Eramus University Rotterdam

Erasmus university of economics

A transformation for the pricing of real estate data

Author

Jordi van Veen (577602)

Supervisor

dr. A. A. Naghi

Second assessor

dr. M. Zhelonkin

May 16, 2023

Abstract

Real estate data commonly contains a positively skewed distribution in the pricing distribution for the valuation of the residencies within the dataset, possibly indicating undesired effects such as heteroskedasticity, outliers, and nonlinearity. A transformation is a traditional method to account for positive skewness in data when using linear regression. However, literature about the effects of transformations to account for skewness in the dependent variable for real estate data using machine learning is inadequate. This study compares the effects of square root, log, and box-cox transformations on the dependent variable for real estate data on the predictive accuracy of machine learning models. We compare the predictive accuracy for penalized linear regression, support vector machine, random forest, and extreme gradient boosting on multiple real estate datasets. Analyses show that transformations can benefit the predictive accuracy of penalized linear regression and support vector machine. Additionally, we observe transformations adversely affect random forest and extreme gradient boosting in predictive accuracy.

Keywords— Transformations, box-cox, log, square root, support vector machine, penalized linear regression, random forest, extreme gradient boosting, real estate data

Table of contents

List of Figures

List of Tables

1 Introduction

The average property in the Netherlands is valued at 425.000 euros with the entire housing market valued at 3.4 trillion euros (Centraal Bureau voor de Statistiek, 2022). There are approximately eight million residences in the Dutch housing market. Annually 75 thousand properties are developed, making this market increasingly appealing for entrepreneurs (Centraal Bureau voor de Statistiek, 2021). As a result of the vast influx of money and, given the present trend, the increasing character of the market, the real estate market receives much attention from researchers and investors. However, the property market is not untouchable, as adverse developments in the housing market can have a worldwide economic impact, such as the financial crisis in 2008.

In 2008 the Dutch GDP fell by about four percent even though the companies in the Netherlands had no direct influence on the occurrence of the global financial crisis (Ministerie van Financiën, 2021). The housing boom in the United States was one primary cause for the Dutch GDP decline. Before the financial crisis began borrowers who in hindsight, were too risky obtained large mortgages for their low overhead. Due to the accessible nature of mortgages, demand for housing escalated quickly, causing housing prices to increase and consecutively excessive construction of housing. However, the "housing bubble" eventually popped due to disproportionate defaults on high-risk-containing mortgages, ultimately causing the financial crisis (Kahn, 2008). Property overvaluation greatly affected banks their decisions to offer risk-containing mortgages to consumers. With the irregularly high demand for housing driving the real estate pricing upward at the time, banks were unaware of issuing mortgages on unreasonable valued homes (Adelino et al., 2018).

Hedonic pricing models are one of the most common methods to asses the monetary worth of commodities for real estate valuation (Yazdani, 2021). Hedonic pricing models provide a tangible framework to estimate housing prices using heterogeneous characteristics such as the number of bedrooms, square foot of living space, and garden size (Kim, 1992), and less tangible environmental variables such as percentage of carbon particles in the air (Din et al., 2001; Smith and Huang, 1993). Researchers commonly employ parametric models to embody the widespread features to decide house prices. Linear regression (LR), among the most popular hedonic pricing models, is one example of the functional form for these parametric models, along with the double-log, exponential, and logarithmic models. However, LR relies on numerous assumptions where violations of these assumptions negatively impact predictive accuracy. A positive skew in the distribution of the dependent variable can indicate the presence of factors that can affect the accuracy of LR, such as heteroskedasticity, outliers, and nonlinearity. Skewed data is common

in real estate data due to the more extensive offer of cheaper housing compared to expensive residences. To account for skewness in data, researchers frequently use transformations to the dependent variable, such as square root, logarithmic, and box-cox transformations, which alter the distribution shape of the dependent variable (Heij et al., 2004).

For capturing linear relationships, LR performs well, but LR falls short of more complex relationships within the dataset. There is extensive research on how to use adapted forms of LR to capture non-linear relationships. (Motulsky and Ransnas, 1987; White and Domowitz, 1984). Nevertheless, interest in machine learning (ML) is rising due to the capability to account for sophisticated relationships in data. As ML can account for the complex interactions within real estate regressors, ML models such as penalized linear regression (PLR) and support vector machine (SVM) show remarkable performance compared to LR for real estate data (Baldominos et al., 2018; Fabozzi et al., 2020). Additionally, Borde et al. (2017) and Guliker et al. (2022) observe that tree-based estimation algorithms such as random forest (RF) and eXtreme Gradient Boosting (XGB) outperform LR when using real estate data.

This research aims to improve the accuracy of real estate pricing using an ML model-based approach. We compare the predictive accuracy of multiple ML models, specifically PLR, SVM, RF, and XGB, to our benchmark model LR. We assess the performance of the models using the mean absolute error (MAE), root mean squared error (RMSE), and Median Absolute Error (MedAE). We compare the performances of our ML models on various dependent variable transformations, including the square root transformation, logarithmic transformation, and box-cos transformation, to account for the highly skewed data in real estate pricing. By comparing the performances of the ML models with various transformations of the dependent variable, we answer which transformations on the dependent variable positively influences the predictive accuracy of ML models using skewed real estate data.

The remaining content of this paper is organized as follows. Section 2 provides an overview of current literature on credit scoring and imbalance. Then, Section 3 describes the data to train and test our models. Subsequently, in Section 4, we discuss the methodology of this research. Section 5 presents the results of our models on the data of Section 3. Finally, we discuss our results and provide a conclusion in Section 6.

2 Literature

The following section provides an overview of current modeling methods in real estate literature for manual appraisal, linear regression (LR), and machine learning (ML) models. Then, we discuss the literature about skewed data and common methods to account for skewed data in

regression.

2.1 Pricing in real estate

Commonly, experts are responsible for rating the monetary value of real estate properties. Northcraft and Neale (1987) show that amateurs and experts use different methods to evaluate properties. First is the concrete referent, which bases pricing on the property's age and listing price. Second, is features-only computation, where experts determine property price based only on the features of the property, such as the property's condition, size of the property, and location of the property. The last and most common is comparison computation. The comparison computation bases property value on the price of closed properties in the neighborhood proportional to each other's square footage. However, as Northcraft and Neale (1987) observe, these valuation methods are heavily biased and subjective as these methods do not incorporate all factors driving housing prices. During the crisis in 2008, the overvaluation of properties by manual appraisal substantially impacted consumer purchasing power (Ben-David, 2011). Kok et al. (2017) argue that manual appraisals of properties can vary around fifteen percent in price and take up to three weeks to determine, costing around 3000 euros extra for consumers, and show an automated model can decrease the variation in property pricing below ten percent providing price indication instantly instead of three weeks.

Researchers commonly employ hedonic pricing models as a model-based pricing method to evaluate real estate price estimates where specifically LR draws considerable attention to research (Abdulhafedh, 2022; Ghosalkar and Dhage, 2018; Ozgur et al., 2016). Researchers use LR substantially in real estate pricing utilizing a model-based approach due to its uncomplicated implementation and interpretability. LR allows for a straightforward interpretation of marginal effects through the explanatory variables, elasticities, and log-odds ratios. LR also supports traditional econometric inference via F-tests and t-tests of the coefficients. However, the accuracy of LR declines for more complex relations within the dataset as, for example, extra bedrooms in apartments have a different pricing influence compared to an extra sleeping room in a stand-alone house. Because LR assumes a linear relationship for feature interactions, it falls short of explaining more complex nonlinear relationships Guliker et al. (2022).

2.2 Machine learning models

ML is a common method of choice for high-accuracy prediction in multidimensional analysis. Its substantial predictive accuracy is well known in econometric literature to engage the most complex problems. Two groups of ML models are among the most common models in econo-

metric literature. First is the individual classifier, which bases predictions on a single model. Researchers use penalized linear regression (PLR), the penalized form of LR, in widespread applications such as real estate pricing Fabozzi et al. (2020). Castelli et al. (2020) use PLR and predict building prices showing competitive results for the lasso penalty instead of regular LR. The penalized form of LR is a feasible solution for regression problems to account for linear relations while compensating for over-fitting using a penalty term. Jamil et al. (2020) favor the ridge penalty over the lasso penalty for prediction if regressors have high colinearity and show competitive results compared to LR in real estate pricing data. The elastic net penalty, a weighted combination of the ℓ_1 and ℓ_2 penalty, allows for the feature space sparsity of the ℓ_1 term and the consideration of multicollinearity using the ℓ_2 term. Ogutu et al. (2012) observe the elastic net penalty to outperform individual penalty terms and positively impact model sparsity while simultaneously addressing colinearity between regressors. By tuning the elastic net weight penalty parameter, the penalty term can still adopt the functional form of the ridge or lasso penalty if it positively impacts model performance. However, the accuracy for PLR declines for higher-order nonlinear interactions. Support vector machine (SVM), a decision boundary-based algorithm, can account for nonlinearity through its flexible hyperplane by kernel choice (Boser et al., 1992). Although SVM can account for nonlinear behavior, the prediction accuracy heavily depends on the proper choice of kernel and hyperparameters but struggles with categorical data (Meyer et al., 2003). For real estate data, Li et al. (2009) show the competitive performance of SVM when compared to LR for real estate data. Yu and Wu (2016) compare the performance of SVM and PLR with a lasso penalty and shows the relatively better performance of SVM.

Homogeneous ensemble learners base decision-making on a pool of likewise models to increase the performance of weak base learners thereby reducing variance and increasing predictive accuracy. Baldominos et al. (2018) compare multiple supervised ML models and find individual classifiers like SVM have a good accuracy. Baldominos et al. (2018) observe homogeneous ensemble learners based upon decision trees that are consistently high-performing models. RF is a robust and highly accurate homogeneous ensemble learner basing prediction upon groups of decision trees, which can account for more complex interactions. Borde et al. (2017) compare the real estate pricing prediction accuracy of LR to RF and show competitive results for RF. In the spectrum of boosted regression trees, Chen and Guestrin (2016) propose eXtreme Gradient Boosting (XGB), which outperforms conventional boosting algorithms, such as generalized boosting regression modeling in terms of predictive accuracy (Nielsen, 2016). Bentéjac et al. (2021) show ore recent models like Catboost and light GBM have comparable predictive accuracy to XGB. For real estate data Zhao et al. (2019), show XGB to outperform more conventional

ML models in predicting house prices.

2.3 Skewed data

Skewed data, specifically in the dependent variable, is common in fields such as real estate pricing, credit card fraud, and mortgage defaults (Bond and Patel, 2003; Diaz-Serrano, 2005; Makki et al., 2019). Skewness in the dependent variable in real estate data occurs as a result of greater demand for less expensive properties compared to more expensive housing. Skewness can have a negative effect on LR because it relies on general assumptions that ensure the estimator is unbiased and efficient, such as the homoskedasticity and linearity assumption (Heij et al., 2004). If skewed data is present, careful consideration is necessary as a violation of the assumptions interferes with the unbiasedness and efficiency of the estimator. Two solutions are common to account for skewness in data. The first solution is a model-based approach to LR assumption violations. If the data does not meet the assumptions of LR, methods such as generalized linear models are more trustworthy compared to LR (Changyong et al., 2014). The second solution is transformations of the dependent variable such as a log, square root, or box-cox transformations Osborne (2010). These transformations notably account for the skewness in the data Benoit (2011). Osborne (2010) argue the box-cox transformation to incorporate many common transformations such as the log, square, and cubic root. Although, Osborne (2010) also argue that the log and square root transformations are still common in literature to ensure the assumptions of LR. However, Krawczyk (2016) argue that the econometric literature lacks research about imbalance for ML models. Even though transformations are not unheard of, researchers such as Potrawa and Tetereva (2022) apply a box-cox transformation for their skewed real estate pricing dataset for prediction. However, Changyong et al. (2014) argue transformations can harm the empirical form of a distribution as, for example, the log transformation offsets the positive skew towards a negative skew in the distribution of the dependent variable. Changyong et al. (2014) recommend careful consideration for applying transformations as they can impose a negative skew instead of the desirable bell curve shape. Kiely et al. (1995) show a square root transformation can force a more regularly distributed dependent variable. In contrast, Kiely et al. (1995) observe a log transformation to change the positive skew of the data to a negative skew, emphasizing the essence of using transformations carefully. Also, Sakia (1992) and Silva and Tenreyro (2006) discuss it is seldom that transformations such as the box-cox or log transformation help fulfill all assumptions of LR. However, Sakia (1992) emphasizes transformations have the potential to help improve model accuracy if adequately used.

3 Data

In the following section, we discuss the datasets used for this research. The five curated datasets originate from Kaggle. For each dataset, we describe the size and characteristics of the dataset. Also, if relevant, we describe pre-processing measures to account for outliers and missing values. We describe the Boston housing dataset, the Russia rent dataset, the housing prices dataset, the advanced house prices dataset, and the Saudi dataset.

3.1 Boston dataset

The first dataset, the Boston housing dataset, contains 511 data points about the median values of privately owned houses of neighborhoods in Boston. The Boston dataset includes information about neighborhood quality, such as pupil-teacher ratios, the concentration of nitric oxide in the air, and tax rate. In the density plot of Figure 1, observe the positively skewed nature of the distribution. The dependent variable of the dataset contains information about the median cost of privately owned houses in the neighborhood. The median values of the homes within the dataset range from 5000 to 67,000 dollars. Appendix A provides an overview of the variables with additional descriptive statistics and a description of the variables.

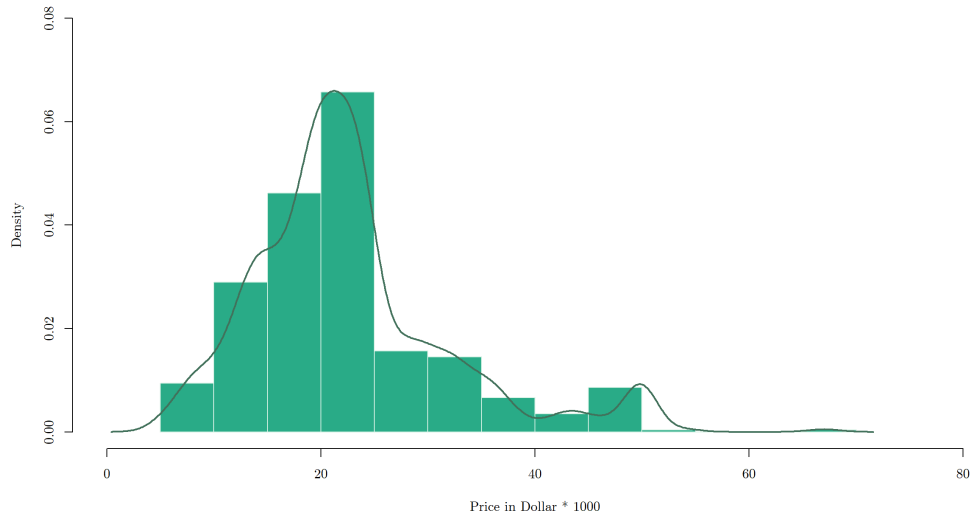


Figure 1: Density plot of the dependent variable of the Boston dataset

3.2 Russia rent dataset

The second dataset in this research, the Russia rent dataset, contains 1446 data points and describes information about the apartment size, public transport near the apartment, and type of the deal for the apartment. Examples of the variables in the dataset include the nearest station name for the metro, the size of the living area, and the number of views the apartment got.

The dependent variable considers the rent price of multiple residencies in Russia. We describe the variables and their respective descriptive statistics in Appendix A. Figure 2 illustrates a density plot of the distribution for the rent prices within the dataset. The density plot shows a high frequency of observations in the tails and does not coincide with the bell-shaped curve. In the dataset, the realtor variable contains various false data points, as these data points did not represent a category. We impute these data points using the mode of the variable.

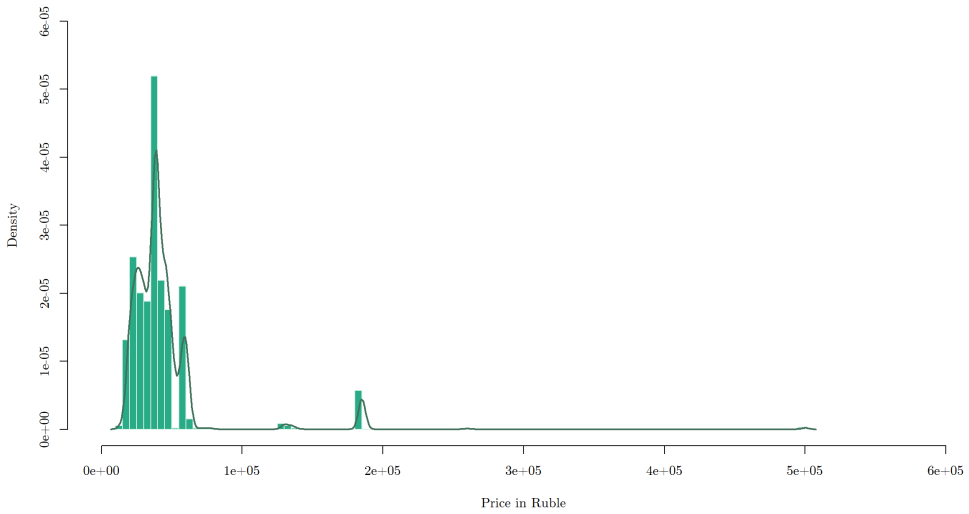


Figure 2: Density plot of the dependent variable of the Russia rent dataset

3.3 Housing prices dataset

The housing prices dataset contains information about various houses where the dependent variable explains the housing prices ranging from 1,75 million to 13,3 million dollars of an undisclosed area. Figure 3 shows the distribution of the house prices in the dataset, indicating a slight positive skew of the dependent variable. The dataset consists of 546 data points where the regressors comprise basic information about a residency. Examples of these variables in the dataset are the number of bedrooms, stories of the building, and parking area size. We further describe these variables and respective descriptive statistics in Appendix A.

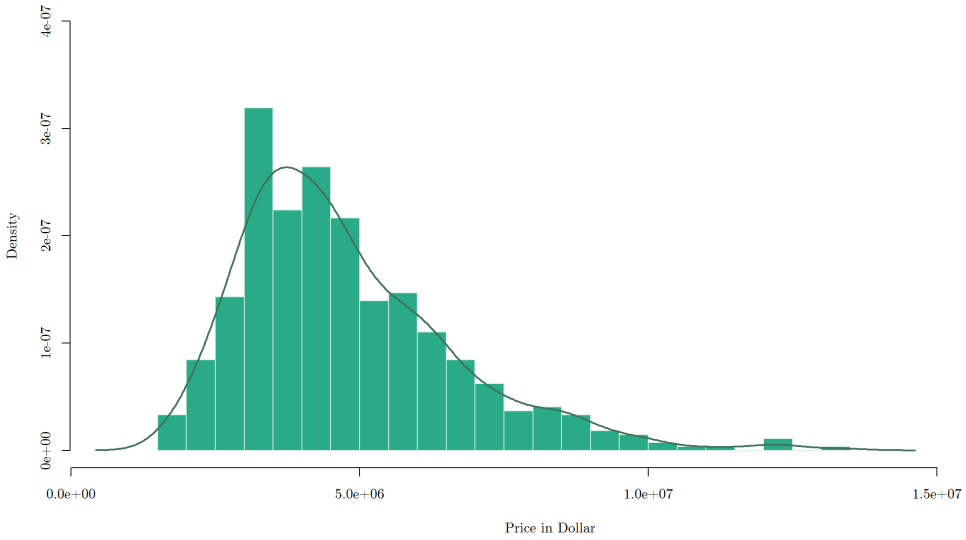


Figure 3: Distribution of the house price in the House prices dataset

3.4 Advanced house prices dataset

The advanced house prices dataset, similar to the housing prices dataset, contains valuation prices of houses ranging from 34,900 to 755,000 dollars of an undisclosed area. However, where the regressors in the housing prices dataset contain more basic characteristics of a home such as the number of bedrooms or bathrooms, the advanced house prices dataset contains more complex information, such as the type of roof or the length of the perimeter. The dataset contains 1460 data points, where 43 variables are categorical, and 37 variables are continuous. As we one-hot-encode categorical variables, the dimensions of the $n \times m$ dataset with n the number of data points and m the number of regressors will result in m and n almost to have similar dimensions, which can have consequences for our models. High dimensionality in data can cause ML models to overfit and diminishes the predictive accuracy of models (Gnana et al., 2016). Therefore we exclude a proportion of the categorical variables based on our reasoning. In Appendix A, we provide an overview of the variables used for this research with their respective description and descriptive statistics. Figure 4 shows the distribution with a positively skewed dependent variable in the advanced house prices dataset.

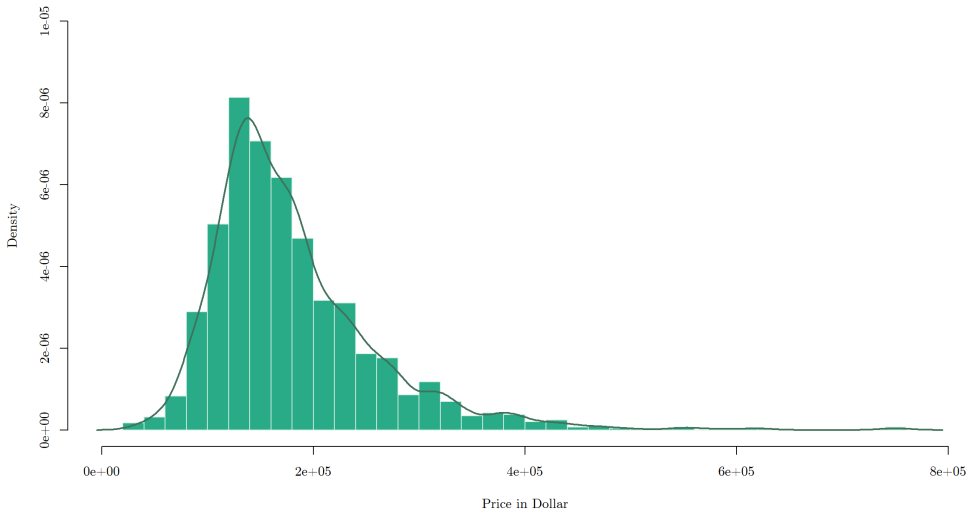


Figure 4: Density plot of the dependent variable of the Advanced house prices dataset

3.5 Saudi dataset

The last dataset used for this research is the Saudi dataset containing the housing prices of various villas in Saudi Arabia. The dataset contains 1417 data points with regressors containing information about the villa's location, size, and additional features. Examples of the independent variables are the city of the villa, if a pool is present, and the number of rooms in the villa. As Figure 5 illustrates, the dependent variables contain large values on the right side of the mean, thus a positive skew of the distribution. We provide the full description of all variables in Appendix A with respective descriptive statistics.

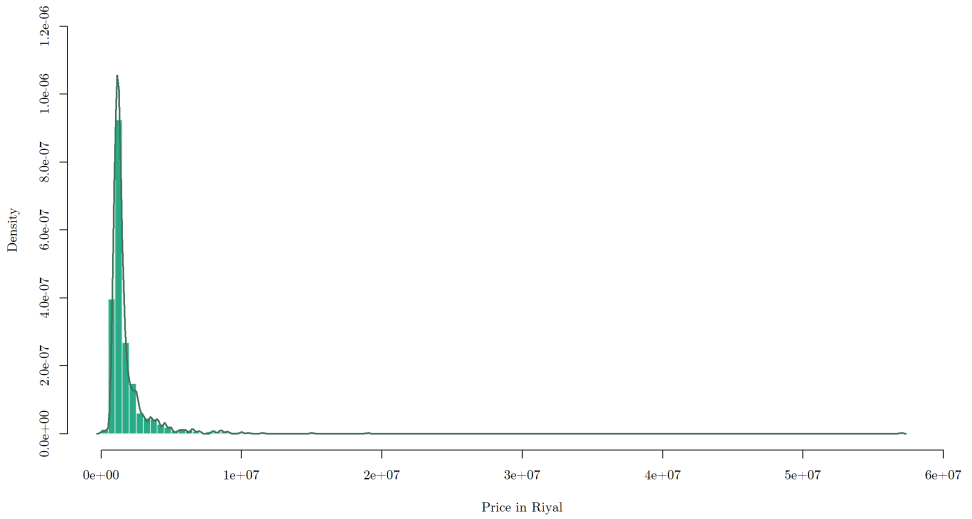


Figure 5: Density plot of the dependent variable of the Saudi dataset

4 Methodology

In the upcoming section, we first detail the transformations to which we subject the dependent variable. Second, we discuss D'Agostino's K-squared test, which we use to detect skewness and assess the normality of the dependent variable after transformations. Then, we elaborate on the mathematical background of the aforementioned models in Section 1, which are LR, PLR, SVM, RF, and XGB. Followed by the introduction of all five models, we will discuss hyperparameter tuning. Lastly, we discuss pre-processing of the data and the performance assessment through statistical testing of the models.

4.1 Transformation of the dependent variable

We use three transformations to account for the skewness of the dependent variable within the dataset. Before tuning and training the ML models, we transform the dependent variable using the root, log, and box-cox transformations. Then, we tune the hyperparameters, train our ML models and predict our test set using 5×2 cross-validation, which we will elaborate on in Section 4.4. After prediction, we transform the predicted values of our models using the inverse of the same mutation that the data was previously transformed with to ensure the magnitude of the errors for the respective transformation is in the same order as the untransformed dependent variable to assure equal comparison.

As mentioned in the previous paragraph, we use the square root, log, and box-cox transformation. For datapoints $i = 1, ..., N$ the square root transformation mutates datapoint y to $\tilde{y}$ as follows

$$\tilde{y_i} = \sqrt{y_i}.$$

The log transformation, a common transformation in econometric literature, attains $\tilde{y}$ using the natural logarithm as follows

$$\tilde{y_i} = log(y_i).$$

Last, the box-cox transformation considers all values of λ between minus five and five to attain the optimal value for the dependent variables distribution to resemble most closely to a normal distribution (Box and Cox, 1964). The box-cox algorithm attains $\tilde{y}$ using

$$\tilde{y_i} = \frac{y_i^\lambda - 1}{\lambda} \qquad if \quad \lambda \neq 0,$$
$$\tilde{y_i} = log(y_i) \qquad if \quad \lambda = 0.$$

For $\lambda = 0$, the functional form of the box-cox algorithm acquires the log transformations' functional form.

Figure 6 illustrates the effects of the square root, log, and box-cox transformation on data. The x-axis illustrates the original value of data, and the y-axis depicts the transformed form. Note that the primary purpose of the transformations is to diminish higher values more heavily, which converts a positive skew towards a more regular bell curve. For these examples, the log transformation is the most severe. However, lower values of λ can impose more severe transformations to higher values. In Appendix D, we provide an overview of the square root, log, and box-cox transformation effects on the dependent variable of our datasets.

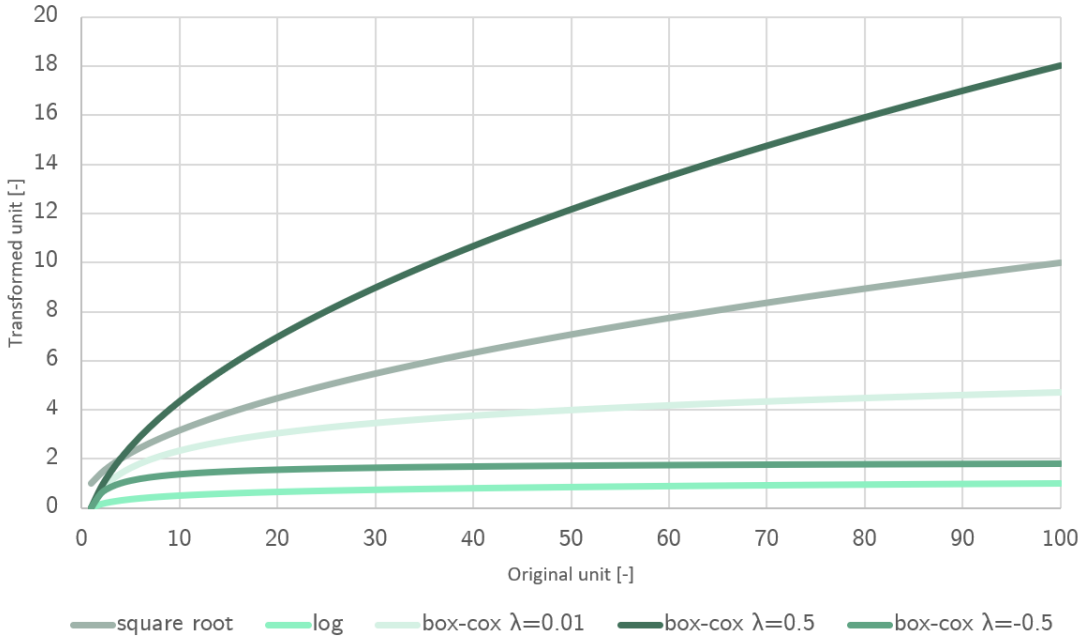


Figure 6: Figure illustrating the effects of various transformations where the original unit, through a transformation, mutates to the transformed unit.

4.2 D'Agostino's K-squared test

After applying the transformations to the dependent variable of the five datasets mentioned above, we perform D'Agostino's K-squared test to evaluate the similarity to a normal distribution of the untransformed dependent variables according to the skewness and kurtosis.

D'Agostino's K-squared test evaluates the goodness-of-fit of a variable according to the kurtosis and skewness compared to a normal distribution (D'agostino and Pearson, 1973). We determine the statistic K^2 for D'Agostino's K-squared as

$$K^2 = W_1(S)^2 + W_2(K_u)^2,$$

with S, a metric for the skewness and K_u, a metric for the kurtosis. The functions W_1 and W_2 represent a transformation of the sample skewness and kurtosis. A sample skewness of

two indicates the dependent variable is moderately skewed (Curran et al., 1996). For the full formulation of $W_1(S)$ and $W_2(Ku)$ we refer to the papers of D'agostino and Pearson (1973) and Anscombe and Glynn (1983). We compare D'Agostino's K-squared test statistic with two degrees of freedom chi-squared distribution at a 0.05 and 0.01 significance level.

4.3 Models

As discussed at the start of this section, we evaluate five models in this research. In the following paragraphs, we further elaborate on LR, PLR, SVM, RF, and XGB.

4.3.1 LR

LR is a common model among widespread academic research fields and industries. The models' simplistic construction allows for easy implementation and good interpretation of decision-making. The model, however, relies upon heavy assumptions, such as homoskedasticity and the independence of residual errors, which has implications for the predictive accuracy of LR. Also, LR is prone to overfitting as it does not rely upon a penalty, which can cause inaccuracies on unseen data. We denote LR, which assesses dependent variable y_i with $i = 1, ..., N$ observations for datapoint $x_{i,j}$ with regressors $j = 1, ..., p$ as

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}.$$

4.3.2 PLR

As mentioned in Section 4.3.1, LR lacks a penalty term to prevent the model from overfitting. Customary options for PLR are a ℓ_1 norm, a ℓ_2 norm, or a weighted combination of both. Where the ℓ_1 norm shrinks regressors ultimately to zero resulting in a sparser model and hence is more interpretable. The ℓ_2 norm shrinks regressors substantially small and, therefore, desirable in possible multicollinearity. The elastic net is a combination of both and has a parameter that controls the proportion of the ℓ_1 and ℓ_2 norm, respectively. As there is no preference for the model sparsity or accounting of multicollinearity property and ultimately, as by tuning of θ, it can still optimize to either the ℓ_1 or ℓ_2 norm, we opt for the elastic net as a penalty. We calculate the loss function of the elastic net penalty as

$$\mathcal{L}(\beta) = \arg\min_{\beta} \left[-\sum_{i=1}^{N} \sum_{j=1}^{p} (y_i - x_i \beta_j)^2 + \lambda \left(\theta \sum_{j=1}^{p} |\beta_j| + (1-\theta) \sum_{j=1}^{p} |\beta_j|^2 \right) \right],$$

where y_i represents our dependent variable for $i = 1, ..., N$ and our parameter vector which we denote with β_j for $j = 1, ..., p$. The parameter λ is the penalty term for elastic net with higher values imposing larger penalties.

4.3.3 SVM

The regression form of SVM, support vector regression (SVR), fits a hyperplane within its margins and maximizes the total amount of data points within the area of the margins. Data points outside of the margins impose a penalty on the cost function of SVR. For SVR, we minimize the Lagrangian function

$$\mathcal{L}(\xi) = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}(\xi_i - \xi_i^*)(\xi_i' - \xi_j^*)K(x_i, x_i') + \epsilon\sum_{i=1}^{N}(\xi_i + \xi_i^*)\sum_{i=1}^{N} y_i(\xi_i - \xi_i^*),$$

for data points i,i'=1,...,N, which are subject to the constraints

$$\sum_{i=1}^{N}(\xi_n - \xi_n^*) = 0,$$

$$0 \leq \xi_n \leq C,$$

$$0 \leq \xi_n^* \leq C.$$

The parameter C represents the regularization parameter and the parameters ξ and ξ^* are the slack variables that allow for a tolerance of the constraints for data points outside of the margin. For this research, we employ the radial basis function where we denote the kernel function $K(x_i, x_i')$ as

$$K(x_i, x_i') = exp(\gamma||x_i - x_i'||^2),$$

where hyperparameter γ determines the flexibility of the hyperplane for SVR.

4.3.4 RF

The RF model of Breiman (2001) aggregates individually trained decision trees and bases prediction on majority voting. RF, also known as a bagging method, trains weak independent learners to form a group. The group of weak learners reduces the variance of the weak individual learners and thus increases the power to predict accurately. Individual decision trees can display high variance in their results. As such, we can lower this variance by bundling a group of decision trees. To lower the variance of individual trees, for a subsample of the features, RF grows splits for the least impure feature of the subset, where RF defines impurity as the best possible split to separate the dependent variable.

4.3.5 XGBoost

For our last model, we consider a common high-predictive performance model: the XGB algorithm introduced by Chen and Guestrin (2016). We adopt the notation of Chen and Guestrin

(2016) for the true outcome $y_i \in \mathbb{R}$ in parameter set $x_i \in \mathbb{R}^m$ the true outcome XGB sequentially builds $k = 1, ..., K$ decision trees f_k and predicts $\hat{y}_i$ for $i = 1, ..., N$ dependent variables by

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i).$$

For consecutive trees, the weighted importance updates for every observation corresponding to the error of the previous decision tree through gradient boosting. By optimizing, we greedily add DT f_k, which best optimizes[1]

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where l is the convex loss function to measure prediction error between y_i and $\hat{y}_i$ and Ω a regularization term to prevent over-fitting of f_k.

4.4 Tuning

The order in how well ML performs substantially correlates with its respective hyperparameters. Before training a ML model, its respective hyperparameters are predetermined and can considerably influence predictions. Since we fix hyperparameters before learning, we tune these hyperparameters.

Wu et al. (2019) compare grid search with bayesian optimization, where grid search slightly outperforms bayesian optimization, but this comes at a cost. Grid search evaluates all possible hyperparameter combinations causing long runtime. random search, a common alternative for grid search, evaluates the parameter space randomly given a predefined amount of evaluations. However, as Wu et al. (2019) discuss, this comes at a cost for bigger dimensionality problems. For tuning complex models, such as XGB (Kapoor and Perrone, 2021), bayesian optimization finds optimal hyperparameter configurations notably quicker than random search and grid search due to its more structured approach. The two-step approach of bayesian optimization makes it suitable for hyperparameter tuning. First, the bayesian optimization algorithm randomly explores the hyperparameter space and proposes a preliminary hyperparameter combination. Then, bayesian optimization seeks the optimal hyperparameter setting by repeatedly updating prior beliefs of the unknown objective function and using the updated beliefs to evaluate the following point to evaluate around the preliminary hyperparameter combination determined in the first step. Using a more direct approach for hyperparameter tuning bayesian optimization saves evaluations compared to random search and grid search. We limit bayesian optimization to 200 steps in the exploratory phase and 50 in the second phase due to computational constraints.

[1] We refer to Chen and Guestrin (2016) for the full mathematical formulation of XGBoost.

For model performance comparison using smaller datasets, Raschka (2018) favor the 5×2 cross-validation method of Dietterich (1998). The 5×2 cross-validation method is a nested cross-validation approach to validating model performance. Commonly nested cross-validation consists of two cross-validation loops consisting of an inner hyperparameter validation and an outer test set loop. For 5×2 cross-validation, the inner loop evaluates the performance of hyperparameter settings in 2 folds, resulting in two splits of 50/50 train/validation folds. In the outer loop, five folds of a train/test split of 80/20 evaluate model performance given optimal hyperparameter settings from the inner loop. Resulting in five evaluations of performance on each dataset for each model.

For the PLR models, we tune the penalization hyperparameter and the elastic net distribution parameter. The penalty term allows control for overfitting, and the adjustable parameter aids model sparsity and multicollinearity (Zou and Hastie, 2005).

SVM lends its suitability to its N-dimensional hyperplane. Proper kernel choice allows for comprehension of the complexity within the dataset using its hyperplane for regression. Sain (1996) observe the RBF kernel to encounter fewer numerical issues than other kernels; hence we opt for the RBF kernel. Probst et al. (2019a) discuss additional hyperparameters for SVM and observe gamma to influence the performance of SVM notably if adequately tuned. The C term is a ℓ_2 penalty, which acts as a strength regularization parameter controlling for overfitting and observations outside of hyperplane margins.

Unlike a model such as SVM, which requires proper tuning for performance, RF achieves notable performances on its standard settings (Probst et al., 2019b). Though, Probst et al. (2019a) shows tuning can improve model performance if computational constraints allow it. We tune the number of estimators, the max depth of the tree, the maximum number of features, and the impurity criterion for the RF model. Standard impurity criteria for RF regression are squared error, absolute error, Friedman mean squared error and Poisson penalty. By keeping the max depth of the tree relatively shallow and by optimizing the size of the set of features, we account for restraining model complexity and consider overfitting for RF.

The XGB model has a notable amount of hyperparameters to tune. As tuning all hyperparameters given the size of our dataset is infeasible to ensure adequate model performance, we select a subset of hyperparameters to tune and hold on to standard hyperparameter settings for the remaining parameters. Probst et al. (2019a) evaluate the tunability for the hyperparameters of various models, tunability indicates the influence tuning of the hyperparameters can have on model performance and find proper tuning of the learning rate and the number of estimators can most notably influence model performance of XGB. Due to personal preference, we

include the gamma and max depth hyperparameters. XGB bases further learning on residual errors of the previously trained tree. Higher gamma values make the model more conservative on splits, hence controlling for overfitting. Max depth is the maximum depth of an individual decision tree. Excessive values of the max depth cause trees to saturate on data impacting model performance negatively. Last, as XGB is a complex model to train, computation time is a factor to consider. Kapoor and Perrone (2021) evaluate the influence of subsamples on model performance and conclude two notable remarks. First, computation time scales linearly with sample size. Second, for subsamples above 75%, model performance is minimally compromised. Hence we set the sub-sample parameter to 80% to reduce computation time. In Appendix C, we provide an overview of the hyperparameters we tune with their respective range and description.

4.5 Pre-processing

In Section 3, we discussed pre-processing methods to consider outliers and missing values for each specific dataset, which we assess and fill manually with either the mode for categorical variables or the median for continuous variables. To ensure similar ranges in our regressors for training our models, which is particularly necessary for PLR and SVM, we use normalization as the method is more robust to outliers in our regressors, which we favor because we don't perform extensive analysis on the distribution of our regressors. We use the min-max scaling as a normalization method since min-max scaling preserves the original distribution of the regressor and is easy to implement. The categorical variables in this research are accounted for by using one-hot encoding.

4.6 Testing model performance

To assess the effects of transformations on the dependent variable for various ML models. We tune the hyperparameters of our ML models PLR, SVM, RF, and XGB using the MAE, RMSE, and MedAE and assess the performance of our models using the respective performance metric with LR. We elaborate on the performance measures of this research in Appendix E. Section 5.1 shows not all dependent variables comply with a normal distribution, so we use non-parametric tests as we can not guarantee all assumptions necessary for a parametric test. As a non-parametric test, we use the Dunn test, a non-parametric ranked multiple comparison test that evaluates group performance.

The Dunn test is a non-parametric rank-based test considering multiple testing, where we use a Bonferonni correction to consider multiple testing due to personal preference (Dunn, 1964). With a null hypothesis of equal performance for groups and an alternative hypothesis of

unequal performance for groups, we compare the model performances of PLR, SVM, RF, and XGB with our reference model LR. We adopt the notation of Dunn (1964) for the pair-wise comparison between model T with samples n_i for $i = 1, ..., n$ of models $s = 1, ..., S$, with $S = 5$ for our comparison. The models are ranked based on their performances on the data. As we perform 5×2 cross-validation to assess the performance of a model for each dataset, we attain $n = 5$ performance metrics. If the performance of models is tied, we average the tied ranks and designate the average to each tie.

$$z_{s,s'} = \frac{\sum_i T_{i,s} / \sum_i n_{i,s} - \sum_{i'} T_{i',s'} / \sum_{i'} n_{i',s'}}{\sigma_{s,s'}} \sim N(0,1),$$

$$\sigma_{s,s'} = \sqrt{\left[\frac{N(N+1)}{12} - \frac{\sum_{v=1}^{w} u_v^3 - u_v}{12(N-1)}\right]\left(\frac{1}{\sum_i n_{i,s}} + \frac{1}{\sum_{i'} n_{i',s}}\right)},$$

with $N = \sum_{s=1}^{S} \sum_{i=1}^{N} n_s$ the total sample size of the S models, u_v the number of observations of all S models for $v = 1, ..., w$ with w the total number of tied ranks across all S models.

5 Results

In the following section, we first discuss the results of the D'agustino K-squared test, which tests normality according to sample skewness and kurtosis. We perform the test on the Boston, Russia rent, housing prices, advanced house prices, and Saudi data set for the untransformed, square root, log, and box-cox transformation of the dependent variable for each respective dataset. We then discuss the performances of LR, PLR, SVM, RF, and XGB on all five datasets for every transformation, and we assess model performance by comparing the MAE, RMSE, and MedAE using the Dunn test.

5.1 D'Agostino's K-squared test

Table 1 shows the D'Agostino K-squared test results, which we use to assess the statistical resemblance of our dependent variable to a normal distribution based on the sample skewness and sample kurtosis. The null hypothesis is of similarity to a normal distribution, with the alternative hypothesis of no similarity. The test is evaluated on a 5% and a 1% significance level.

Table 1: The p-values indicating statistical resemblance to a normal distribution according to the D'Agostino's K-squared test

Transformation	**Boston**	**Russia rent**	**Housing prices**	**Advanced house prices**	**Saudi**
None	0.000**	0.000**	0.000**	0.000**	0.000**
Square root	0.000**	0.000**	0.000**	0.000**	0.000**
Log	0.001**	0.000**	0.332	0.000**	0.000**
Box-Cox	0.064	0.000**	0.792	0.000**	0.000**

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. *An asterisk indicates a significant difference between a normal distribution and the respective distribution of (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, **two asterisks indicate significance on a 1% significance level.

Based on Table 1, we observe that the test statistic for the Box-Cox transformed dependent variable most often provides an indication for statistical resemblance to a normal distribution, compared to the other transformations. The log-transformed dependent variable only statistically resembles a normal distribution for the housing prices data set, as indicated by the insignificance of the test statistic in Table 1. Accounting for the positive skew in the distribution of our dependent variable compared to a normal distribution, as is discussed in Section 2, should improve the predictive accuracy of our models.

5.2 Performance of ML models

In the following section, we discuss the results of LR, PLR, SVM, RF, and XGB on the Boston, Russia rent, housing prices, advanced house prices, and Saudi dataset. The tables below only show the mean performance metrics over all five folds. For exact numbers per fold, we refer to Appendix F.

5.2.1 Boston dataset

Table 2 shows the results of all considered models for every transformation of the Boston dataset, evaluated on MAE, RMSE, and MedAE.

Using transformations on the dependent variable lowers the LR prediction errors based on all three performance measures. However, only the log transformation shows statistical evidence of the increase in performance for the MAE. The statistical difference compared to the untransformed form shows transformations can address for LR assumption violations.

Similar to LR, the PLR and SVM prediction errors are lower when a transformed dependent variable is used. However, we reject the null hypothesis of equal model performance on a 1% and 5% for PLR only for the log and box-cox transformation, respectively, of the MAE metric. The box-cox transformation for SVM of the MAE significantly increases the predictive accuracy of the model on a 1% significance level showing transformations can aid in fitting the hyperplane of SVM based on MAE.

That said, based on the RMSE and MedAE, none of the models statistically improve using a transformation on the dependent variable. Specifically, for a transformation on LR to not statistically improve model performance for the RMSE and MedAE, the transformation does not fully account for possible outliers, heteroskedasticity, or nonlinear effects. Thus, given that RMSE penalizes larger mispredictions more severely than the MAE and the MedAE is a robust performance metric, we cannot conclude that the model performance of our ML models statistically differs from LR based on transformations of the dependent variable, which suggests that the transformations do not aid in model performance in the Boston dataset.

Moreover, Table 2 shows that the RF model statistically outperforms LR on all transformations based on the MAE. The XGB model only outperforms the LR model in case of no transformation on the dependent variable, using the MAE as a measure of performance. Neither of the remaining performance measures shows a statistical difference between the benchmark LR model and the PLR, SVM, RF, and XGB models. As the Boston dataset contains regressors with possible interaction effects, such as the pupil-teacher ratio, the proportion of uneducated people, and the crime rate. We require models to account for nonlinear effects in model construction resulting in RF and XGB capturing the interactions between regressors better than LR, thus significantly outperforming LR. However, transformations can even negatively model the performance of tree-based algorithms as the log transformation for XGB significantly differs from the original data based on the higher value for the MAE.

Table 2: Mean of the 5×2 cross-validated models on the Boston dataset for the MAE, RMSE, and MedAE

Boston mean absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	3.767	3.740	3.516	2.377^{*}	2.340^{*}
Square root	3.443	3.463	2.847	2.380^{*}	2.562
Log	$3.263^{\dagger\dagger}$	$3.281^{\dagger\dagger}$	2.826	2.373^{*}	$3.006^{\dagger}$
Box-Cox	3.318	$3.326^{\dagger}$	$2.768^{\dagger\dagger}$	2.412^{**}	2.599

Boston root mean squared error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	6.183	6.124	5.762	3.876	3.923
Square root	5.819	5.900	5.293	4.010	4.261
Log	5.679	5.793	5.190	4.030	5.028
Box-Cox	5.715	5.819	5.162	4.034	4.598

Boston median absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	6.183	6.169	5.843	3.921	4.008
Square root	5.819	5.864	5.142	3.968	4.411
Log	5.679	5.744	5.120	4.114	5.004
Box-Cox	5.715	5.817	5.088	4.126	4.992

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. †A dagger indicates a significant difference between the respective (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, ††two daggers indicate significance on a 1%. significance level. *An asterisk indicates a significant difference between LR and the respective model on a 5% significance level, **two asterisks indicate significance on a 1%. significance level.

5.2.2 Russia rent dataset

We show the results of all five models on the Russia rent dataset in Table 3. For both the RF and XGB models, the transformations provide no increase in performance for the three performance measures. In addition, the box-cox transformation of XGB shows a statistical difference compared to the untransformed error of the dependent variable, where the error of the box-cox transformations is higher. This statistical difference indicates transformations induce negative effects on the formation of proper splits for the tree-based algorithms impacting predictive accuracy.

For both the LR and PLR models, the log and box-cox transformation show a statistical difference considering the MAE. However, besides the MAE for none of the other performance measures, there is enough statistical evidence to reject the null hypothesis of equal model performance for the square root, log, and box-cox transformation, which is comparable to the Boston dataset. A possible explanation for the ineffectiveness of the transformations is the substantial amount of data points in the tails of the distribution of the dependent variable as described in Section 3, which can cause heteroskedasticity and nonlinearity.

For the SVM model, both the MAE and MedAE statistically differ for the log-transformed

dependent variable compared to the original dependent variable on a 5% significance level. As SVM fits its hyperplane by minimizing the errors outside the margin respective to the hyperplane, the mutation of large values in the tails can positively affect the prediction performance of SVM.

Comparable to the Boston dataset in Section 5.2.1 for the RMSE and MedAE no transformed form indicates statistical difference compared to the original form. For the untransformed and square root transformation, the XGB model outperforms LR on a 5% significance level considering the MAE. For the MAE of the RF model, it outperforms LR with the square root and log transformation on a 5% and 1% significance level. Regressors such as provider together with fee percent and living area along with total area in the Russia rent dataset, have strong interactions as the types of providers have different incentives for profit margins on the fee percentage. In addition, the amount of living area is always proportional to the total area of the apartment and can never be greater than the total area, explaining the statistical difference in the errors of the RF and XGB models compared to the LR model. However, because transformations on the dependent variable do not change these relationships between regressors, tree-based algorithms perform significantly better because they can capture these nonlinear interactions from these relationships.

Table 3: Mean of the 5×2 cross-validated models on the Russia dataset for the MAE, RMSE, and MedAE

Russia mean absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	13,978	13,870	14,076	2,852	2,775*
Square root	11,689	11,736	7,311	2,714*	2,487*
Log	11,020††	11,188†	5,837†	2,720**	3,985
Box-Cox	11,214†	11,195†	15,031	2,919	14,696†

Russia root mean squared error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	23,849	23,953	17,996	13,676	13,376
Square root	23,071	23,343	23,595	11,745	11,207
Log	23,849	23,953	17,996	13,676	13,377
Box-Cox	26,073	26,119	34,102	15,868	33,749†

Russia median absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	24,152	26,055	33,354	11,837	11,438
Square root	23,071	23,069	26,341	12,594	10,772
Log	23,874	28,023	17,600†	14,111	12,886
Box-Cox	26,052	29,561	34,102	16,346	33,740†

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. †A dagger indicates a significant difference between the respective (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, ††two daggers indicate significance on a 1%. significance level. *An asterisk indicates a significant difference between LR and the respective model on a 5% significance level, **two asterisks indicate significance on a 1%. significance level.

5.2.3 Housing prices dataset

We present the results of our ML models on the housing prices dataset in Table 4. In contrast to the Boston dataset and Russia rent dataset of Section 5.2.1 and 5.2.2 respectively, transformations do not aid the predictive accuracy adequately to provide sufficient statistical evidence for a difference in performance considering any of the three performance measures on a 5% significance level. This lack of statistical evidence to differentiate model performance indicates transformations do not aid in improving the accuracy of the LR and PLR models for the housing prices dataset.

Comparing the original errors to the log transformation using SVM shows a statistical difference on a 5% significance level for all three performance measures. The error for the log-transformed dependent variable of the housing prices dataset shows a statistical resemblance to a normal distribution according to sample skewness and kurtosis as described in Section 5.2. The symmetrically distributed dependent variable can help SVM properly fit its hyperplane for the data, as large values in the tails can impact model performance.

Considering the results of Section 5.2.1 and 5.2.2, the LR and PLR model perform substantially better than the RF and XGB models for the housing prices dataset. Occasionally for higher errors of the RF and XGB models, there is a statistical difference between LR and the respective models. This statistical difference suggests the presence of linear interactions within the dataset. Considering the regressors, few relationships can be present between the regressors, which is a possible explanation for the adequate performance of LR and PLR. Additionally, the relatively small size of the dataset (546 observations) can influence models such as RF and XGB, which require an adequate amount of data for tuning hyperparameters.

Table 4: Mean of the 5×2 cross-validated models on the Housing prices dataset for the MAE, RMSE, and MedAE

Housing prices mean absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	806,331	798,063	1,400,930*	803,869	996,333
Square root	785,324	775,927	1,096,404*	829,714	989,275
Log	776,418	777,412	786,162††	847,118	931,368*
Box-Cox	781,186	782,338	1,186,465	836,486	1,400,666**

Housing prices root mean squared error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	1,087,766	1,084,324	1,908,550**	1,187,220	1,273,420
Square root	1,076,395	1,069,330	1,602,060*	1,180,254	1,272,056
Log	1,073,415	1,078,097	1,108,022††	1,190,238	1,310,376
Box-Cox	1,082,541	1,087,233	1,700,165	1,191,146	1,909,215**†

Housing prices median absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	1,087,766	1,084,324	1,908,553*	1,134,353	1,395,486
Square root	1,076,395	1,069,341	1,584,597*	1,154,670	1431333*
Log	1,082,483	1,078,098	1,097,494††	1,152,968	1,316,620
Box-Cox	1,093,463	1,087,211	1,700,165	1,200,477	1,910,627**

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. †A dagger indicates a significant difference between the respective (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, ††two daggers indicate significance on a 1%. significance level. *An asterisk indicates a significant difference between LR and the respective model on a 5% significance level, **two asterisks indicate significance on a 1%. significance level.

5.2.4 Advanced house prices dataset

Table 5 shows the results for our five models on the advanced house prices dataset. For all three performance measures, similar to the housing prices dataset of Section 5.2.3, the log transformation results in lower errors and differs significantly on 5% significance level when compared to the untransformed metric using the SVM model. For both the LR and PLR models, the errors decrease considering the MAE, while for the RMSE, and the MedAE, the transformations indicate an adverse effect on performance. This increase in errors due to transformations on the dependent variable for the RMSE and MedAE indicates LR and PLR mispredict larger errors

more often. The transformations reduce larger values relatively more compared to small values, this can cause the larger values to diminish too much, causing a bad model fit as described in Section 2. As a result of the mispredictions, the RMSE and MedAE increase, although there is not enough statistical evidence of unequal model performance.

Similar increase in the error of the XGB model as for the Russian rent and housing prices dataset of Section 5.2.2 and 5.2.3 respectively, using transformations shows differences in performance. Specifically, a statistical difference for the MAE comparing the box-cox and untransformed dependent variable of the XGB model shows the negative effects transformations can have on the XGB model for the advanced house prices dataset, while the RF model shows little influence for using transformations on the dependent variable as there is not enough statistical evidence to differentiate the transformed forms of RF from the untransformed form.

Table 5: Mean of the 5×2 cross-validated models on the advanced house prices dataset for the MAE, RMSE, and MedAE.

Advanced house prices mean absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	20,688	20,077	55,361	16,842	17,063
Square root	18,824	18,453	23,992	16,874	16,611
Log	19,191	18,446	18,453††	17,028	19,009
Box-Cox	19,631	18,690	22,884	17,040	23,759†

Advanced house prices root mean squared error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	37,402	35,442	80,632	29,393	28,945
Square root	38,934	35,672	41,603	30,021	28,142
Log	53,188	42,829	30,260††	30,195	32,866
Box-Cox	60,117	46,046	36,637	30,042	40,073

Advanced house prices median absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	37,402	34,698	80,571	30,143	30,116
Square root	38,934	38,199	41,603	29,837	28,773
Log	53,150	49,929	30,869††	30,939	32,898
Box-Cox	60,093	54,767	37,649	31,166	41,820

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. †A dagger indicates a significant difference between the respective (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, ††two daggers indicate significance on a 1%. significance level. *An asterisk indicates a significant difference between LR and the respective model on a 5% significance level, **two asterisks indicate significance on a 1%. significance level.

5.2.5 Saudi dataset

In Table 5.2.5, we show the results of our models on the Saudi dataset. Comparing the LR and PLR models considering MAE shows enough statistical evidence for increased performance by using a square root transformation on the dependent variable. On the contrary, log and box-cox transformations adversely affect the predictive accuracy of the LR and PLR models where the transformations excessively shrink large values in the dependent variable, similar to the advanced house prices dataset of Section 5.2.4 impacting the predictive accuracy of the models. However, there is insufficient statistical evidence to reject the null hypothesis at a 5% significance level for any comparisons of the log and box-cox transformations on the dependent variable using the LR and PLR models.

For the SVM model, the RMSE of the log and box-cox transformation shows a similar decrease to the LR and PLR model, although, using the Dunn test, we do not reject the null hypothesis at a 5% significance level. However, considering the MAE for all transformations, the error improves compared to the untransformed form of the SVM model, although we lack statistical evidence to confirm these differences. This inclusive evidence considering the MAE and RMSE indicates the kernel can not fully capture the relationships within the data.

Comparing all transformations using the RF and XGB models considering the MAE shows more accurate predictions compared to the LR model and is not affected by transformations in the same ways that LR and PLR are. For the LR and PLR models, the transformations likely impose some form of nonlinearity in the data, which the RF and XGB models much better account for. Nonetheless, only the errors of the RF model show enough statistical difference compared to LR. Additionally, we do not reject the null hypotheses comparing the LR model to

the RF and XGB models for any transformation by the RMSE and MedAE.

Table 6: Mean of the 5×2 cross-validated models on the Saudi dataset for the MAE, RMSE, and MedAE.

Saudi mean absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	569,533	494,438	716,461	398,168*	416,125
Square root	397,129†	397,862†	590,550	387,639	391,203
Log	2,762,981	2,721,351	556,000	385,055	408,487
Box-Cox	3,481,345	3,392,982	684,246	386,813	411,339

Saudi root mean squared error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	1,166,962	1,166,780	1,917,321	1,254,933	1,267,401
Square root	848,121	846,375	1,786,940	1,271,980	1,266,051
Log	40,424,019	34,324,718	3,676,007	1,309,123	1,278,626
Box-Cox	52,483,112	43,954,354	5,540,968	1,304,281	1,287,609

Saudi median absolute error					
Transformation	**LR**	**PLR**	**SVM**	**RF**	**XGB**
None	1,166,930	1,235,722	1,917,313	1,269,109	1,290,515
Square root	847,901	855,773	1,797,997	1,262,788	1,298,145
Log	4,0149,779	20,375,200	1,999,634	1,287,772	1,559,356
Box-Cox	52,119,316	24,403,280	1,724,667	1,296,320	1,318,775

Notes: We perform the test on the dependent variable of all five datasets for the untransformed, square root, Log, and Box-Cox transformation of the dependent variable. †A dagger indicates a significant difference between the respective (un)transformed forms of the dependent variable for a distinct dataset on a 5% significance level, ††two daggers indicate significance on a 1%. significance level. *An asterisk indicates a significant difference between LR and the respective model on a 5% significance level, **two asterisks indicate significance on a 1%. significance level.

6 Discussion & conclusion

In this research, we assess the effects of transformations on the dependent variable, specifically property prices, of real estate data by evaluating the predictive accuracy of machine learning (ML) models. We evaluate the performance of our ML models on the Boston, Russia rent, housing prices, advanced house prices, and Saudi dataset using the mean absolute error (MAE), root mean squared error (RMSE), and median absolute error (MedAE) metric. Results show

that the predictive accuracy of penalized linear regression (PLR) and support vector machine (SVM) significantly increases by using transformations on the dependent variable. On the contrary, transformations on the dependent variable for random forest (RF) and eXtreme Gradient Boosting (XGB) occasionally show a significant decrease in performance compared to the errors of the original data.

Even though the Dunn test for comparison of equal model performance shows inconclusive results, the use of transformations on the dependent variable occasionally positively influences the performance of PLR and SVM. Using the log and box-cox transformation on the dependent variable using the PLR and SVM models significantly outperforms the original data of the Boston and Russia rent dataset. Because PLR shares a similar model construction to linear regression (LR), we expect PLR to show similar improvement when using transformations on the dependent variable as the cost function of PLR shares a similar construction to LR. The original data of the SVM model occasionally performs worse compared to the LR model, showing a statistical difference in predictive accuracy. As discussed in Section 2, the wrong kernel and categorical data in the datasets can explain bad model performance. Although, transformations on the dependent variable significantly improve the model performance of the SVM model, specifically for the Boston, Russia rent, housing prices, and advanced house prices using either a log or box-cox transformation on the dependent variable. However, D'Agostino's K-squared test indicates normality according to sample skewness and kurtosis using the box-cox transformation for the Boston and housing prices dataset. This indication of a normally distributed distribution did not necessarily translate into a significant improvement in our models.

The RF and XGB models show diminishing performance for transformations of the dependent variable considering our tree-based models. Specifically for Russian rent, housing prices, and advanced house prices dataset, the box-cox transformation enlarges the error considering the MAE to the extent of statistical difference with the untransformed dependent variable. This diminishing performance suggests adverse effects on tree-based algorithms for transformations. As the tree-based algorithms construct splits by minimizing their criterion, the transformation can undermine proper splits, impacting model performance. Comparing models by their MAE, the RF model performs most noteworthy overall, specifically for the Boston and Russia rent dataset. For the XGB model, a model known for its high predictive accuracy, no clear statistical evidence suggests higher performance than the LR model. However, all datasets in this research are small, which is a liability for the model as XGB requires tuning many hyperparameters.

Noteworthy for all datasets, only the MAE metric provides consistent statistical evidence of unequal model performance. Despite the absence of statistical proof, the errors of the RMSE

and MedAE differ substantially, comparing the untransformed data to the log and box-cox transformation using the LR, PLR, and SVM models of the advanced housing prices and Saudi dataset. Because the Dunn test ranks the whole group of performance measures we compare, the test is more conservative in rejecting the null hypothesis. Additionally, as the test only ranks errors within the set and does not consider numerical differences between errors, the Dunn test fails to perceive these notable differences in performance between models, which is a disadvantage of using non-parametric tests. Also, for feature engineering, this research limits itself to a min-max transformation and manual removal of outliers. As the PLR model and the regression model of SVM are susceptible to outliers, if outliers are still present in the data, this could influence results as transformations scale outliers more severely. Additionally, the data sets of this research are relatively small, with less than 1500 data points. Specifically, the XGB model benefits from large datasets for a better decision of splits and feature importance.

Thus transformations on the dependent variable of the SVM and PLR models can occasionally positively affect predictive accuracy, offering a simple and computationally inexpensive solution for positively skewed data. However, carefully consider these transformations because they are only occasionally advantageous. The Saudi dataset shows that transformations can negatively influence model performance substantially. For tree-based algorithms, transformations show adverse effects on predictive accuracy. Although tree-based algorithms benefit from larger datasets, possibly affecting model performance.

For future research, we advise looking into the effects of using transformations with large datasets, exploring different kinds of transformations and their effects, and performing a controlled experiment using generated data for the PLR and SVM models, as these models show the most promising benefits from transformations on the dependent variable. Large datasets can cause different behavior for our models as they are more sensitive to outliers for smaller datasets, and increased sample size, particularly for the PLR model, helps the estimation of coefficients. The controlled experiment can provide more conclusive evidence for using transformations on the dependent variable using the PLR and SVM model as factors impacting model performance, such as noise and outliers, are known in such an experiment. Additionally, a separate analysis with solely numerical variables and varying kernels for the effects of transformations on the SVM model can result in a better exhibition of the performance of SVM for positively skewed data.

References

Abdulhafedh, A. (2022). Incorporating multiple linear regression in predicting the house prices using a big real estate dataset with 80 independent variables. *Open Access Library Journal*, 9(1):1–21.

Adelino, M., Schoar, A., and Severino, F. (2018). The role of housing and mortgage markets in the financial crisis. *Annual Review of Financial Economics*, 10:25–41.

Anscombe, F. J. and Glynn, W. J. (1983). Distribution of the kurtosis statistic b 2 for normal samples. *Biometrika*, 70(1):227–234.

Baldominos, A., Blanco, I., Moreno, A. J., Iturrarte, R., Bernárdez, Ó., and Afonso, C. (2018). Identifying real estate opportunities using machine learning. *Applied sciences*, 8(11):2321.

Ben-David, I. (2011). Financial constraints and inflated home prices during the real estate boom. *American Economic Journal: Applied Economics*, 3(3):55–87.

Benoit, K. (2011). Linear regression models with logarithmic transformations. *London School of Economics, London*, 22(1):23–36.

Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967.

Bond, S. A. and Patel, K. (2003). The conditional distribution of real estate returns: Are higher moments time varying? *The Journal of Real Estate Finance and Economics*, 26:319–339.

Borde, S., Rane, A., Shende, G., and Shetty, S. (2017). Real estate investment advising using machine learning. *International Research Journal of Engineering and Technology*, 4(3):1821–1825.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Box, G. E. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243.

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

Castelli, M., Dobreva, M., Henriques, R., and Vanneschi, L. (2020). Predicting days on market to optimize real estate sales strategy. *Complexity*, 2020:1–22.

Centraal Bureau voor de Statistiek (2021). 8 miljoen woningen in nederland. https://www.cbs.nl/nl-nl/nieuws/2021/31/8-miljoen-woningen-in-nederland.

Centraal Bureau voor de Statistiek (2022). Woningmarkt. https://www.cbs.nl/nl-nl/visualisaties/dashboard-economie/woningmarkt#:~:text=De%20gemiddelde%20verkoopprijs%20van%20een,Pekela%20met%20231%20duizend%20euro.

Changyong, F., Hongyue, W., Naiji, L., Tian, C., Hua, H., Ying, L., et al. (2014). Log-transformation and its implications for data analysis. *Shanghai archives of psychiatry*, 26(2):105.

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*, pages 785–794.

Curran, P. J., West, S. G., and Finch, J. F. (1996). The robustness of test statistics to nonnormality and specification error in confirmatory factor analysis. *Psychological methods*, 1(1):16.

D'agostino, R. and Pearson, E. S. (1973). Tests for departure from normality. empirical results for the distributions of b 2 and b. *Biometrika*, 60(3):613–622.

Diaz-Serrano, L. (2005). Labor income uncertainty, skewness and homeownership: A panel data study for germany and spain. *Journal of Urban Economics*, 58(1):156–176.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Din, A., Hoesli, M., and Bender, A. (2001). Environmental variables and real estate prices. *Urban studies*, 38(11):1989–2000.

Dunn, O. J. (1964). Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252. https://doi.org/10.2307/1266041.

Fabozzi, F. J., Kynigakis, I., Panopoulou, E., and Tunaru, R. S. (2020). Detecting bubbles in the us and uk real estate markets. *The Journal of Real Estate Finance and Economics*, 60:469–513.

Ghosalkar, N. N. and Dhage, S. N. (2018). Real estate value prediction using linear regression. In *2018 fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–5. IEEE.

Gnana, D. A. A., Balamurugan, S. A. A., and Leavline, E. J. (2016). Literature review on feature selection methods for high-dimensional data. *International Journal of Computer Applications*, 136(1):9–17.

Guliker, E., Folmer, E., and van Sinderen, M. (2022). Spatial determinants of real estate appraisals in the netherlands: A machine learning approach. *ISPRS international journal of geo-information*, 11(2):125.

Heij, C., Heij, C., de Boer, P., Franses, P. H., Kloek, T., van Dijk, H. K., et al. (2004). *Econometric methods with applications in business and economics*. Oxford University Press.

Jamil, S., Mohd, T., Masrom, S., and Ab Rahim, N. (2020). Machine learning price prediction on green building prices. In *2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, pages 1–6. IEEE.

Kahn, J. A. (2008). What drives housing prices? *FRB of New York Staff Report*.

Kapoor, S. and Perrone, V. (2021). A simple and fast baseline for tuning large xgboost models. *arXiv preprint arXiv:2111.06924*.

Kiely, P., Bland, J., Joseph, A., Mortimer, P., and Bourke, B. (1995). Upper limb lymphatic function in inflammatory arthritis. *The Journal of Rheumatology*, 22(2):214–217.

Kim, S. (1992). Search, hedonic prices and housing demand. *The review of economics and statistics*, pages 503–508.

Kok, N., Koponen, E.-L., and Martínez-Barbosa, C. A. (2017). Big data in real estate? from manual appraisal to automated valuation. *The Journal of Portfolio Management*, 43(6):202–211.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

Li, D.-Y., Xu, W., Zhao, H., and Chen, R.-Q. (2009). A svr based forecasting approach for real estate price prediction. In *2009 International conference on machine learning and cybernetics*, volume 2, pages 970–974. IEEE.

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.-S., and Zeineddine, H. (2019). An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, 7:93010–93022.

Meyer, D., Leisch, F., and Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1-2):169–186.

Ministerie van Financiën (2021). Ministerie van financiën. https://www.rijksfinancien.nl/miljoenennota/2021/650620#:~:text=Zelfs%20in%202009%2C%20toen%20Nederland,de%20uitvoer%20met%205%20procent.

Motulsky, H. J. and Ransnas, L. A. (1987). Fitting curves to data using nonlinear regression: a practical and nonmathematical review. *The FASEB journal*, 1(5):365–374.

Nielsen, D. (2016). Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2433761/16128_FULLTEXT.pdf.

Northcraft, G. B. and Neale, M. A. (1987). Experts, amateurs, and real estate: An anchoring-and-adjustment perspective on property pricing decisions. *Organizational behavior and human decision processes*, 39(1):84–97.

Ogutu, J. O., Schulz-Streeck, T., and Piepho, H.-P. (2012). Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In *BMC proceedings*, volume 6, pages 1–6. Springer.

Osborne, J. (2010). Improving your data transformations: Applying the box-cox transformation. *Practical Assessment, Research, and Evaluation*, 15(1):12.

Ozgur, C., Hughes, Z., Rogers, G., and Parveen, S. (2016). Multiple linear regression applications in real estate pricing. *International Journal of Mathematics and Statistics Invention (IJMSI)*, 4(8).

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.

Probst, P., Boulesteix, A.-L., and Bischl, B. (2019a). Tunability: Importance of hyperparameters of machine learning algorithms. *The Journal of Machine Learning Research*, 20(1):1934–1965.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019b). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9(3):e1301.

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.

Sain, S. R. (1996). The nature of statistical learning theory. *Technometrics*, 38(4):409–409.

Sakia, R. M. (1992). The box-cox transformation technique: a review. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(2):169–178.

Silva, J. S. and Tenreyro, S. (2006). The log of gravity. *The Review of Economics and statistics*, 88(4):641–658.

Smith, V. K. and Huang, J. C. (1993). Hedonic models and air pollution: twenty-five years and counting. *Environmental and Resource Economics*, 3:381–394.

White, H. and Domowitz, I. (1984). Nonlinear regression with dependent observations. *Econometrica: Journal of the Econometric Society*, pages 143–161.

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40.

Yazdani, M. (2021). Machine learning, deep learning, and hedonic methods for real estate price prediction. *arXiv preprint arXiv:2110.07151*.

Yu, H. and Wu, J. (2016). Real estate price prediction with regression and classification. *CS229 (Machine Learning) Final Project Reports*.

Zhao, Y., Chetty, G., and Tran, D. (2019). Deep learning with xgboost for real estate appraisal. In *2019 IEEE symposium series on computational intelligence (SSCI)*, pages 1396–1401. IEEE.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

Appendices

A Dependent variables

A.1 Included variables

Table 7: Variables of the Boston dataset

Index	Name	Description
1	CRIM	Crime rate per resident.
2	ZN	Land proportionally zoned for residential land for areas above 25,000 square feet.
3	INDUS	Proportion of the total business land, which is non-retail per acre.
4	CHAS	Dummy if the Charles River touches the regional area.
5	NOX	Prts per ten concentration of nitric oxides.
6	RM	Average amount of rooms per residency.
7	AGE	Proportion of the residencies built before 1940 within the suburb.
8	DIS	Weighted distance of the five most approximate centers of employment in Boston.
9	RAD	Index for the accessibility for radial highways.
10	TAX	Tax rate of the property per $10,000.
11	PTRATIO	The ratio of pupil-teacher.
12	B	proportion of African Americans in the suburb.
13	LSTAT	Proportion of the population classified as lower status. Regarding workers with no high school education and the proportion of laborers.
14	MEDV	Median value of the homes within the suburb in $1000.

Table 8: Variables of the Russia rent dataset

Index	Name	Description
1	metro	Name of the nearest metro station.
2	price	The price for the monthly rent of the apartment.
3	A way	Method to indicate how to reach the nearest metro station.
4	views	Number of visits the apartment got.
5	provider	Variable to indicate if a person or agency rents the apartment.
6	fee_percent	Percent of the fee the person or agency acquires for the rent.
7	storey	The story the apartment is located at.
8	minutes	The number of minutes a walk takes for the nearest metro station.
9	storeys	The total amount of stories.
10	living_area	Living area of the apartment in square foot.
11	kitchen_area	The total footage of the kitchen.
12	total_area	Total footage of the apartment.

Table 9: Variables of the Housing prices dataset

Index	Name	Description
1	price	Monetary value of the house.
2	area	Area of the house in square foot.
3	bedrooms	Bedrooms in the house.
4	bathrooms	Bathrooms in the house.
5	stories	Stories of the house.
6	mainroad	yes/no variable to indicate if the residency is connected to the main road.
7	guestroom	True/false variable indicates whether the house contains a guest room.
8	basement	True/false variable indicates whether the house contains a basement.
9	hotwaterheating	True/false indicates whether the house has a hot water heater.
11	airconditioning	True/false variable to indicate whether the house has air conditioning.
12	parking	variable to indicate the number of parking spots designated to the house.
13	prefarea	True/false variable indicates whether the location lies within a preferred area. Regarding workers with no high school education and the proportion of laborers.
14	furnishingstatus	Variable to indicate the furnishing status in the residency.

Table 10: Included variables of the advanced house prices dataset

Index	Name	Description
1	SalePrice	The sale price of the property in dollars.
2	MSSubClass	The class of the building.
3	MSZoning	The zoning classification of the house.
4	LotFrontage	Feet connected in feet to the property.
5	LotArea	The size of the lot in square feet.
6	Street	Road type access.
8	LotShape	Shape of the property.
9	LandContour	Flatness of the property.
10	Utilities	Available type of utilities.
13	Neighborhood	Amount of physical locations that are within the limits of Ames city.
16	BldgType	Type of the dwelling.
17	HouseStyle	Style of the dwelling.
18	OverallQual	Quality of the finish for the material overall in the building.
19	OverallCond	The condition rating of the house.
20	YearBuilt	Construction date of the house.
21	YearRemodAdd	Remodel date.
26	MasVnrType	Type of masonry veneer.
27	MasVnrArea	Area of masonry veneer in square feet.
28	ExterQual	Material quality of the exterior.
29	ExterCond	Current condition of the exteriors material.
32	BsmtCond	Current condition of the basement.
33	BsmtExposure	Walls are garden or walkout level.
34	BsmtFinType1	Finishing quality of basement.
35	BsmtFinSF1	Square footage of finishing area.
38	BsmtUnfSF	Square footage of the unfinished area.
39	TotalBsmtSF	Square footage of basement.
42	CentralAir	Indicated of central air conditioning.
44	1stFlrSF	Square footage of the first floor.
45	2ndFlrSF	Square footage of the second floor.
46	LowQualFinSF	Square footage finishing area of low quality.
47	GrLivArea	Above-ground square footage of the living area.
48	BsmtFullBath	Bathrooms in the basement.
49	BsmtHalfBath	Half bathrooms in the basement.
50	FullBath	Bathrooms above ground.
51	HalfBath	Half bathrooms above ground.
52	BedroomAbvGr	Total sum of the bedroom area.
53	KitchenAbvGr	Total sum of kitchens area.
54	KitchenQual	Quality of the kitchen.
55	TotRmsAbvGrd	Total number of rooms excluding the bathroom above ground.
57	Fireplaces	Total sum of fireplaces.
62	GarageCars	Total number of cars that fit in the garage.
63	GarageArea	Size of the garage in square feet.
66	PavedDrive	Binary variable to indicate if a paved driveway is present.

Table 11: Included variables of the advanced house prices dataset

Index	Name	Description
67	WoodDeckSF	Square footage of wood deck.
68	OpenPorchSF	Square footage of open porch.
69	EnclosedPorch	Square footage of enclosed porch.
70	3SsnPorch	Square footage of three season porch area.
71	ScreenPorch	Square footage of screen porch.
72	PoolArea	Square footage of pool.
76	MiscVal	Value of the MiscFeature variables.
77	MoSold	Month the property is sold in.
78	YrSold	Year the property is sold in.
80	SaleCondition	The integrity of the sale.

Table 12: Variables of the Saudi dataset

Index	Name	Description
1	Villa price	Price of the villa.
2	neighborhood_name	Name of the neighborhood the villa is located.
3	administritive_area	Area of administration for the villa.
4	city	Name of the city where the villa is located.
5	rooms	The number of bedrooms within the villa.
6	bathrooms	The number of bathrooms within the villa.
7	sqm	The size of the villa in square meters.
8	elevator	Binary variable to indicate whether the villa contains an elevator.
9	bool	Binary variable to indicate whether the villa contains a pool.
10	driver	Binary variable to indicate whether the villa contains a driver room.
11	garden	Binary variable to indicate whether the villa contains a garden.

A.2 Excluded variables

Table 13: Excluded variables of the advanced house prices dataset

Index	Name	Description
7	Alley	Alley type access.
11	LotConfig	Lot configuration.
12	LandSlope	The slope of the property.
14	Condition1	Closeness to the main road or railroad.
15	Condition2	Closeness to the main road or railroad if a second is present.
22	RoofStyle	Type of the roof.
23	RoofMatl	Material of the roof.
24	Exterior1st	Covering on the exterior of the house.
25	Exterior2nd	Covering on the house's exterior if there is more than one material.
30	Foundation	Foundation type.
31	BsmtQual	Basement height.
36	BsmtFinType2	Second finishing quality of basement if present.
37	BsmtFinSF2	Square footage of second finishing area if present.
40	Heating	Heating type.
41	HeatingQC	Quality and condition of heating.
43	Electrical	Type of electrical system.
56	Functional	the rating of functionality for the home.
58	FireplaceQu	Quality of the fireplace.
59	GarageType	The location of the garage.
60	GarageYrBlt	Building year of the garage.
61	GarageFinish	Type of interior finishing in the garage.
64	GarageQual	Quality of the garages finishing.
65	GarageCond	Condition the garage is in.
73	PoolQC	Quality of the pool.
74	Fence	State of the fence.
75	MiscFeature	Features not explicitly represented in the dataset.
79	SaleType	Kind of sale.

B Descriptive statistics

Table 14: Descriptive statistics of the Boston dataset

Index	Name	minimum	maximum	mean	median	stddev
1	CRIM	0.00632	88.9762	3.584139	0.26169	8.564433
2	ZN	0	100	11.25245	0	23.23484
3	INDUS	0.46	27.74	11.1511	9.69	6.828175
4	CHAS	0	1	0.068493	0	0.252838
5	NOX	0.385	0.871	0.554757	0.538	0.11531
6	RM	3.561	8.78	6.287589	6.209	0.703802
7	AGE	2.9	100	68.61624	77.3	28.09913
8	DIS	1.1296	12.1265	3.783876	3.1523	2.098631
9	RAD	1	24	9.485323	5	8.688469
10	TAX	187	711	407.4403	330	167.9035
11	PTRATIO	12.6	23	18.5	19.1	2.200348
12	B	0.32	396.9	356.6009	391.34	90.88268
13	LSTAT	1.73	76	12.87955	11.45	7.797416
14	MEDV	5	67	22.68219	21.2	9.484262

Table 15: Descriptive statistics of the Russia rent dataset

Index	Name	minimum	maximum	mean	median	stddev
2	price	14000	500000	43759.51	38000	33240.91
4	views	4	5174	418.1958	103	936.7971
6	fee_percent	0	100	37.97578	50	26.8841
7	storey	1	74	6.670588	6	4.289651
8	minutes	0	47	8.752941	7	4.712275
9	storeys	1	95	13.41453	12	6.319007
10	living_area	6	37	20.58754	20	5.610522
11	kitchen_area	3	37	11.37093	10	8.086495
12	total_area	1	57	37.26367	37	6.145091

Table 16: Descriptive statistics of the Housing prices dataset

Index	Name	minimum	maximum	mean	median	stddev
1	price	1750000	13300000	4766729	4340000	1870440
2	area	1650	16200	5150.541	4600	2170.141
3	bedrooms	1	6	2.965138	3	0.738064
4	bathrooms	1	4	1.286239	1	0.50247
5	stories	1	4	1.805505	2	0.867492
12	parking	0	3	0.693578	0	0.861586

Table 17: Descriptive statistics of the advanced house prices dataset

Index	Name	minimum	maximum	mean	median	stddev
1	SalePrice	34900	755000	181166.7	163500	79589.23
2	MSSubClass	20	190	56.91937	50	42.35913
4	LotFrontage	0	313	57.64369	63	34.73089
5	LotArea	1300	215245	10526.12	9477	10007.81
18	OverallQual	1	10	6.102688	6	1.382576
19	OverallCond	1	9	5.573398	5	1.10701
20	YearBuilt	1872	2010	1971.361	1973	30.18573
21	YearRemodAdd	1950	2010	1984.875	1994	20.64373
26	MasVnrType	0	1600	103.0159	0	180.8379
35	BsmtFinSF1	0	5644	444.8746	384	456.4259
38	BsmtUnfSF	0	2336	567.3301	476	441.8624
39	TotalBsmtSF	0	6110	1058.178	992	439.663
44	1stFlrSF	334	4692	1163.055	1088	387.2214
45	2ndFlrSF	0	2065	347.6885	0	436.983
46	LowQualFinSF	0	572	5.880772	0	48.77156
47	GrLivArea	334	5642	1516.624	1466	526.4613
48	BsmtFullBath	0	3	0.425913	0	0.519141
49	BsmtHalfBath	0	2	0.057202	0	0.238172
50	FullBath	0	3	1.566506	2	0.551069
51	HalfBath	0	2	0.384562	0	0.503378
52	BedroomAbvGr	0	8	2.866988	3	0.81762
53	KitchenAbvGr	0	3	1.046864	1	0.22099
55	TotRmsAbvGrd	2	14	6.518952	6	1.629518
57	Fireplaces	0	3	0.61337	1	0.645462
62	GarageCars	0	4	1.769125	2	0.746327
63	GarageArea	0	1418	473.4328	480	213.6402
67	WoodDeckSF	0	857	94.23639	0	125.4917
68	OpenPorchSF	0	547	46.72777	25	66.35314
69	EnclosedPorch	0	552	21.89593	0	61.191
70	3SsnPorch	0	508	3.430737	0	29.40694
71	ScreenPorch	0	480	14.9111	0	55.55823
72	PoolArea	0	738	2.776017	0	40.30121
76	MiscVal	0	15500	43.44866	0	497.5353
77	MoSold	1	12	6.329428	6	2.702302
78	YrSold	2006	2010	2007.813	2008	1.327789

Table 18: Descriptive statistics of the Saudi dataset

Index	Name	minimum	maximum	mean	median	stddev
1	Villa price	2850	57000000	1770375	1300000	2028901
5	rooms	1	7	4.708009	5	1.199202
6	bathrooms	1	7	5.14387	5	1.344227
7	sqm	22	5450	367.905	312	233.0112
8	elevator	0	1	0.210489	0	0.407801
9	bool	0	1	0.128987	0	0.335304
10	driver	0	1	0.145996	0	0.353227
11	garden	0	1	0.034018	0	0.181341

C Hyperparameters

Table 19: Tuning hyperparameters for PLR

Hyperparameter	Search space	Description
C	[log(0.00001),log(50)]	Penalization parameter on features
l1_ratio	[0, 1]	Weight term for use of l1 and l2 term

Table 20: Tuning hyperparameters for SVM

Hyperparameter	Search space	Description
C	[log(0.001), log(10)]	Regularization parameter
gamma	[0,1]	Gamma is the fitting parameter of SVM. Overall higher values of gamma cause SVM to overfit on individual data points.

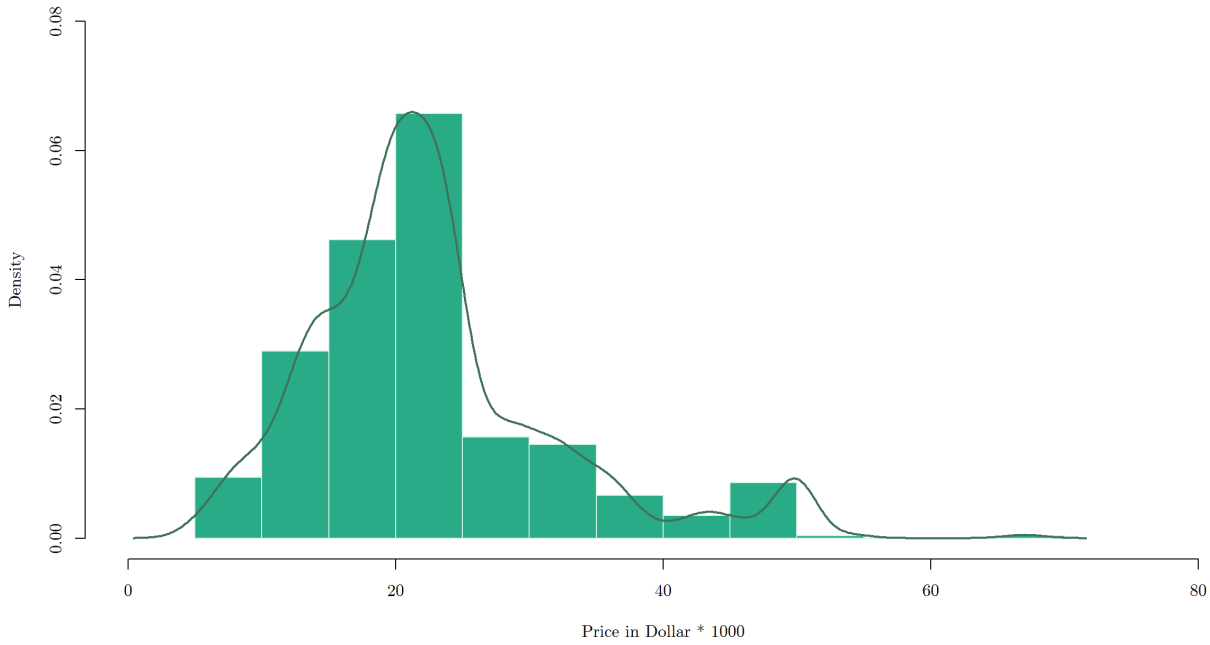
Table 21: Tuning hyperparameters for RF

Hyperparameter	Search space	Description
criterion	[squared error, absolute error, Friedman mse, poisson]	Impurity criterion for splits
max_depth	[2, 10]	Max depth of decision tree
max_features	[2, 20]	Maximum number of features to consider
n_estimators	[50,500]	Number of trees in the forest

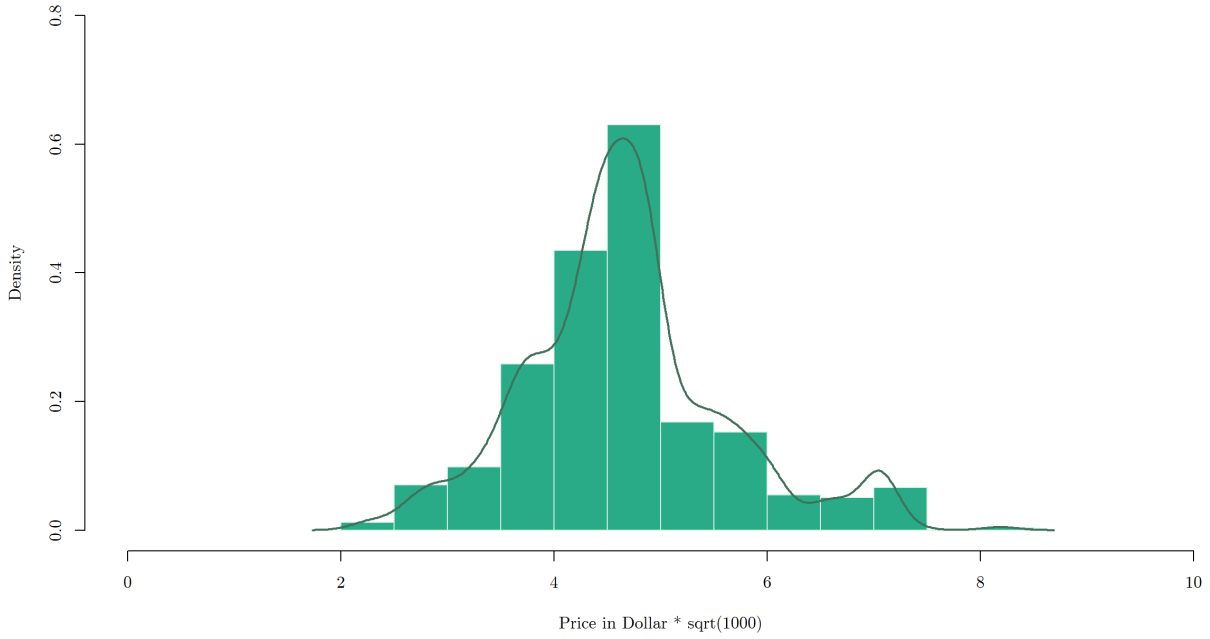
Table 22: Tuning hyperparameters for XGB

Hyperparameter	Search space	Description
gamma	[0.5 , 6]	Penalty term which penalizes XGBoost on excessive greedy splits
learning_rate	[0.2, 1]	Parameter shrink the influence of the sequentially built tree, which trains on the misclassification error of the previous tree. Lower values make XGBoost more conservative.
max_depth	[2, 11]	Max depth of decision tree
n_estimators	[30, 250]	Number of sequentially build trees
subsample	0.8	Sample available for an individual tree to train on. Lower values prevent individual trees to overfit on the data.

D Distributions

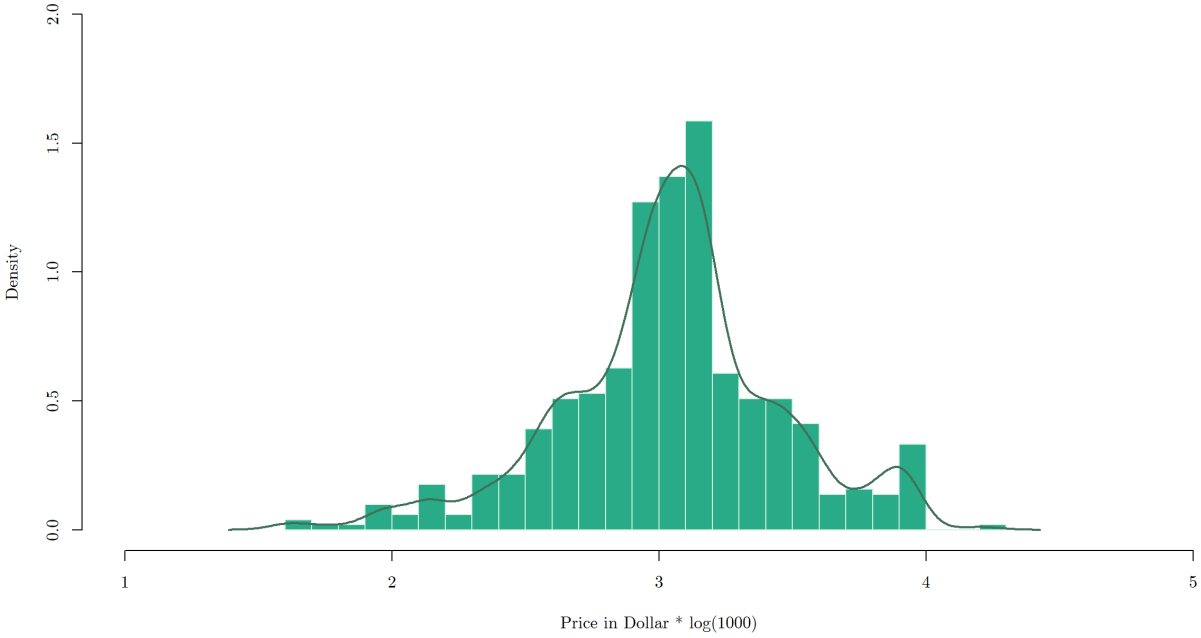


(a) No transformation

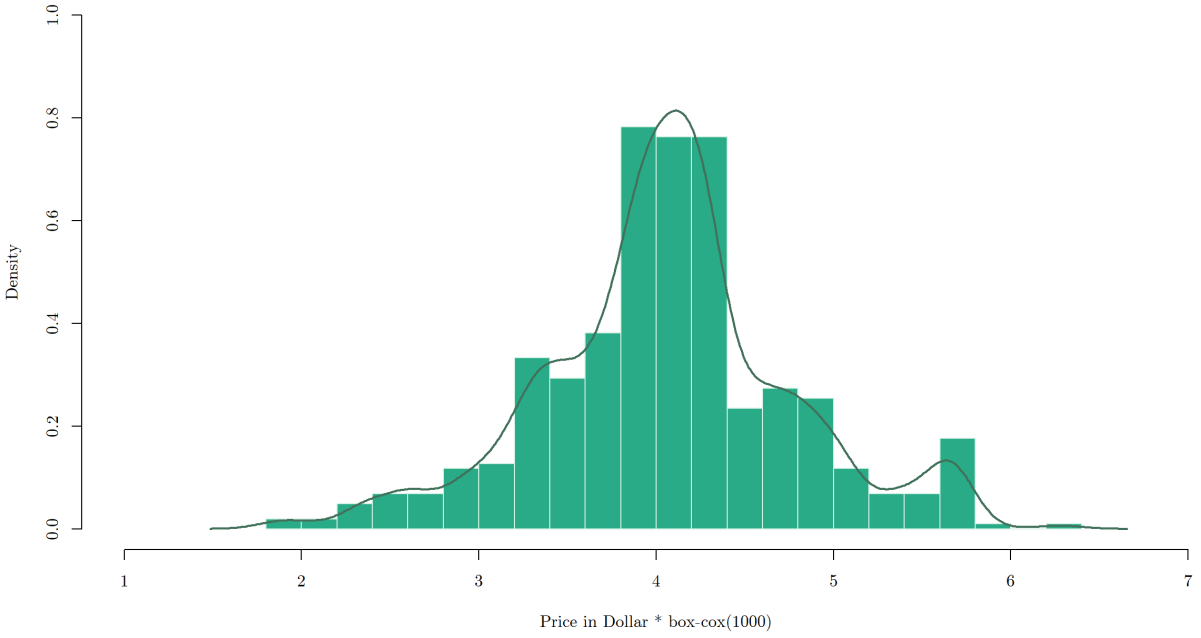


(b) Square root transformation

Figure 7: Density plots of all transformed forms of the Boston dataset

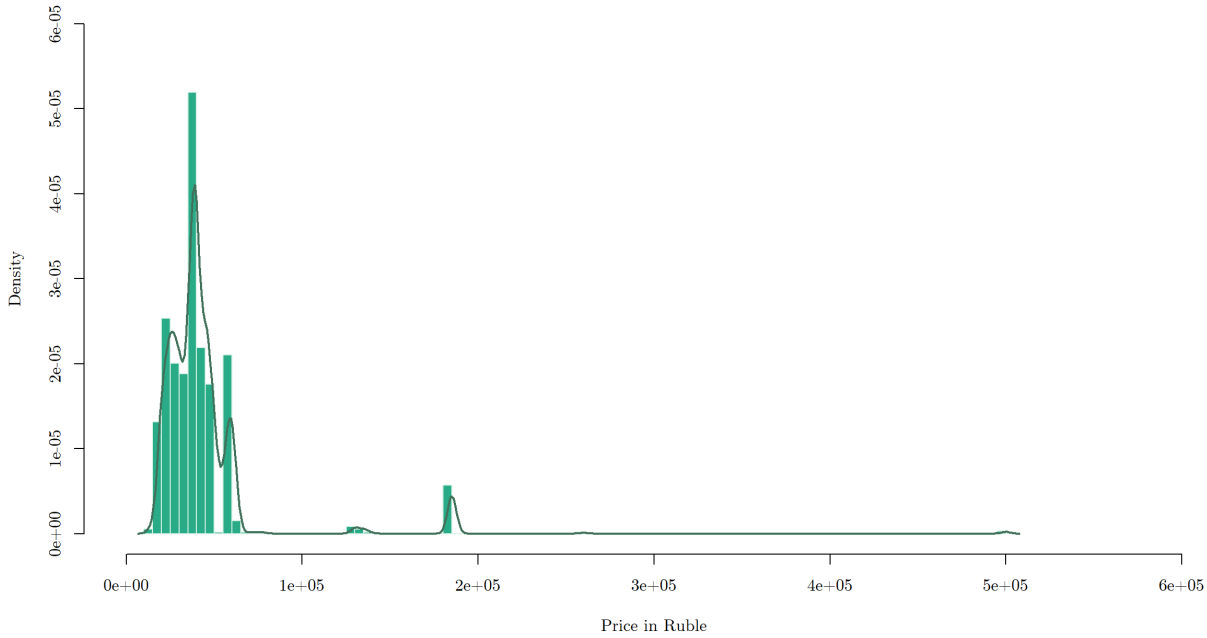


(c) Log transformation

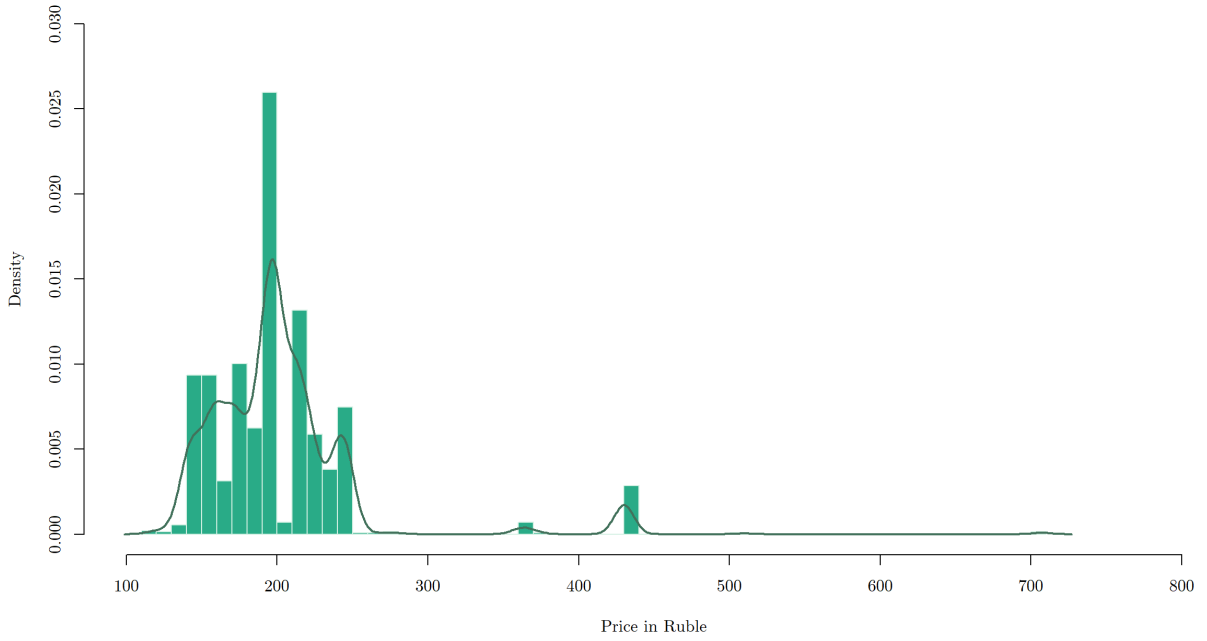


(d) Box-cox transformation

Figure 7: Density plots of all transformed forms of the Boston dataset

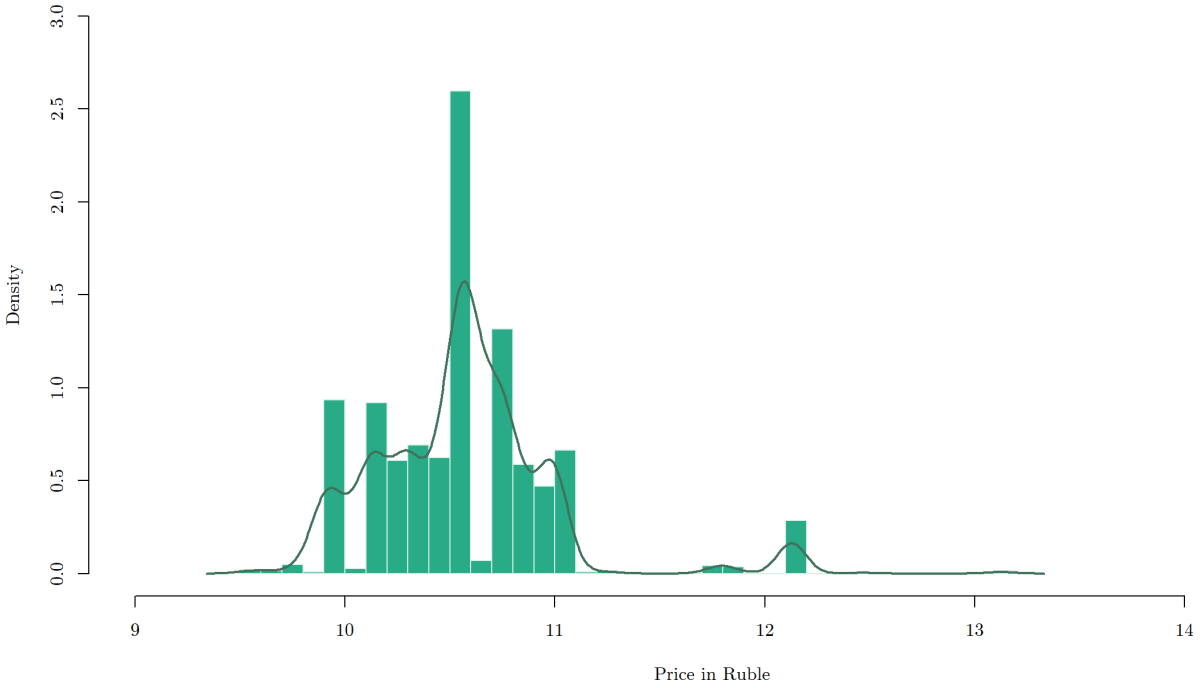


(a) No transformation

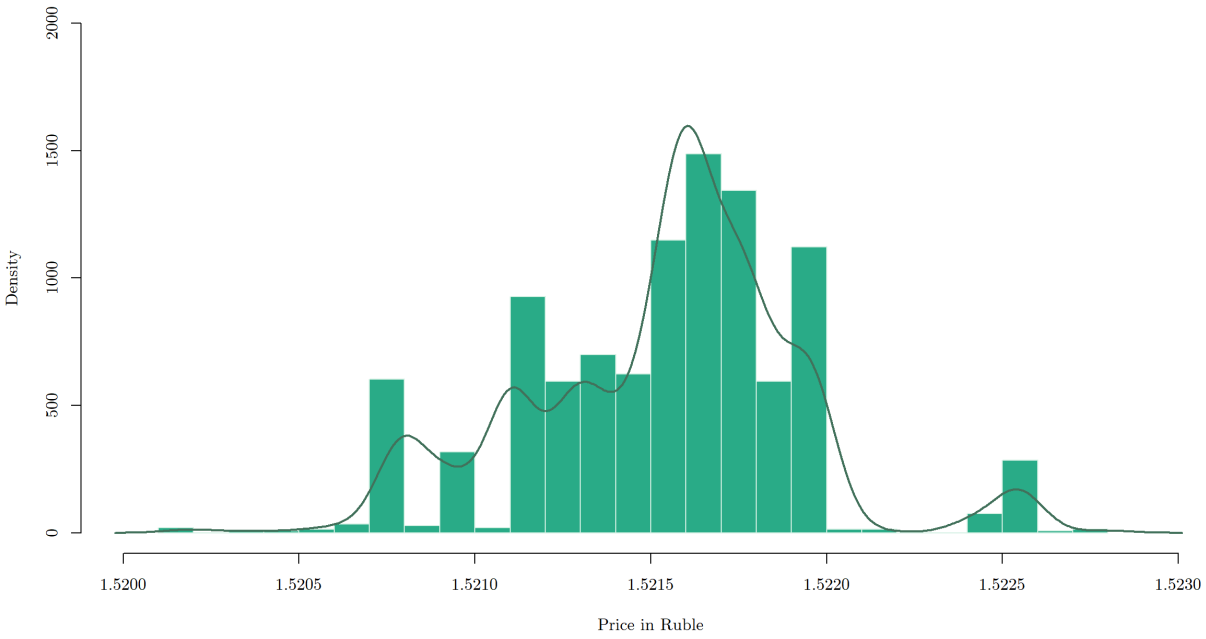


(b) Square root transformation

Figure 8: Density plots of all transformed forms of the Russian rent dataset

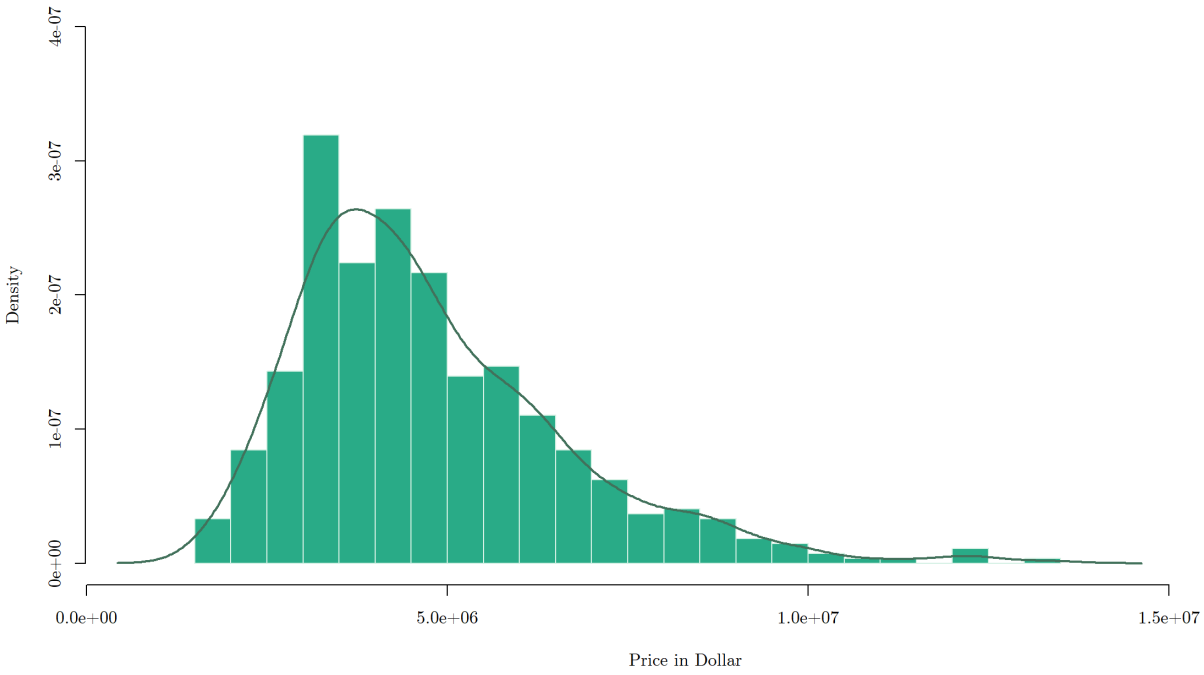


(c) Log transformation

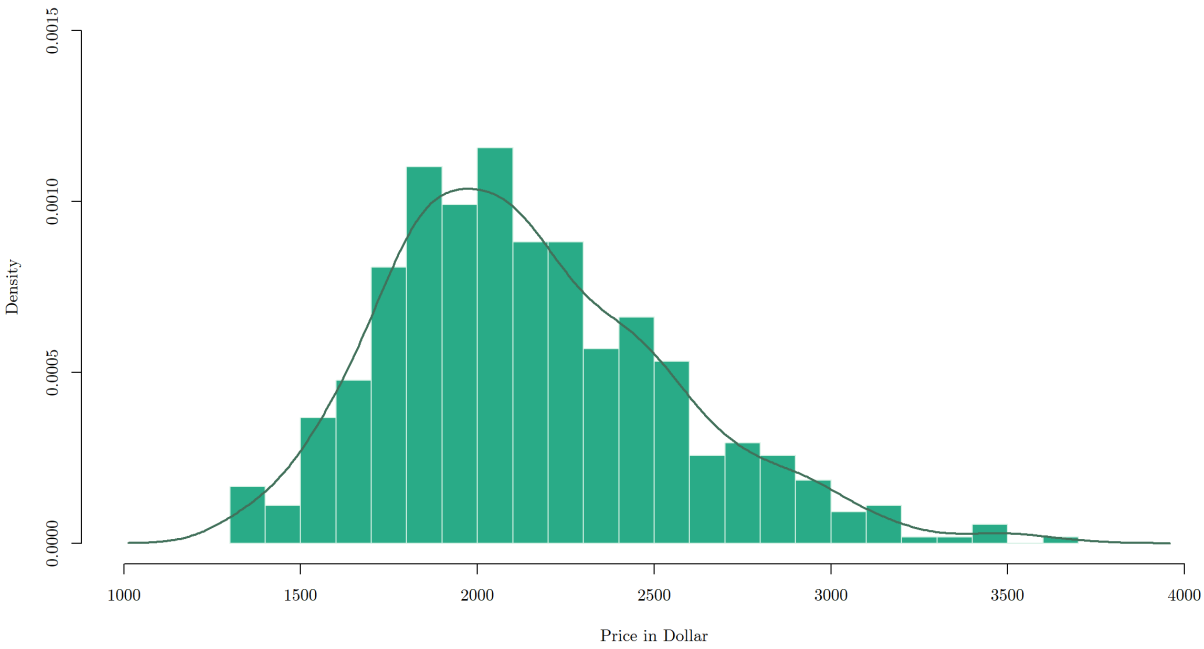


(d) Box-cox transformation

Figure 8: Density plots of all transformed forms of the Russian rent dataset

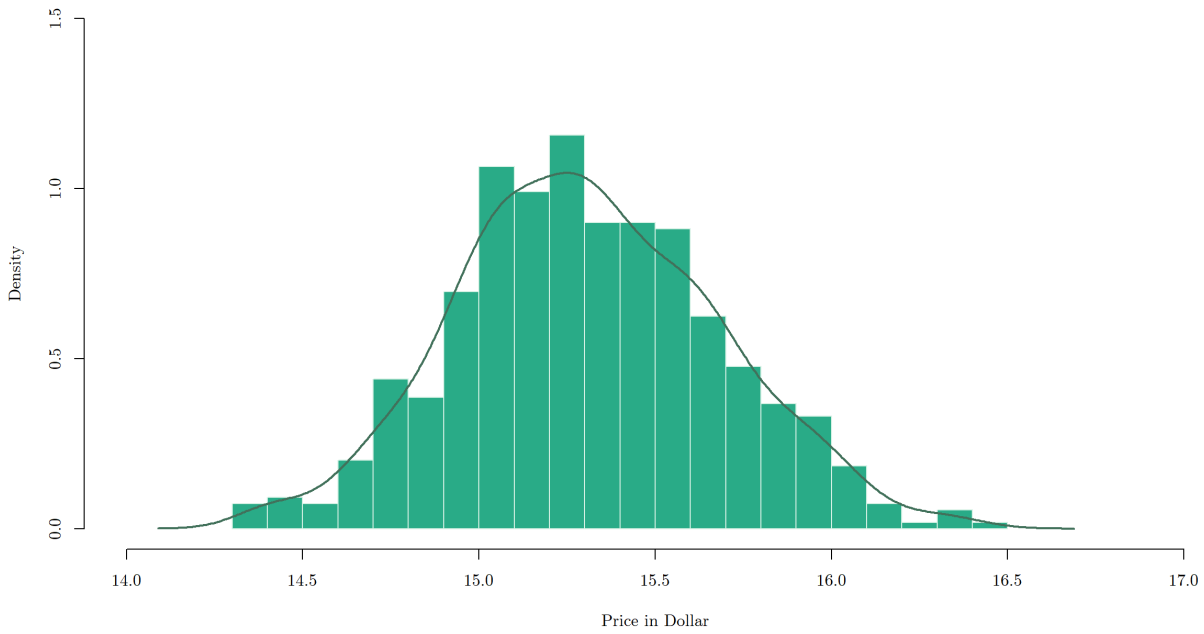


(a) No transformation

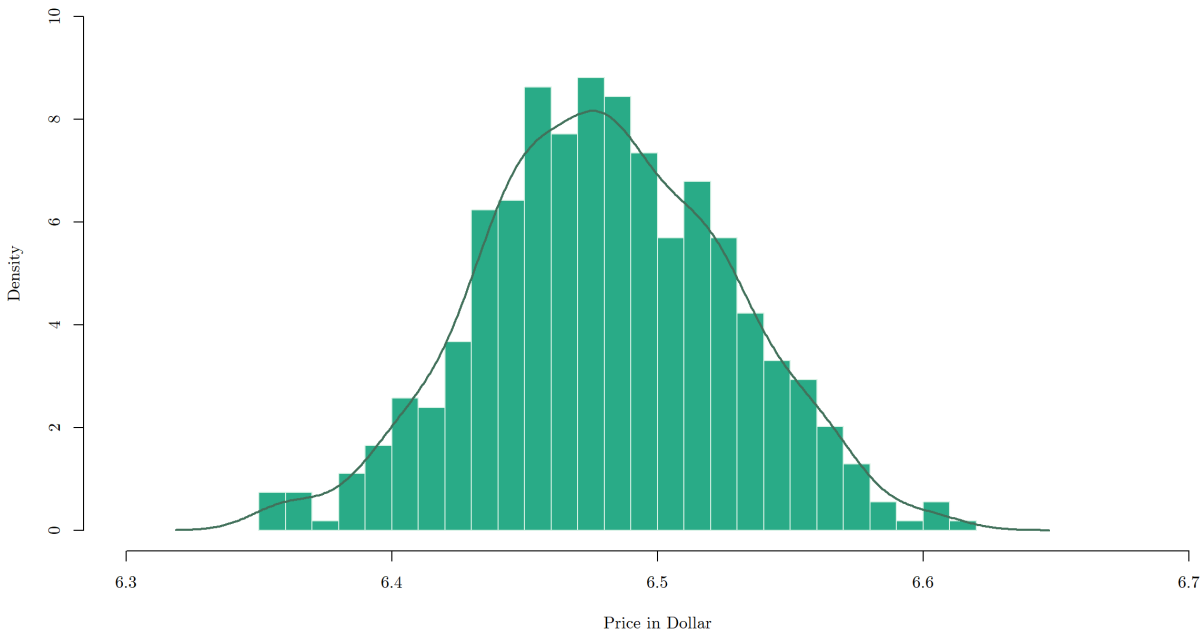


(b) Square root transformation

Figure 9: Density plots of all transformed forms of the housing prices dataset

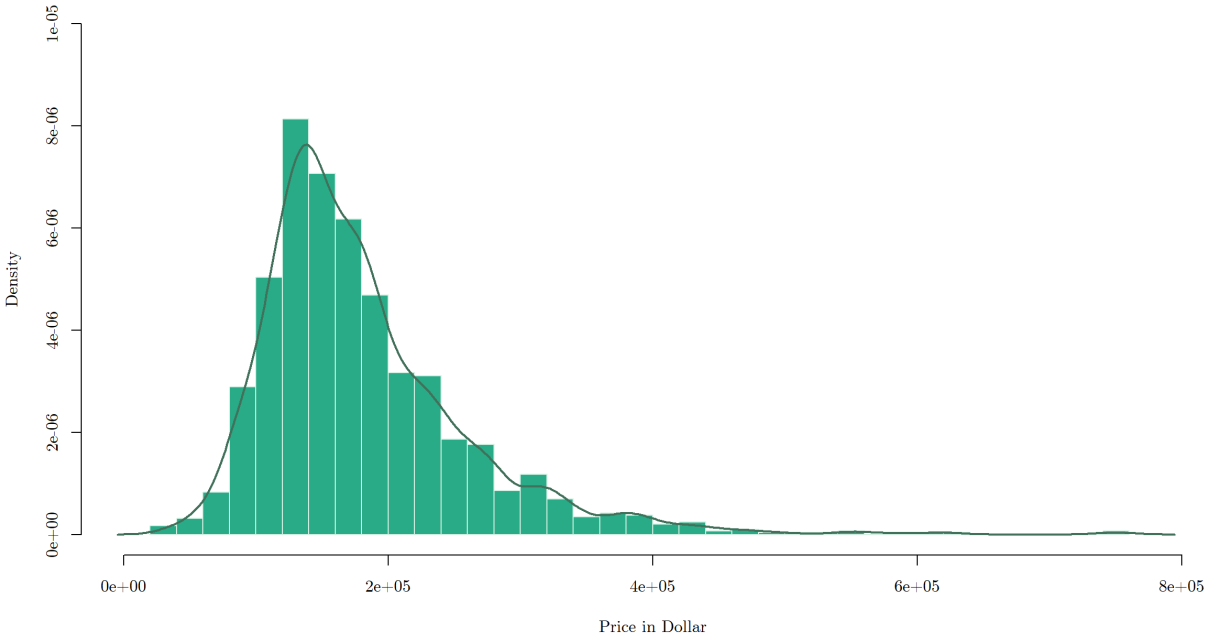


(c) Log transformation

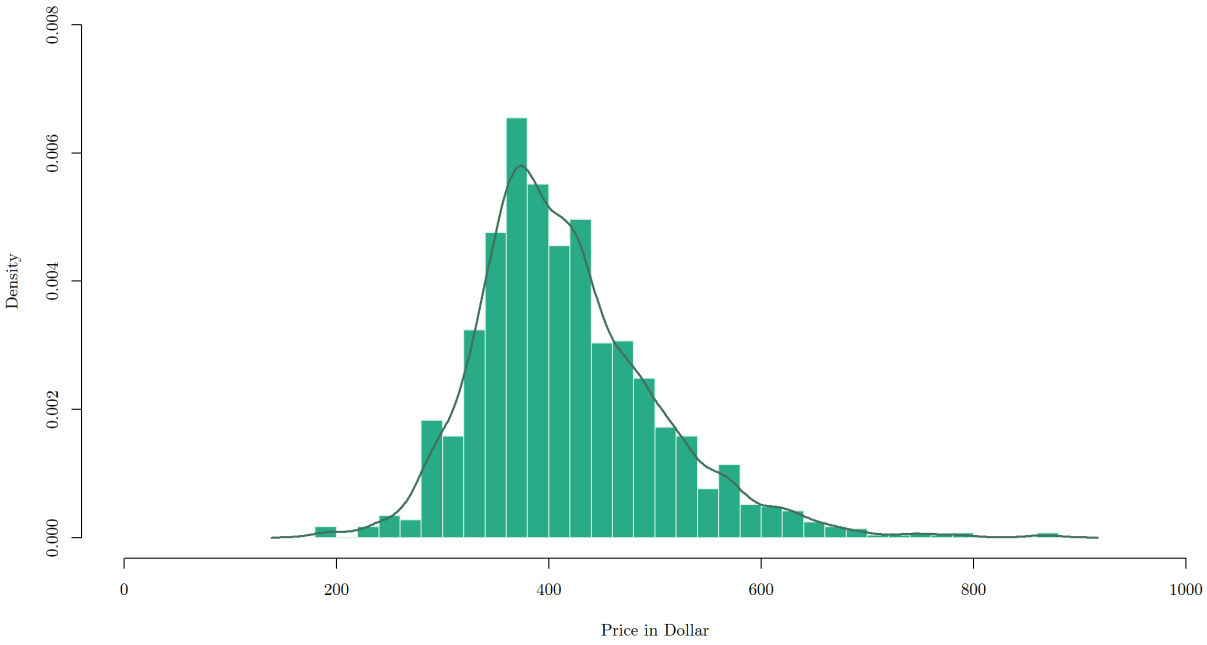


(d) Box-cox transformation

Figure 9: Density plots of all transformed forms of the housing prices dataset

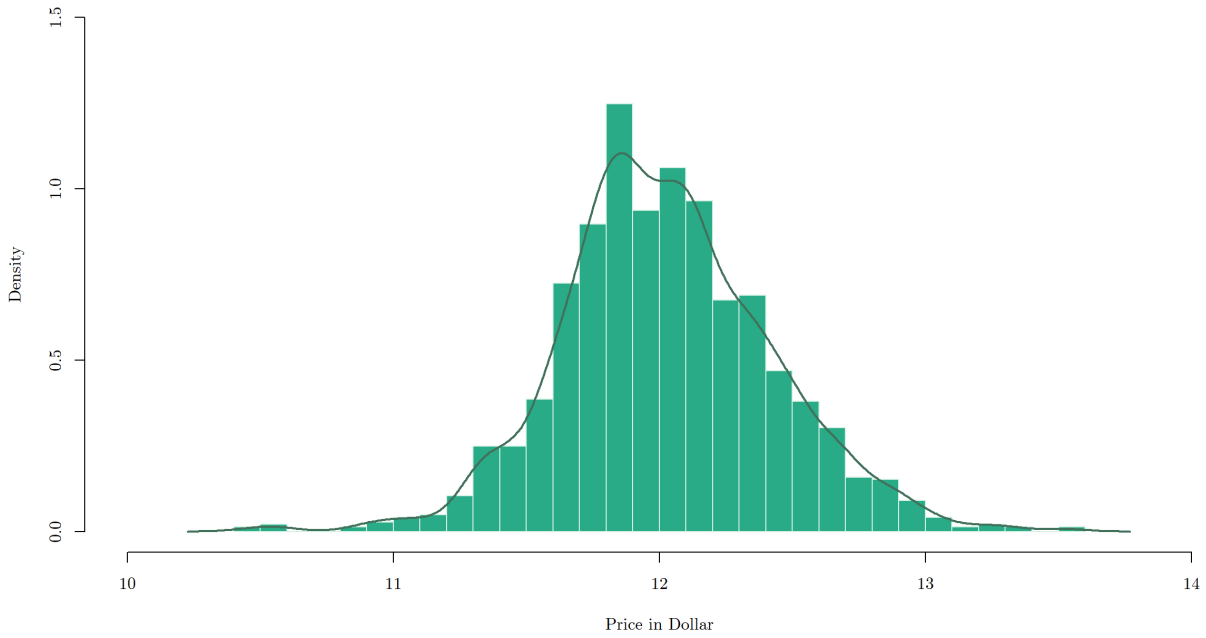


(a) No transformation

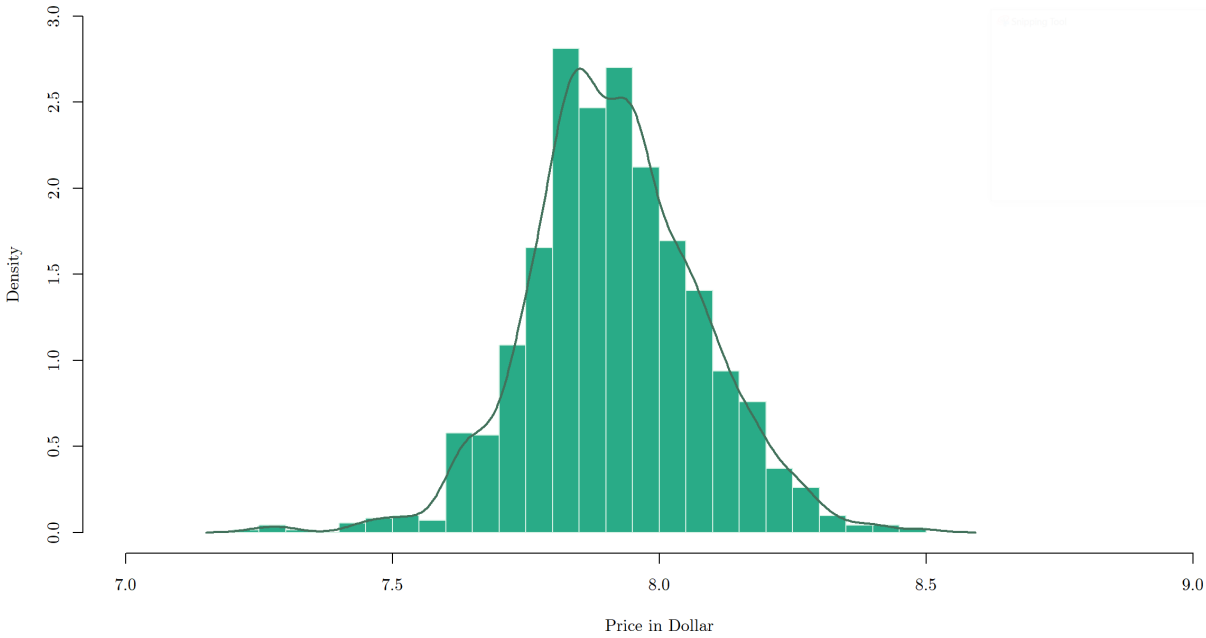


(b) Square root transformation

Figure 10: Density plots of all transformed forms of the advanced house prices dataset

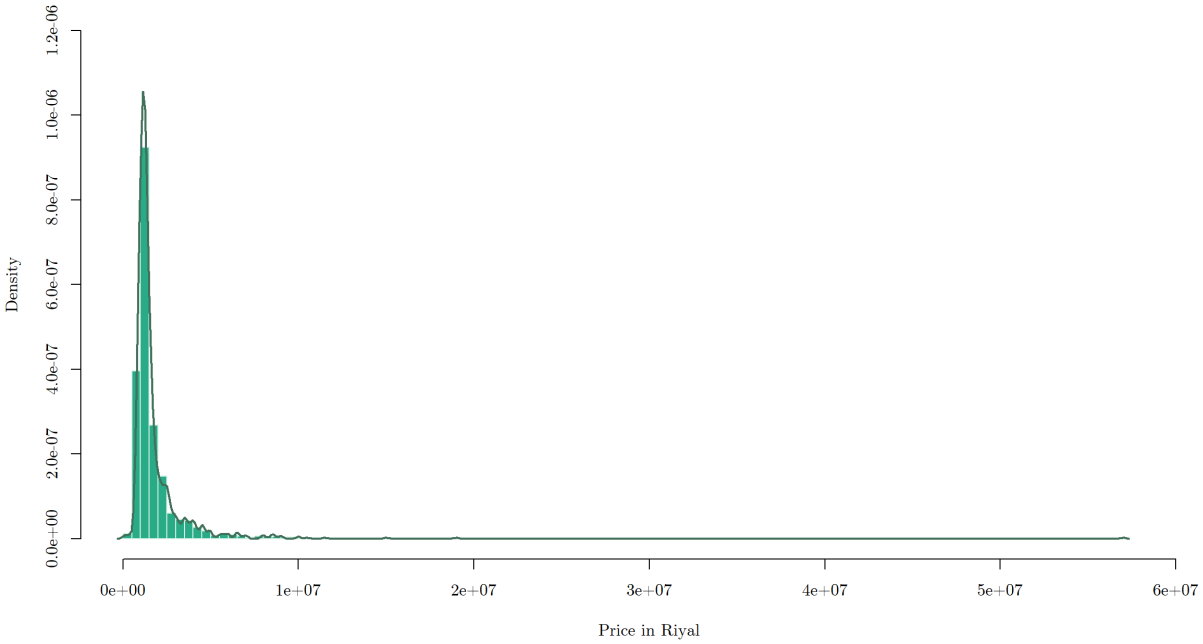


(c) Log transformation

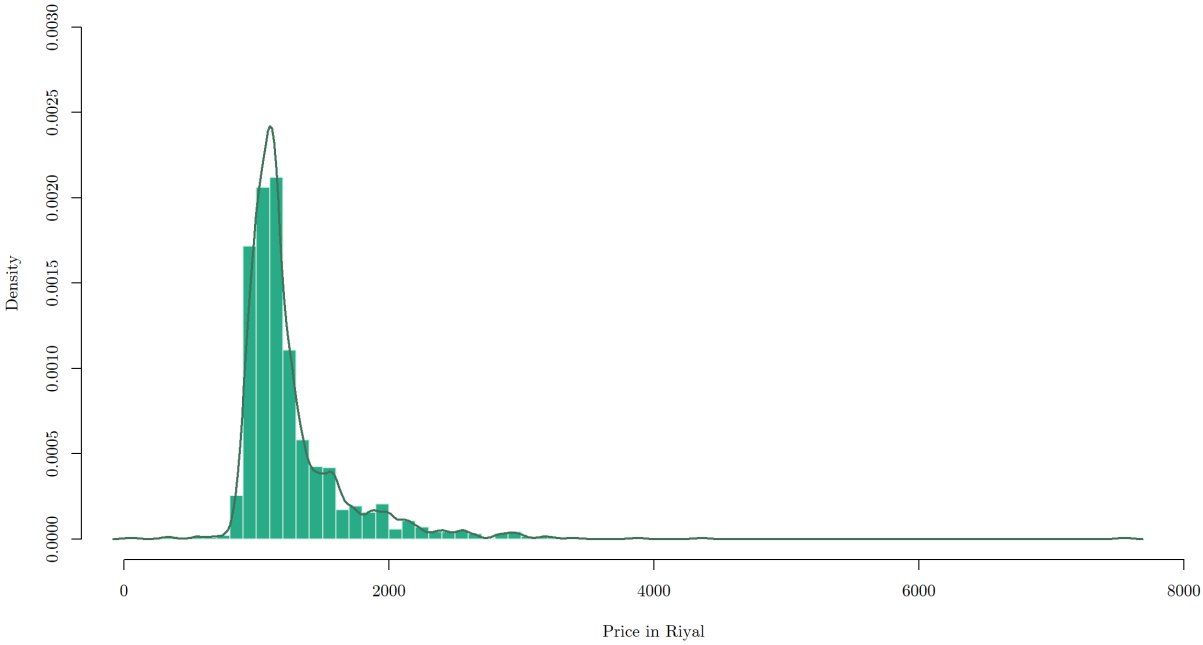


(d) Box-cox transformation

Figure 10: Density plots of all transformed forms of the advanced house prices dataset

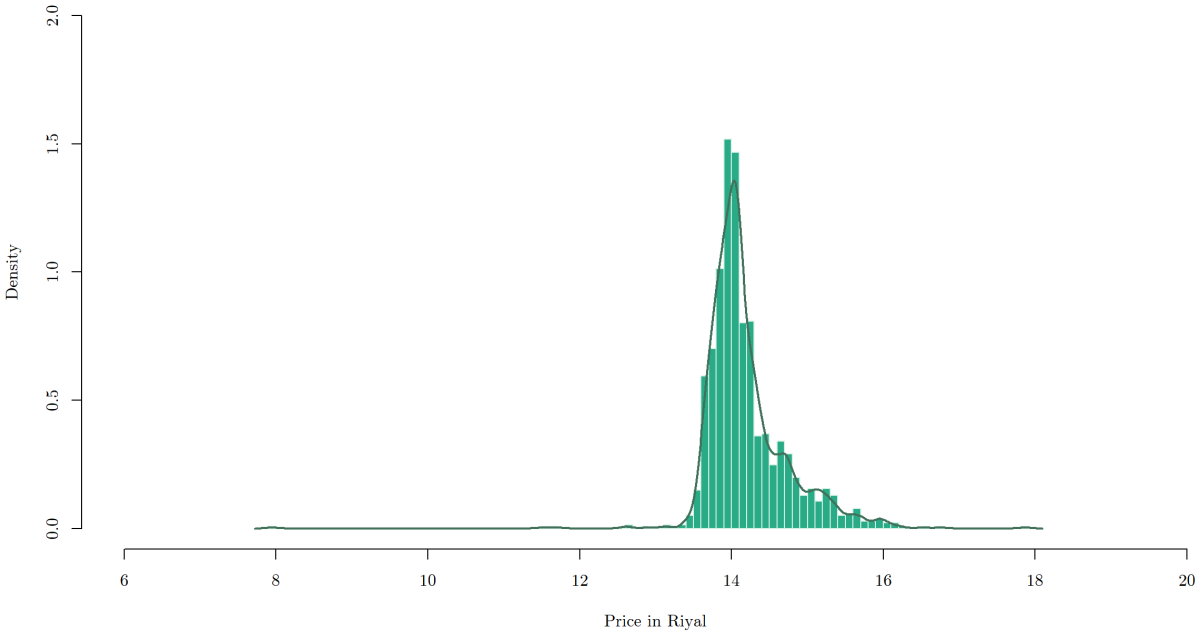


(a) No transformation

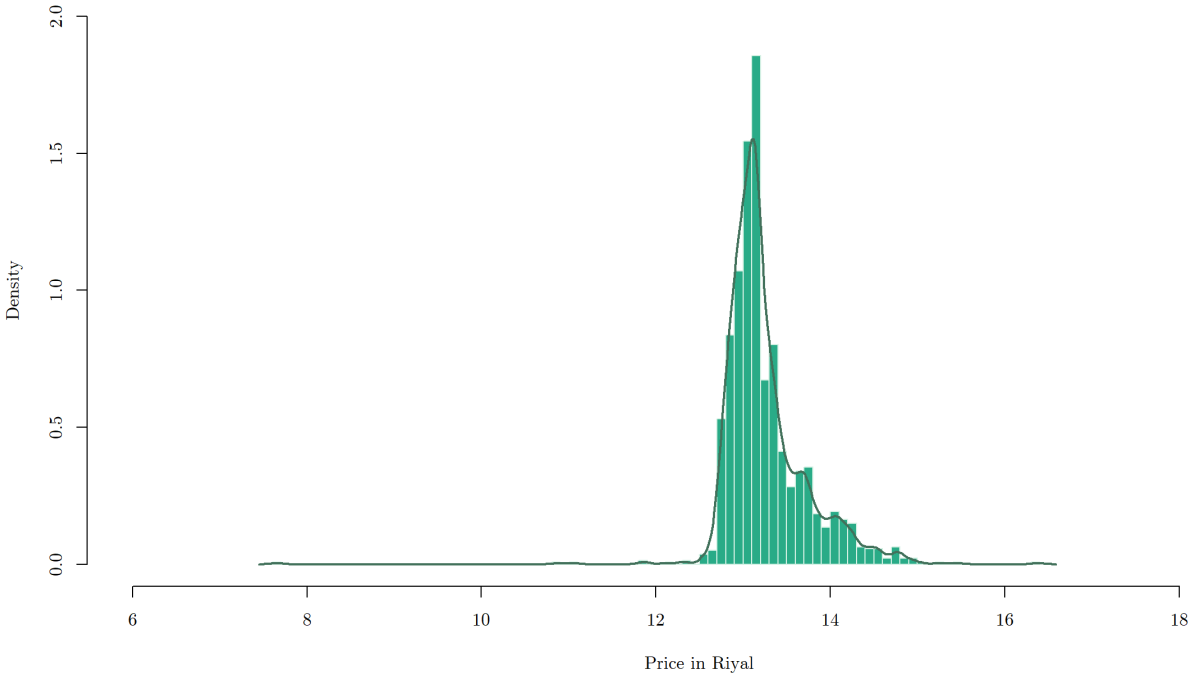


(b) Square root transformation

Figure 11: Density plots of all transformed forms of the Saudi dataset



(c) Log transformation



(d) Box-cox transformation

Figure 11: Density plots of all transformed forms of the Saudi dataset

E Performance metrics

In this research, we opt for four metrics to indicate model performance. These are MAE, RMSE, and MedAE. The MAE determines the error by summing the absolute value of the subtraction for the true outcome y_i with the estimated value $\hat{y}_i$ and dividing by the total number of data points N. Which denotes the error as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|.$$

The RSME takes the root of the mean squared error, calculated as

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

Last, the MedAE compares the group of the errors y_i and $\hat{y}_i$ and then takes the median of the errors as a metric. We calculate MedAE using

$$MedAE = Median(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, ..., |y_n - \hat{y}_n|).$$

F Results

F.1 Boston dataset

Table 23: Results of the 5×2 cross-validated models on the no transformation Boston dataset for the MAE, RMSE, and MedAE

None Boston mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	3.7581	3.8538	3.6583	2.6609	2.5953
2	3.6600	3.6560	3.7056	2.1874	2.8217
3	4.1445	3.9274	3.5959	2.5102	2.3290
4	3.7959	3.6985	3.3280	2.3120	2.2554
5	3.4781	3.5643	3.2936	2.2129	1.9972

None Boston root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.7393	5.7463	5.8131	4.0033	4.2988
2	6.0034	6.1348	6.3015	4.1619	4.9031
3	8.6351	8.0942	6.8785	4.6480	4.3285
4	5.1609	5.2189	4.9612	3.2989	3.0166
5	5.3746	5.4263	4.8552	3.2668	3.0698

None Boston MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.7393	5.8285	5.4948	3.9574	4.2988
2	6.0034	6.0040	6.0015	3.7995	4.4523
3	8.6351	8.3500	7.0229	5.3395	4.4477
4	5.1609	5.2352	5.2468	3.3088	2.9357
5	5.374566	5.4292	5.4470	3.2020	3.9032

Table 24: Results of the 5×2 cross-validated models on the square root transformation Boston dataset for the MAE, RMSE, and MedAE

Square root Boston mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	3.4186	3.5423	2.8589	2.6901	2.8329
2	3.4802	3.5343	3.0697	2.2915	2.6748
3	3.6795	3.5782	3.0369	2.5993	2.8748
4	3.5408	3.5017	2.7416	2.1838	2.4146
5	3.0955	3.1591	2.5284	2.1350	2.0125

Square root Boston root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.3531	5.4840	4.7349	4.1833	4.4683
2	5.8932	6.0327	5.6872	4.2828	4.8720
3	7.9153	7.7447	7.0027	5.2382	5.0368
4	4.9595	5.1211	4.5036	3.1806	3.3810
5	4.9745	5.1190	4.5342	3.1631	3.5466

Square root Boston MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.3531	5.4324	4.6093	3.9387	4.6075
2	5.8932	5.9820	5.7045	4.1182	4.3333
3	7.9153	7.7896	6.2422	5.3172	5.8268
4	4.9595	5.1125	4.5673	3.1527	3.6585
5	4.9745	5.0017	4.5862	3.3121	3.6303

Table 25: Results of the 5×2 cross-validated models on the log transformation Boston dataset for the MAE, RMSE, and MedAE

Log Boston mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	3.2715	3.3669	2.7409	2.5458	3.6598
2	3.4044	3.4217	2.9716	2.3358	2.9052
3	3.3975	3.3210	2.9436	2.6911	3.0418
4	3.3725	3.3564	2.9780	2.2867	2.6467
5	2.8710	2.9380	2.4978	2.0064	2.7783

Log Boston root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.1544	5.3797	4.7067	3.8055	4.6632
2	5.8790	5.9687	5.3878	4.3432	5.4030
3	7.6397	7.5883	6.9069	5.4285	6.3832
4	4.9108	5.1420	4.4161	3.4534	4.4751
5	4.8123	4.8886	4.5342	3.1214	4.2160

Log Boston MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.1544	5.2536	4.6598	3.9036	5.1072
2	5.8790	5.8935	5.3143	4.3280	5.2476
3	7.6397	7.6293	6.8219	5.6563	6.3754
4	4.9108	5.0286	4.2638	3.4276	4.1961
5	4.8123	4.9133	4.5387	3.2540	4.0920

Table 26: Results of the 5×2 cross-validated models on the box-cox transformation Boston dataset for the MAE, RMSE, and MedAE

Box-Cox Boston mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
folder	3.3191	3.4041	2.7193	2.6661	2.8045
2	3.4279	3.4461	2.8417	2.3219	2.6621
3	3.4857	3.3951	2.9490	2.7893	2.7630
4	3.4220	3.4019	2.9451	2.2952	2.4402
5	2.9356	2.9825	2.3838	2.0091	2.3241

Box-Cox Boston root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.2118	5.4179	4.6919	4.0369	4.4629
2	5.8813	5.9715	5.4548	4.3126	5.2037
3	7.7149	7.6362	6.9090	5.3166	5.5561
4	4.9151	5.1349	4.4999	3.2780	3.9527
5	4.8537	4.9371	4.2552	3.2247	3.8126

Box-Cox Boston MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	5.2118	5.3609	4.5928	4.0804	4.5492
2	5.8813	5.9971	5.4204	4.1563	5.3190
3	7.7149	7.7032	6.5560	5.7434	6.0867
4	4.9151	5.0608	4.5232	3.4322	5.0811
5	4.8537	4.9608	4.3478	3.2183	3.9233

F.2 Russia rent dataset

Table 27: Results of the 5×2 cross-validated models on the no transformation Russia rent dataset for the MAE, RMSE, and MedAE

None Russia rent mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	13323.15	13398.09	13253.88	2462.34	2496.21
2	14918.92	14794.12	15275.00	3292.92	3199.59
3	14308.21	14247.61	13820.72	2576.32	2227.27
4	13860.33	13535.82	13155.61	2947.51	2837.23
5	13476.93	13375.58	14872.58	2980.92	3114.62

None Russia rent root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	18542.66	18792.80	28229.32	7722.43	9477.58
2	32414.97	32222.00	39811.61	13080.12	10152.79
3	30636.27	30523.75	36552.75	11151.76	7561.21
4	20902.22	20798.05	29929.97	13965.47	16123.97
5	18265.05	18289.94	31714.68	8546.55	13207.59

None Russia rent MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	18542.66	28034.47	28236.46	7467.66	8565.22
2	32414.97	32282.66	39919.64	14068.40	9258.41
3	30636.27	30640.74	36754.46	11496.87	9217.89
4	20902.22	20986.85	29942.24	14977.24	16419.71
5	18265.05	18329.74	31915.84	11174.81	13726.80

Table 28: Results of the 5×2 cross-validated models on the square root transformation Russia rent dataset for the MAE, RMSE, and MedAE

Square root Russia rent mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	10950.61	11283.13	6537.98	2199.55	2724.33
2	12874.80	12835.71	8496.63	3379.45	2843.01
3	11902.33	11909.07	7271.96	2593.68	1749.58
4	11393.90	11306.15	6906.81	2775.07	2916.29
5	11321.68	11347.29	7442.79	2624.81	2203.38

Square root Russia rent root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	16716.00	17388.71	16932.97	7493.56	9468.37
2	31689.79	31789.78	31200.36	15344.09	10887.11
3	30035.48	30109.52	29058.05	13605.03	8385.83
4	19532.00	19572.78	20666.07	14580.78	15061.55
5	17382.35	17855.83	20119.30	7699.94	12233.11

Square root Russia rent MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	16716.00	16658.32	22248.04	7282.58	8402.41
2	31689.79	31894.94	35017.98	15378.09	8983.34
3	30035.48	30038.41	29292.30	15114.30	7559.62
4	19532.00	19407.66	22538.62	15603.28	15898.51
5	17382.35	17349.11	22607.73	9591.02	13017.38

Table 29: Results of the 5×2 cross-validated models on the log transformation Russia rent dataset for the MAE, RMSE, and MedAE

Log Russia rent mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	10277.49	10636.20	5130.02	2209.27	3566.16
2	12231.25	12229.20	7399.88	3474.34	4837.09
3	11181.48	11301.65	6079.02	2870.43	2900.46
4	10723.42	11020.00	5724.17	2750.79	3538.38
5	10687.15	10753.64	4850.41	2299.34	5081.18

Log Russia rent root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17276.04	17479.07	8331.97	7774.51	8897.19
2	32370.72	32284.24	29340.65	18041.55	22378.86
3	30647.49	30594.36	27026.83	19784.32	8727.79
4	20259.87	20604.57	15888.32	15039.22	14526.89
5	18692.35	18803.34	9390.05	7741.23	12353.09

Log Russia rent MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17274.50	27205.52	8638.97	7482.20	10029.81
2	32291.64	38882.39	28089.44	18452.63	10341.75
3	30579.56	31009.71	27067.42	18240.47	12884.48
4	20404.76	21207.55	14931.25	15452.67	12946.57
5	18820.79	21808.4742	9271.84	10928.85	18229.82

Table 30: Results of the 5×2 cross-validated models on the box-cox transformation Russia rent dataset for the MAE, RMSE, and MedAE

Box-Cox Russia rent mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	10338.93	10381.69	14140.03	2226.34	13799.79
2	12308.71	12224.09	16023.65	3933.92	15765.33
3	11287.25	11196.92	14757.43	3296.99	14373.58
4	11029.72	11040.60	14216.08	2815.70	13856.32
5	11106.99	11133.03	16017.72	2323.60	15686.64

Box-Cox Russia rent root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	19653.89	19690.47	29116.92	8546.16	28742.56
2	34003.22	33924.54	40545.57	23648.86	40227.34
3	32193.48	32101.41	37274.70	23569.30	36970.42
4	22650.90	22806.01	30866.26	15548.30	30511.43
5	21863.57	22071.25	32705.13	8025.03	32293.20

Box-Cox Russia rent MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	19666.15	26917.60	29116.92	8089.89	28732.15
2	33845.27	34392.94	40545.57	23739.30	40184.87
3	32079.49	32422.77	37274.70	24014.09	36922.82
4	22630.04	24250.08	30866.26	15969.94	30525.34
5	22039.66	29822.97	32705.13	9915.93	32334.55

F.3 Housing prices datasets

Table 31: Results of the 5×2 cross-validated models on the no transformation housing prices dataset for the MAE, RMSE, and MedAE

None housing prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	886254.39	892294.95	1496450.93	862925.87	1073114.40
2	781471.82	755698.43	1435879.52	779058.73	1069256.99
3	735796.66	733156.16	1334813.71	793615.42	903901.77
4	845117.27	821948.04	1383356.44	826820.55	910211.13
5	783013.67	787217.21	1354148.12	756926.22	1025181.31

None housing prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1250276.054	1259665.92	2091360.61	1288836.84	1278506.23
2	1036543.83	1014759.19	1988036.28	1182649.00	1624991.04
3	962422.78	959215.66	1675889.47	1057992.60	988677.08
4	1071118.08	1051552.67	1818104.55	1250713.83	1196034.48
5	1118468.27	1136426.93	1969360.43	1155905.60	1278893.45

None housing prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1250276.05	1259665.92	2091377.34	1217120.66	1464049.07
2	1036543.83	1014759.19	1988037.08	1196699.63	1624991.04
3	962422.78	959215.66	1675877.02	924381.28	1177419.47
4	1071118.08	1051552.67	1818109.72	1159206.61	1194670.70
5	1118468.27	1136426.93	1969362.61	1174357.11	1516301.25

Table 32: Results of the 5×2 cross-validated models on the square root transformation housing prices dataset for the MAE, RMSE, and MedAE

Square root housing prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	865000.07	863999.89	1225152.65	866573.49	985895.90
2	753395.05	721744.35	1081480.46	782102.63	972230.42
3	728021.96	727720.21	1013925.66	787677.11	837920.17
4	833727.76	809722.38	1087753.27	893536.07	1098634.24
5	746474.04	756447.34	1073710.07	818682.93	1051693.98

Square root housing prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1223023.72	1225458.72	1773997.87	1210848.65	1267810.20
2	1017615.87	992389.52	1656374.41	1188456.93	1431803.55
3	958208.02	957339.88	1369610.48	1059825.67	1069571.00
4	1088030.89	1065634.88	1532466.03	1241444.02	1138823.97
5	1095095.56	1105828.96	1677850.84	1200695.14	1452268.81

Square root housing prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1223023.72	1225458.63	1735109.52	1207731.58	1371897.95
2	1017615.87	992389.50	1652766.08	1202314.04	1431803.55
3	958208.02	957339.75	1347150.35	939066.52	1245787.25
4	1088030.89	1065634.88	1532466.03	1241444.02	1452904.87
5	1095095.56	1105882.46	1655493.53	1182794.23	1654272.41

Table 33: Results of the 5 × 2 cross-validated models on the log transformation housing prices dataset for the MAE, RMSE, and MedAE

Log housing prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	10277.49	10636.20	5130.02	2209.27	3566.16
2	12231.25	12229.20	7399.88	3474.34	4837.09
3	11181.48	11301.65	6079.02	2870.43	2900.46
4	10723.42	11020.00	5724.17	2750.79	3538.38
5	10687.15	10753.64	4850.41	2299.34	5081.18

Log housing prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17276.04	17479.07	8331.97	7774.51	8897.19
2	32370.72	32284.24	29340.65	18041.55	22378.86
3	30647.49	30594.36	27026.83	19784.32	8727.79
4	20259.87	20604.57	15888.32	15039.22	14526.89
5	18692.35	18803.34	9390.05	7741.23	12353.09

Log housing prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17274.50	27205.52	8638.97	7482.20	10029.80
2	32291.64	38882.39	28089.44	18452.63	10341.75
3	30579.56	31009.71	27067.42	18240.47	12884.48
4	20404.76	21207.55	14931.25	15452.67	12946.57
5	18820.79	21808.47	9271.84	10928.85	18229.82

Table 34: Results of the 5×2 cross-validated models on the box-cox transformation housing prices dataset for the MAE, RMSE, and MedAE

Box-Cox housing prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	837308.15	835842.52	1292871.89	892561.78	1496308.83
2	755293.01	752348.59	1194951.12	789105.15	1431482.34
3	743697.06	747645.53	1114526.32	774413.17	1337781.93
4	814223.98	811712.84	1144752.89	897513.04	1383702.84
5	755407.38	764109.38	1185224.36	828835.51	1354055.23

Box-Cox housing prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1191793.81	1189711.82	1888779.37	1212569.82	2077627.11
2	1036448.52	1039530.01	1770643.25	1217851.76	2006897.38
3	978785.80	984368.23	1442973.35	1062286.59	1672544.12
4	1130921.15	1137492.42	1578339.27	1242856.33	1820314.69
5	1074754.75	1085064.17	1820087.40	1220164.75	1968692.77

Box-Cox housing prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1169209.06	1189711.82	1888779.37	1212569.82	2085713.16
2	1084724.01	1039533.42	1770643.25	1217851.76	2006897.38
3	983151.61	984265.96	1442973.35	1055877.04	1671571.33
4	1156909.42	1137494.46	1578339.27	1242856.33	1820260.86
5	1073321.15	1085049.21	1820087.40	1273231.12	1968692.77

F.4 Advanced house prices dataset

Table 35: Results of the 5×2 cross-validated models on the no transformation advanced house prices dataset for the MAE, RMSE, and MedAE

None advanced house prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	20339.85	20664.60	54695.66	17274.82	16227.89
2	20416.46	19203.48	56057.98	15946.46	17452.55
3	22557.17	22780.02	61830.27	19548.97	19387.00
4	19752.49	18228.35	49835.21	15113.45	14747.48
5	20372.89	19509.37	54383.82	16324.39	17500.50

None advanced house prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	30427.88	30653.00	85707.03	28428.16	25212.33
2	36070.54	33071.78	76206.20	32256.74	34255.17
3	40550.92	42181.47	97352.99	35988.85	36881.37
4	50424.20	42673.60	68981.72	26938.51	23178.79
5	29535.82	28629.30	74911.55	23350.29	25196.93

None advanced house prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	30427.88	31035.35	85592.86	28739.80	26151.67
2	36070.54	33067.83	76206.20	32120.16	34438.37
3	40550.92	41084.71	97316.34	38476.82	39136.28
4	50424.20	39669.44	68975.04	26835.80	24883.92
5	29535.82	28634.87	74764.31	24540.78	25968.35

Table 36: Results of the 5×2 cross-validated models on the square root transformation advanced house prices dataset for the MAE, RMSE, and MedAE

Square root advanced house prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	18689.81	18821.94	25327.71	17409.16	15393.26
2	18537.85	17739.01	23537.47	15761.64	16480.91
3	20633.66	20788.04	28448.00	19846.23	19940.77
4	18146.45	17407.38	19650.79	15143.35	15205.35
5	18109.16	17509.13	22997.77	16211.02	16037.06

Square root advanced house prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	28545.29	29852.86	45021.63	29460.64	27439.77
2	37403.92	33418.02	36025.99	32599.99	31062.01
3	37278.91	37569.10	56574.35	38357.96	35784.71
4	64604.10	51896.64	32662.33	25380.34	23379.03
5	26838.73	25621.04	37732.81	24306.80	23045.29

Square root advanced house prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	28545.29	28603.73	45021.63	29360.65	24894.91
2	37403.92	34662.15	36025.99	30843.71	34732.16
3	37278.91	37276.07	56574.35	39244.36	35777.87
4	64604.10	64038.97	32662.33	25949.88	25417.96
5	26838.73	26415.97	37732.81	23785.09	23045.29

Table 37: Results of the 5×2 cross-validated models on the log transformation advanced house prices dataset for the MAE, RMSE, and MedAE

Log advanced house prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17635.74	17750.95	19624.39	17693.23	19904.50
2	18886.51	18073.00	17218.82	15424.16	16731.37
3	19904.81	19770.34	21375.91	20132.60	22540.37
4	22235.45	19776.11	16049.11	15231.64	17387.16
5	17290.27	16860.74	17995.06	16657.50	18482.13

Log advanced house prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	27148.50	28614.56	30427.53	29305.91	35929.98
2	45423.07	38082.92	26694.32	31895.21	30525.53
3	33983.71	33032.57	41303.08	39965.61	44025.94
4	133884.52	89692.58	25699.02	25458.89	26677.55
5	25498.87	24722.54	27176.87	24351.68	27169.55

Log advanced house prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	27340.24	29393.83	30531.18	29957.38	35802.47
2	45545.50	38506.06	27855.19	30404.44	30532.39
3	33587.76	33559.65	41446.81	40604.66	44377.43
4	133830.79	123201.78	26858.73	28603.91	26717.90
5	25443.95	24984.67	27654.45	25124.52	27060.93

Table 38: Results of the 5×2 cross-validated models on the box-cox transformation advanced house prices dataset for the MAE, RMSE, and MedAE

Box-Cox advanced house prices mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	17420.54	17642.83	23883.63	17802.17	25931.87
2	19119.78	18336.27	21680.63	15637.43	21251.73
3	19862.49	19668.02	27056.86	19640.62	27591.80
4	24473.78	20950.54	19749.72	15393.03	21942.90
5	17280.00	16850.34	22049.63	16725.13	22079.41

Box-Cox advanced house prices root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	26893.17	28579.29	38700.42	30448.59	42626.60
2	47859.88	39040.36	31001.62	31023.94	32775.31
3	33433.38	32198.53	50123.20	39224.00	56933.89
4	166984.82	105885.31	29566.06	24894.66	33306.11
5	25415.95	24527.75	33793.30	24620.08	34721.82

Box-Cox advanced house prices MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	27207.06	29560.31	38767.74	31685.65	43740.01
2	48014.12	43640.14	32767.28	29537.31	41298.32
3	33069.27	32765.62	52294.83	41840.13	55676.82
4	166832.13	143098.28	30361.03	27464.06	33174.98
5	25344.56	24770.70	34053.23	25300.92	35213.01

F.5 Saudi dataset

Table 39: Results of the 5×2 cross-validated models on the no transformation Saudi dataset for the MAE, RMSE, and MedAE

None Saudi mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	593943.23	494414.01	655857.39	368150.23	378454.19
2	512252.32	512833.44	809605.41	469791.45	476181.51
3	563732.83	498564.67	723695.12	388615.43	397157.50
4	592192.46	465126.71	706570.50	372251.36	400737.56
5	585545.84	501252.23	686577.34	392033.98	428093.70

None Saudi root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1058263.40	1057052.05	1380991.90	781494.51	786482.54
2	1840816.64	1842473.72	3524202.82	2807194.00	2648110.74
3	1087525.40	1086484.51	1633752.83	828687.74	1024138.00
4	861970.64	860714.44	1379103.14	744851.65	752494.83
5	986235.61	987176.70	1668552.09	1112438.70	1125781.05

None Saudi MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	1057941.79	983643.29	1381033.49	846010.73	787437.82
2	1841840.40	2330458.24	3524229.19	2827097.84	2649626.75
3	1087290.26	972896.19	1633756.63	824713.59	947126.75
4	861077.11	815386.43	1379033.82	760025.78	774135.23
5	986502.91	1076227.33	1668510.30	1087695.92	1294249.65

Table 40: Results of the 5×2 cross-validated models on the square root transformation Saudi dataset for the MAE, RMSE, and MedAE

Square root Saudi mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	405330.8905	403392.6032	525936.2921	357632.0053	366592.5742
2	358251.8757	356936.759	684856.0277	467046.5576	445134.8176
3	423548.8923	430661.6596	608665.8624	365157.6157	399039.1615
4	387875.4637	387800.5712	576093.5977	364277.5855	360934.4086
5	410636.8369	410517.6814	557198.4203	384080.927	384315.2558

Square root Saudi root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	916220.95	918100.52	1244282.59	836005.71	794749.68
2	733916.33	729329.19	3453575.96	2944921.93	2595098.61
3	979530.14	971669.49	1503397.21	814255.01	1067862.91
4	690600.71	690781.28	1196613.09	709069.76	686856.54
5	920337.95	921993.20	1536833.07	1055646.37	1185686.88

Square root Saudi MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	915991.25	917344.21	1246739.40	844112.21	777440.45
2	729338.96	706658.70	3461846.83	2872764.56	2712933.08
3	980007.35	970269.72	1510621.46	846107.48	1078609.96
4	690780.51	695481.16	1201000.44	691961.90	745209.05
5	923387.49	989109.05	1569775.28	1058992.76	1176532.47

Table 41: Results of the 5×2 cross-validated models on the log transformation Saudi dataset for the MAE, RMSE, and MedAE

Log Saudi mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	412284.97	412949.5787	390974.53	350085.63	370830.00
2	12145021.52	11930454.43	1183439.00	474564.27	486485.57
3	454500.14	455500.35	452960.60	371940.26	407866.23
4	391447.06	395703.95	365192.95	347802.83	357431.32
5	411650.94	412146.16	387430.83	380880.99	419823.02

Log Saudi root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	954827.63	985340.9113	941447.67	835046.95	788407.53
2	198379198.80	167700422.20	14540484.19	3018122.25	2665501.27
3	1072574.81	1080746.38	1118169.60	909016.19	1099174.29
4	730040.26	774075.68	707828.56	716090.95	704418.66
5	983454.91	1083006.07	1072104.14	1067340.37	1135627.37

Log Saudi MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	954461.22	965176.69	937898.21	844140.56	2267818.74
2	196998515.30	98127040.64	5770828.79	2953313.67	2715831.15
3	1074796.56	1043421.03	1598198.56	909229.60	1015735.26
4	730094.46	745144.41	677368.33	666734.48	724175.36
5	991025.50	995215.53	1013875.74	1065441.30	1073217.51

Table 42: Results of the 5×2 cross-validated models on the box-cox transformation Saudi dataset for the MAE, RMSE, and MedAE

Box-Cox Saudi mean absolute error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	412286.36	413377.41	390476.95	353390.20	376413.45
2	15735536.72	15286341.53	1827286.46	473997.05	467710.13
3	455183.47	456219.97	453877.13	371428.05	417682.55
4	391739.02	396531.97	360860.01	347889.36	380805.00
5	411978.73	412440.39	388730.85	387360.13	414085.73

Box-Cox Saudi root mean squared error					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	955454.80	986846.55	938964.65	835679.24	839923.17
2	258669950.10	215841064.40	23873637.63	3028906.19	2742569.48
3	1075349.20	1083919.74	1121596.24	876498.02	1050613.51
4	730792.25	773807.06	698454.89	703912.76	692458.82
5	984013.54	1086133.84	1072187.13	1076407.24	1112480.88

Box-Cox Saudi MedAE					
Fold	**LR**	**PLR**	**SVM**	**RF**	**XGB**
1	956650.80	969829.34	938964.65	844880.80	874040.51
2	256839114.30	118264834.50	4966923.51	2961402.28	2837229.15
3	1077880.94	1042524.90	1121596.24	910116.98	1037500.48
4	730969.89	745231.83	658718.31	679302.08	807271.63
5	991965.47	993980.17	937130.91	1085895.45	1037833.67