Embedding-Enhanced Real Estate Valuation in Non-Metropolitan Sweden

A Hybrid Modeling Approach

Master's thesis in Complex Adaptive Systems

Leonard Smedenman
Teddy Sallén

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Embedding-Enhanced Real Estate Valuation in Non-Metropolitan Sweden

A Hybrid Modeling Approach

Leonard Smedenman
Teddy Sallén

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2025

Embedding-Enhanced Real Estate Valuation in Non-Metropolitan Sweden
A Hybrid Modeling Approach
Leonard Smedenman
Teddy Sallén

Supervisor and Examiner:
Mats Granath, Director, M.Sc Complex Adaptive Systems

Master's Thesis 2025
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Symbolic image of AI in housing. Source: Primary.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Embedding-Enhanced Real Estate Valuation in Non-Metropolitan Sweden
A Hybrid Modeling Approach
Leonard Smedenman & Teddy Sallén
Department of Physics
Chalmers University of Technology

Abstract

Automated valuation of residential properties in sparsely populated regions poses unique challenges due to thin transaction volumes, diverse housing stock, and limited comparables. This thesis presents a hybrid modeling approach combining an embedding-based artificial neural network (ANN) with a LightGBM gradient boosting machine to predict sale prices in six Swedish municipalities, focusing specifically on houses in non-metropolitan areas. The ANN learns dense representations of categorical and geographic features that capture latent spatial and socioeconomic patterns, while the GBM leverages both raw features and ANN embeddings to refine residual errors. Model interpretability is achieved via SHAP values and case studies of embedding dimensions, revealing that distance to regional centers, living area, property condition, and proximity to points of interest are key value drivers, even where market data are scarce. The hybrid model demonstrates competitive accuracy, particularly for mid-priced homes, and offers transparent explanations for each valuation. However, large errors persist for rare, high-end properties and extremely remote dwellings, reflecting fundamental data limitations. The results highlight how AI-driven valuation tools can complement traditional appraisal methods by providing rapid, interpretable estimates for routine cases and flagging high-uncertainty transactions for expert review.

Keywords: Automated Valuation Model, real-estate appraisal, neural embeddings, gradient boosting, SHAP interpretability, non-metropolitan housing.

Acknowledgements

We would like to express our gratitude to everyone who has contributed to the completion of this project. Firstly, we would like to thank our contact persons at Värderingsdata, Magnus Persson, Jon Larborn and Niklas Stenwreth. Without their assistance, guidance and knowledge this project would not be as successful. We would also like to extend our appreciation to our supervisor and examiner Mats Granath for accepting the role and providing input.
Thank you for all your contributions.
Sincerely,

Leonard Smedenman & Teddy Sallén, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
ANN	Artificial Neural Network
AVM	Automated Valuation Model
DeSO	Demographic Statistical Areas
GBDT	Gradient Boosted Decision Trees
GBM	Gradient Boosting Machine
GRP	Regional GDP
HPM	Hedonic Pricing Model
KNN	k-Nearest Neighbors
KTH	Kungliga Tekniska Högskolan
LGBM	Light Gradient Boosting Machine
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
NN	Nearest Neighbor
P10	Percentage of predictions within ±10% of sale price
P20	Percentage of predictions within ±20% of sale price
R2	Coefficient of Determination
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlanations
t–SNE	t–Distributed Stochastic Neighbor Embedding

Nomenclature

Below is the nomenclature of indices, Hyper-parameters and constants, parameters, variables, and metrics used throughout this thesis.

Indices

i	Index for property / transaction in the dataset
j	Index for input feature X_j in the hedonic model
m	Index of boosting iteration / tree (h_m, F_m)
c	Index of price-quantile class in the auxiliary classifier ($p_{i,c}$)

Hyper-parameters and constants

α	Weight of the P10 term in the composite loss (annealed from α_{start} to α_{end})
γ	Focusing parameter of the focal classification loss
δ	Huber-loss threshold that separates MAE/MSE regimes
k	Number of neighbours in the kNN component
m (margin)	Margin in the triplet-loss constraint
ν	Shrinkage (learning-rate) parameter in gradient boosting
B	Mini-batch size used during stochastic optimisation
$w_c,\ w_t$	Fixed weights of classification and triplet losses in L_{total}

Variables

$\mathbf{x}$	Raw feature vector of a property
x'	Standardised feature: $(x - \mu)/\sigma$
$\mu,\ \sigma$	Empirical mean and standard deviation of a feature
y	True log-transformed sale price (target)

$\hat{y}$	Predicted log-price produced by the model
V, $\hat{V}$	Price on the original SEK scale ($\hat{V} = e^{\hat{y}\sigma+\mu}$)
z_j, a_j	Pre-activation and activation of neuron j in the ANN
$\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}$	Weight matrix and bias vector of layer ℓ
e	128-dimensional learned embedding of a property
$p_{i,c}$	Probability that property i belongs to class c (softmax output)
$r_{i,m}$	Residual of sample i at boosting stage m
$h_m(\mathbf{x})$	Weak learner (regression tree) at stage m
$F_m(\mathbf{x})$	Ensemble prediction after m trees

Losses

L_{reg}	P10-aware regression loss (Huber + soft-P10)
L_{cls}	Focal classification loss
L_{triplet}	Triplet embedding loss
L_{total}	Composite training objective $L_{\text{reg}} + (1-\alpha)w_c L_{\text{cls}} + w_t L_{\text{triplet}}$

Evaluation metrics

n	Number of observations in a sample or split
MAPE, MAE, RMSE	Standard error statistics defined in Section 2.7
R^2	Coefficient of determination
P10, P20	Share of predictions within ±10% and ±20% of the true price, respectively

Contents

List of Figures

List of Tables

1
Introduction

Advancements in artificial intelligence (AI) offer new opportunities for real estate valuation, especially in data-scarce markets. This study explores the use of Machine Learning (ML) models, specifically Artificial Neural Networks (ANN) and gradient boosting, to improve valuation accuracy in sparsely populated regions of Sweden where traditional methods face significant limitations.

1.1 Background

Real estate valuation plays a central role in the functioning of the property market and financial system. Accurate property values are needed for a range of purposes, including sales and purchases, taxation, investment analysis, and securing mortgage loans [1]. In Sweden, official assessments of property value ("taxeringsvärde") are determined periodically by the national tax authority (Skatteverket) and are intended to reflect approximately 75% of market value for taxation purposes [2]. These assessments rely on recent sale prices of comparable properties within defined value areas (värdeområden) where properties are assumed to have similar conditions. However, in parts of Sweden that are generally more non-urban, such as the provinces of Östergötland, Småland, Gotland and Blekinge (see A.1 for full list of included counties and municipalities), property transactions occur less frequently, specifically in regards to houses, leading to thin markets with scarce comparable sales data [3]. In these areas, traditional indicators of market value become less reliable or even nonexistent, as noted historically in legal preparatory works that questioned the applicability of a market value concept in locales with virtually no sales activity. This poses challenges for property owners, buyers, and lenders, as valuation uncertainty increases outside urban centers.

Traditional real estate appraisal in Sweden has long been based on professional judgment supported by standard methods. These conventional approaches, while grounded in decades of experience, often struggle to capture market dynamics in real time, especially when data on actual transactions are limited.

1.2 Problem Description

Valuing properties in sparsely populated regions like the ones this thesis focuses on, presents significant challenges due to the limited number of transactions and the diverse nature of the properties. The standard sales comparison approach, which

relies on identifying recently sold comparable properties in a given area, becomes less reliable when few or no truly similar sales exist. In these areas, appraisers may be forced to base valuations on a very small sample of transactions, increasing the risk of error. Moreover, non-urban properties often possess unique features such as old building years or large plots of land, which make direct comparisons difficult. These factors contribute to considerable uncertainty in valuation and highlight the need for more flexible or data-driven approaches in sparsely populated markets.

Due to the limited availability of market data, valuers may be forced to rely on alternative methods or general assumptions. For example, cost based valuations or income based approaches may be used in place of direct market comparisons. These methods, however, may not reflect what a buyer would actually pay, especially if there are intangible values associated with location and amenities that are not captured by cost or income alone. As a result, valuations in these areas carry a higher degree of uncertainty and risk. This is problematic not only for private stakeholders but also for banks and public agencies. Lenders face difficulties in mortgage risk assessment when valuations are uncertain, and municipalities or tax authorities struggle to ensure fairness and accuracy in taxation when comparable sales are lacking [3]. Recent market fluctuations have highlighted this issue, during periods of market downturn or upheaval, transaction volumes can drop sharply, for example, in 2022 the transaction volume in Sweden fell by over 40% year-on-year, creating an extremely thin market [3], making it even harder to gauge true property values in affected regions.

1.3 Traditional Valuation Methods in Sweden

Real estate valuers traditionally employ a few fundamental methods to estimate market value, each with its own assumptions and data requirements. The *sales comparison approach*, as mentioned in the previous section, wherein the appraiser identifies recent sales of similar properties and adjusts for differences to estimate the subject property's value. In Sweden, hedonic pricing models (HPMs) based on multiple regression are used to support both private appraisals and mass appraisal for tax assessment. These models generalize the relationship between property characteristics and market prices within a given region [1]. HPMs represent property value as a function of its attributes, such as size, location, and quality, and have been foundational in valuation theory since 1974 [4]. They are relatively transparent and grounded in economic theory, but they typically assume a linear (or log-linear) relationship and may struggle with complex, non-linear interactions between features.

All these traditional methods require substantial expertise and judgment. Appraisers must carefully select comparables or estimate depreciation. In thin markets, the lack of data forces greater reliance on professional judgment, potentially introducing bias or error. Moreover, manual valuation processes are time consuming and not easily scalable. As the demand grows for rapid valuations, traditional methods show their limitations in terms of speed and consistency [5]. These limitations mo-

tivate the search for more automated and data-driven valuation methods that can complement or enhance the traditional techniques.

1.4 Rationale for ML-Based Valuation

Advancements in ML offer promising opportunities to address the challenges of non-urban property valuation. Automated Valuation Models (AVMs) are increasingly being used in real estate markets worldwide to produce instant value estimates by analyzing large datasets of property features and past transactions [6]. An AVM is a computer-driven algorithm that inputs property data and outputs a value estimate, often very rapidly, making it attractive for both lenders and investors who need quick assessments [5]. The key advantage of ML-based models is their ability to detect complex, non-linear patterns in data that traditional linear models fail to capture. This is particularly relevant for diverse non-urban properties where interactions between attributes, like land size, building condition, and locational factors, may influence value in complicated, often non-obvious ways.

Among ML techniques, one that stands out is *Artificial Neural Networks* models. ANNs are computational models inspired by the human brain, capable of fitting extremely flexible functional forms to data. They have shown promise in house price prediction tasks; for instance, studies have found that neural networks can outperform multiple regression models and other techniques in terms of valuation accuracy [7]. By learning from a broad set of input examples, an ANN can capture subtle relationships. The downside is that ANNs are often criticized as "black boxes", offering little transparency into how they arrive at a given estimate [8]. This lack of interpretability can be problematic for gaining trust in valuations, since proposing a valuation without a clear basis might make for a weak argument for a stakeholder, which is why recent research emphasizes explainable AI methods in real estate applications.

Embracing AI for property valuation in Sweden's non-urban context is not just a theoretical exercise; the industry has already begun moving in this direction. Banks and valuation firms in Sweden are experimenting with AI-driven models to complement traditional appraisals, especially for residential properties. According to Värderingsdata, which is a leading provider of property data in Sweden, AI-based valuation models are already used in practice and can drastically speed up the valuation process, allowing human experts to focus on more complex analysis [5]. In non-urban areas, an ML model might, for example, learn from transactions in a wider region or over a longer time horizon to compensate for the lack of recent local sales.

The rationale for this study is thus clear; by applying ML techniques to the problem of non-urban property valuation in Östergötland, Småland, Gotland, and Blekinge, the thesis aims to assess whether these methods can improve accuracy and consistency over traditional approaches.The practical considerations of using such models will also be examined, including data requirements and the interpretability of results.

This thesis aims to bring meaningful insights into what features, or combination of features, are deemed most important in the chosen focus group and how they differ from, for instance, apartments in urban areas. The ultimate goal is to develop an Automated Valuation Model tailored to non-urban Swedish conditions, or at least to evaluate its feasibility.

The quality of an AI-based valuation is heavily dependent on the quality and quantity of the input data; poor or biased data can lead to misleading estimates [5]. Additionally, stakeholders must be able to trust the output of a model, which yet agin highlights importance of transparency and validation [8]. This study is undertaken with these considerations in mind. By focusing on a geographically specific and data-challenged context, the research will highlight not only the potential accuracy gains from ML, but also the limitations and requirements for deploying such technology in real-world valuation practice.

1.5 Dataset

The dataset used in this study was provided by Värderingsdata and comprises roughly 90,000 residential properties with transactions ranging from 2015 to 2022. Each object represents an individual sale of a property (the same property can thus appear more than once if it was sold multiple times during the timeframe). The data includes approximately 170 features representing various physical, geographic, socioeconomic, and temporal characteristics. There is a column with index-adjusted sale-prices to 2020-06, which allows for a fair comparison across all the transaction years.

The dataset is organized into several feature domains. Object-level features describe each property's individual characteristics, including variables such as living area, construction year, energy class, and water and sewage access etc. Neighborhood-level characteristics capture sociodemographic and economic indicators from the surrounding area, including population age distribution, household types, education levels, income distribution, and local real estate market statistics. Macroeconomic indicators such as interest rates, gross regional product (GRP), and inflation measures are also included, contextualizing each transaction within broader market conditions. Geospatial attributes incorporate detailed locational data, including distances to various urban centers, natural features (like lakes and coast), infrastructure (like roads, rail, airports), and points of interest such as golf courses, schools, and ski resorts. Additionally, temporal variables encode the time dimension of each transaction, with fields like sale year, month, and day of the week.

The diversity and detail of the data offer a rich foundation for statistical learning. While some variables contain missing values, the overall completeness is high. Ideally, the dataset would include even more detailed object-specific features, such as the number of rooms, construction material, window type, roof condition, heating system type, floor material, ceiling height, and the presence of amenities such as a balcony, fireplace, or integrated household appliances, but these types of data

is not easily obtained. Alas, most of the columns pertain to more regional data. More specific data on the condition of the houses and their appliances could lead to more accurate valuation, but the variety of data is sufficient to make meaningful distinctions, though additional detail would still be desirable.

1.6 Objectives / Research Questions

The objectives of the master's thesis are summarized in the defined research questions below.

1.6.1 Objectives

1. To develop an ML-based model for property appraisal.
2. To assess the accuracy and reliability of the model in comparison to alternative methods.
3. To identify the most influential factors in property valuation as determined by the model, and what features contribute to the valuation.
4. To uncover what non-obvious features, and interactions between features, might be specifically important in non-urban housing.

1.6.2 Research Questions

1. How does an AI-driven model compare in performance to benchmark models in terms of accuracy and performance in?
2. What are the key factors influencing non-urban property valuation in the Småland, Östergötland, Gotland and Blekinge regions, as identified by the model?
3. What challenges and limitations arise when applying machine learning techniques to real estate valuation, and how can they be mitigated?
4. What insights can be gained from this study to inform future advancements in property valuation processes?

1.7 Scope and Delimitations

To maintain a clear and manageable scope, the following delimitations were applied:

- **Property Types:** The analysis is restricted to *residential properties*, such as family homes and small non-urban dwellings. However, assessments on very cheap smaller houses are discarded, since their sale prices and features more rarely coincide and thus only provide noise.
- **Temporal Scope:** Transaction data used for training and evaluation will be limited to a defined historical period 2015-2022.
- **Model Focus:** The study will concentrate on a hybrid model consisting of ANN and gradient boosting.
- **Model Inputs:** Only the structured dataset provided by Värderingsdata will be used in this study. No additional data collection from external sources has

been conducted, although lagged features derived from the original data have been created.

- **Comparison Baseline:** The performance of the model is compared to other traditional computational models. Manual expert appraisals are referenced for context but not replicated in this study.
- **Outcome Metrics:** Model performance will be evaluated primarily using statistical measures of predictive accuracy (e.g, RMSE, MAE, MAPE, P10/P20). Broader impacts such as user acceptance and regulatory considerations are not tested.
- **Implementation Context:** The study is exploratory and does not include real-time deployment or integration of the developed models into production environments used by Värderingsdata.

2

Theory

This chapter outlines the theoretical foundations of property valuation and machine learning, providing the conceptual framework for the methods used in the study.

2.1 Price Prediction and Regression Models

Predicting housing prices is a key challenge in real estate economics and data science. This section explores how regression models are applied to estimate property values based on various input features.

2.1.1 Overview of house price prediction as a regression task

House price prediction is the task of estimating a property's market value from its attributes. It is framed as a regression problem because the target output (price) is a continuous variable. In a regression model, the house's features serve as input variables and the output is a predicted price. The goal is to learn a mapping f that relates these features to the sale price by training on historical sales data. House price prediction is therefore a classic example of supervised regression analysis in real estate economics and machine learning [9]. An illustration with three arbitrary features is shown in Fig. 2.1

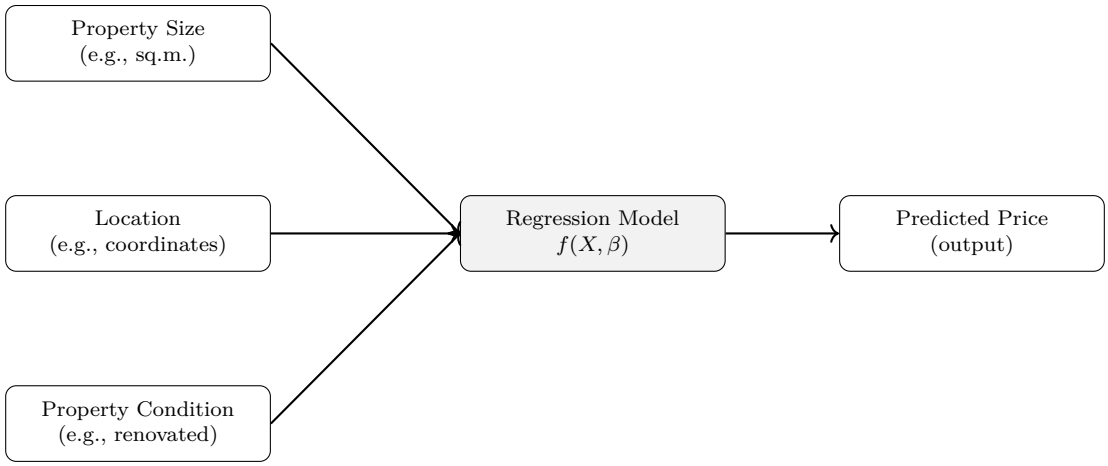

Figure 2.1: Overview of house price prediction as a regression task. Input features are mapped through a regression model to produce a continuous output (price). Source: Primary.

2.1.2 Hedonic regression models

A cornerstone of traditional house valuation is hedonic regression. This method models a property's value

$$V = f(X_1, X_2, \ldots, X_n) \tag{2.1}$$

as a function of its characteristics. In practice, f is often assumed linear:

$$V = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon \tag{2.2}$$

where each X_j is a property feature (size, location, etc.) and β_j its estimated effect on price. Each coefficient thus represents the contribution of that feature, making the model easy to interpret. Hedonic regression has been frequently used for decades in market analysis and mass appraisal [10] because of its simplicity and transparency. However, the linear additive assumptions of hedonic models can be limiting. A simple hedonic model may fail to capture complex or non-linear relationships (for example, varying impacts of property age on market value) or interactions between factors. Moreover, it requires high quality data containing key variables or working with sparse data can lead to biased, unreliable estimates [11]. Furthermore, they are sensitive to multicollinearity, multiple additions of features without careful consideration can therefore lead to lopsided or misleading results, which calls for an informed user in order to get accurate results.

2.1.3 K-Nearest Neighbors

An alternative non-parametric approach to hedonic regression is the K-Nearest Neighbors algorithm (KNN), originally proposed by Fix and Hodges and later formalized by Cover and Hart [12]. Instead of specifying a functional form for the relationship between property characteristics and price, KNNs assume that similar properties have similar market values. For an object with feature vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, the set of its k nearest neighbors is defined in the training data as:

$$N_k(\mathbf{X}) \;=\; \Big\{ (\mathbf{X}^{(i)}, V^{(i)}) : \mathbf{X}^{(i)} \text{ is among the } k \text{ closest points to } \mathbf{X} \Big\}.$$

The predicted value $\hat{V}$ is then computed as the average of the neighbor prices:

$$\hat{V}(\mathbf{X}) \;=\; \frac{1}{k} \sum V^{(i)}. \tag{2.3}$$

By using a distance metric, the model captures non-linear relationships and interactions without explicit model assumptions. The method is intuitive and straightforward to implement, but can become computationally expensive for large datasets and suffer from the "curse of dimensionality" as the feature space grows [13]. It is also sensitive to noise and unevenly distributed data. Nonetheless, KNN remains a popular baseline in real estate valuation studies and in graph–based extensions where local similarity is leveraged.

2.2 Feature Engineering

Effective feature engineering is essential for extracting maximal predictive power from structured data. In real estate valuation, raw inputs can include highly skewed numeric variables, high-cardinality categorical variables and proportional features due to the inherently diverse nature of housing. This section reviews the theory behind each transformation applied in the code.

2.2.1 Logarithmic Transformation of Skewed Variables

Many real estate attributes exhibit a long right tail, where a small fraction of high-end properties inflate the mean and violate Gaussian assumptions. Applying the natural logarithm compresses large values more than small ones, stabilizing variance and often improving both linear and non-linear model performance [14]. In economic contexts, log-errors correspond to relative errors, making them more interpretable when predicting quantities that span multiple orders of magnitude.

2.2.2 Label Encoding

Simple categorical fields with low cardinality can advantageously be converted to integer labels via a "Label encoder", which preserves uniqueness but imposes an arbitrary order. While tree-based models are unaffected by ordinal label codes, neural networks can learn embeddings on these integer indices.

2.2.3 Feature Standardization

Features with heterogeneous scales, for example living area in square meters vs a log-adjusted price can dominate optimization and distance metrics. Z-score standardization,

$$x' = \frac{x - \mu}{\sigma},$$

centers each feature to zero mean and unit variance, facilitating stable gradient descent in neural networks and balanced Euclidean distances in k-nearest neighbors [15].

2.3 Neural Networks for Regression

Artificial Neural Networks are a class of models inspired by the human brain, composed of interconnected units called neurons organized in layers. An ANN typically consists of an input layer, which takes in the features, one or more hidden layers that transform the inputs through weighted connections, and an output layer that produces the prediction. Each connection between neurons has a weight that amplifies or reduces the signal, and each neuron applies a non-linear activation function to the weighted sum of its inputs. Through a learning process, these weights are adjusted so that the network outputs accurate predictions on the training data [16]. In Fig. 2.2 there is a simple illustration of a neural network.

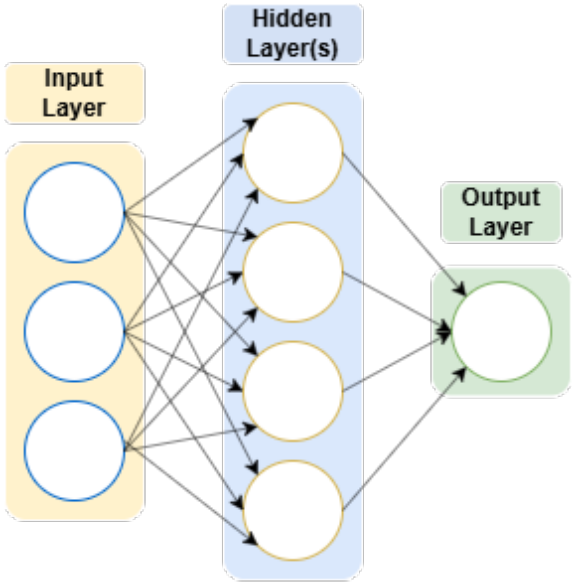

Figure 2.2: A multi-layer feed-forward Artificial Neural Network with an input layer, one hidden layer, and an output layer. Each connection has a weight, and each neuron (circle) computes a function of the weighted inputs. Source: Primary.

Mathematically, a simple ANN with one hidden layer can be described as follows: Suppose there are d input features $x_1, \ldots, x_d$. Each hidden neuron $h^{(j)}$ computes a linear combination:

$$z_j = \sum_{i=1}^{d} w_{ij}^{(1)} x_i + b_j^{(1)} \tag{2.4}$$

and then applies a non-linear activation $a_j = f(z_j)$, where f could be a ReLU or sigmoid function. The output layer then takes these hidden activations and computes the final output:

$$\hat{y} = \sum_{j} w_j^{(2)} a_j + b^{(2)} \tag{2.5}$$

(for a regression network, often a linear activation is used at the output so that $\hat{y}$ is a continuous number). In vectorized form, the network function is:

$$\hat{y} = W^{(2)} f(W^{(1)} x + b^{(1)}) + b^{(2)} \tag{2.6}$$

The key point is that by composing two (or more) linear transformations with non-linear activations, the network can approximate very complex functions. In fact, the Universal Approximation Theorem states that a sufficiently large neural network can approximate any continuous function on compact domains to arbitrary accuracy, given enough neurons in the hidden layer [17]. This theory explains why neural networks are so useful for predicting house prices, they can learn complex relationships between features.

ANNs learn the weights from data through an iterative optimization process called backpropagation combined with gradient based optimizers. The network starts with random weights and then in each training epoch, the predictions are compared to

true prices using a loss function (discussed later more deeply in the Methodology chapter 3). The gradient of the loss with respect to each weight is computed through the backpropagation algorithm, and the weights are adjusted in the direction that reduces the error. Over many iterations, the network hopefully converges to a set of weights that make accurate predictions on the data in question.

One appeal of ANNs in real estate is their ability to automatically learn latent features. For instance, the hidden neurons learn to represent combinations of inputs, for example, a neuron might push for "lakeside rural cottage" properties if such a pattern is present. ANNs are flexible and can theoretically handle interactions and non-linearities better than any predefined regression formula. However, there are challenges and considerations with ANNs. First, they generally require a large amount of data to train effectively, especially compared to many simpler models. In a data-sparse rural context, a complex neural network could overfit, learning quirks of the training data that don't generalize, if not carefully regularized. Simpler network architectures or additional data might be necessary for more enhanced results. Second, ANNs are often criticized as "black boxes" as mentioned in 1.4, because the relationship between inputs and outputs is encoded in many weights in a non-transparent way. It's not obvious why a particular prediction was made, which can be a disadvantage in valuation, where explainability is important. Later in this chapter, interpretability methods which can mitigate the interpretability problem are discussed. Finally, hyperparameter tuning (choosing the number of layers, neurons, learning rate, etc.) is important to get good performance and can be time-consuming. Despite these issues, ANNs remain a promising and highly feasible method for capturing complex value drivers in properties.

2.4 Loss Functions

In training and evaluating regression models, the choice of loss function/error metric is critical. The loss function is the quantitative measure of error that the model tries to minimize during training. Different losses have different properties and can lead to different model behavior, especially important in valuation where one might care about relative error more than absolute error, or want to avoid over-penalizing outliers. Below, common loss functions and specialized ones used in this thesis are outlined.

2.4.1 Huber Loss

The Huber loss is a robust loss function that behaves like mean squared error (MSE) for small errors and like mean absolute error (MAE) for large errors [18] [19]. Mathematically, it is defined piecewise, being quadratic when the absolute residual is below a certain threshold δ and linear beyond that point. This hybrid nature gives Huber loss the advantages of both MSE and MAE. Huber loss is commonly used in robust regression and machine learning settings where the user expects noisy data, providing a balance between sensitivity to small errors and insensitivity to very large

deviations. An illustration of how Huber loss penalizes wrong predictions is shown in Fig. 2.3.

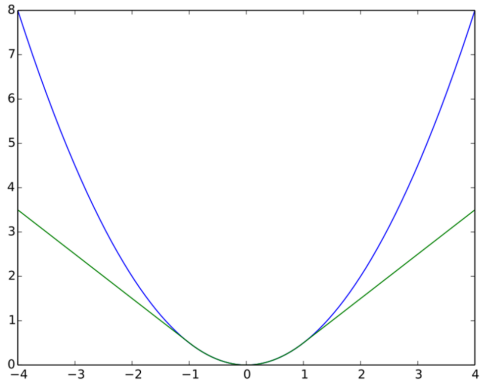

Figure 2.3: Comparison of Huber loss (green) with standard squared error loss (blue) as a function of the prediction residual. Source: Qwertyus `https://en.wikipedia.org/wiki/Huber_loss#/media/File:Huber_loss.svg`

2.4.2 Combining Loss Functions in Regression Models

In practice, a single regression loss may not capture all modeling objectives. Combining multiple loss terms allows the model to balance these priorities. In general, one forms a composite loss as a weighted sum of components, so that each term contributes to guiding the training process. This strategy can improve generalization: prior work has shown that multi-objective loss functions often yield better performance on heterogeneous data and allow practitioners to tune tradeoff hyperparameters between different goals [20].

2.4.3 Combining Loss Functions in Regression Models

In practice, optimizing a single regression loss may not capture all modeling objectives, particularly when the model must also learn a structured or generalizable internal representation. In multi-task learning settings, it is common to combine several loss terms, each with a different purpose. For example, alongside the primary regression loss which predicts the sale price, additional losses such as classification or contrastive objectives can help the model toward learning embeddings that reflect meaningful relationships in the data, for example market segment. This allows each loss term to influence training in proportion to its assigned weight. While the model still predicts a single scalar target, additional losses support generalization by enforcing structure in the learned representation. Prior work shows that such multi-objective training can improve both convergence and out-of-distribution robustness [20].

2.5 Gradient Boosting and LightGBM

Gradient Boosting is an ensemble method that builds a strong predictor by sequentially combining weak learners, typically shallow regression trees. Originally developed for classification, it was extended to regression by Friedman (2001) as Gradient Boosted Decision Trees (GBDT) [21].

2.5.1 The Gradient Boosting Process

Instead of training one complex model, gradient boosting builds a sequence of simple models $(h_1, h_2, \ldots, h_M)$, where each new model tries to correct the errors made by the ones before it. The process starts with a basic guess $F_0(x)$, often just the average sale price, and gradually improves this prediction in steps [22].

1. Compute residuals for each training example. For Mean Squared Error loss, the residual at stage m is $r_{i,m} = y - F_{m-1}(x_i)$.
2. Train a new decision tree $h_m(x)$ on these residuals, learning how the current model errs.
3. Update the model: $F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$, where ν is a shrinkage parameter (learning rate).
4. Repeat until M trees have been added or validation error ceases to improve.

Each tree greedily reduces remaining error, by moving in the negative gradient direction of the loss function, hence the term gradient boosting. The final model is a weighted sum of M decision trees. Although individual trees are generally shallow, the ensemble collectively achieves accuracy and robustness.

2.5.2 Gradient Boosting in Real Estate Valuation

In real estate valuation, gradient boosting models provide distinct advantages. Decision trees naturally handle numerical and categorical features, effectively capturing non-linear relationships among features. For example, trees can specifically model scenarios like rural properties with long commutes or waterfront properties, accumulating adjustments from multiple trees for nuanced predictions.
Despite its strengths, gradient boosting, like all models, has drawbacks. Optimal performance requires careful hyperparameter tuning, like tree number, depth, and learning rate, using techniques like cross-validation [23] to balance overfitting and underfitting. Furthermore, large ensembles can slow predictions with very lagre datasets, though typically manageable in real estate contexts. However, despite these minor drawbacks, gradient boosting remains a powerful, flexible methodology well suited to modeling complex and sparse data.

2.6 Overfitting

Overfitting is a phenomenon in which a model becomes too closely aligned with the training data, capturing noise or unusual patterns that do not generalize well to unseen inputs [24]. This often results in a steadily decreasing training error while the validation error, after an initial improvement, begins to rise. This pattern indicates

that the model is not learning the underlying data distribution but is instead memorizing specific examples from the training set. As a result, the model performs well on the data it has seen but poorly on new or unseen data. Bad fitting typically occurs when a model is either too small to capture the data well i.e underfitting, when model capacity is similar in scale to the training data i.e when it is large enough to memorize patterns without generalizing, the risk of overfitting is often at its highest. Interestingly, work has shown that very large or overparameterized networks often generalize better than moderately sized ones [25]. When the training data is small or contains noise, the risk increases further. For instance, a neural network with a number of parameters similar to the number of training samples can easily memorize the data rather than learn generalizable features. In practice, overfitting is often diagnosed by monitoring the difference between training and validation errors. A growing gap between them signals that the model's generalization ability is deteriorating. To mitigate overfitting, techniques such as regularization, early stopping during training, and the use of a separate validation set are commonly applied.

2.7 Model Evaluation Metrics

In order to make a thorough comparison of the different models, standardized and clear evaluation metrics are needed. The ones used for this thesis are listed in the subsequent subsections, where y is the true price, $\hat{y}$ the predicted price, and n is the total number of properties.

2.7.1 Mean Absolut Percentage Error (MAPE)

MAPE is essentially the average percentage error and is calculated as equation (2.7).

$$\frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \tag{2.7}$$

For instance, an MAPE of 10% means predictions are off by 10% on average. MAPE is scale-independent, which is useful in real estate portfolios with a wide range of prices. It is intuitive and a common metric in appraisal literature.

2.7.2 Mean Absolute Error (MAE)

In contrast to MAPE, MAE measures the average absolute difference between predicted and true values. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right| \tag{2.8}$$

MAE is in the same units as the target, i.e SEK, making it directly interpretable for stakeholders.

2.7.3 Root Mean Squared Error (RMSE)

RMSE penalizes larger errors more heavily by squaring the residuals before averaging and then taking the square root:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{2.9}$$

Because of the squaring, RMSE is more sensitive to outliers than MAE. A lower RMSE indicates fewer large deviations, which is critical when extreme misvaluations carry high risk.

2.7.4 Coefficient of Determination (R^2)

The R^2 metric quantifies the proportion of variance in the true values explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.10}$$

An R^2 of 0.80 means 80% of the variance in sale prices is captured by the model, indicating strong explanatory power. Unlike error metrics, higher R^2 is better, with a maximum of 1.0 for perfect fit.

2.7.5 P10 and P20

In real estate mass appraisal, P10 and P20 are accuracy metrics indicating the share of model predictions that fall within a certain margin of the true property value. In other words, P10 is the percentage of predicted prices within ±10% of the actual sale price, and P20 is the percentage within ±20%. Formally, one can define these as:

$$\text{P10} = 100\% \cdot \frac{(\#\text{ predictions with } |\hat{y} - y|) \leq (0.10 \cdot y)}{n},$$

$$\text{P20} = 100\% \cdot \frac{(\#\text{predictions with } |\hat{y} - y|) \leq (0.20 \cdot y)}{n},$$

2.7.6 SHAP values – feature importance and interpretability

SHAP (SHapley Additive exPlanations) is a method rooted in cooperative game theory for interpreting machine learning predictions by assigning each feature a SHAP value. These values represent how much each feature increases or decreases a prediction relative to a baseline (e.g., the average prediction) [26]. SHAP values extend Shapley values from cooperative game theory to machine learning. They attribute the model's prediction to input features by averaging each feature's contribution

across all possible subsets of features.

SHAP values show the effect of each feature on a specific prediction. For example, a rural house's valuation might decrease due to being farther from a city, but increase with a larger lot size, for example. Summing the baseline and these SHAP contributions explains the final prediction clearly, similar to how an appraiser would justify property valuation.
In this thesis, SHAP values clarify the gradient boosting model's predictions, identifying which features influence house valuations and ensuring the model captures logical patterns (e.g., larger living area positively affecting price). SHAP analyses can also detect any potential spurious correlations and visually demonstrate feature importance and nonlinear effects through SHAP summary and dependence plots. SHAP values are also attributed to neural embeddings if they are used in a gradient boosting, allowing for interpretation of what embeddings, or set of combined features contribute to the valuation.

2.7.7 t-SNE visualizing a high-dimensional representation

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for visualizing high-dimensional data by embedding it into a low-dimensional space, typically 2D, while preserving local relationships [27]. Unlike linear methods like Principal Component Analysis (PCA) [28] that preserve global variance, t-SNE emphasizes local structure: points that are close together in the high-dimensional space are mapped close together in 2D, while dissimilar points are placed farther apart.
The algorithm proceeds in two main steps:

1. **High-dimensional similarities:** Computes probabilities p_{ij} that reflect how similar data points are using a Gaussian distribution.
2. **Low-dimensional mapping:** Computes probabilities q_{ij} in 2D using a Student t-distribution.

3
Methodology

3.1 Data Preprocessing

Accurate and robust data preprocessing is critical to ensure that the models learn meaningful patterns rather than artifacts of noise or absence of data. The following steps were applied to transform the raw transaction records into a fully numeric dataset with consistent scaling and minimal missing values.

3.1.1 Cleaning and Imputation

After loading the Parquet transaction file, columns with more than 30 % missing entries were discarded to avoid distorting model training. For municipality-level attributes such as population, population change rates, migration fractions, gaps were forward-and back-filled within each Municipality code group (see A.1), since within each respective municipality, municipality-level features should be identical. Highly skewed numeric features, identified by a maximum-to-minimum ratio above 10 and strictly positive values were log-transformed and the originals were dropped.

3.1.2 Categorical Encoding and Vocabulary Extraction

Object-dtype columns were first changed to UTF-8 text and nulls replaced with the literal category "Unknown." Each was then converted to a `pandas.Categorical` type [29], allowing LightGBM to treat them natively as categorical features. Simultaneously, integer codes for each category level were extracted into new coded columns, which serve as inputs to the neural network's embedding layers. The code also records each category's vocabulary size, ensuring that each embedding matrix is sized precisely to its feature's cardinality.

3.1.3 Proportion Clipping and Cyclical Date Features

A set of fraction-type variables was clipped to the $[0, 1]$ interval to enforce valid bounds. To allow models to learn smooth seasonal effects, the month of sale was encoded as two cyclical features:

$$\texttt{SaleMonthSin} = \sin\Big(2\pi\,\texttt{SaleMonthOfYear}/12\Big),$$

$$\texttt{SaleMonthCos} = \cos\Big(2\pi\,\texttt{SaleMonthOfYear}/12\Big).$$

This representation ensures December and January are adjacent in feature space, instead of interpreting the months as far apart (1 and 12).

3.1.4 Feature Selection and Scaling

After dropping raw identifiers and geometry columns, the remaining numeric features (original, log-transformed, and cyclical) were split into `FEATURES` for continuous inputs and `CAT_CODE_COLS` for the integer codes of each categorical feature. The continuous features were standardized to zero mean and unit variance using Scikit-Learn's `StandardScaler` [30] fitted on the training split, then applied unchanged to development and test splits, in order to completely avoid any test data leakage into training. Categorical columns retained their `category` dtype for LightGBM, while the corresponding coded columns were fed into the neural network's embedding layers.

After these definitive preprocessing steps, the dataset consists exclusively of:

- Log-transformed and z-score standardized continuous features (`FEATURES`)
- Integer-coded categorical features (`CAT_CODE_COLS`) with known vocabulary sizes
- Validated proportion and cyclical date features.

This scaled representation is favorable for the ANN's embedding layers and the LightGBM's native categorical handling.

3.2 Model Development

In Fig. 3.1 a simple flowchart of the hybrid model is displayed. The following subsections detail the architecture and implementation of each model, as well as the techniques used to improve their efficiency and effectiveness.

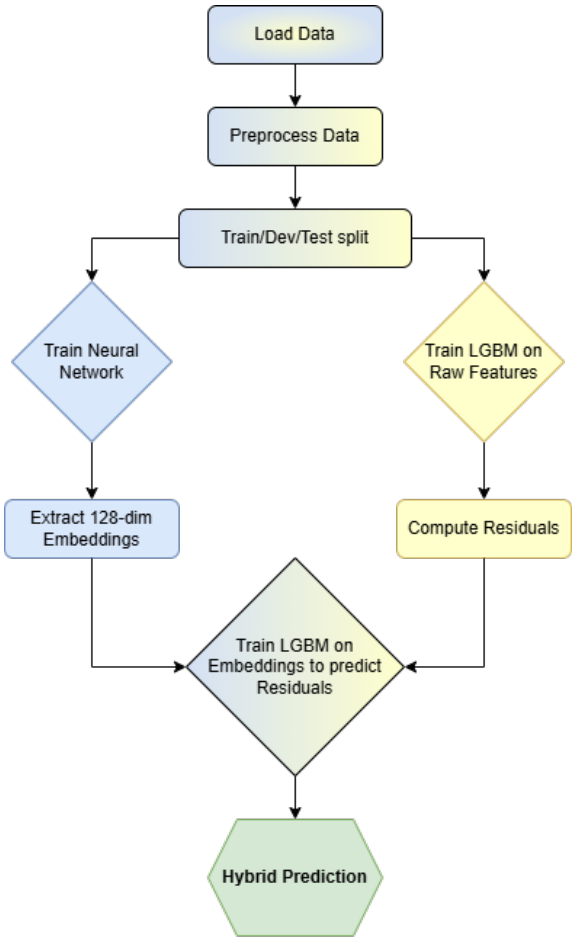

Figure 3.1: A simple diagram, visualizing the steps of the hybrid model. Source: Primary.

3.2.1 Training, Validation, and Test Split

The data is partitioned in three stages to ensure a strict temporal hold-out for final evaluation and a separate development set for model selection and early stopping. The dataset is split chronologically, simulating a real life scenario, the training set consists of transactions from 2015-2020, the validation set consists of transactions from 2021, and accordingly the test set consists of purchases that occurred during 2022. This ensures that no data from 2022 are used in any training or validation step and that the final test set remains completely unseen until the very end.

3.2.2 Artificial Neural Network with Embeddings

One of the core parts of the hybrid model is a multi-task ANN that incorporates learned entity embeddings for categorical features. By mapping each category into a trainable dense vector, the model captures intrinsic similarities among categorical values, avoiding sparse one-hot encodings. The network processes numeric and embedded categorical inputs jointly, feeding them through a deep feed-forward architecture with residual connections. This design allows the ANN to learn a rich

embedding of each data point, which is used for both a continuous value prediction and a class prediction where the classification head teaches the model broader market segments (see 3.2.2.3). The multi-task setup aims to enrich the shared representation by learning from both regression and classification targets simultaneously, improving generalization.

3.2.2.1 Input and Embedding Layers

The input features $\mathbf{x}$ consist of standardized continuous variables and categorical variables. Continuous features are input directly, while each categorical feature is handled via a dedicated embedding layer. Specifically, for each categorical field c with V_c unique values an embedding matrix $\mathbf{E}_c \in \mathbb{R}^{V_c \times d}$ with a small dimension d (e.g. $d = 8$) tuned for the task is included. These are implemented as a `ModuleDict` of embedding layers in PyTorch [31]. During a forward pass, each categorical input is transformed into its d-dimensional embedding vector, and all embeddings are concatenated with the numeric features to form the combined input. By learning embeddings, the model can place similar category values close together in the vector space, reflecting their inherent similarities, such as being in the same region or similar area. This approach not only reduces dimensionality compared to one-hot encoding, but also can reveal meaningful relationships between categories. The resulting input vector (continuous features + all embedding outputs) is then passed into the first hidden layer of the network.

3.2.2.2 Residual Stack and Embedding Head

After the input layer, the network feeds forward through a stack of fully connected layers with residual connections inspired by ResNet architectures [32]. The first layer expands the concatenated input to a high-dimensional hidden state of 512 neurons with batch normalization and ReLU activation. Then, several residual blocks follow: each block is a two-layer MLP that learns an increment $\Delta\mathbf{h}$ and adds it to the block's input via a skip connection. Formally, if a block's input is $\mathbf{h}_{in}$, it produces $\mathbf{h}_{out} = \mathrm{ReLU}(\mathbf{W}_2(\mathrm{Dropout}(\mathrm{ReLU}(\mathbf{W}_1\mathbf{h}_{in})))) + \mathbf{h}_{in}$, with a linear projection on $\mathbf{h}_{in}$ if dimensions differ. The main reasons of using a residual network rather than a purely feed-forward network was:

1. **Stabilizing gradient flow:** In a deep network, gradients can "vanish" or "explode" during backpropagation. By introducing a skip-connection that adds each block's input directly to its output, the network effectively learns only the *residual* function $\delta(h_{\text{in}})$ instead of a full transformation. This identity-mapping shortcut allows gradients to propagate freely from deeper layers all the way to the input.
2. **Non-linear interactions:** House values depends on highly non-linear interactions among features. A deeper architecture can, in principle, capture hierarchical feature interactions. Residual blocks allows the user to train a deeper stack (512 →256→ 128 neurons over two blocks) without suffering from "degradation", where additional layers actually hurt performance. Each block only needs to learn an additive correction on top of its input, so the network can gradually improve representations instead of forcing a large transformation

in one go.

3. **Faster Convergence and Reduced Overfitting.** ResNet-style blocks generally converge faster than plain MLPs of the same depth. This meant requiring fewer epochs and less aggressive regularization. A 20% dropout was used inside each residual block and so was batch normalization after each ReLU to keep embedding magnitudes consistent. This combination reduced overfitting on the training set, leading to a more robust embedding (128-D) for downstream stacking.

After the final residual block, an embedding head was applied which is a linear layer that compresses the last hidden activations into a 128-dimensional embedding vector $\mathbf{e}$. This $\mathbf{e}$ is a compact representation of the input property, integrating signals from all numeric and categorical features. It serves as the input to the subsequent output prediction heads, and also as a learned feature for the hybrid model. The use of a lower-dimensional embedding bottleneck (128-D) encourages the network to distill informative features of the data point, which was used later in the hybrid stacking approach.

3.2.2.3 Multi-Task Output Heads

From the shared embedding vector $\mathbf{e}$, the ANN branches into three output heads, one for regression, one for classification and one for structuring of the embedding space. The regression head predicts the log-adjusted price as a single scalar output. It consists of a small fully connected sub-network: a dropout layer followed by a dense layer (128→32 with ReLU) and a final linear layer to output $\hat{y}$ (log-adjusted sale price) as a single continuous value. In parallel, the classification head predicts a discrete price category indicating the relative price level of the property. Five ordinal buckets are defined by partitioning the training-set prices into 20% quintiles. The classification head is similarly a dropout plus dense layers ending in a 5-logit output $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_5)$. These correspond to the model's confidence that the object's price falls into each quantile range. A focal loss for this classification task to mitigate class imbalance was used, focusing the training on under-represented price ranges. The multi-task design provides an auxiliary learning signal, the classification objective, i.e distinguishing different price-ranges guides the network to learn features that segment properties by value, complementing the exact regression objective. Both heads share the same underlying embedding $\mathbf{e}$, so the gradients from the regression and classification tasks jointly update the preceding layers.

3.2.2.4 Composite Loss Function

The network is trained to minimize a composite loss

$$L_{\text{total}} = \underbrace{L_{\text{reg}}}_{\substack{\text{P10-aware} \\ \text{regression}}} + (1-\alpha) \underbrace{w_c\, L_{\text{cls}}}_{\substack{\text{focal} \\ \text{classification}}} + w_t \underbrace{L_{\text{triplet}}}_{\substack{\text{embedding} \\ \text{triplet}}},$$

where

- $\alpha \in [0, 1]$ is linearly annealed from α_{start} to α_{end} over training epochs.
- w_c and w_t are fixed weights for the classification and triplet losses, respectively.

3.2.2.4.1 P10-Aware Regression Loss L_{reg} Let $\hat{y}$ and y be the network's prediction and ground-truth for the standardized log-price. Define the *Huber* component

$$\delta_{Huber}(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2, & \text{if } |\hat{y} - y| \leq \delta, \\ \delta\Big(|\hat{y} - y| - \frac{1}{2}\delta\Big), & \text{otherwise} \end{cases}$$

and recover original-scale prices $p = \exp(\hat{y}\,\sigma + \mu)$, $t = \exp(y\,\sigma + \mu)$. The *soft P10* term is

$$\text{P10}_{\text{soft}} = 1 \; - \; \frac{1}{B}\sum_{i=1}^{B} \sigma\Big(k\,(0.10 \; - \; \tfrac{|p_i - t_i|}{t_i})\Big),$$

where σ is the sigmoid and k a large constant. Then

$$L_{\text{reg}} = (1 - \alpha)\,\delta_{Huber}(\hat{y}, y) \; + \; \alpha\,\text{P10}_{\text{soft}}\,.$$

An illustration of how the composite loss-function penalizes predictions is displayed in Fig 3.3.

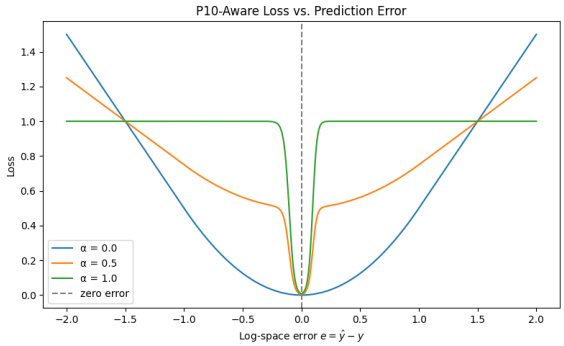

Figure 3.2: An illustration of how the composite loss-function penalizes wrong predictions as α increases. Source: Primary.

3.2.2.4.2 P10-Aware Regression Loss L_{reg} Let $\hat{y}$ and y be the network's prediction and ground-truth for the standardized log-price:

$$y = \frac{\log(\text{price}) - \mu_t}{\sigma_t},$$

where μ_t and σ_t are the mean and standard deviation of log-prices in the training set.
To recover original-scale prices, the inverse transformation is applied:

$$\hat{p} = \exp(\hat{y}\,\sigma_t + \mu_t), \quad t = \exp(y\,\sigma_t + \mu_t).$$

where $\hat{p}$ is the predicted sale price and t is the true sale price. The regression loss consists of two components:

- A *Huber loss* on the standardized log-price:

$$\delta_{\text{Huber}}(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2, & \text{if } |\hat{y} - y| \leq \delta, \\ \delta\left(|\hat{y} - y| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

- A *soft P10 loss*, which softly penalizes predictions that deviate more than 10% from the true price. It is defined using a sigmoid function:

$$\mathrm{P10}_{\mathrm{soft}} = 1 - \frac{1}{B}\sum_{i=1}^{B}\sigma\left(k\left(0.10 - \frac{|\hat{p}_i - t_i|}{t_i}\right)\right),$$

  where σ is the sigmoid function and k is a steepness constant.

The final composite loss is a convex combination of these two objectives:

$$L_{\mathrm{reg}} = (1-\alpha)\,\delta_{\mathrm{Huber}}(\hat{y}, y) + \alpha\,\mathrm{P10}_{\mathrm{soft}},$$

where $\alpha \in [0, 1]$ controls the trade-off between squared-log error and P10-aware supervision.

An illustration of how the composite loss penalizes errors at different α levels is shown in Fig. 3.3.

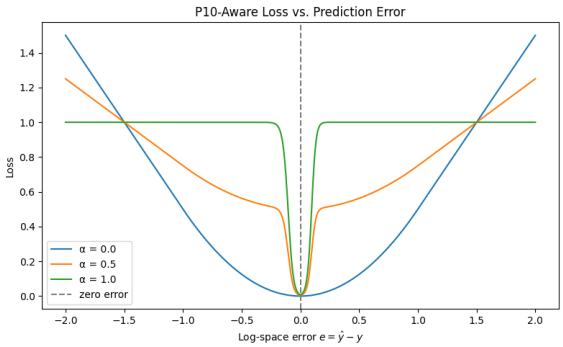

Figure 3.3: An illustration of how the composite loss penalizes prediction wrong predictions as α increases. Source: Primary.

3.2.2.4.3 Focal Classification Loss L_{cls} The classification head outputs logits for n_b price-quantile buckets. After softmax, let $p_{i,c}$ be the predicted probability for the true bucket c of sample i. The *focal loss* with focusing parameter γ is defined by:

$$L_{\mathrm{cls}} = -\frac{1}{B}\sum_{i=1}^{B}(1 - p_{i,c})^{\gamma}\,\log(p_{i,c}).$$

In Fig. 3.4 the focal loss is visualized for different values of the focusing parameter γ.

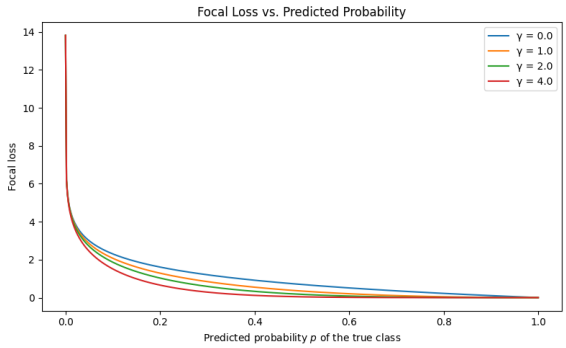

Figure 3.4: Illustration of focal loss, as γ increases, well classified examples (high p) are down-weighted, i.e their loss goes to zero faster, which helps focus the training on more difficultly classified examples (low p). The difference might look small but is quite tangible in practice. Source: Primary.

3.2.2.4.4 Triplet Embedding Loss L_{triplet} To encourage the network to learn an embedding space in which similarly priced properties lie close together and dissimilar properties are pushed apart, a triplet-based loss was implemented in addition to the regression and classification heads.

- **Price Quantile Buckets.** Before training, all sale prices in the training set are sorted and partitioned into five equal-sized buckets / quintiles.
- **Sampling Anchors, Positives, and Negatives.** During each mini-batch, *anchors* are sampled uniformly at random. For an anchor a with sale price in quantile bucket b_a, the model chooses:
  - A positive example p from the same bucket b_a, i.e. a property whose sale price falls into the same quintile as a.
  - A *negative* example n from a different bucket b_n, such that $|b_a - b_n| \geq 1$. In practice, negatives are drawn uniformly from all buckets at least one quantile away, ensuring a clear price separation.

Given embeddings e_a, e_p, and e_n, a standard margin-based triplet loss is implemented by:

$$L_{\text{triplet}} = \frac{1}{T} \sum_{(a,p,n)} \max\{0,\ \|e_a - e_p\|_2^2 \ -\ \|e_a - e_n\|_2^2 + m\}.$$

In Fig. 3.5 the Triplet Embedding Loss is visualized, with $\delta = 0.2$. This makes clear how the margin parameter creates a zero-loss region and then penalizes violations linearly.

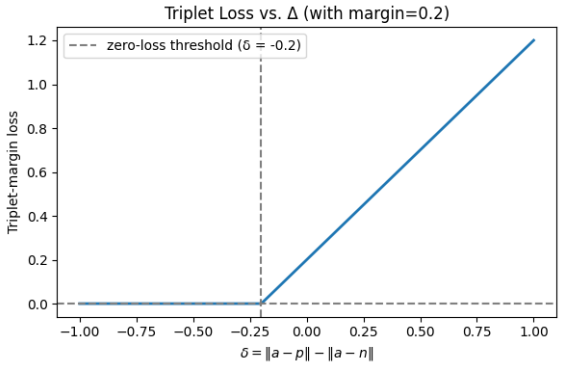

Figure 3.5: Triplet Embedding Loss. For $\delta \leq -margin$, the negative sample is at least "margin" farther than the positive -> zero loss. For $\delta > -margin$, the loss gros linearly with δ + margin.Source: Primary.

3.2.2.5 Optimization and Regularization

The ANN model was trained using the AdamW optimizer (Adam with decoupled weight decay) for efficient stochastic gradient descent. Key training hyperparameters such as learning rate, weight decay, dropout probability, and the loss weight coefficients (w_c, w_t, and the α schedule) were tuned using the Optuna hyperparameter optimization framework [33]. In particular, Optuna's TPE sampler explored ranges for the initial learning rate, the L2 weight decay penalty, the embedding dimensionality for categories, and the starting/ending values of α (which define how quickly the P10 term ramps up). Adopting Optuna [33] allowed for an efficient search of a well-performing configuration. The final chosen parameters (e.g. learning rate $\approx 2 \times 10^{-4}$, weight decay $\approx 1 \times 10^{-2}$, dropout ≈ 0.20) reflect the best trade-offs found. To train effectively, PyTorch's One-Cycle Learning Rate (OneCycleLR) [34] schedule was also applied, which cyclically adjusts the learning rate from a low value up to a peak and back down to a low value within one training run. This method, introduced by Smith [35] for "super-convergence", allows the model to use a relatively high learning rate briefly and often leads to faster convergence and better generalization.

Batch normalization was also applied in each layer to stabilize learning and dropout in the hidden layers and output heads to reduce overfitting by randomly deactivating neurons during training. The trade-off parameter α, which controls the balance between Huber loss and soft P10 supervision, was scheduled to increase linearly over training. Specifically, α started at approximately 0.19 and increased to 0.63 by the final epoch. This gradually shifted emphasis from minimizing squared error on log-price to optimizing the soft P10 metric on original-scale prices. This schedule gave the model time to learn an accurate overall fit before focusing too much on the stricter P10 criterion. Simultaneously, the classification loss weight w_c was effectively scaled by $(1 - \alpha)$, so that as α grew, the classification task was gradually down-weighted to zero towards the end of training. This ensured that in later epochs the model concentrates on P10 and embedding structure, having already benefited from the classification signal early on. Early stopping was introduced on MAPE as

well as increasing validation error. The best model, with lowest validation MAPE was saved for final evaluation. PyTorch's ReduceLROnPlateau scheduler [36] was also used as a fallback, if progress stagnated, the learning rate would be halved after 5 epochs of no MAPE improvement. However, with OneCycleLR in effect, this was rarely needed until the very end of training.

3.2.3 LightGBM Ensemble with Raw Features and ANN Embeddings

In the hybrid valuation framework, two LightGBM regressors are employed in a stacking configuration. LightGBM is known for its efficiency and accuracy, training faster than traditional GBMs while maintaining similar accuracy, making it suitable for the large feature set. The two-stage ensemble is outlined in the following subsections.

3.2.3.1 Stage 1: Raw-Feature GBM

In Stage 1, the LightGBM regressor is trained on the same scaled continuous features as the ANN, but the original categorical columns are kept as pandas Categorical dtype [29], so that LightGBM can handle splits on them natively while predicting the log-adjusted sale price.
The Stage1 LightGBM was tuned with Optuna as well, optimizing hyperparameters like number of leaves, learning rate, feature fraction, and regularization terms. The objective was standard regression, i.e minimizing MAPE as the evaluation metric. The LightGBM was trained with early stopping on a validation set to determine the optimal number of boosting rounds. The Stage 1 model learns a baseline mapping from raw inputs to price. For instance, it can directly learn effects like "houses in region X are more expensive" or "larger living area increases price," and so on, by leveraging decision-tree splits. After training, the Stage 1 predictions are produced. Denote $\hat{y}_0(i)$ as the Stage1 predicted log-price for sample i.

3.2.3.2 Stage 2: Embedding-Based Residual GBM

For Stage 2, a second LightGBM model is trained to predict the residual $r(i)$ using the ANN's learned embedding as input. Essentially, Stage 2 is learning to predict what Stage 1 missed, but only using the information encoded in the embeddings $\mathbf{e}$. The Stage2 LightGBM also uses a set of Optuna tuned parameters. It trains on the dataset $(\mathbf{e}(i), , r(i))$ with an objective of regression on MAPE as well. Because the range of residuals is smaller than the original target, this stage can focus on finer details. For example, the ANN embedding might encode subtle interactions which the model can pick up on by splitting on $\mathbf{e}$ dimensions. Given all the complex signals the ANN captured, the Stage 2 model tries to find the remaining price adjustment that needs to be added to the Stage 1 prediction. Typically, Stage2 required fewer trees than Stage 1, as the residual signal is weaker than the original. After training, the model outputs a residual correction $\hat{y}_1(i)$ for each input embedding. This model effectively boosts the performance of the ensemble by adding back the nonlinear,

interaction-driven effects that a single GBM could not easily find from raw features only.

3.2.3.3 Combined Prediction and Performance

The final prediction for a given property is the sum of Stage 1 and Stage 2 outputs: $\hat{y}_{\text{final}} = \hat{y}_0 + \hat{y}_1$. Where $\hat{y}_0$ is the Stage1 GBM's prediction using raw features, and $\hat{y}_1$ is the Stage2 GBM's predicted residual using the ANN embedding. The two terms together give the full predicted log-price, that is then exponentiated to obtain the predicted sale price in SEK.

3.3 Benchmark Models

Two simple baseline models were constructed to benchmark the proposed hybrid ANN approach: a classical hedonic regression and a straightforward KNN regression. These models serve as interpretable, traditional baselines for comparison.

3.3.1 Hedonic Regression Baseline

The code implements a standard procedure for training and evaluating a hedonic regression model using the same log-adjusted sale price as the target variable. The linear regression model is fitted on the training set, with both the predictors and the log-transformed sale prices. Predictions for the test set are generated in the log-price space and subsequently exponentiated to return to the original price scale. Model performance is then assessed on the natural price scale using the same evaluation metrics as for the other models.

3.3.2 KNN

The kNN regression model predicts property prices by averaging the prices of the nearest training examples in feature space. For this implementation, each property identifies the five most similar neighbors using standard Euclidean distance across identical features. Predictions are generated through uniform weighting, meaning each neighbor contributes equally. Same as for the rest of the models, the KNN regression was applied to the log-transformed price target. Fig. 3.6 illustrates a simple example of this method: the price of a new house of $180m^2$ (indicated by the vertical dashed line) is predicted by averaging the prices of its 5 closest neighbors (marked in orange) within the training dataset.

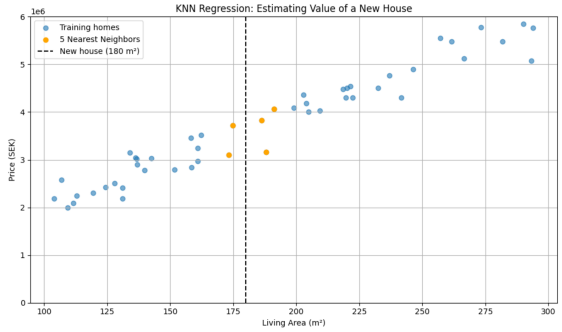

Figure 3.6: Illustration of KNN regression. A new house (vertical dashed line at 180 m^2) is valued by averaging prices of its 5 nearest neighbors (orange points) among a sample of training homes. Source: Primary.

3.3.3 Baseline Model Configurations

Table 3.1 summarizes the key settings of each baseline model. The hedonic regression has no adjustable hyperparameters, while the kNN model's main parameter is K (the number of neighbors).

Table 3.1: Overview of baseline models and their configurations

Model	**Target Variable**	**Key Settings**
Hedonic Regression	`log-adjusted sale price`	OLS linear regression on structural & locational features; no hyperparameter tuning.
KNN Regression	`log-adjusted sale price`	k-nearest neighbors (e.g. k=5), Euclidean distance, uniform weighting.

The comparative evaluation was then carried out on an identical held-out test dataset for all models. Using the same test set for each model ensures a fair, direct comparison of predictive accuracy no model has an advantage from different data splits. The same set of error metrics was applied to each model's predictions. In this way, the hybrid model is benchmarked against both conventional methods.

4

Summary of Findings

4.1 Comparative Evaluation

Table 4.1 summarizes the test-set performance of each modeling approach. The hybrid model emerges as the top performer across every metric. Although its improvements over the raw-only GBM might seem modest, they are consistent and meaningful in a valuation context.

Table 4.1: Test Set Performance Comparison of All Models

Model	MAPE	MAE	RMSE	R^2	P10	P20
Hybrid Model	15.9	431 926	624 460	0.814	41.4	73.6
Raw features LGBM	17.1	464 525	665 020	0.798	38.6	68.0
Embeddings LGBM	19.3	481 560	739 780	0.751	37.3	63.6
Neural Network	18.6	481 290	737 220	0.752	38.9	65.1
KNN	23.6	619 170	963 270	0.577	28.6	53.0
Hedonic Regression	22.8	552 140	799 780	0.709	30.0	56.0

The hybrid model reduces MAPE by over 1 percentage point (pp) relative to the raw-only GBM (15.9% vs. 17.1%), translating into an average error reduction of roughly 33,000 SEK. This means more appraisals fall closer to their true value. The RMSE also decreases by roughly 41,000 SEK. The hybrid model scores an R^2 of 0.814 which means that the model explains 81.4% of the variance in sale price in the test set.

Turning to coverage metrics, the hybrid's P10 of 41.4% signifies that four out of ten valuations lie within ±10% of the sale price, compared to only 38.6% for the raw GBM. Similarly, P20 improves by 5.6pp. These gains reflect a clear tightening of the error distribution, which can translate to stronger confidence intervals in practice.

The embeddings-only GBM and the standalone neural network both underperform the raw GBM, confirming that while learned embeddings excel at capturing complex, nonlinear feature interactions, they do not substitute for the breadth of information contained in the original variables. Embeddings distill higher-order patterns, but require the raw features to ground those patterns in measurable property attributes.

By contrast, the kNN and hedonic regression baselines underperform substantially on every metric. Hedonic regression, relying on linear relationships and pre-specified interaction terms, struggles to accommodate the irregular, multimodal distributions of property characteristics outside major urban centers, without proper, careful and

thorough pre-processing. Likewise, kNN depends on finding truly comparable sales in the training set; in thin markets or highly heterogeneous rural regions, suitable comps may be sparse or distant in feature space, leading to noisy, unstable estimates. This sharp underperformance of classical methods underscores the sheer difficulty of automated property appraisal in diverse, data-sparse contexts. Real estate markets outside metropolitan areas exhibit wide variability in lot sizes, building styles, renovation levels, and locational premiums that differ from the smoothness and homogeneity assumptions of simple regression or nearest-neighbor approaches. In such settings, hybrid models that combine global pattern-learning and local context provide the flexibility and robustness needed to attain practical accuracy.

4.1.1 Model Performance by Price Decile

To understand how valuation accuracy varies across the price spectrum, test-set properties were grouped into ten equally sized deciles by true sale price. For each decile, Table 4.2 reports the average true price, the model's mean prediction, and key error metrics: MAE, MAPE, P10 and P20.

Table 4.2: Hybrid Model Performance by True Price Decile

Decile	True Mean	Pred Mean	MAE	MAPE	P10	P20
0	1,326,602	1,237,231	203,372	13.85%	43.5%	74.8%
1	1,461,995	1,428,079	274,347	18.36%	30.5%	58.3%
2	1,681,692	1,628,669	318,380	18.87%	32.4%	62.0%
3	1,903,869	1,839,429	365,234	19.41%	32.1%	60.5%
4	2,155,763	2,093,139	407,223	19.86%	34.0%	61.0%
5	2,480,797	2,399,840	435,851	18.32%	39.3%	65.5%
6	2,853,116	2,771,432	484,960	18.12%	39.8%	69.3%
7	3,321,370	3,249,310	468,718	14.94%	47.8%	76.8%
8	3,994,205	3,867,537	519,312	13.14%	49.5%	79.9%
9	5,597,320	5,315,956	839,685	14.72%	45.5%	75.6%

Takeaways:

- **Low-to-Mid-market challenge (deciles 1–4):** MAE and MAPE peak in the second through fifth deciles, and P10 dips to its lowest. These low-to-mid-range properties exhibit the greatest disparities in features, making precise valuation more difficult.
- **Improved accuracy at extremes:** Both the lowest decile (0) and the top three deciles (7–9) show stronger P10 and lower MAPE. In the cheapest segment, homes are more homogeneous, while in the mid-to-high-value tiers, the model excels at identifying objects, similar to mid-value properties, but with slight improvements across the object-specific features. However, for the most expensive of objects (found in decile 9) the object struggles to make correct valuation, for the very most expensive properties ($\approx$ 12M+) mostly due to the small sample size of very expensive properties in the training set.
- **P20 stability:** The P20 metric remains above 58% across all deciles, peaking

at nearly 80% for decile 8. This indicates that even when ±10% accuracy is challenging, the model still generally stays within ±20% of sale price.

- **Systematic underestimation:** Across all deciles, the model underestimates the value. The predicted mean in decile 9 (5.32 M SEK) is slightly below the true mean (5.60 M SEK), aligning with the MAPE and MAE increases, suggesting a modest bias that could be addressed by targeted calibration in the highest price bracket.

4.2 Error analysis

This section quantifies how well the hybrid valuation model performs and where it fails. With aggregate views, histograms, and scatter plots that reveal the overall spread and systematic biases of its residuals, and then drill down to illustrative case studies that expose the specific transactions driving the largest misestimations.

4.2.1 Error Distribution

Fig. 4.1 shows the distributions of absolute and relative errors for the hybrid model on the test set. The absolute-error histogram is tightly concentrated: over 50% of predictions fall within ±400 000 SEK, and only 5% exceed 1 000 000 SEK. The relative-error plot confirms that more than 40% of predictions lie within ±10% of actual price (P10), and roughly 75% within ±20% (P20). This error profile indicates that the hybrid model delivers both small typical errors and a compressed tail of large misvaluations critical for reducing risk in automated appraisal.

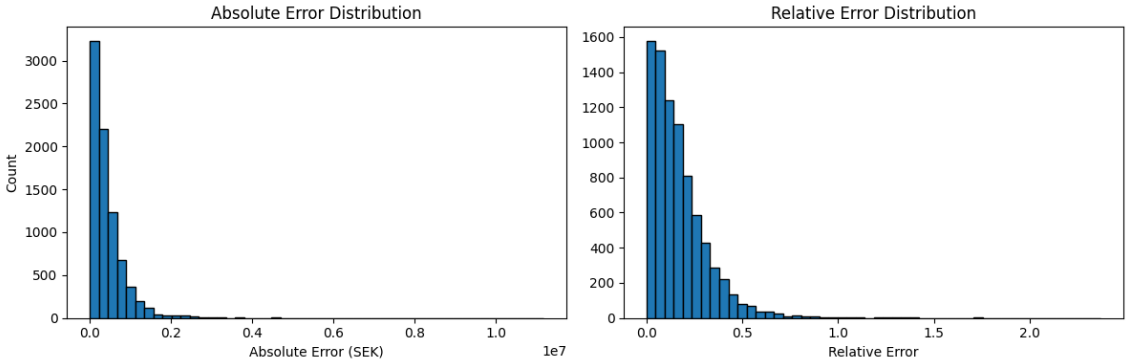

Figure 4.1: These two plots show the Absolute Error Distribution (left) and Relative Error Distribution (right) for the Hybrid Model.

The scatter plot in Figure 4.2 compares true sale prices against model predictions. Most points lie close to the 45° line, demonstrating a relatively accurate fit across price ranges. A heavy under-prediction bias appears as the prices increase, though mostly due to the lack of expensive objects in the entire dataset. Overall, the scatter confirms that the hybrid stack generalizes quite well and maintains linearity between predicted and actual values.

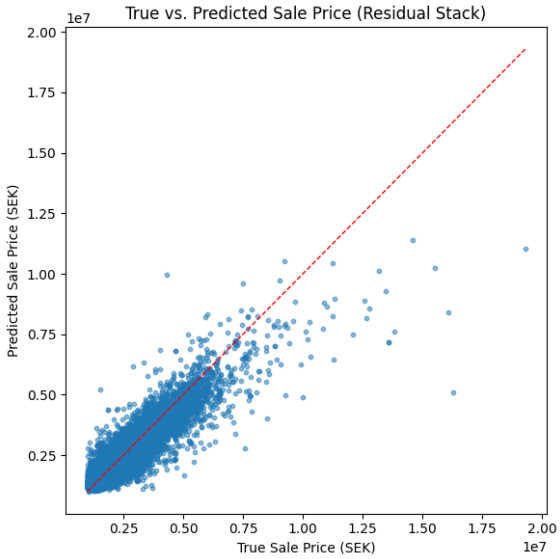

Figure 4.2: Scatter plot of all true (x-axis) and predicted prices (y-axis), if all predictions were totally correct, they would align with the red dotted line.

4.2.2 Case Studies: Best and Worst Predictions

Table 4.3 lists five, respectively seven, examples of the lowest and highest absolute errors. The best-predicted properties tend to be around mid-market, but as was evident in Fig. 4.2, the model successfully makes good estimations even in higher price segments. Conversely, the worst estimations almost exclusively fall into the most expensive range of properties in the test set. This holds for almost all of the 50 worst predictions as well, with two exceptions: one object, whose adjusted sale price was 4,313,017 SEK, was valued at 9,456,976 SEK by the model, almost double the actual price, and another, priced at just 1,548,869 SEK, was overestimated at 5,222,245 SEK.

Table 4.3: The table displays the best and worst predictions made by the model in absolute terms. The sale prices are still adjusted to 2020-06, hence the strange price sequences.

Best Estimates by the Hybrid Model SEK

	Adjusted sale price	Predicted Sale Price	Absolute Error
1	1,466,269	1,466,336	67
2	3,248,320	3,248,389	69
3	1,881,336	1,881,457	121
4	2,481,618	2,481,743	124
5	2,616,279	2,615,984	295

Worst Estimates by the Hybrid Model SEK

	Adjusted sale price	Predicted Sale Price	Absolute Error
1	16,298,297	5,117,908	11,180,389
2	19,281,824	11,055,575	8,226,249
3	16,071,772	8,416,230	7,655,542
4	13,593,189	7,154,120	6,439,069
...	...	...	...
7	4,313,017	9,956,976	5,643,959
14	1,548,869	5,222,246	3,673,377

4.2.3 Case Studies of Selected Transactions

It is expected that many of the worst predictions would fall into the high-end market segment in terms of absolute error. As mentioned multiple times before, the dataset did not contain a sufficient portion of very expensive homes that the model could learn from. However, the two bottom predictions in table 4.3 are not part of the most expensive price segment, something that needs to be investigated. Table 4.4, shows a feature comparison between two instances in the data set, one from the test set (Anomaly) and one from the training set (Similar object).

Table 4.4: Comparison of the model's 7th worst prediction and its nearest neighbors (NN1)(see Table 4.3) between two nearly identical property records, one from the training set (true sale price 12,500,000 SEK) and one from the test set (true sale price 4,313,017 SEK), showing adjusted true vs. predicted sale prices and key features(more of key features in section 4.4), highlighting a likely duplicate entry.

Example of a Potential Duplicate Transaction		
	Anomaly	Similar object
Adjusted Sale Price (SEK)	4,313,017	12,500,000
BuildingAge (years)	113	109
UtilityArea m^2	187	187
Lot Area m^2	548	548
QualityScore	27	27
Closetobeach (1/0)	0	0
DistmediumCity (m)	681.55	681.31
Distcoast (m)	363	363
strand	4	4
Deso Class	C	C

When evaluating the test data, it was observed that the property with a sale price of 4,313,017 SEK was valued by the model at 9,883,873 SEK, resulting in an absolute error of 5,570,856 SEK. Examining the values for both objects in the table reveals that they are highly similar, in fact almost identical, across key features. Furthermore, an analysis of their respective longitude and latitude coordinates confirmed that the two transactions certainly correspond to the same object. The reason behind the same object being sold for almost a third of what it had been sold for just four years prior is not apparent. It could be due to an external action, the property is quite large and could have been subdivided into a two-family building, and that the data for the building simply has not been updated accordingly. It is unclear how many such "identical" or misleading entries exist within the dataset, and no thorough investigations were made.

Another anomaly, i.e a relatively cheap object with a high absolute prediction error, is found in Table 4.5 together with some similar objects.

Table 4.5: Comparison of the model's 14th worst prediction (see Table 4.3), between transaction the *Anomaly* and its five nearest neighbor (NN1–NN5), showing true sale prices and key features.

Summary of Property Transactions						
	Anomaly	NN 1	NN 2	NN 3	NN 4	NN 5
Adjusted sale price (SEK)	1 548 868	6 829 478	6 550 359	6 056 122	5 725 983	4 684 105
BuildingAge (years)	26	24	27	6	14	27
UtilityArea (m^2)	199	168	145	229	220	147
Lot Area m^2	1742	829	816	1795	2671	937
QualityScore	36	26	30	32	44	30
Closetobeach	0	0	0	0	0	0
DistmediumCity (km)	5.04	4.54	2.73	18.70	15.45	5.86
Distcoast (km)	6.66	0.093	126.40	0.39	5.3	39.57
strand	4	3	4	4	4	4
Deso Class	C	C	C	C	B	C

The left-most column in Table 4.5 corresponds to the *Anomaly*, an object priced at 1,548,868 SEK. The model predicts a sale price of $\hat{y} = 5{,}185{,}451$ SEK, whereas the actual price is only $y = 1{,}548{,}868$ SEK. This yields an absolute error of $|\hat{y} - y| = 3{,}636{,}583$ SEK, corresponding to a relative error of 235 %. A row-wise scan of Table 4.5 reveals that the *Anomaly* is not easily separated from its five nearest neighbours (NN 1–NN 5) across all high-importance features. These include `BuildingAge`, `UtilityArea`, `LotArea`, `QualityScore`, the socio-economic `DesoClass`, and categorical indicators like `strand`. There is no standout covariate that would suggest this object should be treated differently by the model. In other words, the *Anomaly* is fully embedded in the typical feature space.

In contrast to its feature similarity, the *Anomaly* is dramatically dissimilar in price. While its neighbors all transacted between 4.7 and 6.8 million SEK, the *Anomaly* closed at just 1.55 million SEK, a discount of 65% to 77 % relative to every peer. Thus, price becomes the only truly anomalous dimension for this transaction.
Additionally, distance-based features such as `DistCoast` offer little help. The *Anomaly* lies 6.6 km from the coast, but the five nearest neighbors span a wide range from just 93 meters to 126 kilometers. The variation within this group weakens the predictive signal in that dimension, making it unlikely that the model can rely on it to adjust for the price outlier.

Feed-forward ANNs are well-suited to learning smooth, high-frequency patterns in feature space. Given that the *Anomaly's* input vector $\mathbf{x}_{\text{Anomaly}}$ closely resembles those associated with sale prices in the 5–7 million SEK range, the model naturally maps it into that price manifold. From the network's perspective, there is no statistical precedent suggesting that a home with such observable features can transact at 1.5 million SEK. It therefore extrapolates upward in a way that is rational from a data-driven standpoint.
Furthermore, an external valuation benchmark was found on the *Anomaly* on *Booli* [37]. Booli is a Swedish real estate platform offering comprehensive housing market data, now owned by SBAB Bank. It provides users with access to current property

listings, historical sale prices, area-level trends, and market statistics across Sweden. Booli's platform serves home buyers, sellers, and investors seeking data-driven insights into the property market.

One of Booli's core services is its automated property valuation tool. In this context, Booli's automated valuation of the *Anomaly* was 3,540,000 SEK, substantially higher than its realized sale price of 1,548,868 SEK but still below the price predicted by the model. This reinforces the notion that, even from the perspective of an independent, market-wide algorithm, the sale price of the *Anomaly* stands out as an extreme outlier [37].

In summary, the *Anomaly* is not a failure of the model but rather a reflection of data limitations. While the model is trained on a rich feature set, the analysis presented here focuses on a carefully selected subset of the most important features, those that contribute most significantly to price prediction according to feature importance metrics. Displaying the entire feature space would obscure the interpretability of the analysis and offer limited additional insight.

As shown, the *Anomaly* is virtually indistinguishable from its nearest neighbours across this high-importance subset, yet its price deviates dramatically. This highlights a fundamental limitation: if two homes appear nearly identical in all observable and influential aspects but sell for vastly different prices, a model, even one trained on a comprehensive feature space, cannot be expected to resolve such discrepancies without access to additional factors. Without access to richer data or targeted algorithmic adjustments, large errors for cases like the ones investigated are not only unsurprising, they are inevitable.

4.3 Embedding Analysis

In the following subsections, the model's embedding space is explored to uncover its key patterns and insights.

4.3.1 Embeddings Clustering

K-means clustering ($k = 5$) was applied to the 128-dimensional embeddings produced by the neural network's projection layer for all training samples.

There is a boxplot in Fig. 4.3, where sale prices by cluster reveals five distinct tiers:

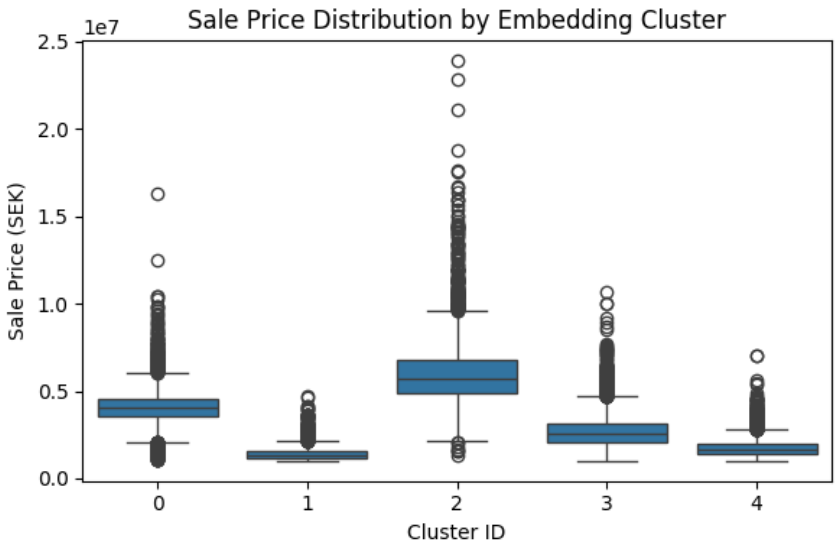

Figure 4.3: Boxplots of sale-price distributions for five clusters obtained by applying K-means to the 128-dimensional neural network embeddings.

- **Cluster 0** Mid-high-market segment (median ≈ 4.0 M SEK) with moderate interquartile range and a few high-price outliers.
- **Cluster 1** Low-priced homes (median ≈ 1.2 M SEK) showing a tight distribution and minimal skew.
- **Cluster 2** High-end homes (median ≈ 5.8 M SEK) with a pronounced long upper tail.
- **Cluster 3** Mid tier (median ≈ 2.8 M SEK) exhibiting the widest overall spread and several extreme values.
- **Cluster 4** Low-mid-level units (median ≈ 1.7 M SEK) with relatively low interquantile range but some upper-end outliers.

The plot shows that embeddings naturally partition the data into value-driven groups beyond any single raw feature.

4.3.2 t-SNE Projection of Embeddings

Figure 4.4 presents a two-dimensional t-distributed Stochastic Neighbor Embedding projection of the 128-dimensional neural network embeddings for all test transactions. Each point corresponds to a single transaction and is colored according to its log-sale price. A smooth gradient from low to high prices is apparent, with higher-priced properties clustering in a distinct region of the embedding space. This continuous organization demonstrates that the learned embeddings capture price information in a structured manner, mapping gradual increases in sale price onto similarly gradual transitions in embedding coordinates.

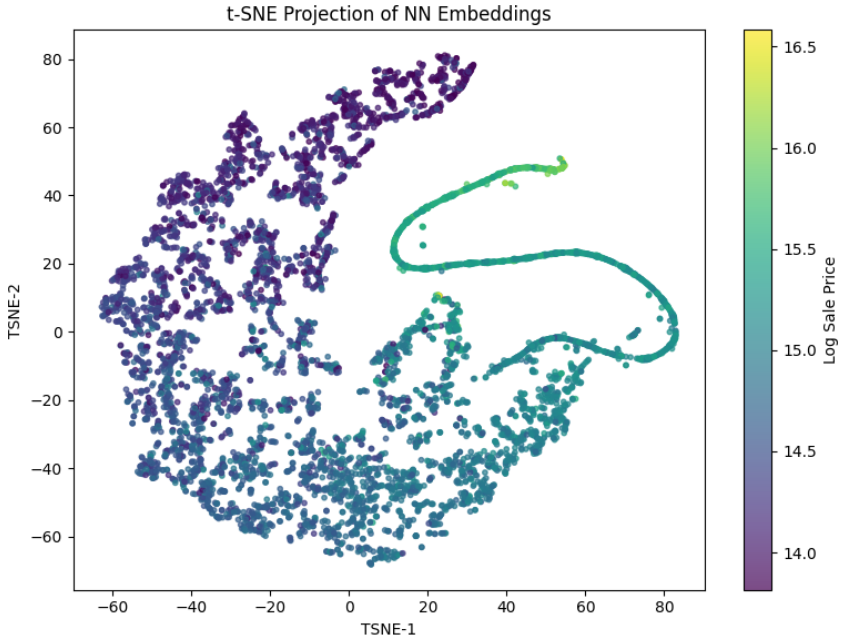

Figure 4.4: Two-dimensional t-SNE projection of the 128-dimensional ANN embeddings for each property, colored by log sale price. Points that cluster together share similar learned representations.

4.3.3 Embedding-Feature Correlation Analysis

To interpret what the learned embedding dimensions capture, the absolute Pearson correlation was computed between each embedding coordinate and all original numeric and categorical-code features. Here are some key takeaways:

- **Educational and demographic signals:** Many of the most important dimensions correlate strongly with educational features, such as the fraction that attended post-elementary education and higher education. These embedding axes clearly encode neighborhood education-level statistics.
- **Population and area metrics:** Many embeddings show high correlation with municipality population, its deviation and its total and its annual (and biannual) change, indicating that latent dimensions capture market size and growth dynamics.
- **Spatial proximity:** Several embeddings also correlate with distance to medium sized cities, and distances to points of interest, such as golf courses, confirming that network-learned axes encode locational gradients.
- **Heterogeneity across dimensions:** While some dimensions focus on socioeconomic factors, others capture built-environment attributes, demonstrating that the embedding space distributes different types of signals across separate axes.
- **Object-specific dimensions:** Many dimensions also focus almost solely object-specific features, such as living area, lot area, distance to beach etc, meaning that the network encodes a combination of attributes that pushes value.

4.4 Model Interpretability

The following subsections explore the model's prediction process by examining feature contributions and importance using various interpretability methods.

4.4.1 SHAP Analysis on Raw Features

The individual feature contribution in SHAP is visualized in Fig. 4.5 for 20 of the most important features.

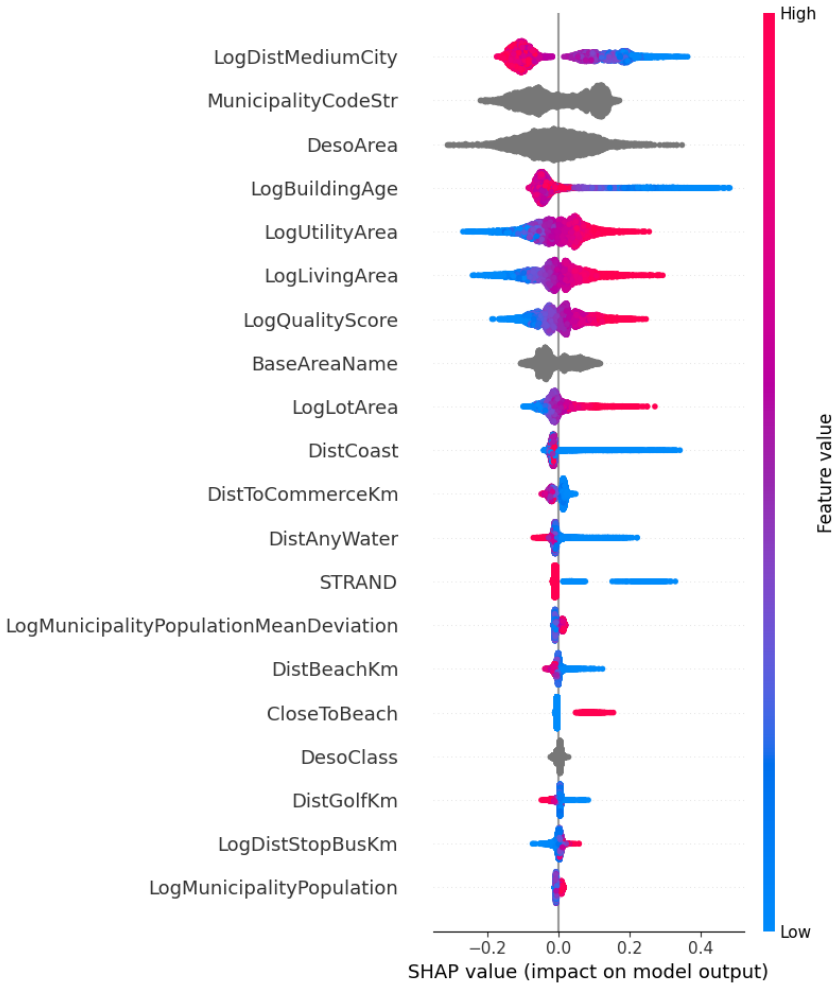

Figure 4.5: SHAP summary plot for the model, showing each feature's contribution to the predicted price (x-axis) and the distribution of feature values (color) across observations

The topmost feature is LogDistMediumCity: its red points (high values) lie on the left (negative impact), and blue points (low values) on the right (positive impact). This indicates that larger distance to a medium-sized city lowers the predicted price, whereas being closer (small distance, blue) raises it. This aligns with empirical findings that home values decline with distance from city centers [38]. The second and third features from the top, `MunicipalityCodeStr` and `DesoArea`, are categorical

location codes. They exhibit nearly symmetric, grey-colored distributions around zero, meaning they adjust baseline prices up or down by municipality or district but show no clear monotonic trend. In other words, different municipalities or DeSO areas simply shift the model output without a single direction of effect since they are merely categorical.

4.4.1.1 Property Attributes and Size/Quality Effects

The model's next most important features are structural attributes of the property. `LogBuildingAge` has high (red) points on the left and low (blue) on the right: older buildings reduce predicted price while newer ones increase it. This aligns with expectations, as older buildings are often valued lower. Similarly, `LogUtilityArea` and `LogLivingArea` show opposite patterns: high values (red) appear on the right (positive SHAP) and low values (blue) on the left (negative SHAP). In plain terms, properties with larger areas push the price higher, whereas very small area properties reduce it. This positive area price correlation is well known and also aligns with what is expected. `LogQualityScore` also follows this trend: high-quality homes (red) drive the prediction up, while low-quality homes (blue) drag it down which. `LogLotArea` shows a much weaker effect: a small red cluster on the positive side suggests larger lots slightly raise price, but the distribution is narrow and mostly centered, implying only a marginal influence, probably due to the fact that very remote homes tend to have larger lots, but will still tend to be cheaper than suburban villas with smaller lot area. This creates a contextual monotone feature, where for example a house with a a *large* value for `LogLotArea` only see a major price push in the positive direction if `LogDistMediumCity` is relatively *low*. All of these clearly monotonic features make for a good sanity check of the model, which clearly picks up on intuitive and expected trends.

4.4.1.2 Location

Proximity to points of interest and natural features is another pattern in the plot. `DistCoast` shows red points (far from coast) on the left and blue points (near coast) on the right, but not completely as monotonic as one might expect due to known coastal premiums. Since the dataset contains houses from Småland, Östergötland, Blekinge and Gotland, this feature might be a bit unfair, since the biggest cities in both Småland and Östergötland are located very much inland, and thus score very low values on `DistCoast`. However, other water related features reinforce the expected waterside premiums and show clearer monotonic trends: `STRAND` (beach indicator), `CloseToBeach`, and `DistBeachKm` all show positive impacts for waterfront properties (blue points on the right) and negative impacts for inland properties (red on the left). The `DistAnyWater` feature similarly clusters with minimal spread but suggests being near any water slightly benefits price. The `DistToCommerceKm` distribution is centered with a weak trend (blue on right, red on left), implying closer proximity to commercial centers can modestly raise value. `DistGolfKm` has a few blue points on the positive side, indicating slight premiums for homes near golf courses. Overall, amenity proximity features tend to form a cluster of mostly blue (high-benefit) points on the positive side and fewer red points on the negative side,

reflecting that proximity to beaches, coastlines or amenities generally increases the model's price prediction.

4.4.1.3 Categorical Location Effects

Several features are categorical indicators of region or class. `BaseAreaName` show wide grey distribution. It points spread moderately on both sides of zero, implying that certain base areas can either increase or decrease price but on average have a small effect. `MunicipalityCodeStr` and `DesoClass` similarly produce nearly symmetric grey clouds. Because these are labels rather than ordered values, high vs low coloring is not meaningful; instead, these features allow the model to capture area-specific adjustments. In practice, a specific base area or DeSO class may carry its own price offset (positive or negative SHAP), but there is no consistent directional gradient across values.

4.4.2 Raw Feature Importance by Gain

The plot of feature importance by gain in Fig. 4.6 for the Raw-BGM model reinforces several key insights already highlighted by the SHAP analysis. Where gain importance for a feature is measured by the total reduction in the loss function that the model achieves when it splits on said feature.

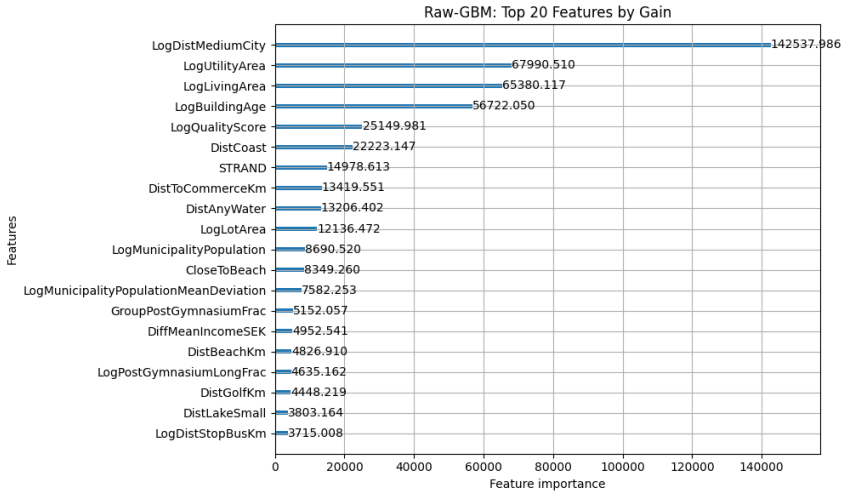

Figure 4.6: Feature importance plot illustrating the top 20 influential raw features in the Light GBM model, ranked by gain (total reduction in the loss function). The horizontal bars represent the relative contribution of each feature to the predictive performance, highlighting LogDistMediumCity, LogUtilityArea, and LogLivingArea as the most impactful features for predicting real estate prices.

Many of the most important features by gain are highlighted in Sec. 4.4.1, features that were not deeply touched upon were demographic and categorical variables such as `LogMunicipalityPopulation` and `LogMunicipalityPopulationMeanDeviation` which show intermediate importance, suggesting the model captures certain regional pricing nuances. While the SHAP analysis showed that these features mainly cause

small price adjustments without a consistent directional effect, the feature importance plot reflects their overall predictive contribution. However, given that the dataset includes only six counties, two of which contain very few observations, these features may largely act as proxies for municipality identity rather than broader demographic effects. This limited variability may inflate their importance and should be interpreted with caution.

Overall, the feature importance plot strongly corroborates the SHAP analysis findings, underscoring the consistent dominance of urban proximity, property structural characteristics, and amenity access in determining real estate prices, while validating the nuanced effects of regional demographic factors.

4.4.3 Quantified Embedding Importance

The neural embeddings learned by the representation network can capture complex, non-linear combinations of raw features that are not directly visible in the original feature space. This section analyzes the importance of the embedding dimensions.

4.4.3.1 Gain-Based Embedding Importance

Figure 4.7 shows the top 20 embedding dimensions ranked by gain (total reduction in the loss function) in the hybrid model.

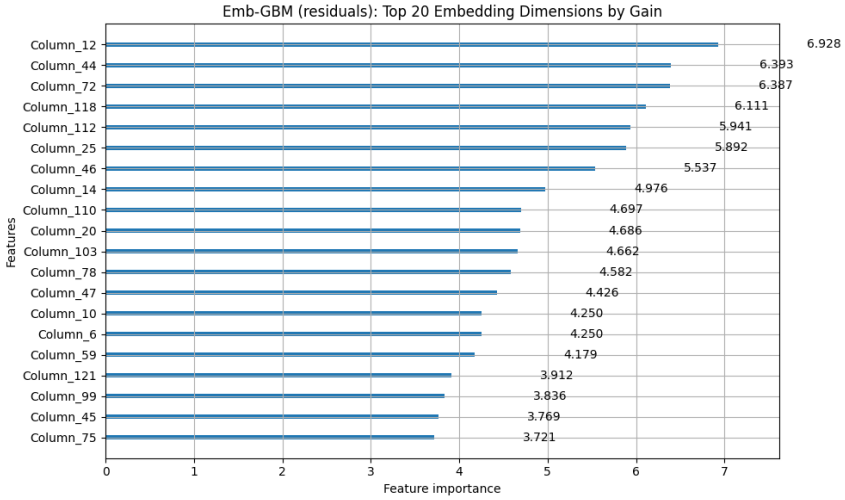

Figure 4.7: The plot shows the embedding gains on the residuals of the Light GBM on raw features. The gain refers to the reduction of loss function

In comparison to the gain plot for the raw features (see Fig. 4.6) the difference in embedding gain are not as distinct, meaning that the values are not as clearly correlated, or driven by a select few of the embeddings.

4.4.3.2 SHAP Analysis on Embeddings

The SHAP summary plot in Figure 4.8 displays the distribution of SHAP values for 20 embeddings, colored by embedding value.

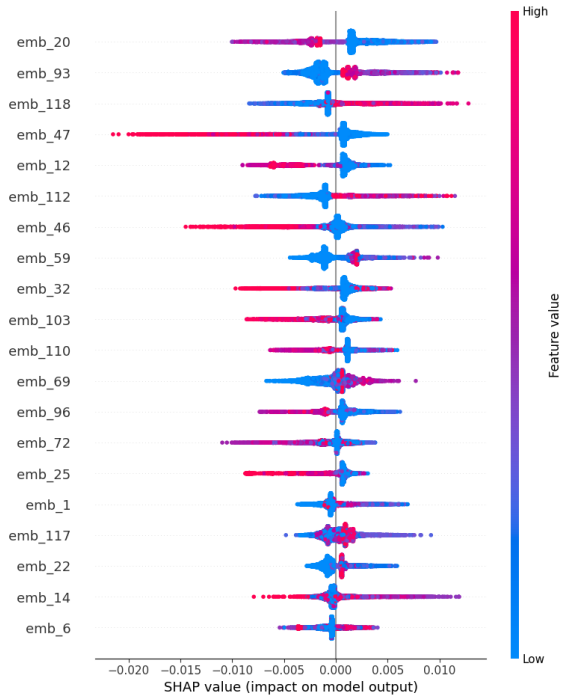

Figure 4.8: SHAP summary plot for 20 embedding dimensions, showing each embedding's impact on predicted price and its value distribution.

Several embeddings exhibit strong positive or negative impacts when their values are high or low. For instance, high values of `emb_112` (red) tend to increase predicted prices, whereas low values (blue) decrease them, suggesting this embedding captures a feature of strong price uplift. However, it is clear the the SHAP values are not as monotonic as for the raw features (see Fig. 4.5) this is expected since the embeddings are complex combinations of a multitude of features and all of these might not formulate a straight forward connection between high/low general values in the embeddings to an increase or decrease of sale price.

4.4.3.3 Case Studies of Three Different Embedding Dimensions

Below are illustrative case studies for `emb_112`, `emb_118`, and `emb_20`, all three driving embeddings but very different in what they consist of. Below is a representation of what the most important features in these three embeddings represent.

Embedding 112 feature correlation:

- `QualityScore`
- `UtilityAre`
- `LivingArea`

Together, these suggest that this dimension encodes a collection of object-specific features which encapsulates a lot of very much tangible features.

Embedding 118 feature correlation:

- `PostGymnasiumLongFrac`
- `GroupPostGymnasiumFrac`
- `GymnasiumFrac`

This indicates that this dimension primarily captures nuances of education level.
Embedding 20 feature correlation:

- `LogDistMediumCity`
- `LogDistToCommerceKm`
- `DistAnyWater`

This dimension appears to mostly encode distances to points of interest, where low values tend to increase sale price.

4.5 Demographic statistical areas analysis

The DeSO area code is constructed from the county-municipality code, followed by the area type as outlined above, a sequential number, and an additional character used in cases where the area is subdivided or modified. Each DeSO code are also given a DeSO code (A,B or C) which serves as an indicator on proximity to major urban areas:

- Type A: These areas are predominantly rural, located far from major population centers.
- Type B: These correspond to semi-urban areas, often smaller suburbs or communities located outside of urban centers.
- Type C: These are subareas within large urban centers. In defining these areas.

These different classes provide diversity in the dataset, and as shown in Tab. 4.6, the model performs quite differently across these different classes on the test set.

Table 4.6: Performance Metrics by DesoClass

DesoClass	N	P10 (%)	P20 (%)	MAE	RMSE	MAPE (%)
A	2529	31.356	58.917	468224.402	734211.666	20.235
B	1318	39.454	68.589	396797.314	573806.253	16.555
C	4385	42.714	72.771	426220.474	643610.158	15.335

The hybrid model achieves its strongest results in Class C areas. In the test set, Class C shows the highest P10 and P20 and the lowest MAE, RMSE, and MAPE of all classes. By contrast, Classes A and have much lower P10/P20 and higher error metrics. In other words, Class C predictions are closest to the true values on average, while Classes A/B exhibit larger absolute and percentage errors.

- Data density: Class C provides a much larger training sample size. More data adds diversity and decreases generalization error. The large number of examples in Class C helps the model learn robust patterns, raising P10/P20 and reducing MAE/RMSE.
- Feature consistency: Dense areas tend to have more uniform, consistent features. The model sees repeatable patterns, so it can predict stably. In B and especially A, features are more varied and irregular across locations, so the learned relationships are weaker.

This supports the intuition that, in more dense markets, with more frequent transactions, the value is more predictable. The sale price is even more so an effect on tangible features, where more competitive markets pushes for a fair market price. Whereas rural housing is much more difficult to predict, the sheer diversity of rural homes, and the sometimes lopsided supply and demand makes accurate prediction of sale prices increasingly difficult, since the valuation might not be intertwined with the inherent features.

4.6 Model Proficiency

Evaluating the proficiency of the model in comparison to alternative approaches is quite a difficult task, due to the lack of previous research on this specific type of data. Furthermore, a straight-up comparison is also near impossible since the datasets can differ a lot. However there exists a related study performed at Kungliga Tekniska Högskolan (KTH), titled Predicting House Prices on the Countryside using Boosted Decision Trees [39]. The KTH project similarly focused on predicting residential property prices specifically for countryside areas in southern Sweden, utilizing various machine learning methods.
In that study, the LightGBM emerged as the best-performing model, achieving a MAPE of 26.11%, considerably higher compared to the MAPE of 15.9% obtained by the model in this research. Thus, in direct comparison, the hybrid model employed here demonstrated notably superior performance. Both studies emphasized similar geographic contexts, predominantly concentrating on countryside and non-metropolitan regions in Sweden. Nevertheless, as previously highlighted, this comparison may not be entirely equitable due to potential substantial differences in the datasets.
.

5
Conclusion

This thesis set out to investigate AI-driven approaches for real estate valuation in non-metropolitan regions of Sweden by combining Artificial Neural Networks and Gradient Boosting Machines. The goal was to develop a data-driven model for property appraisal in sparsely populated markets and evaluate its performance against traditional benchmarks, while also uncovering the key factors influencing property values. The results of the study indicate that these objectives have been met. A hybrid ANN-GBM model was successfully implemented and demonstrated solid predictive performance. Moreover, through model interpretation techniques, the study identified both expected and non-obvious determinants of house prices in the regions studied. In this concluding chapter, the findings are synthesized with respect to the research questions, discuss the model's performance and limitations, and highlight the implications for future valuation practice in low-density housing markets.

5.1 Key Factors Influencing Property Values

One of the primary research questions addressed which factors most strongly influence property valuation in the non-metropolitan Swedish regions. The model results and interpretability analysis showed that several classic real estate drivers remain important even in more rural contexts. Location proved to be a critical factor: properties closer to regional centers or with better access to points of interest tended to have higher valuations, reflecting the premium on accessibility. Property size and characteristics were also influential, size of both house and lot showed diminishing effect the more rural the property was. Property condition and age were important as well; newer or recently renovated homes generally achieved higher prices than very old buildings in need of renovation. These findings confirm that fundamental drivers identified in urban settings (location, size, condition) are still relevant in sparsely populated markets.

Beyond these expected factors, the model uncovered more nuanced predictors of value that might not be obvious in traditional appraisal. Using SHAP to interpret the hybrid model's predictions, evidence that micro-location and neighborhood context have measurable impacts even within generally rural areas were found. For example, the model learned that proximity to certain local features such as lakes, coastlines, golf courses or major roads can subtly increase a property's value, even

if those features are not explicitly listed as variables in a hedonic model. Another not too obvious factor was the local market activity or density, areas with slightly higher transaction volumes and population density showed a price premium relative to truly remote areas, all else being equal. Interestingly, the neural network component of the hybrid model was able to create embedded representations of location that captured these subtleties. In effect, the ANN learned latent location factors from the data: certain neurons in the network's embedding layer strongly correlated with known spatial characteristics. This demonstrates the embedding power of the model it could infer complex geographic value patterns without explicitly being told, highlighting the benefit of the AI approach in uncovering hidden structure in the data. In summary, the first research question can be answered by noting that both traditional value drivers (location, size, condition) and more granular, non-obvious features (like micro-location amenities, population mobility and educational level) influence property valuation in non-metropolitan Sweden.

5.2 Model Performance and Limitations

The hybrid ANN-GBM model achieved a respectable level of accuracy overall, confirming that an ML-based approach to rural house price prediction is feasible and can definitely outperform or at least match more traditional valuation methods. In broad terms, the model outperformed the benchmark models used in this thesis in terms of error rates. Notably, the ensemble nature of the model led to improved calibration in the middle deciles of property prices. This means that for the bulk of transactions in mid-range price levels, the predicted values aligned closely with actual sale prices. Such calibration is important in practice, as it indicates the model is reliable for the "typical" properties that dominate the market. The model also demonstrated stability when generalizing to unseen data, suggesting that the training process successfully avoided overfitting on the limited dataset.

However, the study also identified several challenges inherent to using ML in this context. In rural and small-market settings, the number of property sales is relatively low, and the available data may be noisy or incomplete. The model sometimes struggled with segments of the data that were particularly thin for example, very remote houses or very high-end properties. This sparsity issue can lead to large prediction errors for outlier properties, which were seen both for objects in areas with DeSO class A and also for the most expensive of objects. This highlights a usual challenge, that machine learning models depend on sufficient examples, and when confronted with novel or rare situations, their predictions are less reliable.

An important thing to have in mind is that the *sale price* does not per se mirror the *valuation*. It can all boil down to the most basic of economical theory; an object is only worth what someone is willing to pay for it, and in low-frequency markets, where there might be few, or even only a single speculator the seller of a house might not get what they are looking for, or frankly what the house is worth. Valuation of, for example apartments in city centers, are significantly easier to appraise, where very few factors can lead to a very accurate price prediction. Due to

the high transaction-volume, just comparing the object to a previously sold, similar object might suffice, or just evaluating the object based on location, floor, size and renovation year, very much tangible assets. When it comes to rural houses, a lot of the driving features can be intangible, and might even be impossible to capture effectively in a dataset. Quantifying the house's potential, charm, inherent feeling or ability of meeting a potential buyer's needs is not an easy task, and serves as a key reminder that rural housing valuation is not a straightforward task that can be solved with more extensive regional data. Furthermore, the test set consisted of entries solely from 2022, which was a year that, according to Booli [40], was peculiar, trends that had lasted roughly a decade had started to shift, especially towards the end of the calendar year. Factors like these are hard to capture, despite the fact that the model used adjusted sale prices to 2020-06, willingness to buy and complex local market trends are quite difficult to adjust for in a comprehensive way.

Another challenge in deploying an ML model for valuation is the question of interpretability and trust. Black-box algorithms can be met with skepticism in an industry accustomed to transparent valuation rationales. An attempt to mitigate this concern was made by applying SHAP analysis to the hybrid model, extracting feature importance and individual prediction explanations. This added layer of interpretability is one of the model's strengths: it allows for verification of the model's behavior is sensible (for example, SHAP values confirmed that increasing living area or improving condition generally raises the predicted price, which aligns with intuition or expectations). In summary, while the application of ML to rural real estate valuation does face challenges such as sparse data, high property diversity and the need for transparency, the approach demonstrated ways to partly overcome these issues. By combining models, carefully validating performance, and using explainability techniques, many limitations were mitigated.

5.3 Implications for Low-Density Housing Markets

The third research question asked what insights from this study can inform future property valuation processes. The findings carry several implications for real estate stakeholders, including valuation firms such as this thesis partner Värderingsdata and their clients. First, the performance of the hybrid model demonstrates that unprejudiced data-driven valuation tools can effectively complement or enhance traditional appraisal methods even in rural settings. In practice, this means that an AI-driven model could be used either as a starting point or second opinion for valuing a property where human expertise might be uncertain due to lack of local comparables. For example, a bank evaluating a mortgage application for a countryside home can leverage the model's estimate as an initial guide. Since the model is calibrated well for mid-range properties (which constitute the majority of transactions), it can provide a quick and reasonably reliable valuation for most cases, thereby streamlining the loan approval or appraisal process. This allows human valuers to focus their attention on the truly unusual or high-stakes cases rather than spending equal time

on every valuation. By prioritizing valuations in this way, stakeholders can achieve greater efficiency without sacrificing accuracy on standard properties.

Secondly, the interpretability of the model via SHAP has a direct practical benefit: it can help build trust and transparency in algorithmic valuations. Real estate agents or appraisers using the tool can not only get a price estimate but also see which factors drove that estimate. The ability to point out, for instance, that a property's value was adjusted upward due to its condition and downward due to its remote location provides a narrative that professionals and clients can understand. Hence, integrating such a model into practice could improve the communication of valuations to clients or internal decision-makers.
Moreover, the model's insights into key value drivers offer guidance for future valuation processes and policies. The fact that the analysis highlighted certain features (such as connectivity to towns or the presence of unseeming points of interest) as especially significant in rural pricing suggests that valuation frameworks should explicitly account for these where they might previously have been overlooked. Stakeholders might consider collecting more data on these nuances to further enhance model accuracy. In addition, the demonstrated benefit of a hybrid modeling approach implies that no single method is likely sufficient for complex valuation tasks; combining approaches (ensemble models) should become a best practice when developing new appraisal tools. For instance, valuation firms could deploy a two-stage model: one stage that learns broad market trends and another that fine-tunes predictions for local peculiarities.
Finally, this study emphasizes that rural property valuation requires special consideration. Thin markets and property heterogeneity mean that automated valuations will always have some uncertainty. An AI-driven valuation in a low density market is most useful as an additional data point a scientifically grounded estimate that can flag when a human's initial guess might be off-base, or confirm when it is on the right track.

In conclusion, this thesis has demonstrated the viability and advantages of an AI-driven, hybrid modeling approach to real estate valuation in non-metropolitan Sweden. The model does not only provide accurate price predictions across a range of properties, but also offers interpretability and insights into the factors driving those prices. The research addressed the posed questions by identifying key valuation determinants, acknowledging and mitigating the challenges of applying machine learning in a sparse data context, and extracting lessons to improve valuation practice. The strengths of the model, such as its ability to learn complex feature interactions, and its interpretable output, makes it a promising tool for stakeholders in low-density housing markets. Its weaknesses, including occasional large errors for extremely unique properties, highlight the continued need for human expertise and further refinement in such contexts. Overall, the work contributes to bridging the gap between manual appraisal techniques and modern data driven methods, showing that even in challenging rural markets, AI can enhance valuation accuracy and consistency. This conclusion underscores a positive outlook for integrating hybrid AI models into the property valuation domain, ultimately stakeholders in making

more informed decisions in non-metropolitan real estate markets.

5.4 Future Work

Although this thesis showed true promise in better evaluation techniques for houses in non-metropolitan houses, there is always room for improvement and enhancement. Future work for this study primarily involves enriching the dataset and further exploring advanced interpretability techniques, particularly with spatial considerations. Although the current model delivers robust predictions, its performance could significantly improve through the inclusion of more object-specific features. By incorporating additional unique dimensions, it is likely that the model would become more sensitive to subtle price variations and yield even more precise predictions. Apart from object specific features, more narrow regional features could be helpful, most of the regional indicators were on municipality level, these were likely not as useful as they could be if narrowed down further.
Additionally, extending the temporal scope of the dataset to include transactions occurring between 2022 and the present could offer valuable insights. This expanded dataset would enable testing of the model's temporal robustness and adaptability to shifting market conditions. The real estate market experiences continuous fluctuations influenced by economic cycles, policy changes, and societal trends, making regular updates to the dataset crucial for maintaining predictive accuracy and reliability.

An interesting direction for future research is to apply more advanced spatial interpretability techniques, such as GeoShapley [41]. While SHAP analysis is already used effectively, GeoShapley can provide deeper insights into how spatial variation influences model predictions. It does this by evaluating each observation's contribution based on its geographic location and the spatial relationships between data points. This approach improves the interpretability of models that explicitly incorporate spatial data.
Implementing GeoShapley analysis could allow for deeper insights into why certain predictions deviate significantly from actual prices by pinpointing spatial clusters or anomalies. For instance, regions demonstrating consistently higher prediction errors could be systematically investigated for underlying spatial patterns or external influences such as urban planning decisions or local economic shifts. Furthermore, identifying specific geographical regions where features contribute significantly more or less than expected could guide targeted data collection strategies, thus enriching future datasets.
Another way to mitigate the risk of systematic prediction errors is to make dedicated models for different known variables, such as the DeSO code, this would naturally imply smaller datasets but given the right data could be plausible for all DeSO areas. Since the used dataset, even though filtered on houses in non-metropolitan areas, is still very diverse. Making dedicated models, or bucketizing on specific variables should lead to improved overall performance, this was outside the scope of this thesis however, and would require additional data.
In summary, enriching the dataset, expanding its temporal coverage, and integrating

advanced spatial interpretability methods such as GeoShapley represent promising future directions. These steps would collectively refine predictive performance and deepen the interpretability and utility of the model, ultimately delivering greater value to stakeholders across the real estate sector.

A

Appendix 1

Table A.1: Table of all included counties and municipalities in the dataset.

Regions Included in the Dataset					
SMÅLAND			**Gotland**	**Östergötland**	**Blekinge**
County					
Jönköping	**Kronoberg**	**Kalmar**	**Gotland**	**Östergötland**	**Blekinge**
Municipality					
Jönköping	Uppvidinge	Högsby	Gotland	Mjölby	Karlshamn
Aneby	Lessebo	Torsås		Åtvidaberg	Karlskrona
Gnosjö	Tingsryd	Mörbylånga		Norrköping	Ronneby
Mullsjö	Alvesta	Hultsfred		Ydre	Olofström
Habo	Älmhult	Mönsterås		Finspång	Sölvesborg
Gislaved	Markaryd	Emmaboda		Motala	
Vaggeryd	Växjö	Kalmar		Boxholm	
Nässjö	Ljungby	Nybro		Linköping	
Värnamo		Oskarshamn		Vadstena	
Sävsjö		Västervik		Kinda	
Vetlanda		Vimmerby		Valdemarsvik	
Eksjö		Borgholm		Söderköping	
Tranås				Ödeshög	

References

[1] H. Lind and P. Palm, *Värdering av fastigheter 2024 - en antologi*. Books on Demand, 2024.

[2] Skatteverket, *Värdeområden län för län (2023-2025)*, https://www.skatteverket.se/foretag/skatterochavdrag/fastighet/fastighetstaxering/deklareralantbruk/vardeomradenlanforlan.4.8bcb26d16a5646a14812743.html, Accessed: 2025-04-01, 2023.

[3] J. Sonesson and J. Mattsson, "Fastighetsvärdering i en tunn marknad - en problematik för värderare och banker," Master's Thesis, Lunds universitet (Lund University), Fastighetsvetenskap, Lund, Sweden, 2023.

[4] S. Rosen, "Hedonic prices and implicit markets: Product differentiation in pure competition," *Journal of Political Economy*, vol. 82, no. 1, pp. 34–55, 1974.

[5] Värderingsdata, *AI:s potential att revolutionera fastighetsvärdering*, News article (MyNewsDesk), https://www.mynewsdesk.com/se/varderingsdata/news/ai-s-potential-att-revolutionera-fastighetsvaerdering-484706, Published June 4, 2024, 2024.

[6] A. Baum, L. Graham, and Q. Xiong, "The future of automated real estate valuations (avms)," University of Oxford, Said Business School, Tech. Rep., 2021.

[7] H. Yildirim, "Property value assessment using artificial neural networks, hedonic regression and nearest neighbors regression methods," *Selçuk University Journal of Engineering, Science and Technology*, vol. 7, no. 2, pp. 387–404, 2019.

[8] S. Sanjari and A. Mijac, *Revolutionerande fastighetsvärdering: Påverkan av artificiell intelligens på fastighetsvärderingsprocessen*, Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2023.

[9] E. Pagourtzi, V. Assimakopoulos, T. Hatzichristos, and N. French, "Real estate appraisal: A review of valuation methods," *Journal of Property Investment & Finance*, vol. 21, no. 4, pp. 383–401, 2003. DOI: 10.1108/14635780310483656.

[10] B. Nordlund and E. Persson, *Fastighetsekonomisk analys och fastighetsrätt*. Lund, Sweden: Studentlitteratur AB, 2024.

[11] Office for National Statistics, "Quality assurance of administrative data used in house price statistics," Office for National Statistics (UK), Tech. Rep., 2018. [Online]. Available: https://www.ons.gov.uk.

[12] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. DOI: 10.1109/TIT.1967.1053964.

[13] R. J. Samworth, "Recent progress in the theory of the k-nearest neighbor classifier," *Probability Surveys*, vol. 9, pp. 1–50, 2012. DOI: 10.1214/11-PS188.

[14] J. W. Osborne, "Improving your data transformations: Applying the box-cox transformation," *Practical Assessment, Research & Evaluation*, vol. 15, no. 12, 2010.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[16] IBM, *What is a neural network?* https://www.ibm.com/think/topics/neural-networks, Accessed: 2025-04-20.

[17] GeeksforGeeks, *Universal approximation theorem for neural networks*, https://www.geeksforgeeks.org/universal-approximation-theorem-for-neural-networks/, Accessed: 2025-04-02.

[18] A. V. contributors, *Evaluation metric for regression models*, Accessed: 2025-05-06, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/.

[19] M. E. contributors, *Huber loss - loss function to use in regression when dealing with outliers*, Accessed: 2025-05-06, 2023. [Online]. Available: https://mlexplained.blog/2023/07/31/huber-loss-loss-function-to-use-in-regression-when-dealing-with-outliers/.

[20] J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-González, A. Ramírez-Pedraza, and E. A. Chávez-Urbiola, "A comprehensive survey of loss functions and metrics in deep learning," *Artificial Intelligence Review*, vol. 58, no. 195, 2025. DOI: 10.1007/s10462-025-11198-7. [Online]. Available: https://link.springer.com/article/10.1007/s10462-025-11198-7.

[21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. DOI: 10.1214/aos/1013203451. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full.

[22] J. Carneiro, "Explainable ai and automated valuation models in property appraisals," Accessed: 2025-05-05, M.S. thesis, Faculdade de Engenharia da Universidade do Porto, 2024. [Online]. Available: https://repositorio-aberto.up.pt/bitstream/10216/161255/2/684283.pdf.

[23] GeeksforGeeks, *Cross validation in machine learning*, https://www.geeksforgeeks.org/cross-validation-machine-learning/, Accessed: 2025-04-22.

[24] GeeksforGeeks, *Underfitting and overfitting in machine learning*, https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/, Accessed: 2025-04-26.

[25] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine learning and the bias-variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[26] A. Yadav, *Shap values explained*, Accessed: 2025-05-05, 2024. [Online]. Available: https://medium.com/biased-algorithms/shap-values-explained-08764ab16466.

[27] A. A. Awan, *Introduction to t-sne: Nonlinear dimensionality reduction and data visualization*, https://www.datacamp.com/tutorial/introduction-t-sne, Accessed: 2025-04-26.

[28] A. Kumar, *A step-by-step explanation of principal component analysis*, https://builtin.com/data-science/step-step-explanation-principal-component-analysis, Accessed: 2025-06-09, 2024.

[29] P. D. Team, *Pandas.categorical - pandas documentation*, https://pandas.pydata.org/docs/reference/api/pandas.Categorical.html, Accessed: 2025-05-04.

[30] S. learn Developers, *Sklearn.preprocessing.standardscaler - scikit-learn documentation*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html, Accessed: 2025-05-02.

[31] P. D. Team, *Torch.nn.moduledict - pytorch documentation*, https://docs.pytorch.org/docs/stable/generated/torch.nn.ModuleDict.html, Accessed: 2025-05-02.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.

[34] P. D. Team, *Torch.optim.lr_scheduler.onecyclelr - pytorch documentation*, https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.OneCycleLR.html, Accessed: 2025-04-28.

[35] L. N. Smith, "A disciplined approach to neural network hyperparameters: Part 1—learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2019.

[36] P. D. Team, *Torch.optim.lr_scheduler.reducelronplateau - pytorch documentation*, https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html, Accessed: 2025-04-28.

[37] Booli.se, *Åselstadsvägen 101*, Accessed: 2025-05-24, 2025. [Online]. Available: https://www.booli.se/bostad/2112862.

[38] L. S. D'Acci, "Quality of urban area, distance from city centre, and housing value: Case study on real estate values in turin," *Cities*, 2018. [Online]. Available: https://www.researchgate.net/publication/329171808_Quality_of_urban_area_distance_from_city_centre_and_housing_value_Case_study_on_real_estate_values_in_Turin.

[39] W. Revend, "Predicting house prices on the countryside using boosted decision trees," M.S. thesis, KTH Royal Institute of Technology, 2020. [Online]. Available: https://kth.diva-portal.org/smash/record.jsf?pid=diva2:1464970.

[40] S. Wickert, *Så var bostadsåret 2022*, https://www.booli.se/kunskap/sa-var-bostadsaret-2022, Accessed: 2025-05-24, 2023.

[41] Y. Chen, Y. Ye, X. Liu, C. Yin, and C. A. Jones, "Examining the nonlinear and spatial heterogeneity of housing prices in urban beijing: An application of

geoshapley," *Habitat International*, vol. 162, p. 103 439, 2025, In press. DOI: 10.1016/j.habitatint.2025.103439. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0197397525001559.

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY