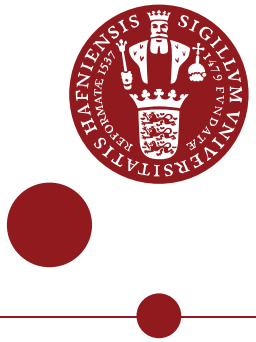
UNIVERSITY OF COPENHAGEN



Course Project

Group 23: He SHI, Mingjie XING, Qi ZHANG

Predicting Housing Prices in Denmark – A Machine Learning Approach

Submitted: August 22, 2023

Keystrokes: 38207 (15 normal pages)

Member contribution:
He SHI: 1 Introduction, 3.2 Data scraping, 3.3 Analysis of scraping log and 3.6 Key statistics;
Mingjie XING: 3.5 Data validation, 4 Methods, 5 Empirical analysis and 7 Conlusion;
Qi ZHANG: 2 Literature Review, 3.1 Data Ethics, 3.4 Data preprocessing, 6 Discussion.

Table of content

1 Introduction

The Danish real estate market over the past decade underwent plenty of ups and downs. However, the market demonstrates resilience and regains momentum soon after crises, even facing the initial impact of the COVID-19 pandemic. By the early 2010s, the market had rebounded from the financial crisis, benefiting from favorable economic monetary policies such as historically low-interest rates. This, coupled with steady housing demand, led to moderate price growth. Notably, Copenhagen experienced a surge in property prices driven by increased demand from urbanization.

Around 2017, moderation in price growth emerged as tighter lending regulations were enforced to prevent housing bubbles. Additionally, measures were introduced to curtail speculative investing, resulting in a more healthy market environment. Despite pandemic-related uncertainties, the market showcased adaptability and stability. Nevertheless, efforts to ensure affordability and curb overheating influenced the trajectory of price growth.

Examining the factors influencing real estate prices remains pertinent. We use data from `https://www.boligsiden.dk/` to conduct a detailed analysis of price determinants using machine learning models. This approach will shed light on the complex interplay of elements shaping the market, contributing to a comprehensive understanding of real estate dynamics and will facilitate informed decision-making.

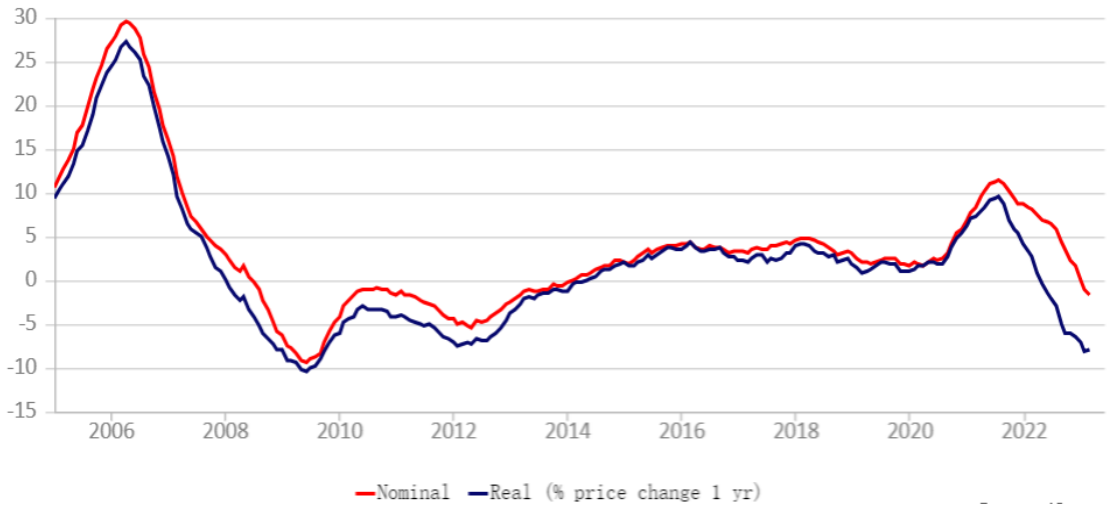


Figure 1: real estate price change over time (globalpropertyguide.com)

When we click on a property link on `https://www.boligsiden.dk/`, we uncover a wealth of valuable details about that property. This encompasses key variables like lo-

cation, size, and price. Additionally, an introduction to the property provides insights into its unique features. We can also track the property's price changes, compare it with neighboring properties, and gauge recent interest. Public comments offer communal perspectives, and internet connectivity data offers practical insight. Hyperlinks reveal even more hidden treasures providing access to broader price trends and more detailed information. Our main target is residential property prices. For this end, we are interested in fundamental property data like size, location, energy rankings, and days after sale. This curated collection of data elements forms the cornerstone of our exploration, allowing us to gain insights into real estate dynamics.

We trained a 3-degree polynomial regression with LASSO regularization and a gradient-boosting model to predict housing prices with the features we scraped from the website. For the boosting model, we experimented three scikit-learn functions, which have different attributes, i.e., `GradientBoostingRegressor`, `HistGradientBoostingRegressor`, and `XGBRegressor`. For all models we conducted 10-fold cross-validation. Our research shows that a tuned extreme gradient boosting model can best predict housing prices with an R2 score of 44% and Root mean squared errors (RMSE) of 2,329,683.66.

2 Literature Review

2.1 Housing Price Prediction by Machine Learning Models

There are several papers recently in top journals in real estate economics predicting housing prices with machine learning models: Deppner et al. (2023) applies a non-parametric machine learning model with k-fold cross-validation to examine the U.S. commercial real estate appraisal to shrink the deviations between market values and subsequent transaction prices. Lin et al. (2023) augments the traditional hedonic model with the Gradient Boosting Machine algorithm to predict housing prices in Beijing. Tchuente and Nyawa (2022) uses artificial neural networks, random forest, adaptive bBoosting, gradient boosting, and K-nearest neighbors algorithms to estimate real estate prices in French cities. Real estate price prediction is challenging and attracts ongoing research.

Mohd et al. (2020) reviews techniques like Neural Networks, Hedonic Price Model, SVM, Linear Regression, Decision Tree, Random Forest, KNN, Fuzzy Logic, among others, for property price forecasting. These studies have provided substantial evidence to support the notion that machine learning models are capable of accurately predicting house prices.

2.2 Independent Variables for House Price Prediction

When examining specific independent variables for house price, researchers commonly choose a range of factors to demonstrate changes in house prices. This diverse selection reflects the understanding that various influences affect the housing market. These key variables includes: Girouard and Blöndal (2001) uses economic indicators, Engelhardt and Poterba (1991) uses demographics, Gelfand et al. (2004) uses location, and Case et al. (1991) uses property attributes, all offer a comprehensive view of the intricate relationship between factors and the dynamic nature of house prices. Pace et al. (1998) uses the characteristics or variables affecting housing prices can be classified according to different criteria, and the classical specification of feature models namely structural features and positional features. Chica-Olmo et al. (2013) indicates that the price of the house mainly depends on the nearby housing prices.

2.3 Distinct Factors in Denmark House Prices

For Denmark house prices, the studies show us the influencing factors are slightly different from other countries. Englund and Ioannides (1997) examines house price trends in 15 OECD nations, uncovering remarkable similarities among these countries. Marsh et al. (2010) indicates that energy consumption of houses holds considerable significance among individuals in Denmark. Using the hedonic price method, Præstholm et al. (2002) finds significant and positive homeowner willingness to pay for forest proximity, often surpassing afforestation costs.

2.4 Contribution

This project makes significant contributions by: 1) meticulously compiling a comprehensive dataset that encapsulates the intricacies of Denmark's residential real estate market, thereby offering a valuable resource for comprehensive analysis; and 2) constructing a robust predictive model tailored to Denmark's various cities, enabling the accurate anticipation of housing price fluctuations. These achievements collectively aim to deepen insights into the complexities of the housing market and furnish stakeholders and researchers with a reliable framework for informed decision-making and exploration.

3 Data Description

We scraped individual-level house prices from `boligsiden.dk`, a major real estate broker in Denmark covering housing prices as monthly data of 36 months scale and detailed features of houses in different regions with open data access. Structural features in data

include area, location, and construction year.

3.1 Data Ethics

The data used in this project is obtained through web scraping from the Boligsiden platform. This platform provides information about the residential real estate market in Denmark. The data we have collected represents the status quo of the Danish residential real estate market.

Throughout the process of data collection and analysis, we have consistently adhered to the principles of legal compliance, particularly in accordance with the regulations set forth in the European Union's Digital Services Act Dec (2022). This act ensures the transparency and legality of our data usage. Moreover, to safeguard individual privacy, the data we have utilized is strictly limited to publicly available information, devoid of any personal sensitive data. When presenting the data, we have employed aggregation techniques to ensure the privacy and anonymity of individual observations.

We are also committed to upholding the seven principles of the General Data Protection Regulation (GDPR). These principles encompass legality, fairness, transparency, purpose limitation, data minimization, accuracy, and storage limitation. Our data collection and usage adhere to legal boundaries while respecting the rights of data subjects. The purposes of data usage are well-defined, restricted to the objectives of this project, ensuring data accuracy and currency, and placing strict limitations on data retention.

By adhering to these principles, our aim is to ensure the secure, legal, and transparent use of data for valuable research purposes while safeguarding the privacy rights of data subjects.

3.2 Data Scraping

Our data collection process started on August 15^{th} and took about three days to complete. During this time, we faced some challenges. One was making sure that the data we collected from the website matched up correctly. Another challenge was figuring out where exactly to find the different details on the website. But as we worked through these challenges, we actually got better at understanding how websites are structured and how to get the information we need.

At the outset, we collected details about eleven different property features that we believed could aid in predicting prices. This effort resulted in a dataset containing 3650 entries after cleaning. Initially, we had a list of 10,000 website links (URLs) to work with, which we considered a sufficient number for our analysis. By filtering for links

with 'address', we were left with 7619 URLs. However, after investigating errors, we discovered that the main obstacle preventing us from accessing all 7619 URLs was the error message 'NoneType' object has no attribute 'text'". As a result, our dataset consisted of 3754 entries before cleaning, carefully curated to align with our research objectives. Additionally, during our journey, we recognized that certain collected information was not essential for our goals. Consequently, we chose to exclude these irrelevant details from our analysis.

A total of nine features have been deemed pertinent for our analysis. To facilitate our subsequent analysis, we have transformed 'city', 'energy', 'area_code' and 'type' into dummy variables. This transformation enables us to effectively incorporate these variables into our analytical framework. Numerical variables include 'saledays', 'living_space', 'ground_space', 'rooms', 'age'.

3.3 Analysis of Scraping Logs

This section will analyze the logs created in the scraping process, with the intention of ensuring the data quality. Logging during data scraping serves as a fundamental practice to document and monitor the process of retrieving information from websites. It involves capturing relevant details and events throughout the scraping operation and saving them for reference, analysis, and troubleshooting purposes. In our logging process, We mainly acquire the time of scraping, the status code of the request-response, the length of the output, and the path to the output file.

Response Status Code HTTP status codes logged during the scraping process can indicate the success or failure of requests. Non-200 status codes, such as 404 (Not Found) or 500 (Internal Server Error), may suggest issues with the availability or integrity of the data being accessed.

As observed in the Figure 2, the response codes consistently indicate a '200' status. While a total of 7619 requests were initiated, only 5418 of these have been successfully logged. This discrepancy can be attributed to several factors. Firstly, instances of requests timing out may arise due to sluggish or unresponsive servers. Consequently, these instances might result in incomplete responses that remain unrecorded if not adeptly addressed. Secondly, the presence of duplicate URLs within the request list may lead to redundant requests, potentially resulting in incomplete logging if mechanisms to circumvent redundant scraping are not in place. Lastly, it is plausible that the website being scraped enforces rate limiting or other constraints, limiting the number of requests permissible within a short time span.

Response Length and Content The length of the response content logged can offer in-

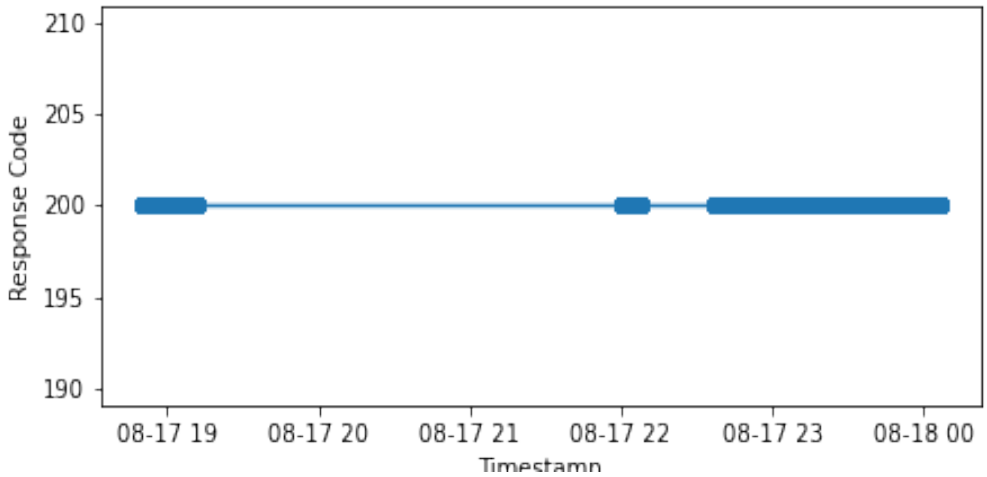


Figure 2: response codes over time

sights into the amount of data retrieved from the website. Drastic variations in response length across similar requests might indicate irregularities in the website's structure or discrepancies in data presentation.

Figure 3 plots response sizes throughout the duration of the web scrapings. The response sizes vary throughout the scrape. Figure 4 presents the distribution of response sizes, showcasing a distribution pattern that closely resembles a normal curve. Notably, instances of notably large response sizes are infrequent within this distribution. The presence of inconsistent response sizes for similar requests potentially signifies variations in the content served by the server. Interestingly, our dataset lacks extremely small response sizes, which is often indicative of crucial data or content absent from the response. This indicates that our data collection process has managed to avoid instances of missing information. This absence is a positive aspect as it implies that our scraping process has been effective in capturing relevant data. Conversely, larger response sizes might point to the server transmitting extensive content, potentially encompassing multimedia elements, attachments, or supplementary resources. Scrutinizing the dynamics of response sizes helps provide a comprehensive evaluation of the data quality landscape and allows us to identify patterns or anomalies that might require further investigation.

Response Time Response time, also known as latency, refers to the amount of time it takes for a system, application, or service to respond to a given request. In the context of web scraping and data extraction, response time typically pertains to the duration between sending an HTTP request to a web server and receiving the corresponding response. Response time is a crucial metric in various scenarios and can provide valuable

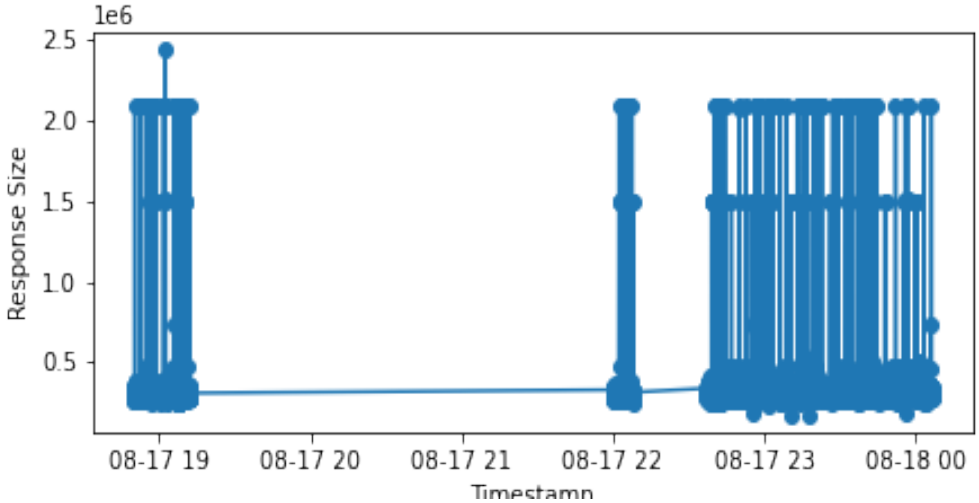


Figure 3: response sizes over time

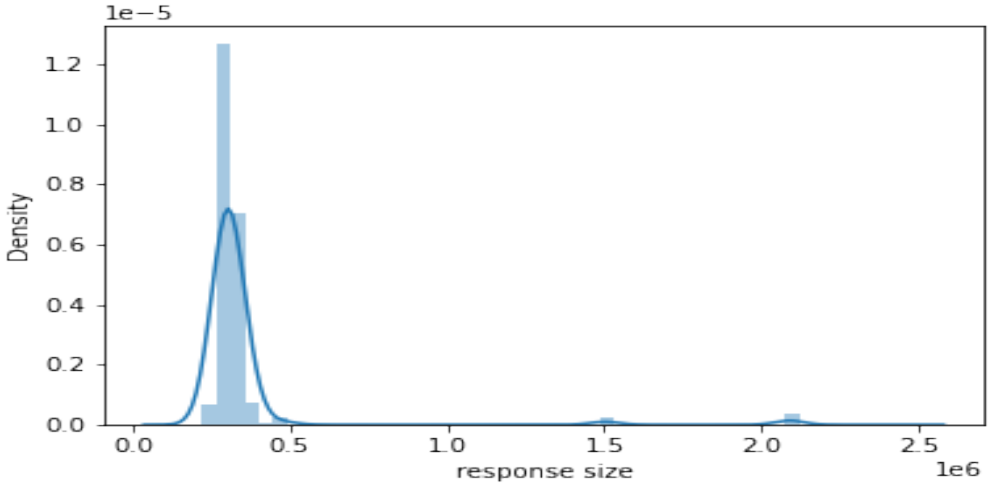


Figure 4: response sizes distribution

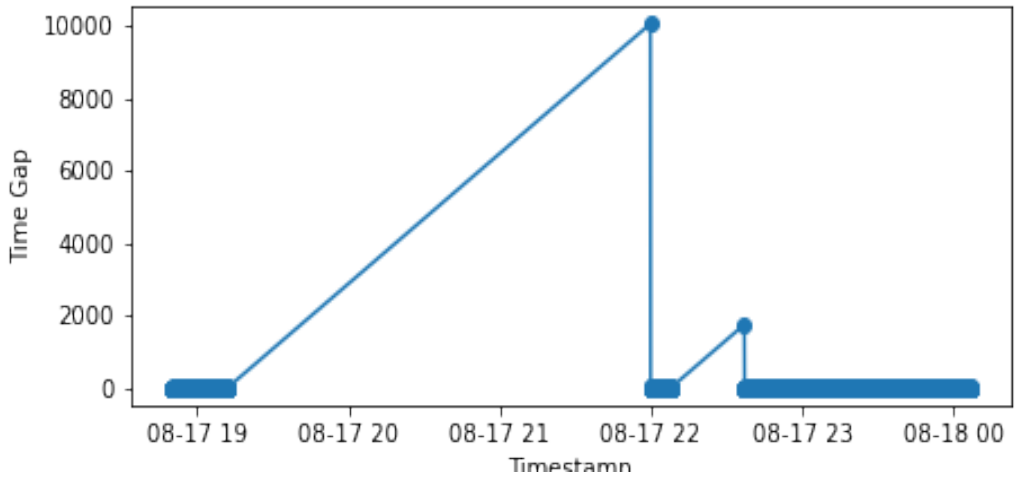


Figure 5: response time

insights into the performance and efficiency of systems and services.

As depicted in Figure 5, the response time distribution generally follows a normal pattern with most responses clustering around the 1-second mark. However, the figure also illustrates two instances of notably large response times. Several reasons can contribute to the occurrence of large response times. Firstly, when the web server faces an influx of requests or experiences heavy traffic, it may need more time to process each request. Consequently, response times can be extended due to the increased load on the server's resources. Secondly, response times can be prolonged when data has to traverse multiple network nodes before reaching the requesting system. Slow network connections or congested routes can introduce delays in data transmission. Thirdly, some websites and APIs enforce rate limits on the number of requests that can be made within a specific time interval. Exceeding these limits can result in lengthened response times as the server restricts the frequency of responses to meet the imposed limits. Understanding the underlying reasons for these large response times is essential for optimizing the web scraping process.

In our analysis, it's important to acknowledge certain limitations that affect our ability to pinpoint the exact reasons for large response sizes and extended response times. Due to the absence of request-specific information in our logging file, we are unable to precisely identify the factors leading to these occurrences. However, we have diligently sought to identify potential explanations that enable comprehensive analysis. The graphs presented above play a significant role in shedding light on the quality of the data. Despite the aforementioned limitations, we have effectively demonstrated the data

quality through various visualizations and observations. The distribution of response sizes and response times, combined with insights derived from response code analysis, provides valuable indications of the integrity and consistency of the collected data.

3.4 Data preprocessing

The original data after scraping needs to be cleaned for machine learning.

Split 'data' variable Firstly,for the 'data' variable of original database including detailed data,we split it to 'living space', 'ground space', 'rooms', 'owner expenses', and 'year'. The 'living space' includes only those rooms that serve for living, living and residence in the house or apartment. The 'ground space' shows the total area that belongs to the home/plot. The 'owner expenses' are the expenses that the new owner of the home must pay in the first year in addition to interest and installments on the loan. The owner's expenses cover i.a. property value tax, property debt and home insurance as well as expenses for owners' association, renovation etc. But we find out the 'owner expenses' would cause the data leakage problem because the value is estimated from house price. The 'rooms' and and 'year' mean the number of total rooms and the year when the house was built.

String method We utilize string methods to replace and extract data on the 'city', 'price', and 'saledays' columns.

Create dummy variable Then we categorized different types of houses as "Villa," "Apartment," "Terraced house," "Holiday home," and so on, helps simplify the dataset and make it more conducive to analysis and understanding, and created five dummy variables about the house types. We found an independent variable of interest named energy label which is from A to G, indicating how energy efficient the house is. We replaced certain energy labels with a unified category, creating dummy variables for 'energy' columns, and then concatenating these dummy variables with the original dataframe for further analysis. Even though dropping 'none' value processing would drop the observations which do not have an energy label, we still reserve this as a feature because Marsh et al. (2010) shows that people place significant importance on house energy consumption in Denmark.

Other cleaning procedure for ML For ground space, we replace 'none' with 0 rather than delete them ,because they belong to the 'apartment' and the ground space of the apartment is not in consideration when buying and selling consumption behavior. We convert building year to age for ML. We split 'area' to 'area code' to 'area address' by extracting numbers from 'area name'. Some 'area code' are empty because some postal codes between text, it's populated with the extracted numbers from 'area name'. Using

	Detached/ terraced house	Owner-occupied flat	Holiday home
2023M07	29 553	6 394	4 647

Table 1: Dwellings for sale in Denmark at the end of 23M07

the 'area code' as independent variable would cause the over-fitting problems because the area code is a category rather than a continuous variable, so we exact the first digit of the area code to create it dummy variable. But after practical analysis, we rule out first digit of the area code dummy variable because it can only delicate the zone in Denmark, city dummy variables can be more explanatory.

The table of contents and description of each variable is in the appendix. Before the cleaning, we have more than 7600 observations, and after the cleaning, we have 3651 observations.

3.5 Data validation

To make sure that the data scraped from the Boligsiden is representative of the residential real estate market in Denmark, we compare scraped data with the average property prices by zip code area in Denmark obtained from Finans Denmark database on `https://rkr.statbank.dk/statbank5a/default.asp?w=1470` under label `BM011`. Finans Denmark provides detailed housing prices in terms of supply, price, and mortgage lending statistics over years. We can find in this database the average housing prices by house type in each postal area in Denmark. From the same database we can draw the number of dwellings for sale by type under label `UDB010`. In table 1 we can see dwellings for sale at the end of July 2023 in all of Denmark. Our sample takes up about 10% of houses on sale in Denmark.

Figure 6 plots the mean price per squared meter by three house types: Detached/ terraced house, owner-occupied flat, and holiday home over postal areas across Denmark. The `BM011` data is of 2023Q1 and the boligsiden data is up-to-date. Detached/ terraced house takes up 70.68% of observations, owner-occupied flat 19.34%, and holiday home 9.97%, also in accordance with the same proportion of each type of houses on sale under lable `UDB010`. By eyeballing, we can find that the two datasets behave similarly over all three types. Our scrapped data is most similar to the data-set in the detached house type, which constitutes the majority of our observations, and the least in holiday home, which should be a result of little sample size. However, in each case, we can find the two data-sets share the same high and low values in the same area. To sum up, it is safe to say that our data-set is representative of the housing market of Denmark to this date.

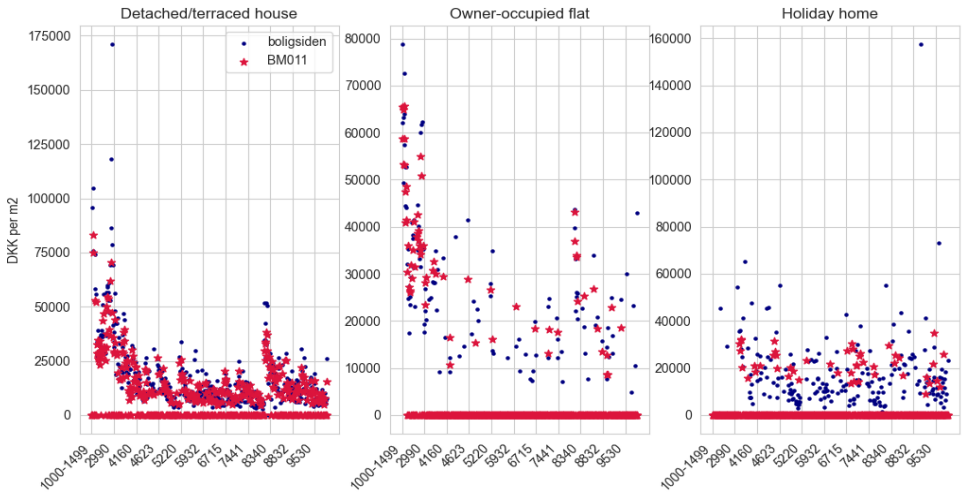


Figure 6: Validating scrapped data with property price from Finans Denmark

3.6 Key Statistics

Table 2 presents the statistical measures pertaining to the chosen variables. It is evident that there exists a substantial disparity in real estate prices. Specifically, the range spans from a minimum of 20,000 DKK to a maximum of 78,000,000 DKK. The average and median prices converge around 3,000,000 DKK. Similarly, the variances in living space and number of rooms are notably extensive, likely attributable to the diverse array of real estate types. Houses generally encompass larger spatial dimensions compared to apartments, and houses typically incorporate a greater number of rooms. Figure 8 and 9 shows non-apartment (value = 0) have larger median dimensions and rooms than apartment (value = 1)

Metric	price	living_space	rooms
Mean	3100033.7	141.34	4.55
Standard Deviation	3387762.9	60.65	1.76
Minimum	200000	10	1
25% Percentile	1375000	102	3
50% Percentile (Median)	2295000	134	4
75% Percentile	3795000	169	5
Maximum	78000000	683	18

Table 2: Summary Statistics

In Figure 7, an evident disparity emerges as the median price of real estate within Copen-

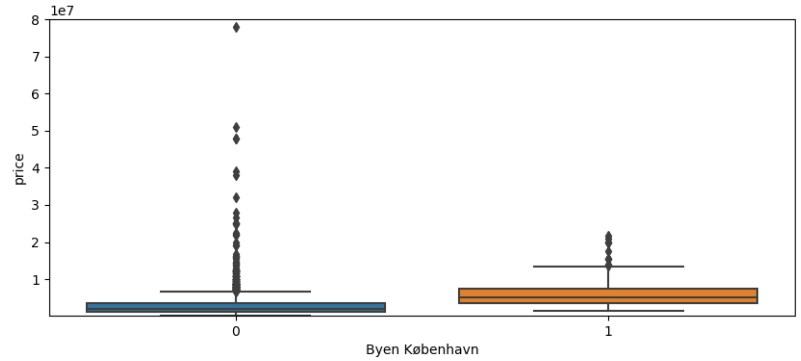


Figure 7: price in different areas

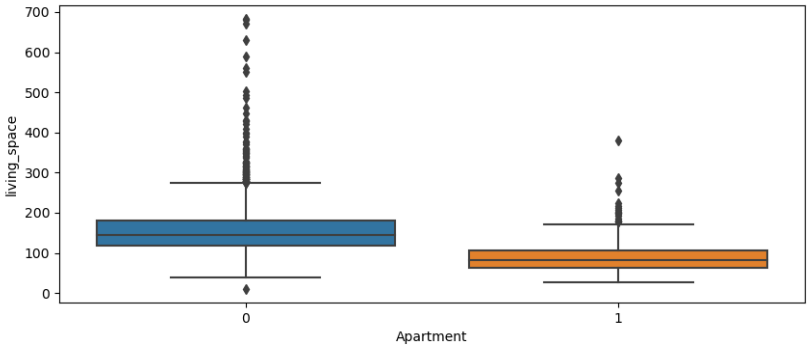


Figure 8: sizes of different types

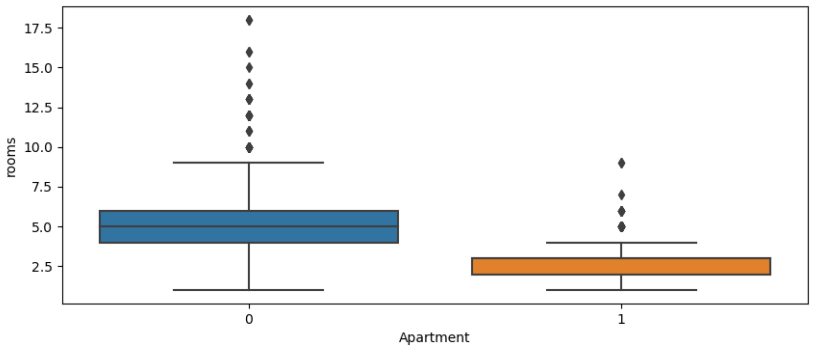


Figure 9: rooms of different types

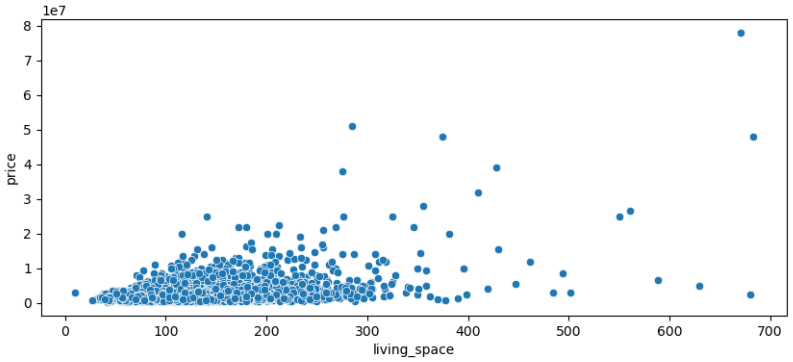


Figure 10: relationship between size and price

hagen city (value = 1) surpasses that of other regions (value = 0). In contrast, other areas exhibit a considerable number of instances with notably elevated prices. It is presumed that these instances may owe their high prices to their remarkably expansive dimensions. As depicted in Figure 10, a discernible relationship materializes between the price and the size of properties. Remarkably expensive properties often boast substantial dimensions; notably, these properties find themselves situated outside the confines of Copenhagen city.

4 Methods

We trained two machine learning models on residential housing prices in Denmark at the moment in each region. The feature variables include structural features such as house area and age and energy type, and external variables like location. The train and test datasets are split in a 70:30 way due to the limitation of available observations.

4.1 Polynomial Model

The first model we trained is a 3-degree polynomial model with LASSO regularization and 10-fold cross-validation. This is the common practice in the literature in the prediction of housing prices because of its simplicity and explainability. We minimize the standard workhorse LASSO formula

$$L_{LASSO}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - \hat{y}_i(\beta))^2 + \lambda \sum_{j=1}^{p} |\hat{\beta}_j| \tag{1}$$

where λ is the penalty hyperparameter which we are going to tune with cross-validation. We also examined the under-fitting and over-fitting problems by inspecting the learning curve and validation curve of linear regression.

4.2 Gradient Boosting Model

Following a vast literature implementing machine learning algorithms on the prediction of housing prices such as Lin et al. (2023) and Deppner et al. (2023), the second model we applied is the gradient boosting model (GBM) as it is arguably the most efficient and effectively predictive algorithm in the prediction of house prices. We also conduct 10-fold cross-validation by splitting the data into 10-fold even-sized validation bins. For each bin we fit the model on the data outside the validation bin. We transform and predict the target in the validation bin. We pick the model which performs the best on the validation data during cross-validation.

The gradient boosting model is a type of ensemble method combining different regressors that have better generalization performance than each individual regressor. As a boosting model, gradient boosting boosts weak learners to strong learners. It is different from adaptive learning with regard to how weights are updated and how weak regressors are combined. We follow the algorithm given in Raschka and Mirjalili (2019) in Chapter 7 and the authors' lecture note on ensemble models on `https://pages.stat.wisc.edu/~sraschka/teaching/stat479-fs2019/`.

Gradient boosting model works as follows:

1. Construct a base tree by computing the prediction with the average of all training examples:

$$\hat{y_1} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)} \tag{2}$$

and compute

$$h_0(\mathbf{x}) = \arg\min_{\hat{y}} \sum_{i=1}^{n} L(y^{(i)}, \hat{y})$$

where $L(y^{(i)}, h(\mathbf{x}^{(i)}))$ is a pre-chosen differentiable loss function

2. Build next tree based on errors of the previous tree: first with pseudo residual

$$r_1 = y_1 - \hat{y_1} \tag{3}$$

$$= -[\frac{\partial L(y^{(i)}, h(\mathbf{x}^{(i)}))}{\partial h(\mathbf{x}^{(i)})}]_{h(\mathbf{x}^{(i)})=h_{t-1}(\mathbf{x}^{(i)})} \tag{4}$$

where the second equation is the general form. And secondly, build a new decision tree for the residuals with terminal nodes $R_{j,t}$ where $j = 1, ..., J_t$.

3. Combine trees from step 1 and 2, predict one entry with

$$\hat{y_2} = \hat{y_1} + \eta * r_1 \tag{5}$$

$$= \arg\min_{\hat{y}} \sum_{\mathbf{x}^{(i)} \in R_{i,j}} L(y^{(i)}, h_{t-1}(\mathbf{x}^{(i)}) + \hat{y}), \tag{6}$$

where η is the learning rate, and the second equations are the general form. And then go to step 2 and compute

$$r_{2,2} = y_2 - \hat{y_2} \tag{7}$$

where $r_{i,t}$ is the pseudo residual of the t-th tree and i-th example and update

$$h_t(\mathbf{x}) = h_{t-1}(\mathbf{x}) + \eta * \sum_{j=1}^{J_t} \hat{y_{j,t}} \mathbb{I}(\mathbf{x} \in R_{j,t}) \tag{8}$$

The procedure is repeated until meet the maximum depth of tree. The model returns the final $h_t(\mathbf{x})$.

We evaluate the predictive efficacy of models with R-squared and Root Mean Square Error (RMSE). We also inspect the importance of each feature in predicting the housing price. It turns out that GBM model outperforms linear model by 3 times in terms of RMSE and its R-squared is around 40% higher than linear models.

5 Empirical Analysis

In applying the model, certain features are omitted or included for better performance. Figure 11 shows the correlation matrix of independent variables. Most variables are safely uncorrelated. Villa and land are more correlated to ground space due to the obvious reason that compared to holiday home, flat and detached house, these two types are more likely to have bigger ground place in the form of grassland or garden. However, we keep the two variables because we believe that they have separate predictive power. It should be noted as well that we include age squared in the linear regression as common practice to reduce bias. It is highly correlated to the feature 'age' but no concern should be raised.

Figure 12 shows the correlation matrix of categorical features used as codes but not explicitly in the form of dummy variables. The conclusion is generally the same as above. Following we introduce the results of our prediction models. Throughout the models, the random seed is set at 17082023, the starting date of our machine-learning exercise.

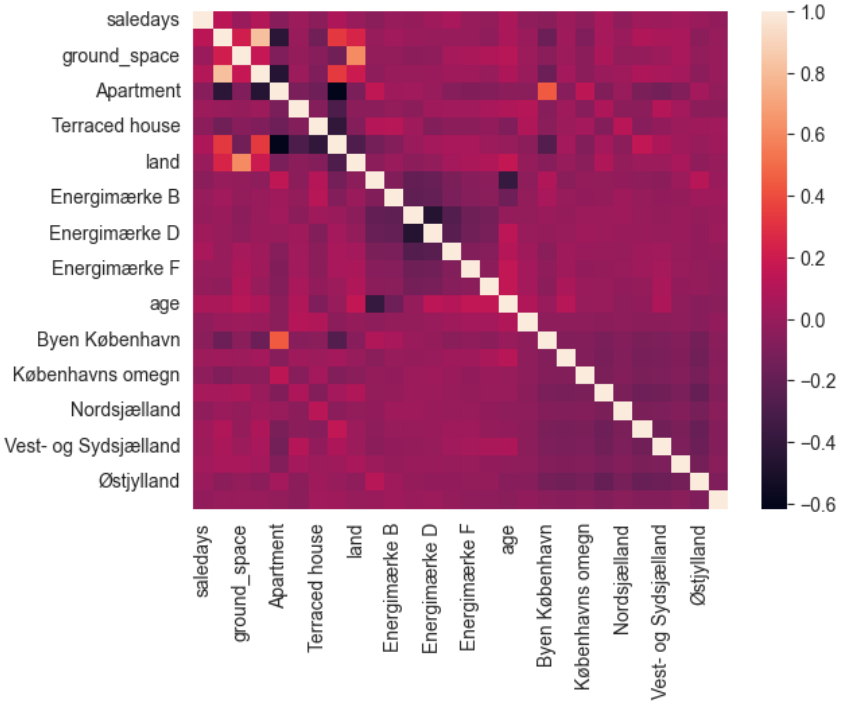


Figure 11: Heatmap of X variables correlation matrix

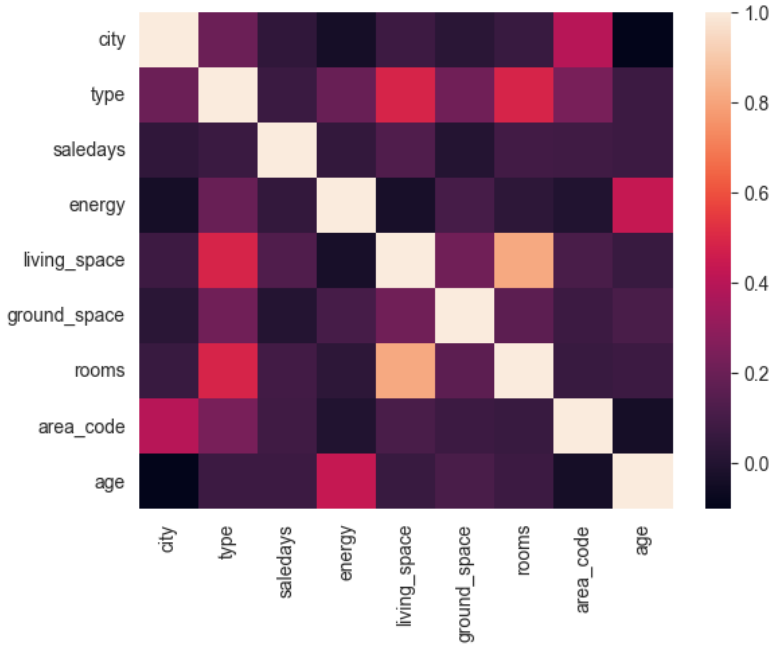


Figure 12: Heatmap of X variables correlation matrix with categories

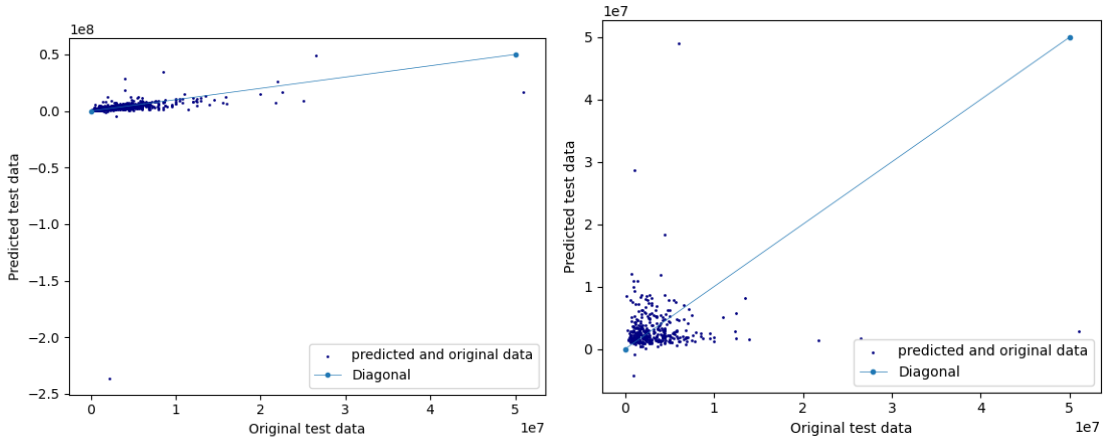


Figure 13: Original and Predicted data in optimal LASSO

	Lambda	**Validation RMSE**
3-degree Lasso	10,000.00	249,834,112.73
3-degree Lasso w/ 10-fold CV	10,000.00	8,438,023.01

Table 3: Tuned Hyperparameters for LASSO and LASSO with 10-fold CV

5.1 Polynomial Model

A common starting point in modeling house prices in the literature is polynomial linear regression. We conduct a 3-degree polynomial linear regression. We also conduct LASSO regularization penalizing parameters to tackle the problem of over- and underfitting and run 10-fold cross-validation to tune for the best hyperparameter λ given the small size of our dataset. We select root mean squared errors as the criteria. Figure 13 plots the original and predicted test data with optimal hyperparameters given in table 3, and the line is a diagonal line. We can see that the LASSO model behaves largely unsatisfactory in prediction in the LHS. After dropping 746 `NaN` values and 3 outliers (larger than 10^8), the prediction is much more satisfactory, as is shown in the RHS of figure 13, though it can be criticized that too many observations are omitted.

We test 66 evenly-distributed λ from 10^{-4} to 10^4 in LASSO and conducted 10-fold cross-validation. The best hyperparameter from LASSO and cross-validation and the corresponding RMSE from validation against each hyperparameter tuned is given in table 3. Note that the λ's are extrema, and therefore the real optimal value may be out of our chosen scale. This means that we need higher penalty strength for parameters and more unimportant features are pushed to zero. This raises bias in tackling overfitting and may cause underfitting. We will further look into this with learning and validation curves.

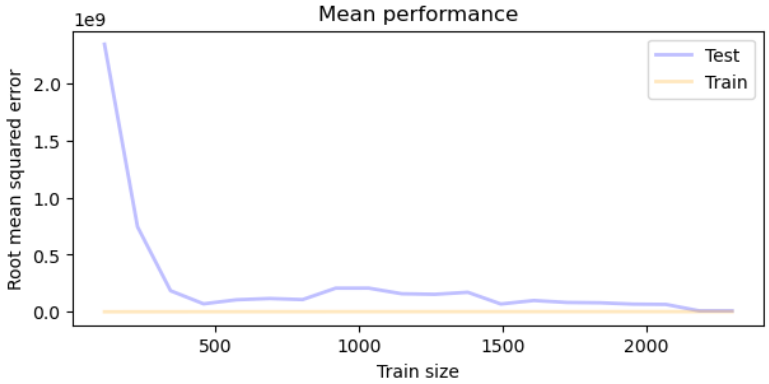


Figure 14: Learning Curve

5.1.1 Learning Curve

We examine the learning curve to find out if our model faces overfitting (high variance) or under-fitting (high bias) problem (pp.201-6, Raschka and Mirjalili (2019)). In figure 14, the x-axis is the training data size, and the y-axis is the model training and validation accuracies. The train dataset plot remains flat through the change in train size, whereas the root mean squared error of the test data dampens quickly around 300-400 data points, remains flat, and converges to the train plot with the increase of train size. This implies that with the increase in train size, that is, more data points, we can expect our model to behave generally better in predicting housing prices in Denmark.

5.1.2 Validation Curve

We also leverage the validation curve to address over- and under-fitting in figure 15. The y-axis is the same as the learning curve, while the x-axis is the model hyperparameters. The RMSE falls tremendously when λ is larger than 1000, and converges to the training data. This indicates that with a higher strength of regularization, we can expect a higher accuracy of prediction.

5.2 Gradient Boosting Model

In practice, the histogram-based gradient boosting model is an enhanced model of gradient boosting `GradientBoostingRegressor` (GBM) in scikit-learn toolkit by the name of `HistGradientBoostingRegressor` (HGBM). It runs faster than ordinary gradient boosting but does not support the search for feature importance. However, the first does not support categorical features. We also experimented with `XGBRegressor` (XGB) which is essentially the same gradient boosting model yet faster and more memory-efficient and more importantly, supports both categorical features and feature impor-

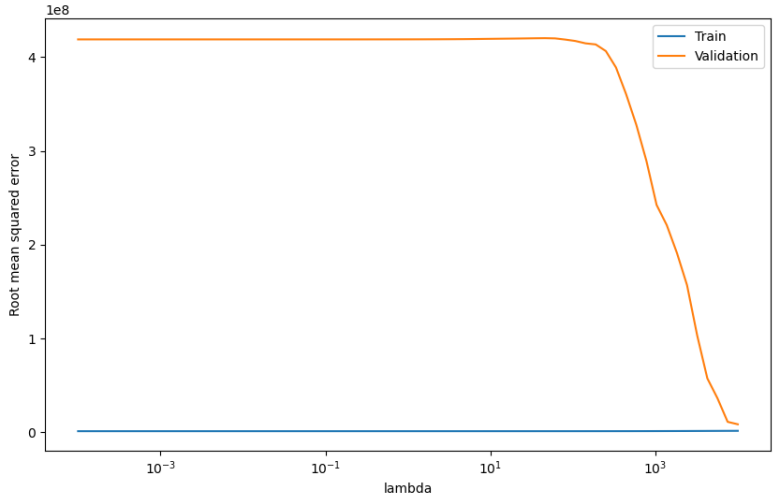


Figure 15: Validation Curve

tance plotting.

The first two models are tuned against `max_depth`, `learning_rate` and `min_samples_leaf`. The `max_depth` refers to the maximum depth of the tree we build. The deeper, the more splits we have and the more we learn from the predictors. We tune against [1, 2, 3, 4]. And the `learning_rate` shrinks the contribution of each tree by the designated amount. With a higher learning rate, we are more likely to meet the over-fitting problem. We tune it in the range [0.1, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6]. `min_samples_leaf` controls the minimum number of samples required to be at a leaf node. It has the effect of smoothing the model and is used to control the overfitting problem. We tune against [10, 30, 50, 70, 90]. There is no `min_samples_leaf` for `XGBRegressor` and thus we tune it against `max_leaves`, meaning the maximum number of leaves, which is essentially the reciprocal of `min_samples_leaf`. We tune it against [0, 20, 40, 60, 80, 100].

We can refer to table 4 for the optimal hyperparameters. We tested the three types of gradient boosting models. We tested ordinary GB with city as a categorical variable, XGB with city and zip, and HGBM with city, zip, and city and zip combined.

The maximum tree depth of histo-based gradient boosting is 1, which is the smallest possible value, indicating that the model opts for a simple and shallow tree model. The optimal value of the maximum leaf for XGB is 0, which means no restriction is put on the maximum number. In all cases, the learning rate is the extremum in our tuning scale. This means that models tend to learn more from each tree and raises the worry of overfitting.

	Max depth	Min samples leaf/ Max leaf	Learning rate
GB w/ city	2	90	1.6
Extreme GB w/ city & zip	2	0	1.6
Histo GB w/ city	2	70	1.6
Histo GB w/ zip	1	1	1.6
Histo GB w/ city & zip	2	70	1.6

Table 4: Tuned Hyperparameters for gradient boosting models with 10-fold CV

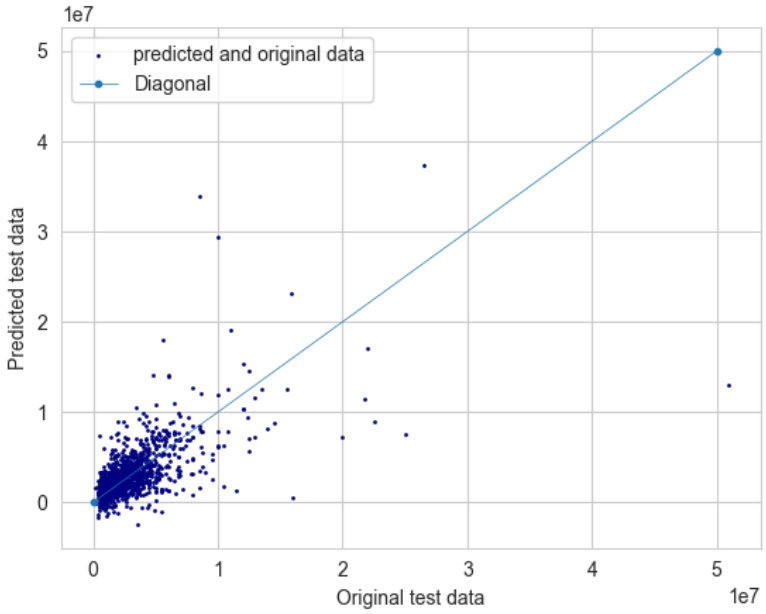


Figure 16: Original and Predicted data in optimal HGBM

Figure 16 plots the original and predicted test data from HGBM with city with optimal hyperparameters given in table 4, and the line is diagonal line. We can see that the GBM behaves largely well in prediction.

5.2.1 Contribution of features

In figure 17 we can find the importance of each feature to the final prediction in an optimal GBM `GradientBoostingRegressor`. We find that the feature that contributes the most is living space, which contributes about 48%. The second is Nordsjælland, which contributes 11%. This means that there is a substantial difference in housing prices between Nordsjælland and other parts of Denmark.

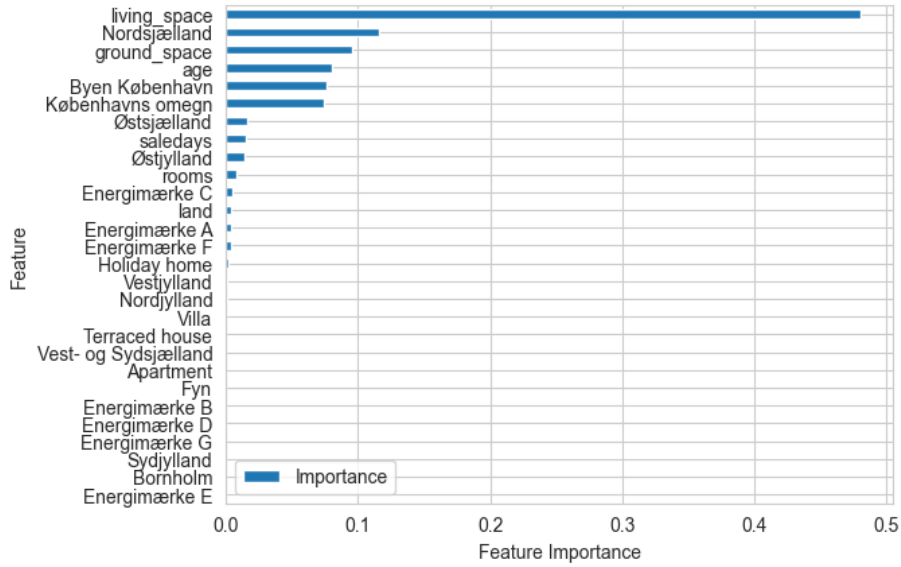


Figure 17: Optimal GBM feature importance

In figure 18 we can find the importance of each feature to the final prediction in a baseline XGB `XGBRegressor` with default parameters, which as we shall later see is the best model in prediction. We find that the feature that contributes the most to the prediction is the living space with 39% importance. The second largest is zip code by about 29% and following rooms, city, and energy all around 7%.

On the whole, it should be argued that the most powerful predictors are living space and location, while energy type and house types are not the most salient features in determining house prices.

5.3 Model Comparison

We compare the root mean squared errors (RMSE) and r-squared (R2) of each model to see how well they perform in prediction. Table 5 concludes the relative performance of each model in a baseline with default parameters and optimal parameters after 10-fold cross validation. We can find that gradient-boosting models outperform LASSO models by around 3 times in terms of RMSE. The best-performing model is baseline `XGBRegressor`, with 0.44 R2 score, indicating that we can explain and predict 44% of the variations in the residential real estate prices in Denmark, whereas polynomial regressions hardly have an R2 value. The second best is the histogram-based gradient boosting model with city as a dummy variable. The reason why models with `area_code` as dummy variable is not the best, though it is a more detailed variable and in intuition

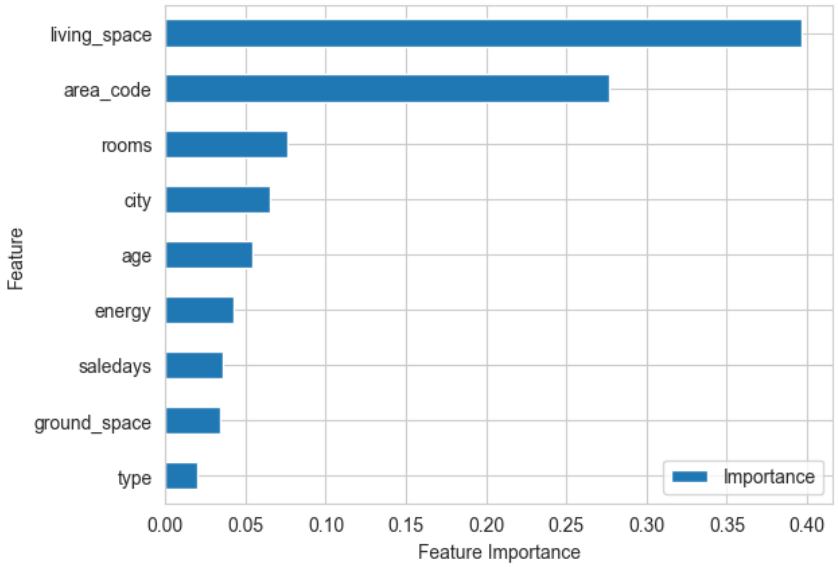


Figure 18: Baseline XGB feature importance

is a better predictive feature in the housing prices, is the limit of available data. For each area code, there are only about 1-5 observations, which tremendously increases the variation and uncertainty of prediction. We can overcome this by increasing the dataset size.

6 Discussion

6.1 Comparison with Previous Studies

Our findings corroborate and extend the results of prior research that has emphasized the significance of energy consumption in house valuation. The house price is notably impacted by various factors, particularly the living space, along with other significant influences like the city, group space, house age, house rooms, and house type. Among these, living space tends to wield a particularly strong influence on the overall pricing dynamics.

6.2 Limitations and Future Research Directions

While our study provides valuable insights, it is essential to acknowledge its limitations.

Limited dataset size makes our models may attempt to fit noise in the data. There are several reasons for the insufficient data volume.The boligsiden website shows there are

Model comparison	**RMSE**	**R_2**
3-degree polynomial Linear	1.74E+19	0.05
3-degree Lasso	7,619,160.64	0.05
3-degree Lasso w/ 10-fold CV	7,619,160.64	-
Gradient boosting w/ city baseline	2,643,662.58	0.28
Gradient boosting w/ city optimal	2,458,819.78	0.38
Extreme gradient boosting baseline	**2,329,683.66**	**0.44**
Extreme gradient boosting optimal	2,806,964.29	0.19
Histo-based gradient boosting w/ city baseline	2,442,247.15	0.39
Histo-based gradient boosting w/ city optimal	2,697,510.92	0.25
Histo-based gradient boosting w/ zip-code baseline	3,102,302.28	0.01
Histo-based gradient boosting w/ zip-code optimal	3,249,872.72	(0.09)

Table 5: Model Performance Comparison

totally 47,728 houses for sale, however, we can only see 200 pages which is 10,000 houses observations. During the scraping process, we encountered certain challenges. Specifically, when attempting to scrape the "energy ranking" variable, our code consistently encountered errors. The specific error message received was "'NoneType' object has no attribute 'text'". Within our code, we've successfully managed to scrape observations without encountering errors within a loop. However, it's worth noting that due to the error encountered during the scraping of the "energy ranking" variable, our overall data collection process has been constrained, resulting in a reduced number of observations obtained.

The relative location of the houses is not considered because we do not have enough time to get relative location data in `hvorlangterder.dk`. Real estate markets can exhibit significant price variations within relatively small geographic areas. Failing to account for local market dynamics can result in generalized and imprecise price predictions that don't accurately reflect the intricacies of the neighborhood or district. The optimal approach to our feature selection, would be using relative location data in `hvorlangterder.dk`.

6.3 Practical Implications

Using machine learning to predict house price in Denmark can benefit a lot for buyers and sellers in the real estate market. Prospective buyers, armed with accurate price predictions, can make well-informed choices aligned with their budget and preferences, leading to more satisfying property acquisitions. Sellers can strategically price their properties based on data-driven insights, optimizing their listing strategies to attract

potential buyers effectively. Moreover, our predictive model can be leveraged by policymakers, real estate professionals to make informed decisions.

7 Conclusion

We web-scraped up-to-date residential real estate prices and features in Denmark from leading danish real estate broker boligsiden to perform machine learning exercises to predict the prices of housing prices in Denmark. Our workhorse model is a traditional hedonic model, with internal features such as living space and rooms, and external features such as location and days on sale. We successfully scraped 7619 observations from a list of 10000 website links, and conducted thorough data cleaning to get a final data set of 3754 entries.

We executed machine learning with polynomial regression with LASSO, and gradient boosting models, both with 10-fold cross-validation. Our research shows that a tuned gradient boosting model with scikit-learn function `XGBRegressor` can best predict housing prices with a satisfactory R2 score of 44% and RMSE of 2,329,683.66. The feature that contributes the most to prediction is living space, with an importance score of 39% and the second is zip code at 29%. The major determinants of housing prices include living spaces and zip code. Energy type and hous types have little influence on the residential real estate prices. Further exercise can extend to obtaining more observations and including more features to improve prediction accuracy.

Bibliography

(2022). Regulation (eu) 2022/2065 of the european parliament and of the council of 19 october 2022 on a single market for digital services and amending directive 2000/31/ec (digital services act).

Case, B., Pollakowski, H. O., and Wachter, S. M. (1991). On choosing among house price index methodologies. *Real estate economics*, 19(3):286–307.

Chica-Olmo, J., Cano-Guervos, R., and Chica-Olmo, M. (2013). A coregionalized model to predict housing prices. *Urban Geography*, 34(3):395–412.

Deppner, J., von Ahlefeldt-Dehn, B., Beracha, E., and Schaefers, W. (2023). Boosting the accuracy of commercial real estate appraisals: An interpretable machine learning approach. *The Journal of Real Estate Finance and Economics*.

Engelhardt, G. V. and Poterba, J. M. (1991). House prices and demographic change: Canadian evidence. *Regional Science and Urban Economics*, 21(4):539–546.

Englund, P. and Ioannides, Y. M. (1997). House price dynamics: an international empirical perspective. *Journal of housing economics*, 6(2):119–136.

Gelfand, A. E., Ecker, M. D., Knight, J. R., and Sirmans, C. (2004). The dynamics of location in home price. *The journal of real estate finance and economics*, 29:149–166.

Girouard, N. and Blöndal, S. (2001). House prices and economic activity.

Lin, W., Shi, Z., Wang, Y., and Yan, T. H. (2023). Unfolding beijing in a hedonic way. *Computational Economics*, 61(1):317–340.

Marsh, R., Larsen, V. G., and Kragh, M. (2010). Housing and energy in denmark: past, present, and future challenges. *Building Research & Information*, 38(1):92–106.

Mohd, T., Jamil, N. S., Johari, N., Abdullah, L., and Masrom, S. (2020). An overview of real estate modelling techniques for house price prediction. In *Charting a Sustainable Future of ASEAN in Business and Social Sciences: Proceedings of the 3 International Conference on the Future of ASEAN (ICoFA) 2019—Volume 1*, pages 321–338. Springer.

Pace, R. K., Barry, R., and Sirmans, C. F. (1998). Spatial statistics and real estate. *The Journal of Real Estate Finance and Economics*, 17:5–13.

Præstholm, S., Jensen, F. S., Hasler, B., Damgaard, C., and Erichsen, E. (2002). Forests improve qualities and values of local areas in denmark. *Urban Forestry & Urban Greening*, 1(2):97–106.

Raschka, S. and Mirjalili, V. (2019). *Python Machine Learning - Third Edition*. Packt Publishing.

Tchuente, D. and Nyawa, S. (2022). Real estate price estimation in french cities using geocoding and

machine learning. *Annals of Operations Research*, 308(1):571–608.

8 Appendix

Variable	Description	Transformed to Dummy Variables	Dropped
city	City names	TRUE	FALSE
energy	Energy ranking from A to G	TRUE	FALSE
area_code	Zip code	TRUE	FALSE
type	Housing type	TRUE	FALSE
expenses	Owner expense every month	FALSE	TRUE
living_space	Size	FALSE	FALSE
address	Street and number	FALSE	TRUE
saledays	Days after sale	FALSE	FALSE
ground_space	Size of land	FALSE	FALSE
rooms	Number of rooms	FALSE	FALSE
age	Housing age	FALSE	FALSE

Table 6: Variable Descriptions and Handling