Anchoring the Price: Machine Learning applied to Hedonic Pricing of Boat Listings

Philipp Gottschalk *595725* Henry Heppe *597651* Timothy Nijhuis *611398*
Igor Pradhan *598064*

April 2024

FEWSIBEOR

Team 8

Abstract

This paper aims to predict the listing prices of boats on online marketplaces using the hedonic pricing approach. The listing price is predicted based on information about the boat's characteristics, the location of the sale and images made available in the online listing. For our study of the boat market, we collect a new dataset of over 14,000 used boats from boat24 including sailing and motorized vessels. Image data is included in the model by extracting characteristics through image classification. For price prediction, we build our approach around implementing and extending the tree-based methods considered in previous hedonic pricing literature. Additionally, we implement various post-hoc interpretability methods to analyse the diverse local and global price effects of different boat characteristics. We find that these boat prices are most accurately predicted by an XGBoost model - a tree-based ensemble method. For this model, our results indicate that boat prices are mainly determined by age, size and the motorization of the boat. Furthermore, our application illustrates the need for a careful differentiation between price effects on a global model scale and local individual price explanations in a heterogeneous market with many sparse variables. With these insights, we contribute to hedonic pricing literature through our novel exploration of the uncharted waters of boat pricing.

1 Introduction

With the ever-expanding presence of online marketplaces, the ability for private individuals to (re)sell goods has become ubiquitous, with new challenges stemming from these increased capabilities. In contrast to large-scale, commercial sellers, individual sellers are unlikely to always possess the level of expertise required to determine a listing price which is simultaneously appropriate and competitive. With professional appraisal being both costly and time-consuming, our paper seeks to present a model to address this pricing problem.

The question of determining appropriate pricing has concerned economists for decades. A popular approach to determining pricing is the hedonic pricing model, first introduced by Court (1939). The main stipulation of this model is that the price of a good can be directly attributed to a sum of utilities derived from the attributes of the aforementioned good. We believe this to be the most appropriate model for our research, considering that we explore boat listing prices.

There is a strong case to be made that boats are an appropriate good within the framework of the hedonic pricing model. Consider that boats have isolated and distinct attributes that are likely to contribute to the boat's utility. Customers will have clear preferences for various characteristics of a boat. They may want a boat of a certain length or with a certain minimum engine horsepower. The synthesis of the utility that customers receive from fulfilling their requirements is likely to be the most significant factor in determining the price customers are willing to pay for the boat, in line with the assumptions of the hedonic pricing model.

The motivation behind analysing the listing prices of boats is multi-fold. Although the listing price may differ from the final price, it serves as a crucial anchoring point for future negotiation. A further argument derives from the practicalities of our data source, the online marketplace boat24. Whilst there is some guidance on appropriate listings, to the best of our knowledge, this is limited to an archive containing examples of old ads. This archive contains 300,000 listings, which highlights the necessity of this study in two fundamental ways. Firstly, it indicates the size of the market and consequently the potential impact of this study. Secondly, it further highlights the difficulty of determining an appropriate listing price through any non-data-driven approach.

We seek to implement and extend upon the framework introduced within Potrawa and Tetereva (2022), which uses hedonic pricing to explore the housing market. The three-step framework consists of *Feature Extraction*, *Predictive Modelling* using machine learning and *Explainable AI*. We apply this framework to the boat market, which, despite being worth $35 billion in 2023 (see Verified Market Research (2024)), has a significant scarcity of literature.

For our *Feature Extraction*, we scrape data from boat24 and use image classification to find features. For our *Predictive Modelling*, we mainly use tree-based methods, although other models

are also tested due to various potential advantages they may offer. Lastly, for *Explainable AI*, we use LIME, PDP and variable importance to explain which characteristics of a boat are most significant in determining its listing price. Key contributions to literature throughout our paper include: finding evidence that machine-learning implementations of hedonic pricing models outperform standard regressions, detecting which characteristics impact boat prices and helping close the gap in boat pricing literature. In particular, determining which characteristics impact the price is helpful for sellers who want a general indication of their boat's listing price.

Within our investigation, we observe that machine-learning models have significant predictive abilities regarding the listing prices of boats on an online marketplace. We see that the XGBoost model performs best in terms of predictive ability. Our post-hoc interpretability models indicate that age, boat size and horsepower are the most relevant factors for boat price. Within our interpretation, we further learn that a careful differentiation between global and local price effects of variables is appropriate when dealing with heterogeneous markets.

Section 2 summarises the literature concerning hedonic pricing models, machine learning applications within this framework and boat pricing. Section 3 provides an overview of our dataset. Subsequently, Section 4 describes the methodology and models utilised within our paper. Within Section 5, we show the results of our modelling. Lastly, Section 6 contains a conclusion and discussion of the paper.

2 Literature Review

Court (1939) is considered the first piece of literature to deal with hedonic price analysis, focusing on the automotive industry. Goodman (1998), finds that Court (1939) holds up relatively well under the standards of contemporary hedonic price analysis. Wing and Chin (2003), which aims to provide a summary of hedonic pricing literature, notes two influential papers conventionally used as the basis of modern hedonic price analysis: Lancaster (1966) and Rosen (1974).

Whilst all approaches stipulate that goods possess attributes that form bundles that the consumer values, there are key differences. Lancaster (1966) focuses on so-called consumer-theory. This suggests that all goods are members of a certain group and that combinations of goods are consumed subject to the customers' budget. In contrast, Rosen (1974) assumes that a range of goods are consumed discretely. Note that hedonic pricing, similar to the theory in Rosen (1974), does not require goods to be consumed jointly. Further, Rosen (1974) highlights how willingness to pay is dependent on customers' budget constraints, which may be non-linear. In our investigation of boat pricing, Rosen (1974) is the most appropriate basis, as boats are highly unlikely to be subject to any form of combined consumption except for the richest of customers.

The literature specifically concerning the hedonic pricing of boats seems to be limited at best. Akyurek (2013) looks at a hedonic pricing model of 'mega yachts', finding that the length of the yacht seems to have the largest effect on the yacht's price, in the sense that a percentage change in length corresponds to the largest percentage increase in price. A paper that explores the issue of boat pricing, although not utilising a hedonic pricing model, is Chen et al. (2024), which carries out sailing boat pricing using Principal Component Analysis and Back-propagating Neural Networks. However, this study is limited to sailing boats alone.

Conventionally, the standard approach to hedonic pricing is the use of Ordinary Least Squares (OLS) based models, as noted by Potrawa and Tetereva (2022). However, particularly in the last few years, the use of machine-learning models in literature has expanded drastically, especially for the study of housing and Airbnb pricing. Table 1 provides a selection of papers that utilise hedonic pricing in a machine-learning context, as well as an overview of our paper as a comparison. Note that whilst this overview is non-exhaustive, most other papers combining machine learning and hedonic pricing employ similar approaches with different datasets.

Authors	**Purpose**	**Source Type**	**Summary/ Contributions**
Potrawa and Tetereva (2022)	Proposes a general framework for using ML tools within hedonic pricing research.	Journal article	The proposed framework is applied to housing data from Rotterdam. For *Predictive Modelling*, random forests are used, which are found to have higher predictive accuracy than OLS models. Another contribution is the application of image classification and text analysis in *Feature Extraction*. For *Explainable AI*, LIME, PDP and variable importance are used.
Chen et al. (2020)	Adds interpretability to non-linear Machine Learning models to help explanation in the context of hedonic pricing.	Journal article	This paper finds XGBoost models to outperform random forests, as well as linear and gradient-boosted regressions. The introduction of SHAP-methods allows the paper to explain the contribution of different environmental features on housing, feeding into the interpretability of hedonic pricing.
Hong et al. (2020)	Compares random forests as a predictor to a traditional OLS-based hedonic pricing model.	Journal article	The paper finds that random forests predict with significantly higher accuracy than OLS, as non-linearities and complexities of the housing market are adequately captured. It suggests using machine learning models as a complement to traditional hedonic pricing models.

Garcia (2023)	Explores the influence of image features on the performance of predictive hedonic pricing models and tests this using multiple ML models.	Master's Thesis	This paper finds that of the ten tested models, XGBoost performs best in predicting hedonic pricing for Airbnb bookings. The paper finds that including image features does not significantly increase the performance of XGBoost, but it does indicate non-negligible influences on the decision-making process. Explanatory methods for ML models used include Shapley values, H-statistics or LIME.
Our Paper	Applies and extends the three-step framework from Potrawa and Tetereva (2022) to study the boat market using various state-of-the-art machine-learning methods.	-	We close a significant gap in the literature through our hedonic pricing analysis of the boat market. We find evidence that XGBoost is the model with the best predictive ability. Further, our results also indicate which features of a boat are most significant in determining its price, as well as emphasizing the need to differentiate between global and local interpretation in diversified markets.

Table 1: A selection of papers that utilise machine-learning methods in the context of hedonic pricing models

3 Data

To give potential sellers the best insight into their boat's listing price, we estimate the pricing model on a new dataset of recent boat listings. By collecting this large body of information on the characteristics of real boats in connection with the prices they are listed for, the pricing model can utilise advanced machine learning methods. This enables us to ground the analysis of price sensitivity to different boat attributes in a model with strong predictive power. One important aspect of optimizing the predictive power of the model is to extract features from the raw data that can be fed into the model, conveying as much signal as possible. The precise description of the feature extraction process can be found in Section 4.1. Here, we focus on providing context on the data's origin, how it was collected and the roles of different data types within the model.

3.1 Data Context

The origin of the data is the online boat marketplace boat24. This type of website functions as a modern-day catalogue for boats. Potential sellers can upload a sales advertisement of their boat and potential buyers use the platform to find candidate boats. A transaction is not completed

on the platform; it is solely used to connect sellers and buyers. The advertisements provide information on the boat characteristics, images, location of the boat and the price.

The listing price is often subject to negotiation and may thus differ from the actual sales price. In our analysis we focus on the listing prices, as only these are available publicly. It is these prices that potential buyers are exposed to when deciding on whether a boat is worth an inquiry. This implies that setting this price correctly is crucial to the seller. Based on this anchor, a seller can then subsequently negotiate what price they are willing to accept in an actual sale.

The data were collected in March 2024 and provide a snapshot of a sizeable subsection of the boat market. The boat market represented in the dataset consists of used recreational boats, mostly sold by non-professionals in Europe. It includes both motorized as well as sailing vessels. A boat in this market could be anything between a super-yacht with multiple rooms or a small sailing dinghy.

After scraping the website and preprocessing the data as described in Appendix C, the dataset consists of 14,439 boats. This includes only boats with a price below £500,000, as the number of observations exceeding this price is too low to reliably predict prices for those boats. A histogram of prices and an example image - taken from a listing - are shown in Figure 1.

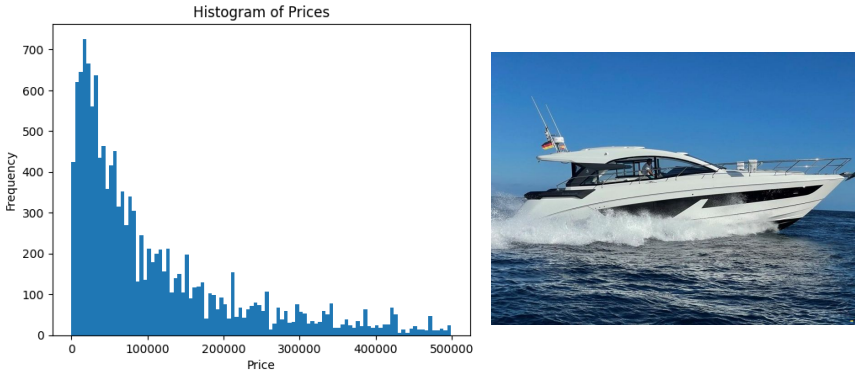

Figure 1: Distribution of prices for boats below £500,000 and an example image of a boat on sale

3.2 Images, Locations and Missing Values

A useful byproduct of how the data is collected and enters the prediction model is that this process mimics the data collection behaviour of a potential buyer, as argued by Potrawa and Tetereva (2022), but takes advantage of all the information simultaneously. We therefore also account for images within our model. The first 3 images per boat are collected since those are the only pictures immediately visible when a listing is opened up. They can thus be conjectured to have the largest impact on the viewer's impression of the boat. In total, approximately 42,000 images are collected, 1,000 of which are used as a training set for the image classification, whilst our test set for classification numbers 200.

It is worth further analysing how the location of a boat sale could affect the price predictions. In other hedonic pricing studies like Potrawa and Tetereva (2022), geographical data proved to have a strong impact on model performance. Additionally, this data adds a useful geospatial interpretation to the model, especially for tree-based models. Since a decision tree partitions the feature space into regions, it also effectively partitions the geographical space. This partition can then be interpreted as different regions, which make for a useful proxy variable for a set of unobserved variables that might have a great influence on the price. In boat markets, such latent factors can be general local market dynamics, the diversity of mariners' preferences in various locations and differences in boat demand induced by local environmental conditions. Although boats are certainly not bound to their sale location, transportation to a more suitable location is costly and could thus also be included in the listing price. In the Appendix (Figure 5), we visualise the stark differences in median prices for boats across European countries.

Since a large set of heterogeneous boats are listed on the marketplace, only a subset of collected variables apply to each given boat, resulting in many missing values. We choose a simple imputation approach, where a missing value is replaced with the most frequent value for the respective variable. Previous research (see Le Morvan et al. (2021)) suggests that for strong learners, such as the machine learning models applied here, the exact imputation method is not relevant to the asymptotic model performance. Results for an advanced imputation method are included in Appendix G.

4 Methodology

The setup of the hedonic pricing framework for the boat market is described in detail in the following sections. We start by elaborating on how the raw data is turned into meaningful information and then proceed to introduce the wide set of models used for predicting prices, followed by the methods that are employed to gain insight into the black-box models. Especially these last two steps are of crucial importance to boat sellers, as they allow both good predictions and an understanding of the features that impact boat prices.

4.1 Feature extraction

In this section, we elaborate on how data is entered into our models. To include the information images convey to a viewer within our model, we need to extract a signal from these images. This is explained in the Section 4.1.1. In Section 4.1.2, we give an overview of the model features.

4.1.1 Image Classification and Brightness

We approach image classification by extracting pre-specified image features. The advantage of this method is that it enables us to choose potentially relevant image characteristics based on domain knowledge. For example, a viewer may subconsciously have a higher willingness to pay for a boat portrayed in an aesthetically pleasing way. Since aesthetics are relatively subjective, we extract two features that are both measurable and could be a simple proxy for this perception.

The first feature measures whether a boat is depicted on water or land. We hypothesise that a boat shown on land is less engaging than a boat shown on water, thereby affecting the price. Since we account for three images, we extract two binary variables *image water* and *image land*, with these set equal to 1 if the boat is shown on land or water in at least one image respectively.

These two variables are synthesized by classifying all images into land/water. For this classification task we finetune a pre-trained deep learning image classification model. We use the *ConvNextV2* model architecture by Woo et al. (2023). The fine-tuning process makes use of the abstract feature space in which the pre-trained model converts the images and trains a classification layer on top of these representations. By fine-tuning the model on 1000 labelled images it reaches an out-of-sample prediction accuracy of 91.8%, making it highly effective for the extraction of this feature. Additional information on the model architecture, training parameters and other approaches can be found in Appendix B.

The second feature is *image brightness* of the first image. We can infer that image brightness may be relevant as brighter images of boats are likely to be perceived as more aesthetically pleasing overall, catching a viewer's attention. One can argue that these effects are mainly relevant to the first impression of the boat. For this reason, image brightness is only measured for the first image, as this serves as the thumbnail for the listing. The brightness of an image is calculated by converting the image to greyscale and taking the pixel level mean of the brightness values, yielding values between 0 and 244 for our data.

4.1.2 Numerical and Categorical Data

The complete set of characteristics that are used to predict the boat prices consists of 68 variables. Among them are numerical features such as *Age, Length, Beam* and *Total Horsepower*, binary features like *Motorized Vessel*, unordered categorical features and the image-based and location-based features. To give an impression of the data, both summary statistics and an example observation are included for a selection of boat characteristics in Table 2. A thorough description and full list of the variables can be found in Appendix C and J.

	Price (£)	Age (years)	Condition	Length (m)	Beam (m)	Engine Hours	Image Land
Example	218300	2	7	12.35	3.96	120	0
Mean	107788.2	18.1	5.6	9.6	3.1	908.7	0.46
StD	109913.9	18.7	0.8	3.7	1.1	1206.3	0.50
Max	499200.0	157.0	8.0	50.0	55.0	20500.0	1.0
Min	1.0	-1.0	1.0	1.0	0.3	1.0	0
NA	0	0	8195	0	0	7476	0

Table 2: Example boat characteristics and summary statistics for a subset of variables of the boat pricing data.

Our dataset shows two characteristics that could pose a problem to some of the machine learning models. It has a large number of columns, some of which are highly sparse and/or correlated. Further, some of the variables suffer from perfect multicollinearity (see also correlation heat map in Appendix A). We perform two transformations of the variable space to deal with these problems separately and compare the predictive performance on both datasets.

The first transformation deals with two sets of sparse and highly correlated variables. *Certified No. of Persons, Cabins, Berths, Bathrooms, Toilets* and *Showers* can all be conjectured to convey similar information about boat price. We replace them with *Boat Size*, the first component of a Principal Component Analysis model estimated on these variables. Similarly, we reduce *Mainsail Area, Jib Area, Genoa Area* and *Spinnaker Area* to one *Sail Area* component. These components explain over 60% and 75% of the respective variance (see scree plots in Appendix C).

The second data transformation is aimed at the linear and penalized regression models. Since the unordered categorical variables are encoded as binary dummy variables, we remove one dummy variable each, serving as baseline category in the model. This yields the two datasets *freqPCA* and *freqBaseCat*, with further data configurations being analysed in Appendix H.

4.2 Predictive Modelling

This section is dedicated to summarizing the main characteristics of the models that are employed for predicting boat prices based on their characteristics. A wide range of models is tested to find the model with the best predictive accuracy, which is subsequently analysed.

Note that we define some common notation across all of our models. We denote a training set $(\mathbf{X}, \mathbf{y})$, which can also be written as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ for N observations. We write the mean square error (MSE) $\mathcal{M}(\mathbf{X}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^N (y_i - \hat{y}_i)^2$, where $\hat{y}_i$ is the model prediction for y_i.

4.2.1 Linear Regression Model

First a standard linear regression model estimated with ordinary least squares (OLS) is applied to the data. This has the advantage that it sets a benchmark on predictive performance against

which the advanced non-parametric methods can be compared. A power transform on the regressors is used to reduce the variance and make the data more normally distributed, as described by Yeo and Johnson (2000). The objective function to be minimised in an OLS model is

$$\mathcal{L}(\mathbf{X}, \mathbf{y}) = \mathcal{M}(\mathbf{X}, \mathbf{y}).$$

4.2.2 Penalized Regression

An attempt to improve the predictive performance of the standard linear regression model is to use penalized regression models (see James et al. (2013)). We use these models as an additional baseline, serving as an indication of whether potentially poor OLS performance is truly attributable to non-linearity or simply due to high data dimensionality. To avoid overfitting, these models regularize the coefficient estimates by introducing a penalty term which shrinks the regression coefficients towards zero. We consider Ridge regressions, Lasso regressions and the Elastic Net, a combination of the former two models.

Ridge regressions minimise the standard residual sum of squares loss from OLS with an additional penalty term, denoted as L_1. Combining the penalty term with the original loss function of a linear regression gives the following objective function, which is minimised

$$\mathcal{L}(\mathbf{X}, \mathbf{y}; \lambda) = \frac{1}{2}\mathcal{M}(\mathbf{X}, \mathbf{y}) + \lambda \sum_{j=1}^{p} \beta_j^2,$$

for regressors $j = 1, \ldots, p$, where $\lambda > 0$ is a hyperparameter tuned via cross validation.

The Lasso regression also contains a penalty term, which we call L_2. This penalty term is combined with the OLS loss function to give an objective function to be minimised, written as

$$\mathcal{L}(\mathbf{X}, \mathbf{y}; \lambda) = \frac{1}{2}\mathcal{M}(\mathbf{X}, \mathbf{y}) + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Elastic net includes both the L_1 and L_2 penalties at the same time, each with their separate hyperparameters λ_1 and λ_2. We then minimise the following objective function

$$\mathcal{L}(\mathbf{X}, \mathbf{y}; \lambda_1, \lambda_2) = \frac{1}{2}\mathcal{M}(\mathbf{X}, \mathbf{y}) + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2.$$

4.2.3 Optimal Sparse Regression Trees

An Optimal Sparse Regression Tree (OSRT) is a regression tree model introduced by Zhang et al. (2023b). This model is attractive for hedonic pricing for multiple reasons. Unlike tree-based ensemble methods, sparse regression trees are considered relatively interpretable. This is because

there is a clear set of splits that determine a prediction. Further, the sparsity of the model, induced by a penalty term, ensures that the tree does not become too large for interpretation. The optimality of the resulting OSRT, proven by Zhang et al. (2023b), is theoretically advantageous over standard regression trees as it prevents suboptimal solutions caused by greedy heuristics.

The approach builds on the model first introduced by Bertsimas et al. (2017), where trees are created by solving a Mixed-Integer Optimisation (MIO) problem rather than using greedy heuristics. Zhang et al. (2023b) proposes a dynamic-programming-with-bounds approach to construct provably-optimal sparse regression trees. The intuition behind this approach is an attempt to reduce the search space by creating tight bounds on the objective function. To do so, a lower bound based on an optimal solution to the k-Means clustering algorithm is created. The technical details of this algorithm can be found in Zhang et al. (2023b), which claims that this results in a fast, consistent and interpretable model.

To define the objective function of the MIO as a whole, let $\mathbf{x}'_i$ be a binary feature vector, obtained from binarising $\mathbf{x}_i$. This is necessary for a timely computation of the OSRT. We do this by guessing potential splitting thresholds of continuous features, following the methodology used in McTavish et al. (2022). Whilst this can impact optimality, this is necessary for the computational feasibility of the OSRT. Consider MSE $\mathcal{M}_f(\mathbf{X}, \mathbf{y})$, where $\hat{y}_i$ is predicted from x'_i using tree f. Zhang et al. (2023b) defines the objective function to minimize as

$$\mathcal{L}_f(\mathbf{X}, \mathbf{y}; \lambda) = \mathcal{M}_f(\mathbf{X}, \mathbf{y}) + \lambda H_f.$$

Here, H_f is the number of leaves in tree f, whilst the hyperparameter λ is chosen to penalize model complexity. See Appendix G for further information on the OSRT configuration.

4.2.4 Tree-Based Ensemble Methods

In our research, we utilize three different tree-based ensemble models for predictive analysis. An ensemble model comprises a collection of weak learners, which are decision trees that individually achieve predictions slightly better than chance. When aggregated, these weak learners provide accurate and robust predictions. The objective function of an ensemble model includes two primary components: (i) the loss term, which penalizes prediction errors using the MSE, although custom loss functions could also be employed; and (ii) the regularization term, designed to penalize model complexity to prevent overfitting. The regularization term differs between models and is discussed in their respective sections.

The ensemble models we investigate are: (i) XGBoost, (ii) Bayesian Additive Regression Trees (BART), and (iii) Random Forest. For the first two employed models, the prediction $\hat{y}_i$

for an observation i can be expressed as the sum of the predictions from K trees, where $\mathbf{x}_i$ is the input vector for observation i, and each tree f_k is a member of the set of all possible trees $\mathcal{F}$. Hence, the model prediction is formulated as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}.$$

As the Random Forest predicts the average of its constituent tree predictions, the sum is multiplied by $1/n$ for this model. We expect the sum-of-trees models to be highly effective. Due to the redundancy these models incorporate, allowing multiple tree combinations to achieve the same target function, they are both robust and flexible. This is further evidenced by their superior out-of-sample performance reported in Potrawa and Tetereva (2022).

Gradient Boosted Trees: XGBoost

XGBoost, an implementation of gradient boosting, stands out for its exceptional predictive performance and widespread adoption in various machine learning applications. The algorithm, originally proposed by Chen and Guestrin (2016), iteratively constructs an ensemble of regression trees, with each subsequent tree refining the model's predictive capability by learning from the errors of its predecessors. By focusing on areas of high residual errors, the XGBoost model can capture complex relationships within the data and potentially outperform other tree-based methods in terms of predictive accuracy. At iteration t, XGBoost minimizes an objective function to determine the structure of the next tree f_t to be added to the ensemble. This objective function is defined as

$$\mathcal{L}_t(\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k).$$

Here, N stands for the total number of observations. The loss function, denoted by l, is selected to be the squared error. The regularisation function, represented by ω and specified later, governs the complexity of the learned model. By incorporating the constant "const" to account for the influence of previous trees on the objective function, the objective for iteration t and the new tree f_t can be expressed as

$$\mathcal{L}_t(\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \omega(f_t) + \text{const},$$

where $\hat{y}_i^{(t-1)}$ denotes the predicted value at iteration $t-1$ and $f_t(x_i)$ represents the prediction of

the new tree f_t for observation x_i. The regularization function ω used in XGBoost is defined as

$$\omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2.$$

The regularization technique employs the L_2 parameter λ to regulate the score of individual leaf values, represented by w_j, within a tree consisting of T leaves. This parameter penalizes larger leaf values, thereby diminishing their impact on the ultimate prediction. Additionally, the model mandates a minimum loss reduction γ for each new split incorporated into the tree. The objective function is minimized using gradient descent to find the optimal parameters for the new tree f_t. In the XGBoost algorithm, each tree level is optimized sequentially by assessing the net gain from a split as the reduction in prediction loss minus the regularization cost. Splits proceed only if the calculated net gain is positive. Another similar model, the Light GBM model, is discussed in Appendix E.

Bayesian Additive Regression Trees

We incorporate a Bayesian Additive Regression Trees (BART) model as outlined by Chipman et al. (2010). BART, a sum-of-trees model, employs a regularization prior to ensure trees act as weak learners. The non-parametric nature of the model ensures robustness against overfitting through its probabilistic framework and integration over many possible trees, potentially giving us better out-of-sample (OOS) performance compared to our other tree methods. For a single-tree model, Bayesian Model Averaging would entail assigning a posterior probability to each possible configuration of a single tree, which would then be used to weigh predictions Chipman et al. (1998). BART extends this notion to encompass all configurations of tree ensembles.

The overall prior probability is modeled as the product of independent probabilities for each tree and the variance. Each tree's probability is composed of its structure (nodes and splits) and the parameter values at its terminal nodes. The model assumes independence among trees, between the trees and the variance parameter σ, and among the leaf node scores. The prior form is assumed to be equal for all trees. Hence, specifying the ensemble prior involves defining three distinct priors: for the variance, for a tree structure, and for a leaf parameter conditioned on a tree structure. In our modelling approach, we have employed the default specifications proposed by Chipman et al. (2010). Having set a number of trees K, BART uses an iterative Markov Chain Monte Carlo (MCMC) backfitting algorithm to derive a posterior distribution based on observed data y. This distribution covers all unknowns in the model, formulated as:

$$\mathbf{P}(\text{trees}, \sigma \mid y) \propto \mathbf{P}(y \mid \text{trees}, \sigma) \cdot \mathbf{P}(\text{trees}, \sigma)$$

Here, $\mathbf{P}(\text{trees}, \sigma \mid y)$ is the posterior and $\mathbf{P}(\text{trees}, \sigma)$ the prior, while *trees* encapsulates the set of all trees and their parameters. The MCMC procedure employs a Gibbs sampler, sequentially updating each tree and its parameters while conditioning on the remaining trees, their parameters, and σ. Finally, it updates σ.

$$\begin{array}{lll} \text{sample} & \text{tree}_k \mid \text{others}, \sigma, y & \text{for each } k \text{ from 1 to } K, \\ \text{sample} & \sigma \mid \text{all trees}, y & \end{array}$$

In our BART model implementation, we initialize two chains with $K = 200$ single-node trees. Each chain undergoes a burn-in of 200 iterations to converge to the posterior, followed by 200 sample iterations. During each iteration, each tree can increase or decrease its number of terminal nodes by one, or modify one or two decision rules. The final prediction averages these 400 samples, weighted by their posterior probabilities.

Random Forest

We also train a Random Forest model as developed by Breiman (2001). Here, each decision tree is constructed using a unique subset of the dataset, selected through bootstrap sampling where data points are randomly chosen with replacement. This approach ensures that each tree is built on a slightly different set of data, allowing each tree to capture unique aspects and variances inherent in the dataset. By doing so, it reduces the likelihood that all trees will make identical errors or overfit the same features, thereby enhancing the overall generalization capability of the model. Furthermore, to increase the diversity of the decision trees, the model employs feature subsampling at each node split during tree construction. Instead of using all available features, a randomly selected subset is considered. This method prevents the dominant features from being repeatedly used at the top splits across different trees, which not only helps in reducing the correlation among trees but also diminishes model variance, providing a more generalized and robust predictive performance. Hence, the advantage of also training a Random Forest model is that it tends to be less prone to overfitting compared to gradient boosted models.

4.2.5 Neural Network

When building a model that estimates a complex and non-linear relationship in a high-dimensional feature space, one cannot claim to have optimized predictive power without including an artificial neural network in the comparison. Due to their theoretically unlimited flexibility to approximate any function, we include such a model to compare to the other machine learning methods.

The model we train here is an artificial neural network with a standard feed-forward archi-

tecture for this regression task. It has three fully connected hidden layers of 32 neurons each. The output layer consists of a single neuron to obtain a single price prediction. The following is a short overview of the components of an ANN, with Hastie et al. (2009) having an in-depth introduction. At each layer for a given node, a linear combination of the output of the previous layer is computed. The weights in the linear combination and an added bias term are the model parameters to be estimated during training. The result is then passed through an activation function which in our model is the Rectified Linear Unit function. It is of the form $max(0, z)$, where z is the intermediary result. After the output layer, the predicted price is plugged into a loss function; we use the MSE. In training, the performance in terms of loss is back-propagated through the neural network and the weights and biases are adjusted accordingly via stochastic gradient descent. We use the Adam optimizer by Kingma and Ba (2014). It uses an adaptive learning rate, calculated based on estimates of the first and second moment of the gradient.

Since different training runs showed problems with overfitting, regularization is added to the model. Activity regularization and kernel regularization are applied to the second and third hidden layers. Both work with L_2-regularization, which introduces a penalty of the same form as a Ridge Regression, namely a sum of squares. Activity regularization applies to the output values of a layer and thus pushes the weights and the bias of a layer to produce smaller outputs. Kernel regularization applies directly to the weights of the respective layer, penalizing large weights.

4.3 Model Interpretation (Explainable AI)

Once the model with the best predictions is found, the actual hedonic pricing analysis occurs. To the benefit of both sellers and customers, we identify which boat characteristics drive the price, what the relationship between those drivers and price is and how single prices are estimated. For each of these explanatory goals, we employ a separate post-hoc interpretation method. The advantage of post-hoc methods is that they are model-agnostic and can thus be applied to whichever model performs best at predicting. We utilise the same interpretation methods as Potrawa and Tetereva (2022) to explain how different boat characteristics impact listing prices.

4.3.1 Variable Importance

To address the first question, we employ Permutation Feature Importance (see Fisher et al. (2019)). It measures the importance of each variable to the overall predictive accuracy of the model. This is done by estimating how much the prediction accuracy declines when the model cannot rely on the information of a given variable. To effectively remove a variable without having to retrain the model, the values of the variable in question are shuffled randomly (permuted).

This creates a new dataset for which the model predicts prices. The difference in the loss function is then taken as the importance score of the variable. In this paper, we show the variable importance in terms of the difference in explained variance; a function of the MSE loss.

4.3.2 Partial Dependence Analysis

For the second question, a graphical approach using partial dependence plots (PDP), introduced by Friedman (2001), is employed. These plots provide a way to visualize the effect of one variable on the price after the effects of all other variables are accounted for. For a given variable, the plot can be obtained by averaging the prediction function over all values found for the other variables in the dataset. Using notation from Molnar (2022), the partial dependence function is

$$\hat{f}_s(x_s) = E_{X_c}\left[\hat{f}_s(x_s, X_C)\right] = \int_\Omega \hat{f}_s(x_s, X_C) d\mathbb{P},$$

where we marginalize over the variables X_C to get the effect of the variables x_s. In practice, this expectation is estimated by averaging over samples from the original data. This works both for a single variable as well as for two variables simultaneously, which then produces a contour plot. One can then quickly see if the effect is somewhat linear, piece-wise linear or non-linear. Hastie et al. (2009) emphasize that this approximation of the effect differs from the partial effect interpretation within a simple regression model, where you ignore all other variables.

4.3.3 Local Interpretable Model-Agnostic Explanations (LIME)

Whilst the two previous interpretation methods focus on global interpretation, in practice, it is also vital to consider why the model assigns a certain price to a specific boat (local interpretation). To investigate individual observations, we utilise the Local Interpretable Model-Agnostic Explanations (LIME), introduced by Ribeiro et al. (2016). The intuition behind this model is as follows: even in a non-linear black-box model, similar observations should produce similar outcomes. Essentially, if two boats are almost identical, this should be reflected in their prices. Hence, there should be a small enough local area which can be approximated as a linear model.

This approach is explained in a relatively simple manner in Molnar (2022). First, we choose the observation for which we wish to explain the black-box prediction. We then draw a random sample of data points around this observation and predict their prices. Here Gaussian sampling is used, where each feature is perturbed individually based on a normal distribution with the mean and standard deviation of the feature in the training data. We then weigh the new samples based on the distance between the original observation and the observations of the perturbed data. Subsequently, we fit an interpretable model to this weighted data set. In our paper, we

use an Elastic Net regression, as described in Section 4.2.2. The coefficients of these models can then be interpreted towards the local effect of a certain feature.

5 Results

In this section, we present a detailed analysis of the findings from the application of the explained methodology to our boat data. We provide insight into which machine learning model is best suited for prediction and what we can deduce regarding the price effects of the different boat characteristics.

5.1 Predictive modelling

We find that the best-predicting model for boat listing prices is the XGBoost model, closely followed by the Random Forest model. The best model has a root mean squared error of £39,372 for out-of-sample prediction. Given that the average boat price in the dataset is around £107,000, the magnitude of this deviation is acceptable. The relative prediction errors of all models are listed in Table 3. It compares model performance on the two differently transformed datasets. Further models and comparisons are listed in Appendix H.

	Datasets	
Models	freqPCA	freqBaseCat
OLS	-	0.00
Ridge Regression	-	-0.26
LASSO	-	0.00
Elastic Net	-	0.00
XGBoost	35.66	**36.51**
Random Forest	34.81	34.93
OSRT	-21.18	-17.96
BART	18.27	19.13
Neural Network	21.20	13.83

Table 3: % decrease in out-of-sample prediction error relative to benchmark model which is OLS with *freqBaseCat* and the prediction error metric is the Root Mean Squared Error.

It is noteworthy that the linear and penalized regressions all perform remarkably similarly. This implies the high dimensionality of the dataset is likely not an issue for the predictive accuracy of these models. Hence, this indicates that the relationship between the boat characteristics and the price is complex and non-linear, a hypothesis from the Hedonic Pricing literature. Thus, this relationship can likely be better captured by a more flexible model, as indicated by the significant performance difference between OLS and XGBoost.

Another insight gained is that not all tree-based methods are similarly strong predictors within our approach. While the Random Forest is nearly as good as XGBoost, the BART

model performs less well, albeit better than the benchmark model. Additionally, the OSRT model underperforms the benchmark substantially. One possible explanation for this finding is that XGBoost and Random Forests are well-known for their robustness against different types of challenging dataset characteristics. The same cannot be said of BART and OSRT. Although BART is similar to the better-performing tree-based models in that it is also an ensemble method, its performance is sensitive to the exact prior distribution choices. Coupled with the fact that estimating BART takes about 3 hours, our ability to test other priors is limited. It is thus possible that we are unable to maximise prediction performance for this model, which warrants future exploration. A similar argument applies to OSRT, with the additional complication that it is a single-tree model. Consequently, it is inherently more fragile than an ensemble method. While in theory, its optimality property should cause it to outperform a greedy tree, its practical disadvantages within our application are substantial. Furthermore, if the model does not predict well, its inherent interpretability serves little purpose.

The neural network does not perform as well as could be expected given its theoretical flexibility. Considering the dimensionality of the data, it is possible that a neural network would outperform the other methods if the quantity of training data were increased.

For both feature space transformations, we find that the XGBoost model manages to capture the complexity well. These results do not, however, warrant the conclusion that this strong performance on both datasets is due to the robustness of the model against such transformations. *Boat Size* and *Sail Area* may simply not have a major effect on the model overall. The second dataset is used for model interpretation due to its increased performance with XGBoost.

5.2 Hedonic Pricing Analysis

We apply the interpretation methods to the XGBoost model to analyse how certain boat characteristics influence listing prices. An impression of the overall predictive behaviour of the model is found in Figure 2, with Plot 1 indicating that the model is better at predicting cheaper boats.

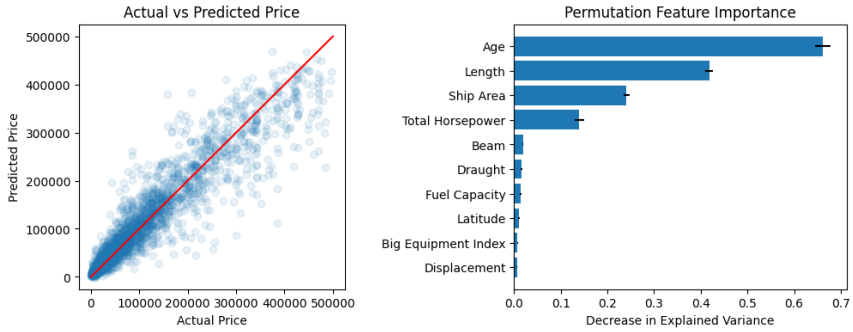

Figure 2: 1. Plotting the actual price against the model prediction over the test set. 2. Plotting the 10 most important features.

From the second plot in Figure 2, which visualizes the permutation feature importance of the ten most important variables, we observe that the most impactful predictors for the price of a boat are its age and size. The total horsepower is also highly relevant, which is likely both due to the high price of motors and that horsepower impacts the practical usages of a boat. The latitude is also among the most important variables. Although the difference in overall explained variance is small, this still hints at pricing differences for similar boats in different locations. As a seller, one might want to consider listing the boat in another country.

It is worth noting that the decrease in explanatory power, associated with effectively removing the information of a given feature one at a time, is substantial for the first four variables and only minor for the remaining factors. Beyond the tenth feature, we do not observe any measurable decrease in explained variance. However, we cannot conclude that the remaining features are irrelevant and should be removed from the model altogether. The permutation feature importance concerns the overall model performance over the whole test set. Some features may be highly relevant, but only to a small subset of observations. Hence, they could only have a minor influence on total model fit whilst contributing substantially to local explanations of individual boats. This phenomenon can be further investigated using PDP and LIME.

The price effect of a whole range of variables is shown in Figure 3. From the summary of the partial dependence of price on a set of binary variables, we learn that aluminium, wood and carbon fibre are heavily associated with a higher price. As expected, price also seems to be affected by whether taxes were already paid. The largest association is between sailing vessels and price. Whether a boat is a sailing vessel is associated with an average price drop of £8,000. Finally, inviting negotiation on the price is related to a lower price, which seems counter-intuitive but could provide easily actionable advice to sellers.

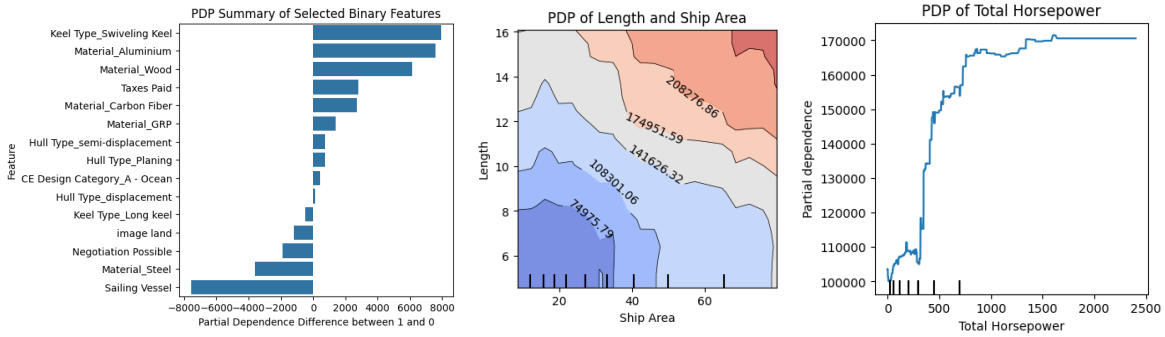

Figure 3: 1. PDP Differences for Binary Features. 2. Joint PDP of Length and Ship Area. 3. PDP of Total HP

The *sailing vessel* variable exemplifies a challenge faced in this analysis, that being the flip side of the previously mentioned low number of globally relevant features. The PDP analysis of this feature suggests a pronounced price effect, however, this is not mirrored in the global feature

importance. Similarly, a LIME analysis of this feature would show that it is always omitted through the regularization of the local penalized regression. One possible reason for this, which may apply to the sparse features as well, is that their variance in the data is not sufficiently high to provide explanatory power next to the main variables. As context, note that 28% of boats are classified as sailing vessels. Another potential explanation is that, because of the large heterogeneity in boat types, many sparse features which only apply to a small subset of boats are outweighed by the substantial explanatory power of variables that apply to all boats.

For the image-related features, neither feature importance nor PDP indicate a large impact on pricing. Two possible interpretations are that either images do not affect prices, which cannot be verified through these results, or that the price signal in the images is not captured well in the extracted features. The latter could be verified by finding more powerful features to extract.

The second PDP shows the joint dependence of price on area and length. We see that for boats shorter than 8 meters, a small increase in the ship area (i.e. the beam) does not come with a higher price, contrary to longer boats. Finally, the PDP of *Total Horsepower* indicates that one could achieve a higher price by adding another motor, without accumulating over 800 hp.

The locally explained effects of total horsepower for different predicted prices in Figure 4 vary substantially but within a similar interval for all price levels. The second plot in the figure illustrates the local effects of three variables that are not deemed highly relevant by the permutation feature importance. Although these features lack global importance, they can have a wide range of effects with high peaks, especially for the fuel capacity of a boat.

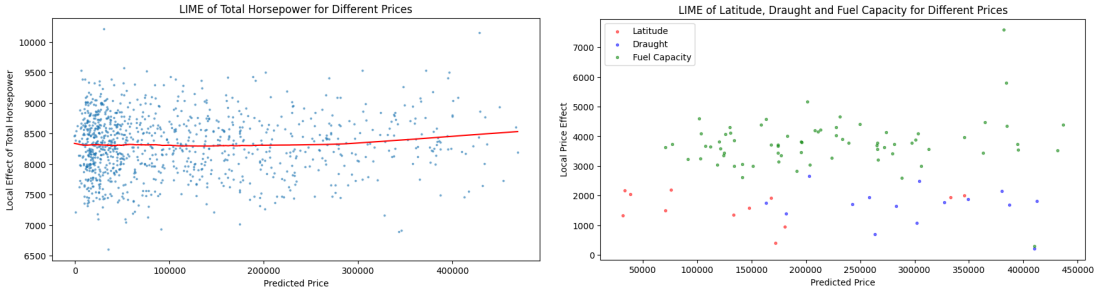

Figure 4: 1. Plotting the coefficients of the total horsepower variable as estimated in local Elastic Net regression models based on sampled neighbourhoods around test data observations (LIME). 2. Plotting the same for Latitude, Draught and Fuel Capacity.

6 Conclusion

In this paper, we perform hedonic pricing analysis on the listing prices of used boats offered on an online marketplace. Our model approach provides price predictions for a wide range of boat types, including both sailing and motorized vessels, providing aid to sellers without the expertise to price their boats. Additionally, the effects of different boat attributes on its listing price are

analysed to illustrate what features of a boat could be invested in to achieve a higher price.

We collect a new dataset of over 14,000 boats consisting of extensive data on their characteristics, together with their price. In addition to numerical data, we make use of geographical data and extract two features from the boat images. Subsequently, we test a diverse range of machine learning models to find the approach with the highest predictive power. We then utilise variable importance and PDP to analyse the effect of boat characteristics on price at a global level and use LIME to further supplement this interpretation on a local level.

The best prediction performance is achieved by an XGBoost model, followed closely by a random forest model. Note that nearly all machine-learning models, with the exception of the OSRT, outperform the linear baseline models, supporting the hypothesis of non-linearity in the relationship between boat features and listing price. Based on permutation feature importance, we find that age, size and total horsepower are the most important features determining boat prices. The third finding is that, for the hedonic pricing analysis, a clear distinction between global feature importance and the magnitude of local price effects of variables is advisable. This is especially true for heterogeneous markets, such as the boat market.

Our findings are limited by the scope of the approach and the level of information extracted from the data. Due to the lack of any reference point, it is not possible to conclude whether our model performs well enough for individual sellers to usefully apply this in practice. This would warrant further market research on a larger scale. Another limitation of our paper that should be considered is the question of interpretability of black-box machine learning models. Whilst post-hoc *Explainable AI* methods give good indications of certain features' impacts, methods such as LIME frequently make predictions that differ from those of the actual prediction model, as noted in Dieber and Kirrane (2020). Hence, the reliability of the feature effects calculated by these models is questionable. Dealing with this limitation falls beyond the scope of our investigation.

For additional research, an implementation of this approach to boats with focus markets outside of Europe, to new or non-recreational boats or other durable luxury goods could provide an interesting foundation. Moreover, a targeted investigation of the feature extraction could produce more informative image features, yielding even better model predictions. Lastly, incorporating a time dimension into the analysis could be of use for example to build a quality-adjusted boat price index to enable proper market timing of sales.

Through our investigation, we contribute to charting the previously unexplored waters of boat pricing. Not only do we find strong evidence regarding the predictive ability of machine-learning models with respect to boat listing prices, but we also determine which features likely impact an appropriate boat listing price; a useful result for both sellers and customers alike.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Akyurek, E. (2013). Pricing of mega yachts. Master's thesis, Sosyal Bilimler Enstitüsü.

Bertsimas, D., Dunn, J., and Paschalidis, A. (2017). Regression and classification using optimal decision trees. In *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–4.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Carriere, D. (2017). geocoder. https://github.com/DenisCarriere/geocoder.

Chandra, R. V. and Varanasi, B. S. (2015). *Python requests essentials*. Packt Publishing Ltd.

Chen, L., Yao, X., Liu, Y., Zhu, Y., Chen, W., Zhao, X., and Chi, T. (2020). Measuring impacts of urban environmental elements on housing prices based on multisource data—a case study of shanghai, china. *ISPRS International Journal of Geo-Information*, 9(2).

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Chen, Y., Li, Z., and Jia, D. (2024). The sailing boat price study based on principal component regression analysis. *Highlights in Business, Economics and Management*, 25:189–196.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1).

Chollet, F. et al. (2015). Keras.

Clark, A. (2015). Pillow (pil fork) documentation.

Coltman, J. (2020). bartpy: Bayesian additive regression trees in python. https://github.com/JakeColtman/bartpy.

Court, A. (1939). Hedonic price indexes with automotive examples. *The Dynamics of Automobile Demand*, pages 98–119.

Dieber, J. and Kirrane, S. (2020). Why model why? assessing the strengths and limitations of lime.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81.

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

Garcia, S. G. (2023). Evaluating the impact of image features on airbnb price predictions: A machine learning approach to hedonic pricing. Master's thesis, Norwegian School of Economics.

Goodman, A. C. (1998). Andrew court and the invention of hedonic price analysis. *Journal of Urban Economics*, 44(2):291–298.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585:357–362.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hong, J., Choi, H. H., and Kim, W.-S. (2020). A house price valuation based on the random forest approach: the mass appraisal of residential property in south korea. *International Journal of Strategic Property Management*, 24:1–13.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning*, volume 112. Springer.

Johnson, R. A. (2024). quantile-forest: A python package for quantile regression forests. *Journal of Open Source Software*, 9(93):5976.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of Political Economy*, 74.

Le Morvan, M., Josse, J., Scornet, E., and Varoquaux, G. (2021). What'sa good imputation to predict with missing values? *Advances in Neural Information Processing Systems*, 34:11530–11540.

Lhoest, Q., von Werra, L., del Moral, A. V., and Sasko, M. (2022). Huggingface datasets. https://github.com/huggingface/datasets.

Lin, J., Zhong, C., Hu, D., Rudin, C., and Seltzer, M. (2022). Generalized and scalable optimal sparse decision trees.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986.

McKinney, W. et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

McTavish, H., Zhong, C., Achermann, R., Karimalis, I., Chen, J., Rudin, C., and Seltzer, M. (2022). Fast sparse decision tree optimization via reference ensembles.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999.

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, 2 edition.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Potrawa, T. and Tetereva, A. (2022). How much is the view from the window worth?: Machine learning-driven hedonic pricing model of the real estate market. *Journal of Business Research*, 144:50–65. Publisher Copyright: © 2022 The Authors.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

Ribeiro, M. T. C. (2019). Lime python package. https://github.com/marcotcr/lime.

Rokem, A. (2023). lowess: Locally weighted scatterplot smoothing. https://github.com/arokem/lowess.

Rosen, S. (1974). Hedonic prices and implicit markets: Product differentiation in pure competition. *Journal of Political Economy*, 82(1):34–55.

Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Stekhoven, D. J. and Buehlmann, P. (2012). Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Verified Market Research (2024). Global boat market size by propulsion system, by material, by end-user, by geographic scope and forecast. https://www.verifiedmarketresearch.com/product/boat-market/.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wing, C. K. and Chin, T. (2003). A critical review of literature on the hedonic price model. *International Journal for Housing Science and Its Applications*, 27:145–165.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., and Xie, S. (2023). Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16133–16142.

Yeo, I.-K. and Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959.

Zhang, R., Lin, J., and Zhong, C. (2023a). optimal-sparse-regression-tree-public. https://github.com/ruizhang1996/optimal-sparse-regression-tree-public.

Zhang, R., Xin, R., Seltzer, M., and Rudin, C. (2023b). Optimal sparse regression trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11270–11279.

Appendix

A. Data

This section provides further insights on the data. The listed boat prices show substantial heterogeneity across different countries in Europe. An indication for this can be found in Figure 5 showing the median listed boat price for different countries. The size of the marker represents the number of boats listed in the respective country. Figure 6 shows the complete set of pairwise correlations for the original *man* dataset. Based on this figure the PCA datasets were developed. In the correlation heatmap, the two sets of variables that are combined can be identified as groups of red boxes.

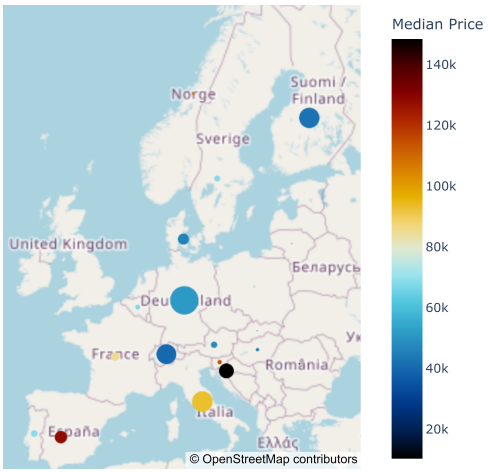

Figure 5: Median boat price per country in Europe. Size of the marker represents the number of boats listed in a country.

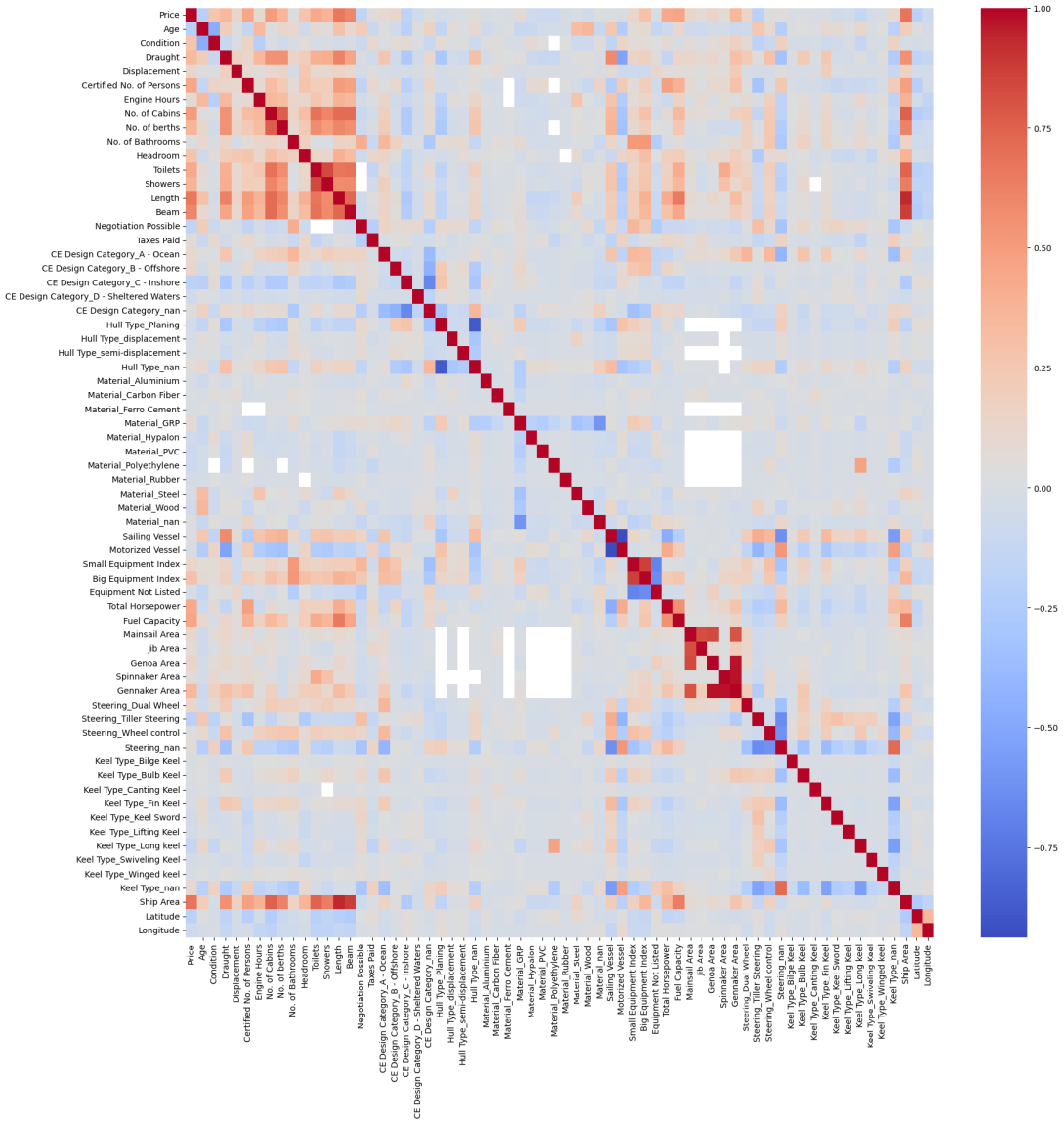

Figure 6: Heatmap of correlations in the unimputed dataset without any variable transformations. White box are missing values.

B. Finetuning and Inference with ConvNextV2

For finetuning, we use the ConvNextV2-tiny model checkpoint published by Facebook AI research (see Woo et al. (2023)). This model has 28.6 million parameters, works on an image resolution of 224×224 and has publicly available weights on GitHub and via the HuggingFace Transformers API (see Wolf et al. (2019)).

The pre-trained model architecture is an update to the ConvNext Liu et al. (2022) version of convolutional neural networks (CNN). It introduces a self-supervised learning component to the originally purely supervised-learning-based CNNs and pairs it with a new type of normalization layer called *Global Response Normalization*. The self-supervised learning-based part of the architecture is a *fully convolutional masked autoencoder* which learns to encode an image as

an abstract representation. This is done by giving the model an image and a partially masked version of that image and having it learn to reconstruct one from the other.

The model is fine-tuned on 1000 manually labelled images with a validation set of 52 images and a test set size of 200, achieving an OOS prediction accuracy of 91.8% after only a few minutes of training. For fine-tuning and inference, we also use the model's image processor normalization and random resizing and cropping of the images to avoid overfitting. For hyperparameter settings, we use a learning rate of 0.00005, a per-device training and evaluation batch size of 16 with 4 gradient accumulation steps. The model is trained for 10 epochs with a warm-up ratio of 0.1. Our finetuned model weights are publicly available at huggingface.co/henry-heppe/finetuned_convnextv2.

Using this model for inference on large-scale image data has two limiting factors: the power of the GPU for the actual model inference and the latency of the internet connection needed for accessing the images in real time. To avoid having to save 40,000 images to a hard drive both were done at the same time, which resulted in a computation time of about 6 hours for all images. This time effort also limited our possibilities in possibly taking into account more than 3 images per boat.

We also fine tune a Vision Transformer (Dosovitskiy et al. (2020)) and a DINOv2 model (Oquab et al. (2023)) both of which required much longer training times but were not able to improve their predictions beyond 85% out-of-sample accuracy. Taking into account the imbalance in our training dataset (only 17.6% of images show a boat on land) the goal was for the model to make predictions that are better than the approximately 82.4% accuracy one would get by always predicting water.

C. Preprocessing and Data Transformations

This section describes the preprocessing steps applied to raw data to produce a dataset suitable for our modelling, which may still contain NaN values, as it represents the dataset before imputation. We start with raw data from boat24, resulting in 27,393 rows and 1,188 columns, indicating potential unique boats and explanatory variables, respectively. To refine the dataset, we first eliminate the duplicate entries and removed rows without a specified price. We also exclude non-relevant categories like houseboats, pedalos, and equipment such as trailers or engines. We categorize the remaining boats as either sailing boats or motorized vessels, eliminating the original boat category column and introducing two dummy columns to indicate the type of each boat. We further reduce the dataset's dimensionality by discarding columns with fewer than 200 defined values, which are impractical for imputation due to the low percentage of defined

values. For feature engineering, we calculate *Age* by subtracting the boat's build year from 2024, dropping rows where age was undefined. We restructure the *Condition* column, which contained categories ranging from *damaged* to *new*, into an ordered linear scale of integers from 1 to 8. We split the *length x beam* column into separate *Length* and*Beam* metrics in meters and introduced a *Ship area* column by multiplying these two values. Additionally, we create two dummy columns: *Negotiation possible* and *Taxes paid*, which are set to 1 if true and 0 otherwise. For columns with a limited set of string values like *CE Design Category* and *Material*, we create individual dummy columns for each unique value. We remove non-essential columns such as *iPod interface* and categorized equipment features into small equipment and big equipment based on their estimated cost, summing up the categories to form a *Small equipment index* and *Big equipment index* Lastly, we used the Python geocoder package with the *Location* column to determine *Latitude* and *Longitude* values for the boats. They point to a representative point in the country where the boat is sold. That point is recorded in terms of coordinates to make use of the geospatial modelling capabilities of the machine learning models taking into account distances and similarities. The final step involves cleaning the columns to ensure they contained integer values, thus preparing the dataset for subsequent analysis stages.

The scree plots for the principal component analysis on two groups of variables of the dataset as introduced in Section 4.1.2 can be found in Figure 7.

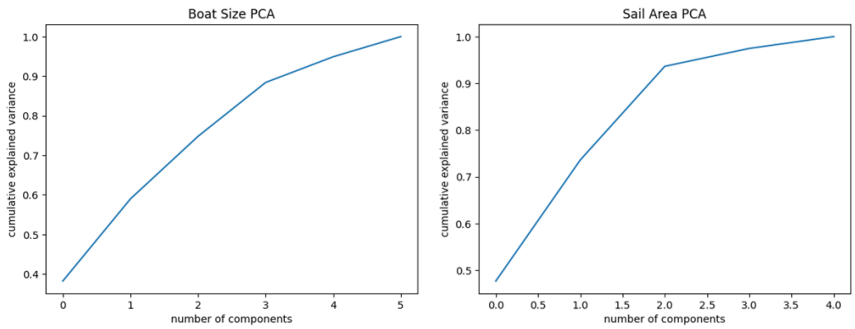

Figure 7: Scree plots for PCA of *freq* dataset.

D. Quantile Regression Forests

Whilst conducting our research, we also implement quantile regression forests, as described in Meinshausen (2006). Quantile regression forests allow us to estimate conditional quantiles in an accurate and non-parametric way. The technical details on the implementation can be found within Meinshausen (2006).

There are a variety of different advantages to predicting these conditional quantiles. For example, as noted by Meinshausen (2006), these allow us to gain a better over of a distribution Y as a function of the predictor variable X, whilst also allowing us to construct prediction

intervals for our data. Furthermore, we can also use the calculated quantiles as a method for outlier detection.

Note that Quantile Regression Forests are not directly comparable to the models used within the rest of the paper. This is as they return the 50% quantile, i.e. the median, when making a prediction, rather than the conditional mean that is used in the other methods. Due to this lack of comparability, we do not use quantile regression forests within our main methodology.

To illustrate the potential use of the quantile regression forests, Table 4 contains the 95% confidence interval ranges for our different datasets, calculated using k-fold cross-validation, with $k = 5$.

Dataset	**man**	**manPCA**	**manBaseCat**	**freq**	**freqPCA**	**freqBaseCat**
95% Prediction Interval	129139.91	127612.16	128618.68	122902.52	123357.92	122777.01

Table 4: The size of the 95% prediction interval (in £) calculated using quantile regression forests on each of our datasets.

Further, we can plot the actual price of the test observations against the size of the 95% prediction interval, as we have done in Figure 8.

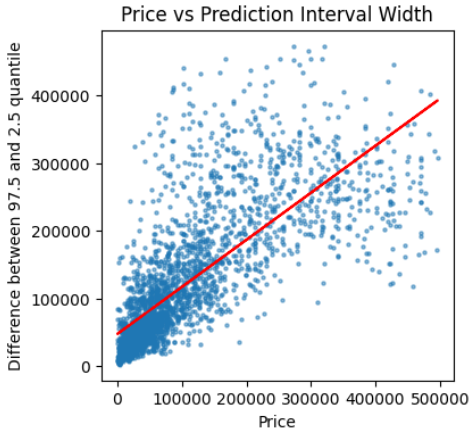

Figure 8: Plotting the 95% prediction interval against the actual listing price of the test observation.

From this figure, we can see that in general, the higher the price, the larger the 95% prediction interval. This result is to be expected: the larger the price, the more likely that factors less captured by our modelling are to be included in a model. Particularly for boats that fall within a luxury market, it is more likely that individual consumer preferences determine price over measurable attributes. As already noted in Potrawa and Tetereva (2022), this falls under the second stage of the model envisioned by Rosen (1974), which requires a very wide methodological toolkit to potentially address.

E. Light GBM

In contrast to XGBoost, LightGBM uses a leaf-wise growth strategy rather than level-wise, splitting the leaf that promises the highest gain. This can lead to more effective learning in scenarios where complex data structures exist, though it comes at the risk of excessively fitting to noise and outliers. The results for this method are included in Appendix H.

F. Hyperparameter Tuning

For the penalised regression models grid search is used. The selection of the parameters was decided on the lowest out-of-sample MSE score. The parameters shown in Table 5 were found.

Model	Parameters	Value	Search Space
Ridge Regression	λ	10	$\{0.1, 1, 10, 100, 100\}$
Lasso Regression	λ	0.001	$\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$
Elastic Net Regression	(λ_1,λ_2)	(0.01, 1)	$\{(i, j)\|i \in \{0.001, 0.01, 0.1, 1\}, j \in \{0.1, 1, 10, 100\}\}$

Table 5: Regression parameters

For the tree-based methods we found the optimal parameters for XGBoost through the grid search package optuna in Python (see Akiba et al. (2019)). To find the parameters, we did 100 trial runs based on 5-fold cross-validation. We use the parameters for which the trial run gave the lowest root mean squared error. These parameters were then used for the LightGBM package too, due to their similar inner workings. For both the BART and the Random Forest model grid search would have taken too much computation time. Hence, for these models, we report the parameters that worked the best based on our experiments. Let colsample_bytree, learning_rate, max_depth, alpha, n_estimators as c, l, d, a and n for XGBoost and LightGBM. For BART let n_trees, n_chains, alpha, beta be n, c, a, b. For Random Forest let n_estimators be n. The parameter values can be found in Table 6.

Model	Parameters	Value	Search Space
XGBoost	c, l, d, a, n	0.85, 0.096, 8, 8.31, 164	[0.5,1], [0.01,0.1], [3,12], [5,15], [50,200]
LightGBM	c, l, d, a, n	0.85, 0.096, 8, 8.31, 164	
BART	n, c, a, b	200, 2, 0.95, 2	
Random Forest	n	200	

Table 6: Tree Methods Parameters

Due to technical constraints, for the OSRT we decided to choose $\lambda = 0.05$.

For the neural network the hyperparameters were chosen as listed in Table 7.

Parameter	Value	Search Space
learning rate	0.01	{0.001, 0.01, 0.1}
batch size	10	{5, 10, 20, 40}
epochs	30	{20, 30, 40, 80, 100, 500, 1000}
kernel regularizer	0.01	{0.01, 0.02, 0.1}
activity regularizer	0.01	{0.01, 0.02, 0.1}

Table 7: Tree Methods Parameters

Regarding the structure of the neural network, experiments were also performed with two additional hidden layers and various dropout layers at different positions. The performance was also compared for the sigmoid activation function.

G. Additional Information OSRT Configuration

For the implementation of OSRT, adding a depth constraint and limiting the depth of the resulting tree is possible, but also introduces the possibility of sub-optimal values of the objective, should the optimal tree depth exceed the depth constraint.

Further, note that in our implementation, we allow the OSRT to use similar support bounds in the creation of the tree. As described in Lin et al. (2022), if two features in the dataset are similar, bounds obtained for one feature to create a split in the tree can be leveraged to create bounds for the second feature in the tree, if the second feature were to replace the first one. This allows us to further reduce the search space.

H. Additional Results and Analysis

Additional results can be found in Table 8. All datasets that start with *freq* are based on the simple imputation method used within the main paper. The *man* datasets were completed with a more thorough imputation method. This semi-manual approach uses domain knowledge to impute values where possible. For example, missing values in variables representing sail area are set to zero for boats that are not sailing boats. For the remaining missing values, imputation is done by estimating the relationship between the variable in question and related variables with a Random Forest. This is done with the Stekhoven and Buehlmann (2012) R package, where the idea is to treat the imputation problem as a supervised learning problem. The Random Forest takes the variable in question as a dependent variable and all other given variables as independent variables. By estimating this interplay on the non-missing values it can then predict a missing value based on the variable values of the other variables in the same row. For imputing one variable only a subset of all other variables is taken into account because of computation time limits. This subset is determined based on domain knowledge as well. For example draught and

displacement can be hypothesized to be imputed well based on each other and length and beam of the vessel.

man and *freq* are based on the full feature space, while the other ones are either transformed based on PCA as described above or they have one dummy column per categorical variable removed as a base category.

The LightGBM model and Quantile Regression Forests are included and the non-baseCat datasets of the linear regression models are estimated using the Moore-Penrose inverse to still be able to find a solution in spite of the perfect multicollinearity.

	Datasets					
Models	man	manPCA	manBaseCat	freq	freqPCA	freqBaseCat
OLS	0.70	-1.14	0.70	-0.05	-0.86	0.00
Ridge Regression	0.58	-1.44	0.58	-0.26	-1.19	-0.26
LASSO	0.70	-1.10	0.70	0.00	-0.84	0.00
Elastic Net	0.70	-1.10	0.70	0.00	-0.84	0.00
XGBoost	36.27	35.61	36.10	36.37	35.66	**36.51**
Random Forest	34.37	34.46	34.40	34.95	34.81	34.93
OSRT	-20.53	-38.97	-16.24	-23.10	-21.18	-17.96
BART	18.65	18.27	18.35	17.77	18.27	19.13
Neural Network	22.13	22.42	20.24	-2.54	21.20	13.83
Quantile RF	33.41	33.67	33.42	33.78	33.72	33.80
LightGBM	35.89	36.13	35.88	36.11	36.05	35.84

Table 8: % decrease in prediction error relative to benchmark model where the benchmark model is OLS with *freqBaseCat* and the prediction error metric is the Root Mean Squared Error. The underlined results are the best performance for each respective model. The bold result indicates the best overall performance.

Figure 9 shows an individual conditional explanation (ICE) plot. This is similar to the PDP approach but does not average over the individuals but instead displays them separately. It is useful to see how uniform the partial dependence is across individuals. For example, concerning age, we can see similar behaviour for most individuals, but in some instances the price increases for increasing age. In the same figure, the PDP of length and beam provides an interesting comparison to the PDP of length and area found in the main part of the text. Whilst we see here that increasing the beam for the same length does not increase the price, one can see in the plot in the main text that increasing the ship's area while keeping the length fixed is associated with a price increase. Since the ship's area is length multiplied by beam and we keep length fixed, we are increasing the beam of the boat and observing a price increase. This shows a limitation of the PDP method, as it cannot fully separate correlated variables. As ship area and length (and beam) are correlated we get unclear results for these two PDP.

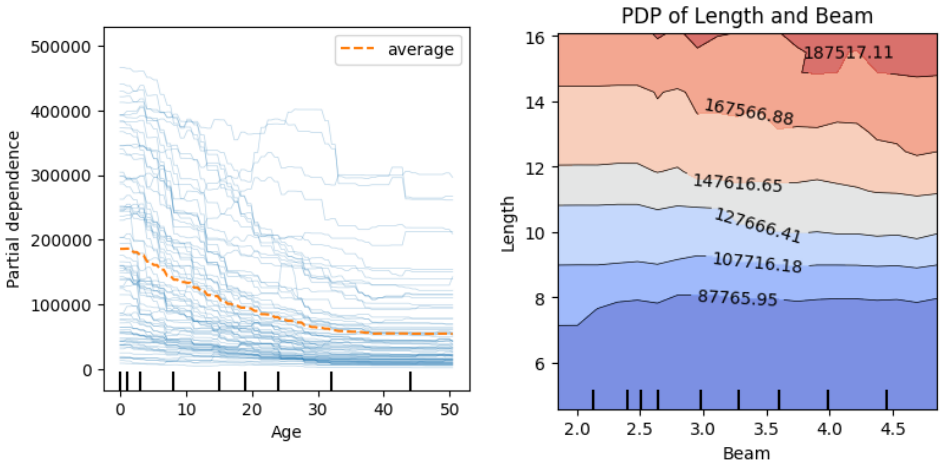

Figure 9: 1. ICE plot of age. 2. Joint PDP plot of length and beam.

If one makes use of the experimental *enable_ categorical* feature in the XGBoost package in Python (see Chen and Guestrin (2016)), the model can be estimated without having to separate unordered categorical variables into dummy variables. The package then makes the decision tree split based on subsets of the full set of categories. This method results in more concise partial dependence plots, which are shown in Figure 10. Like this, one can see at a quick glance that for a wheel control boat a swiveling keel is associated with a much higher price than a lifting keel.

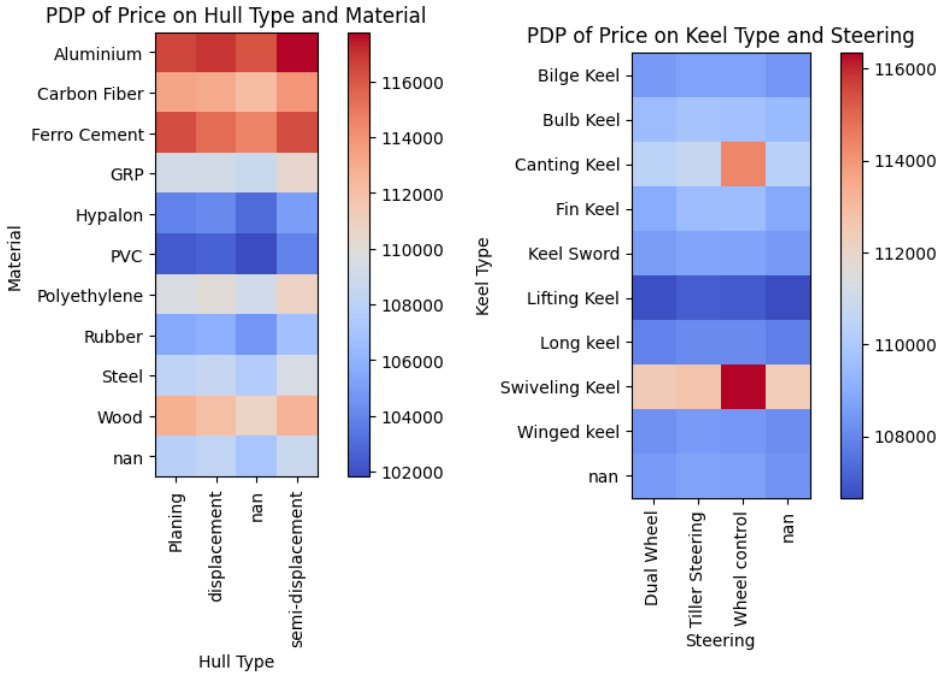

Figure 10: PDP heatmaps of categorical features based on an XGBoost model estimated on *man* dataset where the categorical features where not encoded as binary variables.

In Table 9 one can see what a local explanation of a boat price would look like in practice based on LIME. To keep the local model interpretable it is limited to a maximum of 10 features.

One can then interpret that for example the age is estimated to have a strong negative impact (consistent with the global interpretation of age) and length has a positive impact.

Variable	Value	Regression Coefficient
Age	41.00	-26927.40
Length	7.70	18322.74
Ship Area	22.18	12915.86
Total Horsepower	7.60	8510.64
Material_Polyethylene=0	True	-366.58
Material_Carbon Fiber=0	True	-289.52
Hull Type_displacement=0	True	256.89
Keel Type_Winged keel=0	True	-111.57
Keel Type_Bilge Keel=0	True	-64.31
Material_Ferro Cement=0	Tru	-40.83

Table 9: Explanation of an example boat with an estimated price of £7579.6, using the coefficients of a locally estimated Ridge Regression.

I. Software Packages

Table 10 indicates which Python (see Van Rossum and Drake (2009)) package or GitHub repository was used to implement each respective model.

Model or Method	**Python Package**	**Reference**
OSRT	osrt	Zhang et al. (2023a)
XGBoost	xgboost	Chen and Guestrin (2016)
LightGBM	lightgbm	Ke et al. (2017)
BART	bartpy	Coltman (2020)
Random Forest	sklearn	Pedregosa et al. (2011)
Linear Regressions	statmodels, sklearn	Virtanen et al. (2020), Pedregosa et al. (2011) Seabold and Perktold (2010)
LIME	lime	Ribeiro (2019)
Quantile Regression Forests	quantile-forest	Johnson (2024)
Neural Network	keras, sklearn	Chollet et al. (2015), Pedregosa et al. (2011)
Data Scraping	requests, Pillow	Chandra and Varanasi (2015), Clark (2015)
Image Classification	transformers, datasets, evaluate, accelerate, pytorch	Wolf et al. (2019), Lhoest et al. (2022) Lhoest et al. (2022), Gugger et al. (2022), Paszke et al. (2019)
Data preprocessing	geocoder, ast	Carriere (2017), Van Rossum and Drake (2009)
Data visualization	matplotlib, lowess	Hunter (2007), Rokem (2023)
Ubiquitous	pandas, numpy	McKinney et al. (2010), Harris et al. (2020)

Table 10: A table that indicates which python package was used in the implementation of each model.

The additional data imputation was done in R (R Core Team (2023)) using the *missForest* package by Stekhoven and Buehlmann (2012).

J. Feature List

Variable name	Type
Age	Numeric
Condition	Ordered Categorical (8 Layers)
Draught	Numeric
Displacement	Numeric
Certified No. of Persons	Numeric
Engine Hours	Numeric
No. of Cabins	Numeric
No. of berths	Numeric
No. of Bathrooms	Numeric
Headroom	Numeric
Toilets	Numeric
Showers	Numeric
Length	Numeric
Beam	Numeric
Negotiation Possible	Binary
Taxes Paid	Binary
Sailing Vessel	Binary
Motorized Vessel	Binary
Small Equipment Index	Numeric
Big Equipment Index	Numeric
Equipment Not Listed	Binary
Total Horsepower	Numeric
Fuel Capacity	Numeric
Mainsail Area	Numeric
Jib Area	Numeric
Genoa Area	Numeric
Spinnaker Area	Numeric
Gennaker Area	Numeric
Ship Area	Numeric
image water	Binary

Variable name	Type
image land	Binary
image brightness	Numeric
Latitude	Numeric
Longitude	Numeric
CE Design Category	Unordered Categorical (5 Levels)
Hull Type	Unordered Categorical (4 Levels)
Material	Unordered Categorical (11 Levels)
Steering	Unordered Categorical (4 Levels)
Keel Type	Unordered Categorical (10 Levels)

Table 11: List of all variables.