journal of Property Investment & Finance*, *21*(4), 383-401.

Poursabzi-Sangdeh, F., Goldstein, D. G., Hofman, J. M., Wortman Vaughan, J. W., & Wallach, H. (2021). Manipulating and measuring model interpretability. Proceedings of the 2021 CHI conference on human factors in computing systems, 1-52.

Putatunda, S., & Rama, K. (2018). A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost. Proceedings of the 2018 international conference on signal processing and machine learning, 6-10.

Rajapaksa, D., Athukorala, W., Managi, S., Neelawala, P., Lee, B., Hoang, V.-N., & Wilson, C. (2018). The impact of cell phone towers on house prices: evidence from Brisbane, Australia. *Environmental Economics and Policy Studies*, *20*(1), 211-224.

Rajapaksa, D., Gono, M., Wilson, C., Managi, S., Lee, B., & Hoang, V.-N. (2020). The demand for education: the impacts of good schools on property values in Brisbane, Australia. *Land Use Policy*, *97*, 104748.

Reyes-Bueno, F., García-Samaniego, J. M., & Sánchez-Rodríguez, A. (2018). Large-scale simultaneous market segment definition and mass appraisal using decision tree learning for fiscal purposes. *Land Use Policy*, *79*, 116-122.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 1135-1144.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206-215.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: fundamental principles and 10 grand challenges. *Statistics Surveys*, *16*, 1-85.

Rudin, C., Wang, C., & Coker, B. (2018). The age of secrecy and unfairness in recidivism prediction. *arXiv preprint arXiv:1811.00731*.
Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*(2), 197-227.
Schulz, R., & Wersing, M. (2021). Automated valuation services: a case study for Aberdeen in Scotland. *Journal of Property Research*, *38*(2), 154-172.
Selbst, A., & Powles, J. (2018). "Meaningful information" and the right to explanation. Conference on fairness, accountability and transparency, 48-48. PMLR
Selim, H. (2009). Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. *Expert systems with Applications*, *36*(2), 2843-2852.
Semenova, L., Rudin, C., & Parr, R. (2022). On the existence of simpler machine learning models. 2022 ACM conference on fairness, accountability, and transparency, 1827-1858.
Shehata, W., Abu Arqoub, M., Langston, C., Elkheshien, R., & Sarvimäki, M. (2021). From hard bed to luxury home: impacts of reusing HM Prison Pentridge on property values. *Journal of Housing and the Built Environment*, *36*(2), 627-643.
Steurer, M., Hill, R. J., & Pfeifer, N. (2021). Metrics for evaluating the performance of machine learning based automated valuation models. *Journal of Property Research*, *38*(2), 99-129.
Štubňová, M., Urbaníková, M., Hudáková, J., & Papcunová, V. (2020). Estimation of residential property market price: comparison of artificial neural networks and hedonic pricing model. *Emerging Science Journal*, *4*(6), 530-538.
Swartout, W. R., & Moore, J. D. (1993). Explanation in second generation expert systems. In *Second generation expert systems* (pp. 543-585). Springer.
Tajani, F., Morano, P., Locurcio, M., & D'Addabbo, N. (2015). Property valuations in times of crisis: artificial neural networks and evolutionary algorithms in comparison. International conference on computational science and its applications, 194-209. Springer
Talaga, M., Piwowarczyk, M., Kutrzyński, M., Lasota, T., Telec, Z., & Trawiński, B. (2019). Apartment valuation models for a big city using selected spatial attributes. International conference on computational collective intelligence, 363-376. Springer
Tchuente, D., & Nyawa, S. (2022). Real estate price estimation in French cities using geocoding and machine learning. *Annals of Operations Research*, *308*(1), 571-608.
Telec, Z., Trawiński, B., Lasota, T., & Trawiński, K. (2013). Comparison of evolving fuzzy systems with an ensemble approach to predict from a data stream. International conference on computational collective intelligence, 377-387. Springer
Thackway, W. T., Ng, M. K. M., Lee, C.-L., Shi, V., & Pettit, C. J. (2022). Spatial variability of the 'Airbnb effect': a spatially explicit analysis of Airbnb's impact on housing prices in Sydney. *ISPRS International Journal of Geo-Information*, *11*(1), 65.
Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273-282.
Trawiński, B., Lasota, T., Kempa, O., Telec, Z., & Kutrzyński, M. (2017). Comparison of ensemble learning models with expert algorithms designed for a property

valuation system. International conference on computational collective intelligence, 317-327. Springer
Trawiński, B., Telec, Z., Krasnoborski, J., Piwowarczyk, M., Talaga, M., Lasota, T., & Sawiłow, E. (2017). Comparison of expert algorithms with machine learning models for real estate appraisal. 2017 IEEE international conference on innovations in intelligent systems and applications (INISTA), 51-54. IEEE
Valier, A. (2020a). The cross validation in automated valuation models: a proposal for use. International conference on computational science and its applications, 585-596. Springer
Valier, A. (2020b). Evaluating avms performance. Beyond the accuracy. In *International symposium: new metropolitan perspectives* (pp. 1155-1164). Springer.
Veale, M., Van Kleek, M., & Binns, R. (2018). Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. Proceedings of the 2018 chi conference on human factors in computing systems, 1-14.
Vedachalam, S., Hitzhusen, F. J., & Mancl, K. M. (2013). Economic analysis of poorly sited septic systems: a hedonic pricing approach. *Journal of Environmental Planning and Management*, *56*(3), 329-344.
Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harvard journal of law & technology*, *31*(2), 841-887.
Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing*, *415*, 295-316.
Yang, Z., Zhang, A., & Sudjianto, A. (2021). GAMI-Net: an explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, *120*, 108192.
Yee, L. W., Bakar, N. A. A., Hassan, N. H., Zainuddin, N. M. M., Yusoff, R. C. M., & Ab Rahim, N. Z. (2021). Using machine learning to forecast residential property prices in overcoming the property overhang issue. 2021 IEEE international conference on artificial intelligence in engineering and technology (IICAIET), 1-6. IEEE
Yilmazer, S., & Kocaman, S. (2020). A mass appraisal assessment study using machine learning based on multiple regression and random forest. *Land Use Policy*, *99*, 104889.
Zhang, M., & Shukla, J. (2023). Measuring the impact of heavy rail transport infrastructure on house prices in Melbourne, Australia: a case study of Mernda rail extension project. *Property Management*, *41*(1), 97-110.