Universität St.Gallen

School of Management, Economics, Law and Social Sciences

Comparative Analysis of Machine Learning-Based Valuation Models for Predicting Residential Real Estate Prices

Research Seminar: Real Estate Finance

submitted by

Tim Graf 16-609-257

Kilian Gerding 15-606-577

to

Prof. Roland Füss

on

April 21, 2021

JEL Classification: C21, C31, C53, R32

Keywords: US Real Estate Prices, Real Estate Price Prediction, Hedonic Regression, Spatial Autoregressive Model, Machine Learning, Random Forest, Random Tree, XGBoost, Bagging Predictor, Stacked Generalization, Ensemble

Contents

List of Tables.....III

List of Abbreviations.....III

1. Introduction.....1

2. Literature Review.....2

3. Model Selection.....3

3.1 Multivariate (Non-)Linear Regression Model: Hedonic Pricing.....3

3.2 Machine Learning Models.....5

3.2.1 Bagging.....5

3.2.2 Random Forest.....6

3.2.3 Boosting with XGBoost.....7

3.2.4 Stacked Generalization.....8

4. Empirical Analysis.....9

4.1 Data.....10

4.2 Data Cleaning and Descriptive Statistics.....11

4.3 Hedonic Model Estimation and Prediction.....11

4.3 Machine Learning Prediction.....15

5. Conclusion.....19

Literature.....20

Appendix.....22

Declaration of Authorship.....39

List of Figures

Figure 1: Construction of decision trees.....6

Figure 2: Example how Stacked Generalization is implemented.....8

Figure 3: Visualization of House Locations (H3 Hexagons).....14

Figure 4: Boxplot of prediction errors.....18

List of Tables

Table 1: Hedonic Regression 12

Table 2: Linear Prediction 13

Table 3: Comparison of Prediction Performance 15

Table 4: Feature Importance 16

List of Abbreviations

AHS	American Housing Survey
ML	Machine Learning
OLS	Ordinary Least Squared
RF	Random Forest
RMSE	Root Mean Squared Error
XGB	XGBoost

Abstract

This paper evaluated the methods and predictive power of several machine learning algorithms in comparison to a linear hedonic regression model to predict residential house prices in Los Angeles County, CA in 2016. This research provides a method in combining the benefits of regression models with machine learning methods. The empirical analysis shows that the boosting algorithm XGBoost stands out in a holistic assessment. It outperforms the traditional hedonic regression model and nearly all other machine learning algorithms in predictive power having the lowest *Root Mean Square Error* while remaining computationally effective and rather robust to potential overfitting issues. Contrary to other machine learning algorithms, such as neural networks, XGBoost allows for interpretability of feature importance due to its particular use of decision-tree ensembles.

1. Introduction

Hedonic real estate pricing theory claims that real estate prices are a function of the object's heterogeneous characteristics. This modeling approach has been dominating real estate pricing. The practicability of the concept proposed by Rosen (1974) found application mainly in multivariate (non-)linear regression settings. However, this approach is more steered towards inference rather than prediction. It further demands the analysis of statistical assumptions, such as the normality of the residuals, homoskedasticity, independence, and multicollinearity. (Pérez-Rave, Correa-Morales & González-Echavarría, 2019 p. 59; Mullhainathan & Spiess, 2017, p. 93)

The upcoming of ever-larger data sets, enabled by digital real estate marketplaces, a growing number of hedonic variables, and a shift of interest more towards prediction of real estate prices fostered the use of Machine Learning (ML) Methods in real estate pricing. These methods can not only enhance predictive power (Mullhainathan & Spiess, 2017, p. 94) but also help to select the important hedonic variables (Pérez-Rave, Correa-Morales & González-Echavarría, 2019 p. 60). While installing and applying ML methods is rendered easy through out-of-the-box and open-source packages, its difficulty lies in the correct application and a clear inference of the underlying algorithm.

On the one hand, some ML methods use regularization techniques to prevent overfitting in out-of-sample data, however, this can lead to omitted variable bias or wrongly specified models. On the other hand, some ML methods need to be scaled or normalized data as inputs to produce satisfactory predictions. Finally, some ML models excel only at high dimensions and observation spaces, while working well with few data points. (Pérez-Rave, Correa-Morales & González-Echavarría, 2019 p. 61; Mullhainathan & Spiess, 2017, pp. 91—93)

Thus, the objective of this research is to evaluate whether machine learning approaches, in particular, ensemble methods, such as Bagging predictor, Random Forest, XG Boost or Stacked Generalization perform better than attribute-based models using hedonic regression. Further, the application of ML spatial regressions is discussed. A real estate dataset was therefore retrieved from Kaggle, which originally was issued by the real estate platform Zillow containing real estate valuations and multiple hedonics in 2016 and 2017 for the county of Los Angeles, USA.

The analysis is structured into four parts. First, a short literature review on machine learning in real estate is provided. Second, the model choice, their parameters, and the benefits and disadvantages of the models are discussed. Following, details and issues of the dataset, the procedure for data cleaning, and variable selection are explained. In the fourth part, the empirical results of the training and predictions are presented and discussed. The focus lies on comparison metrics for regressions and the individual model performances.

This paper contributes to how machine learning approaches and hedonic pricing models can be used for the house price analysis in Los Angeles, USA. Furthermore, it suggests which hedonic variables are important in prediction. However, the containing research does not provide sufficient proof in transferring the knowledge to other markets, as other geographies might lay their importance on different hedonics and external factors. Therefore, there is a need in applying the methods of this paper to national and even international housing data.

2. Literature Review

With the growth of publicly available data and the ease of use of applying machine learning approaches with packages in programming languages such as R and Python, research in forecasting real estate prices using machine learning algorithms in combination with hedonic pricing theory is becoming more frequent.

To begin with, Mullainathan and Spiess (2017) discuss the importance of machine learning for econometrics and show an application using data from the American Housing Survey (AHS) of 2011. They report an outperformance of machine-learning-based algorithms such as random forest and ensemble over multi-variate regression under ordinary least squares (OLS) in the out-of-sample dataset. However, the authors stress that (1) while ML-methods can produce better predictions, it often comes with a lack in inferential and explanatory power of ML-model, (2) such methods tend to overfit the training sample and thus do not necessarily generalize well for out-of-sample predictions, which can be partly mitigated by using cross-validation, and (3) the choice of initial variables can greatly enhance or decrease the model's predictive performance. Lastly, they mention that various combinations of the feature set can produce similar predictions. It is therefore important to reason which variables to include rather than use all available variables. (Mullainathan & Spiess, 2017, pp. 88—95)

Pérez-Rave, Correa-Morales & González-Echavarría (2019, pp. 59—62) contribute to the methodology on big data regression analysis of real estate using incremental sampling with resampling to partly circumvent the computational issues when working with large datasets. The used data for comparing various algorithms was built on transaction data from web advertisements in homes in Colombia and the Metropolitan AHS of 2011. Their research reports how hedonic regressions can be limiting when working with big data for prediction purposes and does not guide the selection of the important variables. However, they use the importance plots of ML-algorithms to optimize the variable selection in their hedonic regression.

A recent study on comparison of the predictive performance of ML-based algorithms in the context of real estate is performed by Čeh, Kilibarda, Lisec, & Bajat, B. (2018, pp. 2—4) using apartment prices in Ljubljana, Slovenia. The findings report an outperformance of random forest algorithms over hedonic regressions, partly thanks to its non-linear nature, which confirms the prospective of this machine learning technique on apartment price predictions. Oladunni & Sharma (2016, p. 522) study the use of principal component analysis, support vector machines, and k-nearest neighbors as algorithms to predict housing prices in eight counties of Washington, USA, and further validate the use of the algorithms in hedonic pricing theory. Park & Bae (2015, pp. 2928—2930) compare various ML-algorithms such as decision trees and ripper to data of transaction prices in Virginia, USA.

Lu, Li, Qin, Yang, and Goh (2017, p. 319) use a hybrid Lasso and Gradient boosting regression model to predict individual house prices using US housing data from an international competition posted on Kaggle. Their findings report the importance of data quality which include properly scaling data, selecting features, and tuning the parameters of the algorithms. Antipov & Pokryshevskaya (2012, pp. 1772—1773) use ML-methods for mass appraisal of Russian residential apartments and find random forest as a competitive technique in predicting prices. Lastly, Shahhosseini, Hu & Pham (2019, pp. 87—89) demonstrate on real estate prices from Boston, USA, how ensembles can even outperform single ML-algorithms.

To conclude, considering the publication years and computational advances, research on ML-algorithm application in real estate price/rent prediction is gaining more traction in academia. Overall, many results confirm some advantages of ML methods to the linear regression model in their predictive power, however, sacrificing inferential power. Moreover, there appears to be a consensus in the analyzed papers, that enhancing linear model capacity and variable selection can be supported by applying ML-methods.

3. Model Selection

This chapter provides an extensive overview of the used methods and models, as well as their strengths and limitations.

3.1 Multivariate (Non-)Linear Regression Model: Hedonic Pricing

Multivariate Regression Model provides a suitable framework to estimate the economic significance of housing characteristics on a house’s price. Generally, a house price is a composition of and to a certain degree bounded by its characteristics. These, also referred to as hedonics, include house characteristics such as the number of bedrooms, bathrooms, the size of the living area and the housing lot; if the house has a garage, pool, or balcony; location characteristics such as the neighborhood; or environmental characteristics such as air pollution. These characteristics are treated as independent variables in a multivariate regression setting estimating the house price, as a dependent variable. That said, hedonic pricing models reflect representative parameters used to make pricing, selling, or purchasing decisions. (Rosen, 1974, pp. 34-39; Sirmans, Macpherson & Zietz, 2005, pp. 3-6)

A general hedonic price model with a specific property *n* and *k = 1, 2, …, K* hedonics can be written as follows:

$$\mathrm{P}_{\mathrm{n,t}} = \mathrm{f}(\mathrm{c}_{\mathrm{k,t}}, \mathrm{l}_{\mathrm{k,t}}, \mathrm{e}_{\mathrm{k,t}}, \varepsilon_{\mathrm{t}}), \qquad \text{where} \tag{1}$$

$$\mathrm{c}_{\mathrm{k,t}} = (\mathrm{c}_{\mathrm{1,t}}, \mathrm{c}_{\mathrm{2,t}}, \dots, \mathrm{c}_{\mathrm{k,t}}) \text{ are building characteristics,}$$
$$\mathrm{l}_{\mathrm{k,t}} = (\mathrm{l}_{\mathrm{1,t}}, \mathrm{l}_{\mathrm{2,t}}, \dots, \mathrm{l}_{\mathrm{k,t}}) \text{ are location characteristics,}$$
$$\mathrm{e}_{\mathrm{k,t}} = (\mathrm{e}_{\mathrm{1,t}}, \mathrm{e}_{\mathrm{2,t}}, \dots, \mathrm{e}_{\mathrm{k,t}}) \text{ are environmental characteristics,}$$
$$\text{and } \varepsilon_{\mathrm{t}} \, is \text{ an error term}$$

The general expression can be specified in a linear multivariate parametric model:

$$\mathrm{P}_{\mathrm{n,t}} = \beta_{0,\mathrm{t}} + \sum_{\mathrm{k=1}}^{\mathrm{K}} \beta_{\mathrm{k,t}} \mathrm{c}_{\mathrm{k,t}} + \sum_{\mathrm{k=1}}^{\mathrm{K}} \gamma_{\mathrm{k,t}} \mathrm{l}_{\mathrm{k,t}} + \sum \delta_{\mathrm{k,t}} \mathrm{e}_{\mathrm{k,t}} + \varepsilon_{\mathrm{t}} \tag{1.1}$$

This parametric model allows estimating the hedonics' marginal contributions $\beta_{k,t}$, $\delta_{k,t}$ and $\gamma_{k,t}$ to the house price (Rosen, 1974, pp.34-39; OECD et. al., 2013, pp.50-52). The marginal contributions, however, depend on the level of house prices. An additional bedroom has a decreasing absolute marginal impact with ever-higher house prices. Therefore, hedonic pricing models are often quoted as log-linear models with the log house price as a dependent variable. Unit changes of bedrooms or bathrooms then translate into percentage changes of house prices (Sirmans, Macpherson & Zietz, 2005, pp. 3—5). Furthermore, the model setting leverages the statistical regression framework, statistical testing and inference can be conducted. Moreover, transformation to a non-linear regression via interaction terms between variables is possible.

However, the model relies heavily on a correct variable selection, which can be challenging before estimating and therefore is exposed to a *Specification Problem* (Sirmans, Macpherson & Zietz, 2005, pp. 3—5). This also raises the issue of omitted variable bias, when excluding certain hedonics (OECD et. al., 2013, pp. 50-52; Sirmans, Macpherson & Zietz, 2005, pp. 3—5).

Next to the *Specification Problem*, there are limits incorporating the uniqueness of a house mainly driven by its unique lot location. Attempts to incorporate location parameters as hedonics such as distance to downtown district or distance to schooling were proposed by Ottensmann, Payton & Man (2008) show. However, the limits of these location hedonics are that they do not account for spatial spillover effects and autocorrelation in contrast to Spatial Autoregressive Models. Moreover, the values of certain characteristics depend on their locations. Whereas a garage or brick construction might be higher valued in cold regions, a balcony might be higher valued in warmer regions (Sirmans, Macpherson & Zietz, 2005, pp. 3-5).

Lastly, hedonics tend to be correlated among each other e.g., the number of bedrooms and the living area in square feet. Hence, hedonic pricing models are prone to multicollinearity. This might cause higher standard errors and unstable coefficients. (OECD et. al., 2013, pp. 50-52)

3.2 Machine Learning Models

ML aims to optimize a single or multiple performance criteria based on historical or exemplary data. For regression tasks often the Root Mean Squared Error (RMSE) between predicted and actual data is to be minimized. This is conducted via specifying model parameters and programming models to optimize those parameters on a subset of the available data, known as training data. This results in either a descriptive outcome to expand knowledge yielded from data or in a predictive outcome yielding a prediction to some uncertain event. (Ethem, 2010, pp. 1-3)

There exists an extensive variety of ML models and a seemingly endless number of variations of such. In this house price prediction approach, the focus is on *meta-learning* models, also known as *ensemble* models, trying to enhance learning robustness. Among these *ensemble* models *Bagging, Random Forest, Boosting,* and *Stacked Generalization* were used in the empirical comparison.

While there are multiple ensemble models, all share a similar mechanism: First, a sample of or the full training data is allocated to one or multiple learning model(s). After the learning models predicted their outcomes based on this training data, results are combined via a combination function. This yields an ensemble model. (Lantz, 2013, pp. 337-339)

Problems such as overfitting, performance on very large datasets, and synthesizing predictions from different domains can be addressed by these methods, which will be explained with more rigor in the following sub-chapters.

3.2.1 Bagging

The name Bagging originates from **B**ootstrap **Agg**regation, which is essentially a resampling method where N samples are drawn from a given training set S with replacement to fit N decision trees. It can be applied to classification or numeric regression tasks. After bootstrapping subsamples, the leaner estimates one decision tree over each of those N subsamples. The predictions yielded by this show high variance. However, Bagging then aggregates the estimates in the numeric regression case by averaging the subsample estimators, thus yielding a robust estimate. (Ethem, 2010, pp. 360)

Tuning bagging parameters comprise the amount of N bootstraps to be drawn, interchangeably with the number of trees, and the depth of each of the N trees. The former is impractical to tune as the RMSE decreases with decreasing marginal effects as N increases. However, it can be determined practically by testing iterations of N to see, where the RSME becomes stable. The latter can be hyper-tuned.

3.2.2 Random Forest

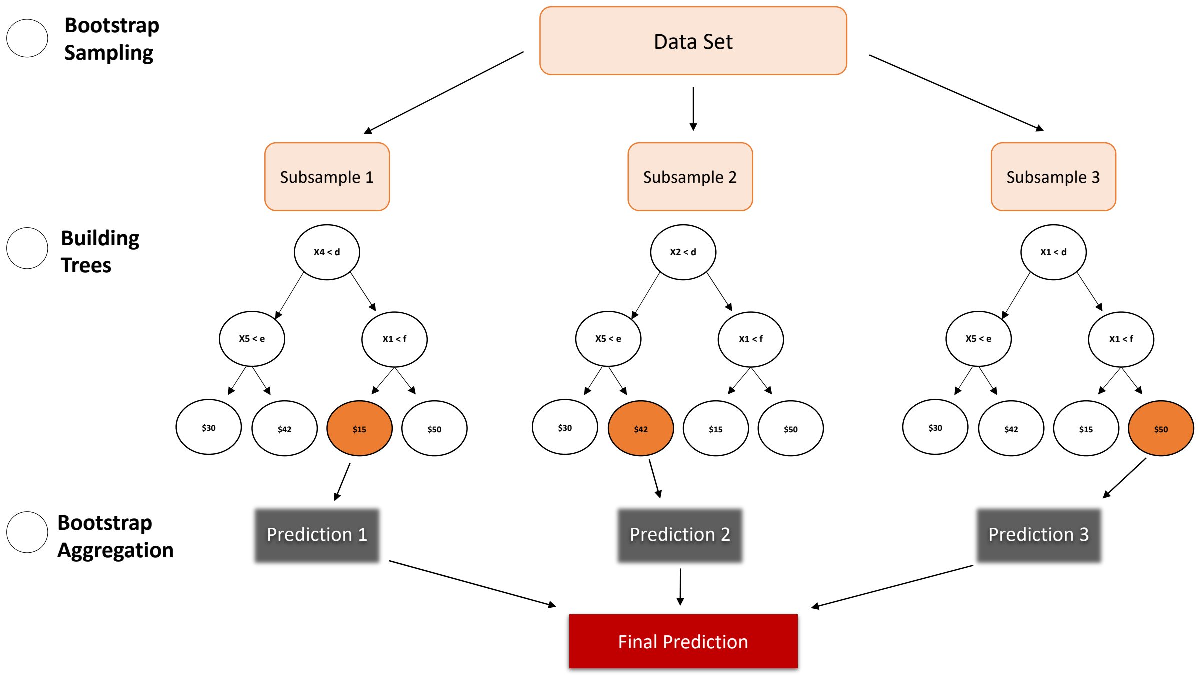


Figure 1: Construction of decision trees

Random Forest (RF) belongs to the ensemble learning methods and can be used for both classification and numeric regression analysis. It is a predictor consisting of a collection of randomized regression trees, which are for example created by constructed trees from randomized subsamples of the feature set or by randomly splitting nodes. This randomness is what differs RF from bagging predictors and simple regression trees. First, the given dataset is randomly split into equal subsamples. Then for each subsample, a decision tree is constructed based on a random set of m features of the whole feature set. In a third step, the predictions from all individual trees are aggregated into a "forest" which renders a final prediction (see figure 1). Through the nature of the tree structure, it can fit non-parametric data well. This process is comparable to the effects of k-fold cross-validation. (Ho, 1995, p. 278; Breiman, 2001, pp. 5-6)

The main parameters in optimizing Random Forests are the total amount of trees to be computed, the number of variables randomly sampled as candidates at each split, and the minimum sample per leaflet. Typically, the random subset of variables is computed as *1/3* of all variables for regressions. The other parameters must be tuned individually based on the use case and data set. This can be done by setting a range of the possible parameters, computing the prediction given a combination of parameters, and choosing those parameters which result in the best metric, e.g., minimizing the mean square error of the prediction. (Biau & Scornet, 2016, pp. 201 - 205)

As the number of trees increases, it does not always result in better performance compared to a model with fewer trees and rends the interpretability of the model difficult. Furthermore, it can lead to overfitting the training dataset and not finding general patterns needed for out-of-sample prediction, but the risk of overfitting is smaller with Random Forest than in other decision-tree-based methods (e.g., bagging). Therefore, this parameter needs to be manually tuned. (Biau & Scornet, 2016, pp. 199 - 203)

The main benefit of Random Forest is that it can handle large, high-dimensional data sets and is computationally effective while finding complex and non-linear patterns in the given data. Another benefit is that feature interpretation remains possible. This is done by computing the frequency of used features in splitting nodes over the whole forest and plotting the aggregation. (Biau & Scornet, 2016, p. 198 – 199; Breiman, 2001, pp. 6 & 28; Oshiro, Perez & Baranauskas, 2012, pp. 154 -155)

3.2.3 Boosting with XGBoost

XGBoost (extreme gradient boosting) is a form of Gradient-Tree-Boosting which is computationally efficient through its novel tree learning algorithm for handling sparse data, the use of parallel and distributed computing, cache optimization, and the way it handles weights among other reasons. XGBoost has been used extensively in ML-competitions and research, as it tends to be computationally efficient while leveraging the benefits of gradient boosting and Random Forest. (Chen & Guestrin, 2016, p. 785; Nielson, 2016, pp. 1-3).

XGBoost is an ensemble technique by combining simpler models, which are often referred to as base learners or weak learners. These simpler models are single decision trees that are combined to predict the final value (see Figure 1). Models are added sequentially until no further improvements can be made. It uses gradient descent (hence the name gradient-tree-boosting) to minimize the loss when adding new models (see equation 2). The most commonly used base learner for XGBoost is a regression tree algorithm, which is also used in Random Forest. In comparison to Random Forest, XGBoost builds the individual tree one at a time, while Random Forests constructs them independently. Additionally, Random Forest combines results only at the end, while XGBoost continuously combines trees and the resulting prediction. The final prediction for a given example is the sum of predictions from each tree. (Chen & Guestrin, 2016, pp. 785 - 786)

$$\mathcal{L}(\Phi) = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k), \tag{2}$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

$$l(\hat{y}_i, y_i)\ is\ the\ training\ loss$$

XGBoost uses three main methods to minimize overfitting: a regularized learning objective, shrinkage, and column subsampling. The regularization term, which is Ω in the above equation, 2, penalizes the model for choosing a more complex model (e.g., more base learners). Shrinkage, like the learning rate *eta*, scales newly added weights by a factor n after each step of tree boosting while leaving space for future trees to improve the model. Column subsampling is column-wise sampling of the features to build the tree model, as seen in the algorithm Random Forest. (Chen & Guestrin, 2016, pp. 786 - 787).

Finally, XGBoost has two main parameters which are the number of iterations in which the model is improved and the learning rate. Generally, with high numbers of iterations, the complexity of the model will tend to increase. This leads to overfitting the training data set and not performing adequately in an out-of-sample data set. The learning rate is the weight associated with the new tree added in the iterative approach of XGBoost. Thus, a low learning rate does not put much weight on new trees and results in the model converging slower to the actual values, aiding in the generalizability of the model. (Nielsen, 2016, p. 44)

3.2.4 Stacked Generalization

Stacked Generalization, also known as Super Learning, is a meta-ensemble method of using a high-level model to combine lower-level models to achieve greater predictive accuracy. The algorithm has been introduced in 1992 by Wolpert for the use case of neural networks and further developed by Breiman (1996). There have since been further contributions and variations in the method of Stacked Generalization, but the general procedure remained the same. The method can be best explained by providing an example. (Ting, K.M. & Witten, 1997, pp. 2—3; Wolpert, 1992, pp. 4—6)

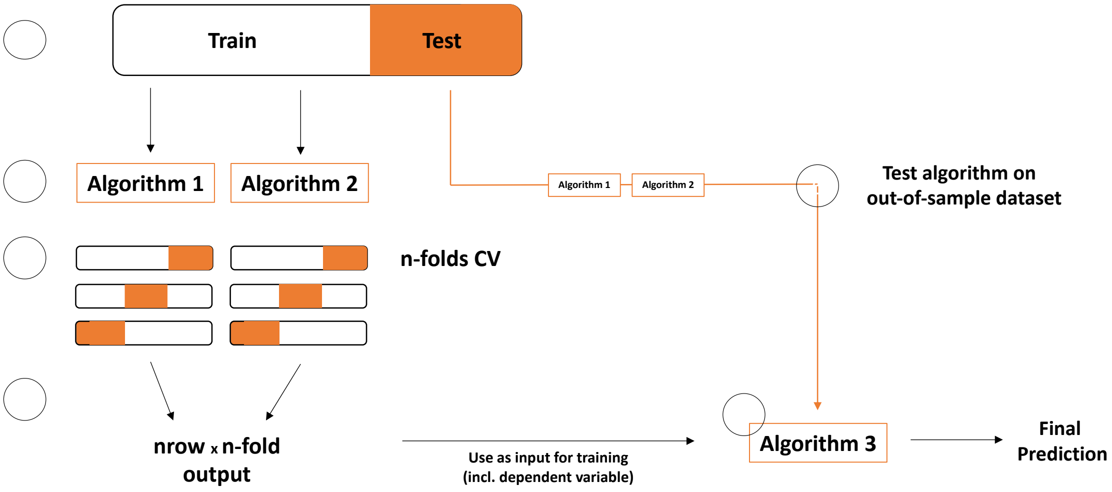


Figure 2: Example of how Stacked Generalization is implemented

1. First, the dataset is split into a training and testing (i.e., out-of-sample) set.
2. The second step is to choose a number *m* of algorithms (e.g., Linear Regression, Bagging, Random Forest) as base learners (i.e., lower-level models) for the first layer and one algorithm (e.g., XGBoost) for the second and final prediction layer.
3. Third, the base learners are trained with the training set using a *n*-fold cross-validation method.
4. The resulting output will be a *n x m* matrix, which is a weighted prediction of the cross-validations.
5. The output will be used as input for the algorithm in the second layer while adding the dependent variable of the training set. The algorithm is then trained on the input.
6. Lastly, the base learners are trained on the full training dataset at once and then applied to the testing dataset. The resulting predictions are given as input for the algorithm in layer two to produce the final prediction.

(Ting, K.M. & Witten, 1997, pp. 2-6)

In the original form, Stacked Generalization uses a linear regression as the algorithm in the second layer (step 5) to evaluate which of the base learners explains the dependent variable best. The base layer algorithms are then trained on the whole training dataset and their predictions weighted according to the coefficients obtained from the regression to form the final prediction value. Newer methods allow also other algorithms, e.g., XGBoost, in the second layer. (Naimi & Balzer, 2017, pp. 460—462)

Like Bagging, Random Forest, and XGBoost, Stacked Generalization is ideal for parallel computation as each tree construction and cross-validation in the first layer can be computed independently. However, when applying these together in Stacked Generalization it becomes computationally intense as multiple algorithms must be trained. The implementation of general Stacked Generalization is found to produce mostly the same if not marginally better quality of predictions, which shall be tested in this paper (Van der Laan, 2007, pp. 16 –17). This comes, on the other hand, at the cost of interpreting the inner workings of the algorithms and interpretability. (Ting, K.M. & Witten, 1997, pp. 16-17)

4. Empirical Analysis

This chapter presents the data processing, as well as the estimated models and results of the empirical analysis. Given that the similarity for 2016 and 2017 data is very high, ML models estimated beyond the multivariate linear regression will be only conducted on 2016 data.

For each of the algorithms, the dataset has been split equally into a training set (75% of observations) and a testing set (25% of observations). In an ideal scenario, the dataset is split randomly, and the predictions repeated multiple times following a Monte-Carlo simulation. However, this would result in a huge computational burden and has thus been skipped. To allow comparative results, the splitting has been fixed with a seed, resulting in all algorithms using the same training and testing dataset.

4.1 Data

The data used to conduct the empirical analysis had to meet certain requirements: (i) the data had to be sufficiently distributed in housing prices to avoid a sample bias, (ii) there needed to be a time dimension of at least 2 periods, (iii) some spatial component, (iv) a sufficient amount of hedonics to specify a realistic model, (v) as close as possible to an objective pricing such as a transaction price or filed tax amount rather than a subjective assessment, and lastly (vi) a certain number of observations is required and an upper threshold should not be exceeded to remain computationally effective.

Open-source data that fulfills all those criteria is difficult to obtain. Two potential datasets were closely assessed in their fit to this empirical study: (i) American Housing Survey (AHS) and (ii) Zillow Housing Platform Data.

The AHS (AHS Census Data) has already been widely used as a dataset for hedonic estimation and ML pricing prediction (see Mullainathan & Spiess, 2017). One approach would have been to evaluate hedonic pricing and ML performance on the three most recent surveys (2015, 2017, 2019). However, there were multiple obstacles identified: (i) the AHS is missing a spatial component in its open-source format, (ii) there are is bounded objectivity to the stated prices as the survey format asks for individual assessment of the house prices by its owners, and (iii) while a large variety of hedonics was accessible, the number of observations post data cleaning was on the lower end of the threshold.

In contrast, Zillow's open-source Housing Platform Data (Zillow Data) has not been widely used for scientific research purposes yet. Still, this data provides a (i) spatial component, (ii) a time dimension (2016, 2017), (iii) sufficient hedonics and observations post data cleaning, and (iv) a pricing variable which is based on the building tax filed for the house. This building tax amount, which was due for all observations in 2015 post data cleaning, made the impression of a recent and reliable house price approximation. The latter characteristics convinced this study to further work with Zillow Housing Platform Data.

Nevertheless, there is a trade-off in trying to minimize the assessor's subjectivity in housing prices. On the one hand, the tax filing approach used by Zillow Housing Platform might underestimate transaction prices, on the other, this approach omits cyclical pricing levels.

4.2 Data Cleaning and Descriptive Statistics

The initial number of observations, house prices equivalently, for 2016 and 2017 was 2’985’217. Despite the amount the observations contained in the dataset, there were many missing observations for important hedonics and conceptual irregularities. The following hedonics were considered, taking missing values and conceptual importance into account:

> Hedonics: *Building Price (tax value), Living Area (sqft), Lot Area (sqft), Garage Area (sqft), No. Bedrooms, No. Bathrooms, No. The story, No. Garage, Area Garage (sqft), No. pools, Fireplace, Tub or Spa*
>
> Spatial Variables: *Latitude, Longitude, ZIP Code, County Code*

All the above-mentioned hedonics were used as predictive variables. According to the meta-study of Sirmans, Macpherson & Zietz (2005, pp.10) all are featured in the *Top 20* most used hedonics in academia. Next to that, all the selected hedonics represent basic budling characteristics in contrast to more detailed hedonics such as heating systems, rooftop materials, or base materials.

The first step in cleaning the data was to ensure that we were indeed estimating a building structure. This meant filtering prices for observations with ≥1 bedroom(s) and ≥1 bathroom(s). Empty lots, building structures, which are not finished, and unspecified building structures are therefore excluded. This left us with almost all Single-Family Residential Homes (type of housing). The remaining housing types (11 obs.) were omitted as they were not representative. Another conceptual error was that a building structure could have a garage with 0 sqft garage area or had a garage but >0 sqft garage area. This was cleaned as well. After conceptual cleaning, all observations for missing values in building price, living area, no. bedrooms, no. bathrooms, no. the story, no. garage, no. pools were excluded (see Appendix A1 and A2). Lastly, obvious outliers were carefully detected and omitted. See the histograms and plots in Appendix A3 to A7 for 2016 as a reference. The remaining observations in 2016 were 562’983 and in 2017 563’973. Find the histograms of certain hedonics in Appendix A8 and descriptive statistics in Appendix A10. For a short overview, the mean price for a house in 2016 was 148’732 USD with a standard deviation of 132’372, average living area was 1’901 sqft with 768 sqft std. deviation, the average age of a house was 53 years with 3.5 bedrooms and 2.3 bathrooms.

4.3 Hedonic Model Estimation and Prediction

To keep the hedonic price estimation in appropriate scaling, House Price (are log-normally distributed see A8), Living Area, Lot Area, Garage Area, and Age are transformed in log values. This helps to interpret non-unit variables better. The marginal contributions from unit-hedonics such as bathrooms and garages also possess now a decreasing marginal impact. Percentage changes in House Price are also stated by percentage changes in Living Area or Age. Other units hedonic are in a linear format, yielding a *100%*coefficient* impact on percentage changes in House Prices. The following hedonic regression model is estimated:

$$\begin{aligned}&\log(House\ Price)\\ &\quad = \beta_0 + \beta_1 \log(Living\ Area) + \beta_2 \log(Lot\ Area) + \beta_3 \log(Garage\ Area)\\ &\quad + \beta_4 \log(Age) + \beta_5 No.Bedrooms + \beta_6 No.Bathrooms + \beta_7 No.Story\\ &\quad + \beta_8 No.Garage + \beta_9 No.Pool + \beta_{11} D(Fireplace) + \beta_{12} D(Tub\ or\ Spa)\end{aligned} \tag{3}$$

Almost all coefficients in the 2016/17 regression were significant on the 1% level except the number of garages and the fireplace dummy. For judging the significance of the coefficients, robust standard errors were used, as the Breusch-Pagan Test for heteroscedasticity was highly significant. Furthermore, both regressions yielded a R^2 of ~65%. Coefficients across the two years are nearly identical. For 2016, which is the base year in later analysis, yielded the following insights for 2016:

Table 1: Hedonic Regression

	2016		2017	
Variables	Estimate	Std. Error†	Estimate	Std. Error†
Intercept	*7.078 ****	*0.024*	*6.811 ****	*0.024*
log Living Area	*0.891 ****	*0.003*	*0.905 ****	*0.003*
log Lot Area	*0.062 ****	*0.001*	*0.063 ****	*0.001*
log Garage Area	*0.115 ****	*0.006*	*0.112*	*0.006*
log Age	*-0.699 ****	*0.002*	*-0.652 ****	*0.002*
No. Bedrooms	*-0.057 ****	*0.001*	*-0.061 ****	*0.001*
No. Bathrooms	*0.107 ****	*0.001*	*0.109 ****	*0.001*
No. Story	*-0.028 ****	*0.001*	*-0.028 ****	*0.001*
No. Garage	*0.002*	*0.003*	*0.001*	*0.003*
No. Pools	*0.109 ****	*0.001*	*0.105 ****	*0.001*
Dummy Fireplace	*0.012 **	*0.005*	*0.012 **	*0.005*
Dummy Tub or Spa	*0.096 ****	*0.002*	*0.094 ****	*0.002*
N	*562983*		*563973*	
Breusch-Pagan Test	*16745 ****		*16765****	
Adj. R	*0.6592*		*0.6551*	

**** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$; †Robust Standard Errors*

The log-log relationship can be interpreted as elasticities. Specifically, a 0.891% change in the Living Area (sqft), a 0.062% change in the Lot Area (sqft), a 0.115% in the Garage Area, and lastly a -0.699% in the houses Age would constitute a 1% change in the House Price. Furthermore, unit-hedonics can be interpreted in a semi-log fashion where one added bedroom would raise the house price by an estimated -5.54% ($e^{-0.057}$-1), one added bathroom by 11.29%, one added story by -2.76%, and one added garage by 0.2%. The remaining dummy variables, the number of Pools was essentially just 0 or 1, are interpreted similarly: possessing a pool increased the house price by an estimated 11.52%, a fireplace by 1.21%, and a hot tub or a spa by 10.08%. The coefficients for 2017 can be interpreted analogously.

Exemplary house with 1500sqft living area, 5000sqft lot area, 300sqft garage area, 30 years of age, 5 bedrooms, 2 bathrooms, 2 stories, 1 garage, 1 pool, 1 fireplace, and 1 spa would have an estimated price of 266,370.47 USD. This mechanic is also used to predict the test data set containing 25% of the observations of the full 2016 data set. The estimated coefficients are used to predict the actual house prices. This basic linear prediction exercise resulted in the following:

Table 2: Linear Prediction

	Root Mean Square Error		*Adjusted R²*			
Algorithms	Train	Test	Train	Test	*ø Prediction Error*	*p-value*
Linear Regression	*0.4010*	*0.4001*	*0.6587*	*0.6607*	*-0.00051*	*0.632*

The obtained R^2 of around 0.66 and RMSE of around 0.4 from our test subsample were nearly identical to the initial estimation. The RMSE can be seen as the average deviation from predicted to actual values. A simple t-test testing for the average difference to be zero (ø Prediction Error) yielded that one cannot reject the Null-Hypothesis.

Investigating the hedonic regression, Variance Inflation Factors (see Appendix A11) were computed for each hedonic to control for potential multicollinearity. This factor indicates how much predictors inflate the explained variance by collinearities (see Appendix A9). A Variance Inflation Factor of 1 indicates no collinearity, while values above 10 should be corrected. The Variance Inflation Factors for 10 out of 12 predictors is below 5, however, the Garage Area and the number of garages seem to possess a rather high Variance Inflation Factor around 9.6 indicating potential multicollinearity.

Furthermore, QQ-Plots (see Appendix A12) and Breusch-Pagan Test indicated potential heteroskedasticity. Therefore, robust standard errors were used. Another potential threat to the model is being exposed to omitted variable bias. Since the dataset contained large amounts of missing values in excluded hedonics (see Appendix A1 and A2), the analysis was constrained in this regard.

Moreover, instrument variables to account for endogeneity and control variables to account for structural differences in the data should be discussed. First, control variables did not prove to yield more conceptual robustness as there are no significant differences in entity types, regions, or time to control for. Second, a potential endogeneity problem was not further investigated as this paper focuses on differences in predictive power among linear and ML models. The implementation of an instrument variable would be beyond the scope of this paper. However, it was assured that all models are based on identical model specifications, meaning the integration and scaling of hedonics.

Lastly, during the hedonic regression estimation, the data was controlled for spatial autocorrelation effects as the data provided spatial characteristics. As the data concerned single house price observations the data had to be aggregated to avoid unstable weight matrices. However, aggregation on ZIP level or county level is both deemed to be unprecise and impractical. The H3 open-source library provided by UBER (H3 Library) provides a pragmatic solution by netting hexagons over spatial data. More importantly, the size of hexagons can be defined. Aggregating 562’983 observations in 2016 into 114 hexagons proved to have significant spatial autocorrelation measured by Moran’s I (0.1180057, p-value <0.0001). While the spital coefficient was significant on the 1% level, indicating that price increase between areas has spillover and feedback effects, aggregating and averaging house prices and hedonics to these areas resulted in a severe loss of predictive power of aggregated hedonics for a 6-nearest neighbor weighted SAR model (see SAR model results in Appendix A17). Moreover, a trade-off between the granularity of the hexagons and the resulting instability of the weight matrix posed a problem to estimate ML-algorithms effectively. Hence, spatial autoregressive models were not further analyzed. Below the hexagons are plotted indicating higher concentration (hexagons become red) of observations in the Long Beach/Orange County area.

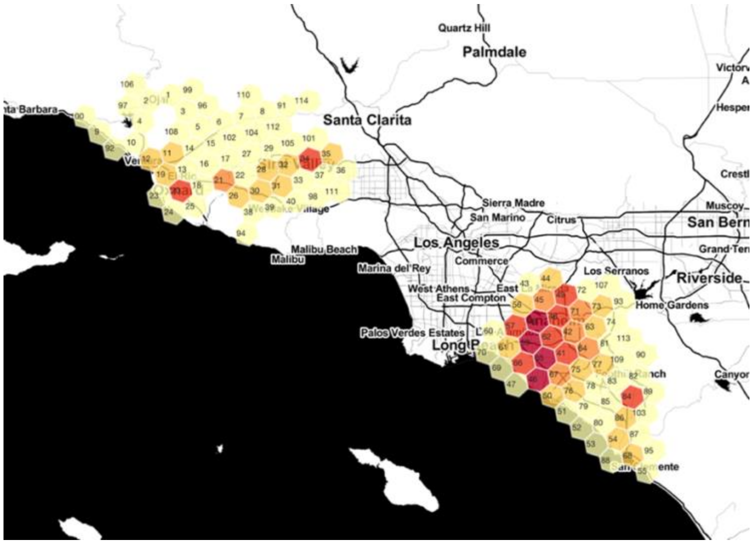


Figure 3: Visualization of House Locations (H3 Hexagons)

4.3 Machine Learning Prediction

Each algorithm has its own specific set of parameters. To ensure optimal predictions by the applied ML methods, the parameters were tuned using a fixed or a random grid search. By leveraging *n*-fold cross-validation various or all combinations of the parameters were tested and the best among them, the ones minimizing the *root mean squared error*, were used for the prediction of the test subsample of the initial dataset. Ideally, all possible combinations of parameters should be tested in a "brute-force" procedure, but this becomes quickly computationally intense, so the range of the hyperparameters had to be limited. These tuned parameters were finally used for specifying the base learners in the stacked generalization model.

The results of the considered models are as follows:

Table 3: Comparison of Prediction Performance

Performance of Different Algorithms in Predicting Building Values

	Root Mean Square Error		*Adjusted R^2*			
Algorithms	Train	Test	Train	Test	*ø Prediction Error*	*p-value*
Linear Regression	*0.4010*	*0.4001*	*0.6587*	*0.6607*	*-0.00051*	*0.632*
Bagging	*0.3785*	*0.3788*	*0.6963*	*0.6953*	*0.00088*	*0.385*
Random Forest	*0.2688*	*0.3600*	*0.8467*	*0.7254*	*-0.00345*	*0.0003 ****
XGBoost	*0.3238*	*0.3572*	*0.7776*	*0.7296*	*-0.00087*	*0.358*
Stacked Generalization	*0.3113*	*0.3529*	*0.7945*	*0.7361*	*-0.00218*	*0.02 ***

Looking at the results from Table 3, we can first and foremost see that each model slightly outperforms the other sequentially with increasing sophistication (or complexity) in the respective ML algorithm. Stacked Generalization remains to be both for the training and test data and on both metrics the best predictor for housing value. The optimized weighting of the considered ML algorithms was (i) 0% of the linear model, (ii) ~69% of the Random Forest, (iii) ~31% of the XGBoost, and (iv) 0% of the bagging predictor. As mentioned in Chapter 3.2.4 Stacked Generalization should perform better or equally as the other learners, as it simply finds an optimized weighting across the predictions of the base learners thus leveraging on predictive of each model.

This insight can be confirmed with the given dataset. A 12% decrease in the RMSE for the test data in comparison to the simple linear prediction, while an increase of around 11.4% in R^2 was achieved on the test data again in comparison to the simple linear prediction. However, the t-test of the prediction error shows that on the 5% significance level the Null-Hypothesis of having a zero mean in the average prediction error (actual minus predicted value) can be rejected. This is due to the high deviations in the prediction in low or high extreme regions of house prices causing a high t-value or low p-value respectively. The second-best prediction algorithm is XGBoost.

While predictive measures are just slightly lagging behind the Stacked Generalization method, it withstands the t-test on the *mean prediction error*.

As the base prediction model, the linear Regression performs rather well for its simplicity and can explain 66% of the out-of-sample values. However, as it cannot account for non-linearities, its predictive power underperforms any other ML Model – led by Stacked Generalization with a difference of 11.4%.

Besides evaluating predictive performance, another important issue is purely focused on differences in adjusted R^2 between the training and testing set. Both linear prediction and bagging have low variance (~0.01) between the training and testing dataset. In contrast, XGBoost with an absolute of ~0.05, Stacked Generalization with an absolute of ~0.06, and Random Forests having the largest difference with an absolute of ~0.12 report all rather high variance. It can be inferred that Random Forests tends to overfit and adapt to the noise instead of generalizing the data in this empirical analysis. Bagging, on the other hand, has its strength in robustness as it aims to reduce the variance of estimators by averaging the redrawn samples with replacement.

Following the feature importance in the models are compared in Table 4. For each algorithm except Stacked Generalization, the contribution of a feature to the final prediction can be computed. This is thanks to the tree-based structures of Bagging, Random Forest, and XGBoost, in which the frequency of a variable across trees or its contribution to R^2 is calculated. Stacked Generalization, on the other hand, is harder to interpret through the combination of multiple algorithms and thus has not been included in this analysis.

Table 4: Feature Importance

	Linear Model	*Bagging*	*Random Forest*	*XGBoost*
Measurement Method	Part*	Frequency**	Permutation***	Gain****
Top 1	*Living Area*	*Living Area*	*Living Area*	*Living Area*
Top 2	*No. Bathrooms*	*Age*	*Age*	*Age*
Top 3	*No. Pool*	*No. Bathroom*	*No. Bathroom*	*Area Garage*
Top 4	*Lot Area*	*No. Story*	*Area Lot*	*Area Lot*
Top 5	*Dummy Tub or Spa*	*Area Lot*	*No. Bedroom*	*No. Bathroom*

**Part: measures the unique contribution of independent variables to R^2*

***Frequency: measures the frequency of a feature across all trees.*

****Permutation: measures the decrease in a model score when a single feature value is randomly shuffled, which breaks the relationship between the feature and the true outcome.*

*****Gain: represents the fractional contribution of each feature to the model based on the total gain of this feature's splits.*

Both linear prediction and machine learning models find *Living Area* to best explain the house price. The high emphasis of all models on *Living Area* can also be shown by a simple scatterplot (see Appendix A16). There seems to be a sound positive linear relationship between the *log* House

Price and the *log* Living Area. Furthermore, for the linear model, this importance is also supported by the correlation contribution analysis in Appendix A13.

However, next to the most important predictor ML algorithms favored *log Age* over *No. Bathroom* selected by the linear model. More interestingly, XGBoost amongst no other ML algorithm uses *Garage Area* to predict. This has to be considered in the evaluation, as *Garage Area* appears to be a variance inflating factor. As mentioned in Chapter 2, there are multiple possibilities in combining the features to produce a similar R^2. Overall, all ML-methods show high similarity in the used features for predictions. This is not surprising as all tested ML algorithms revolve around tree-based models. Hence, it may also explain their somewhat similar predictive power.

Moreover, a note from the importance feature plots (see Appendix A15) we see how a few selected features largely account for the predictive power of the algorithm. The additional predictive power of the *No. Bedrooms* or *No. Story* is small compared to the five most important variables.

While the importance feature plots help in assessing the contribution of a variable to the prediction, we cannot infer economic importance or significance as is done in a linear regression context. However, this feature of ML algorithms could further help to adjust hedonic models, in general, to become more parsimonious.

Finally, while the results report relatively high predictive quality of the algorithms in terms of adjusted R^2, plotting the mean, median, and interquartile ranges provide a different picture (see figure 4). It can be seen that all algorithms contain many outliers, even after the data cleaning process applied in this paper. In another view (see Appendix A14), plotting the predicted vs. actual results, the bottleneck can be identified clearer. All algorithms predict poorly in extreme ranges of housing prices (both low and high). This may be in part due to misleading hedonics, missing variables explaining the difference, or simply an outlier event. One method to counter this issue is to separately analyze and build models for the extreme values.

Overall, considering the only slight difference in predictive power, the advantage in computational effectiveness, and its non-significant t-test of the mean prediction error, XGBoost is the most favorable candidate as an algorithm.

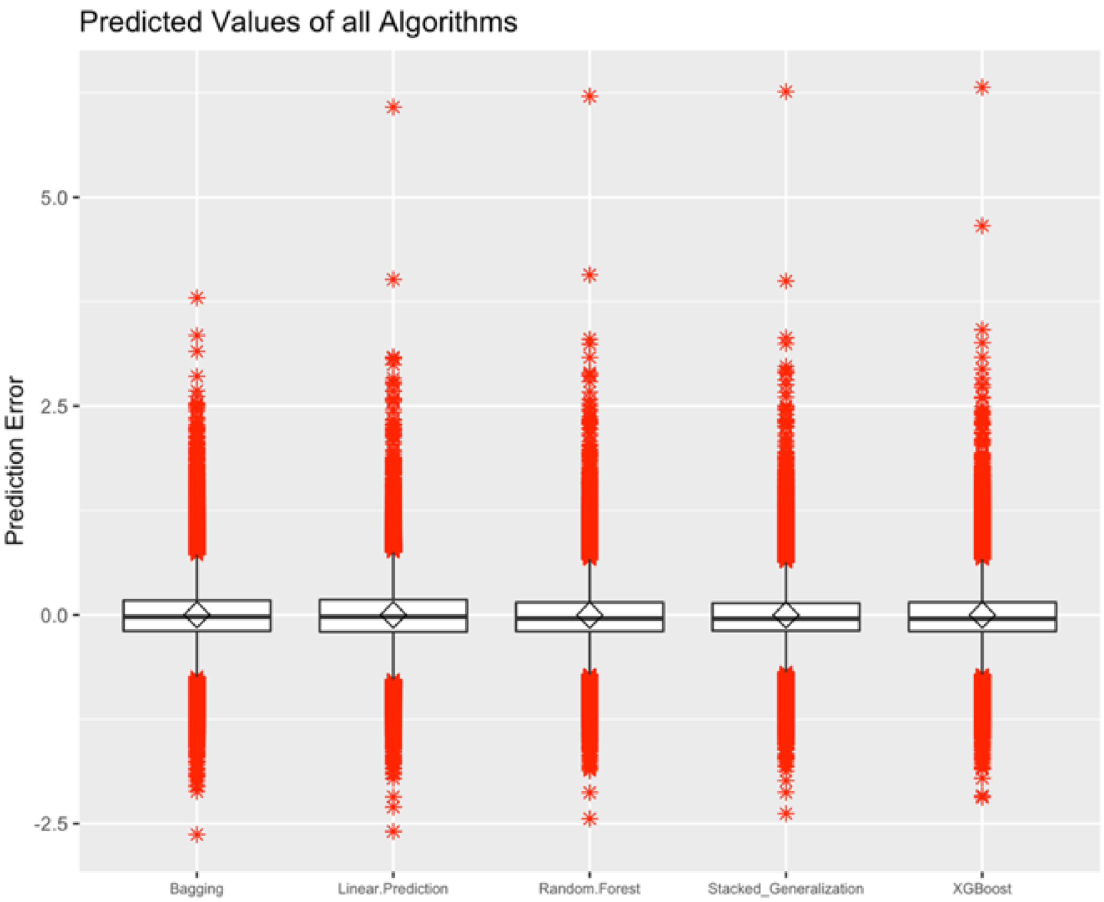


Figure 4: Boxplot of prediction errors

5. Conclusion

Overall, the findings show that machine-learning methods can provide significant outperformance compared to linear regression analysis. All of the tested machine-learning methods have provided better prediction results in terms of return mean squared error (RMSE) and adjusted R^2. Out of the tested algorithms, XGBoost is found to produce the best out-of-sample prediction with minimal variance in the training and testing dataset. This shows how XGBoost successfully mitigates to overfit the training dataset thanks to multiple regularization techniques.

The combination of hedonic pricing and machine-learning models has proven to be of value in the context of prediction. Linear Regression was used for selecting relevant variables, which then served as a foundation for training the machine learning algorithms. The obtained hedonics have been reported significant predictive power in the housing value. In particular, the *log Living Area* in square feet provided the best proxy to predict the dependent variable, closely followed by the *log Age* of the building structure.

While machine learning can provide improved predictive results, it comes with a disadvantage in its interpretability. In comparison to linear regression, the economic importance and significance cannot be easily inferred with machine learning models. Hence, there exists a tradeoff between prediction and inference.

In general, this study contributes to the usage of machine-learning in predicting US housing values. This analysis has narrowed the application to a housing dataset in Los Angeles LA, and the assessed building value for taxes. To transfer the results of this research into a more general context, a national housing dataset and various data publishers should be analyzed. The methods and produced results cannot be simply inferred in an international context, as other markets might lay importance on different hedonics. Moreover, with an increased size of the dataset, the calculations, and tuning of the algorithm hyperparameters present various computational challenges which need to be considered.

An outlook to further research could go in ML algorithm application in spatial regression models focusing on house price prediction. While there are already applications of spatial models in ML, the application in Real Estate Pricing is somewhat sparse. Recent attempts by Kiely & Bastian (2020) and Zhang, Zhang & Miller (2021) show that there are potentially interesting applications and insights yet to be researched.

Literature

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

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*(2), 197-227.

Breiman, L. (1996). Stacked regressions. *Machine learning*, *24*(1), 49-64.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Čeh, M., Kilibarda, M., Lisec, A., & Bajat, B. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. *ISPRS international journal of geo-information*, *7*(5), 168

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

Ethem, A. (2010). Introduction to Machine Learning (2nd. ed.). The MIT Press.

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

Kiely, T. & Bastian, N. (2020). The spatially conscious machine learning model. Stat Anal Data Min: The ASA Data Sci Journal, 13: 31– 49. https://doi.org/10.1002/sam.11440

Lantz, B. (2013). Machine Learning with R. Packt Publishing Ltd.

Lu, S., Li, Z., Qin, Z., Yang, X., & Goh, R. S. M. (2017, December). A hybrid regression technique for house prices prediction. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 319-323). IEEE.

Mullainathan, S. & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. Journal of Economic Perspectives, 31 (2), 87-106.

Naimi, A. I., & Balzer, L. B. (2018). Stacked generalization: an introduction to super learning. *European journal of epidemiology*, *33*(5), 459-464.

Nielsen, D. (2016). *Tree boosting with xgboost-why does xgboost win" every" machine learning competition?* (Master's thesis, NTNU).

OECD et al. (2013). Handbook on Residential Property Price Indices. Eurostat, Luxembourg.

Oladunni, T., & Sharma, S. (2016, December). Hedonic housing theory—a machine learning investigation. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 522-527). IEEE

Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012, July). How many trees in a random forest? In *International workshop on machine learning and data mining in pattern recognition* (pp. 154-168). Springer, Berlin, Heidelberg

Ottensmann, J., Payton, S. & Man, J. (2008). Urban Location and Housing Prices within a Hedonic Model. The Journal of Regional Analysis & Policy, 38 (1), 19-35.

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert systems with applications*, *42*(6), 2928-2934.

Pérez-Rave, J., Correa-Morales, J. & González-Echavarría, F. (2019). A machine learning approach to big data regression analysis of real estate prices for inferential and predictive purposes. Journal of Property Research, 36:1, 59-96, DOI: 10.1080/09599916.2019.1587489

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

Shahhosseini, M., Hu, G., & Pham, H. (2019, June). Optimizing ensemble weights for machine learning models: a case study for housing price prediction. In *INFORMS International Conference on Service Science* (pp. 87-97). Springer, Cham.

Sirmans, S., Macpherson, D. & Zietz, E. (2005). The Composition of Hedonic Pricing Models. Journal of Real Estate Literature, 13 (1), 3-43.

Ting, K. M., & Witten, I. H. (1997). Stacked Generalization: when does it work?

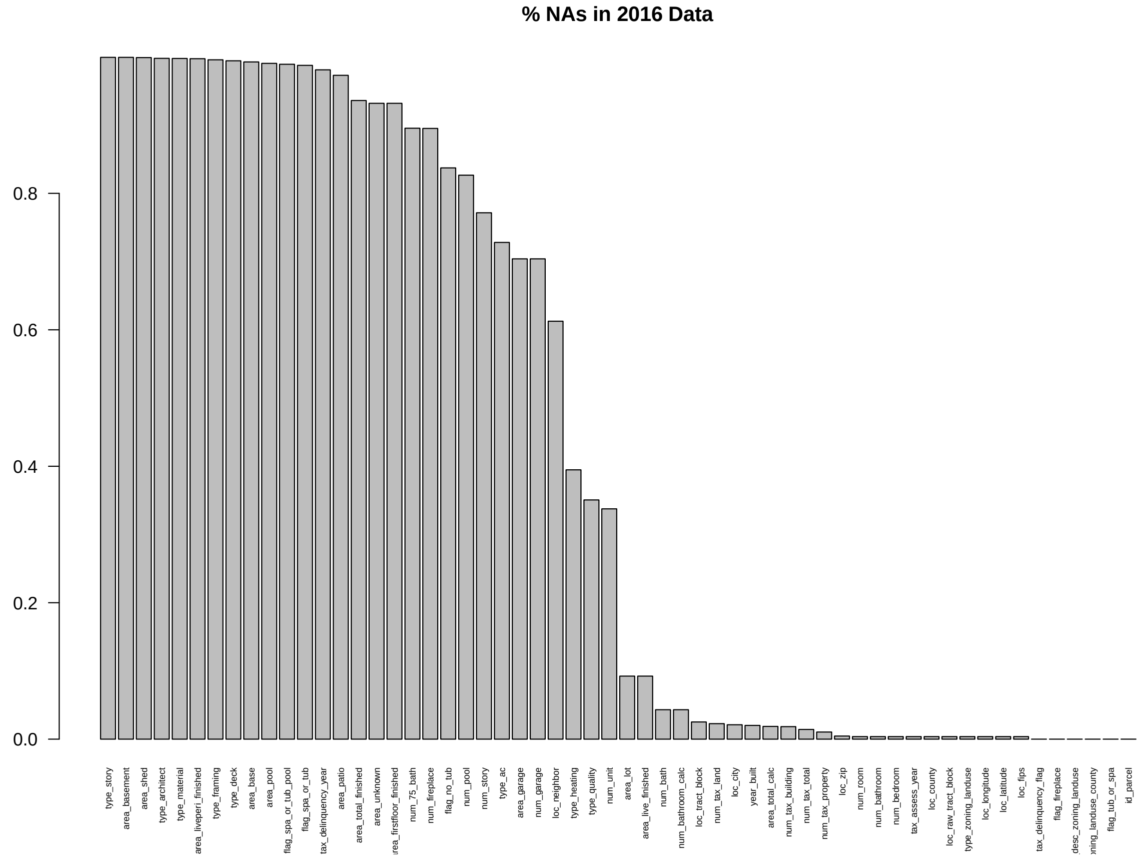
Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical applications in genetics and molecular biology*, *6*(1).

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241-259.

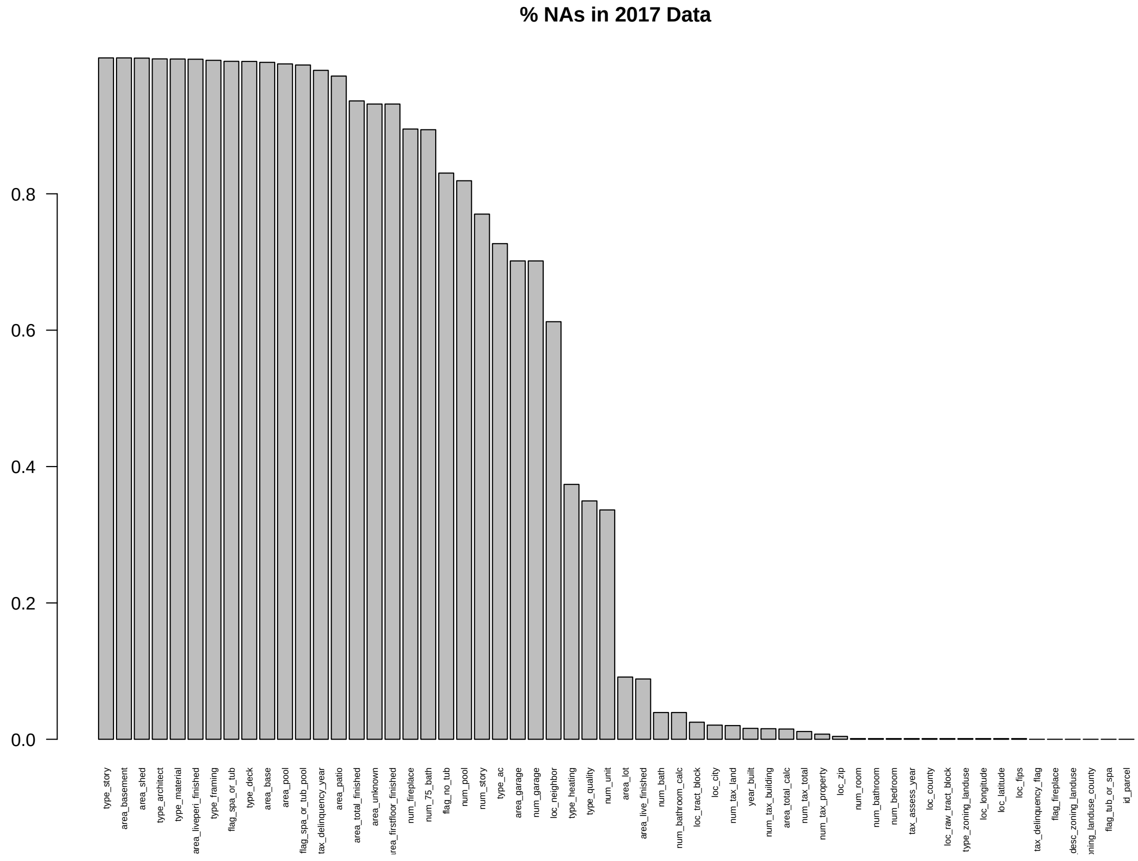
Zhang, Y., Zhang, D. & Miller, E. (2021). Spatial Autoregressive Analysis and Modeling of Housing Prices in City of Toronto. Journal of Urban Planning and Development, 147 (1).

Appendix

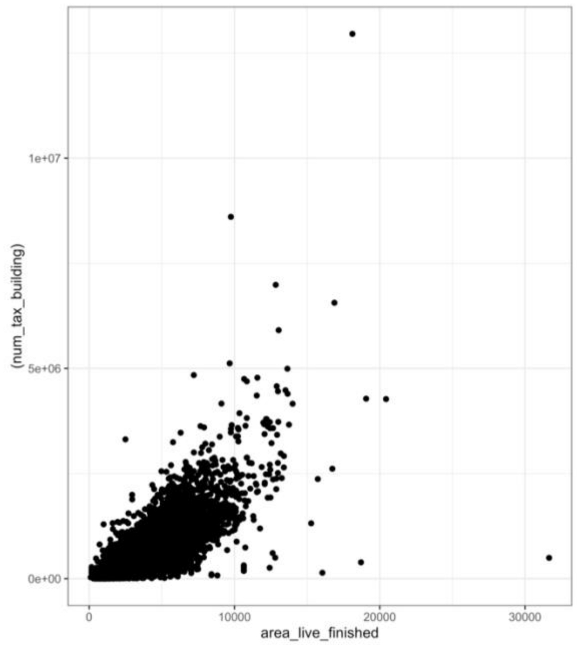
A1: %NAs in 2016 Dataset



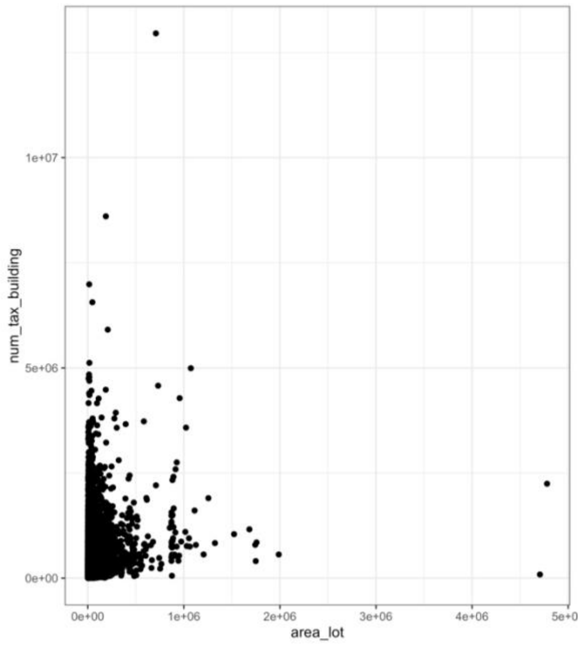
A2: %NAs in 2017 Dataset



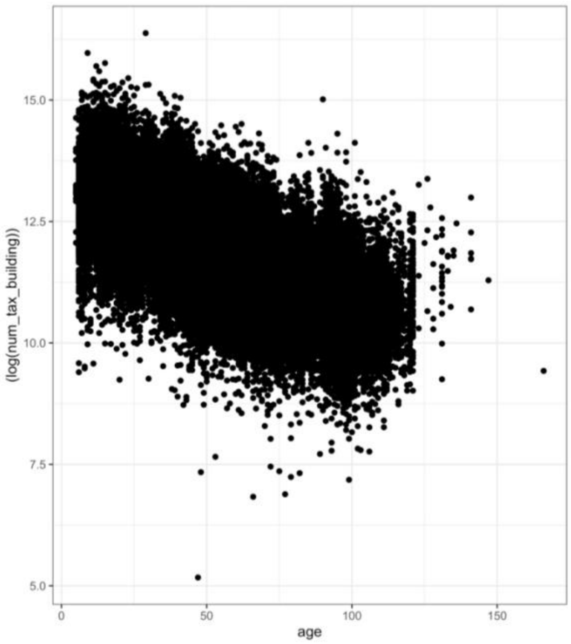
A3: 2016 House Price vs Living Area



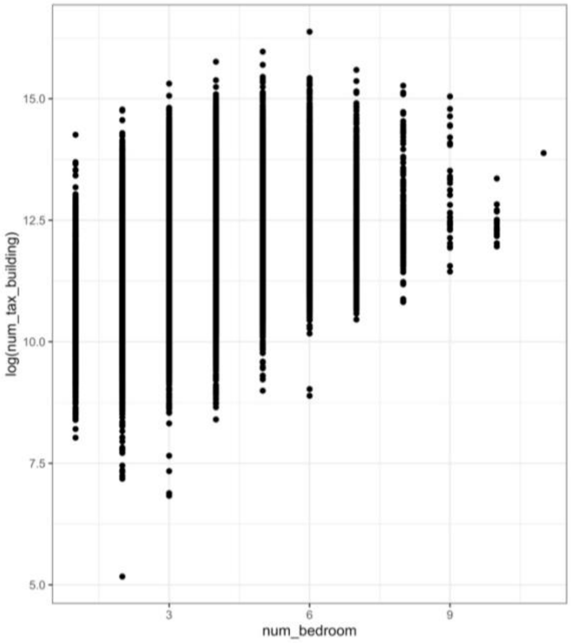
A4: 2016 House Price vs Area Lot



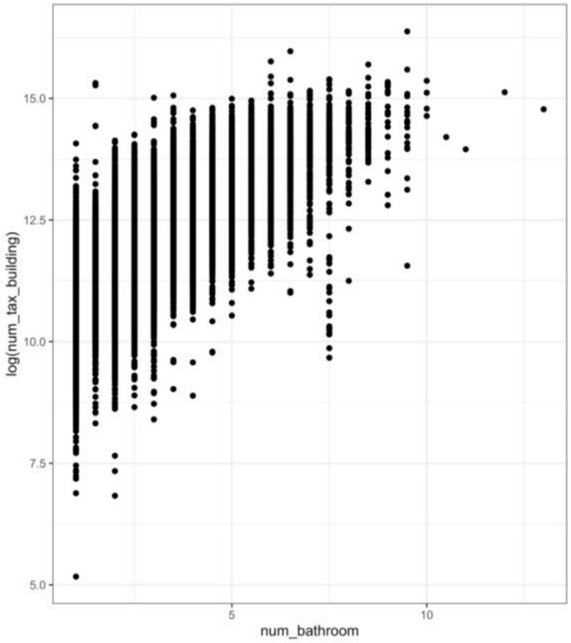
A5: 2016 log House Price vs Age



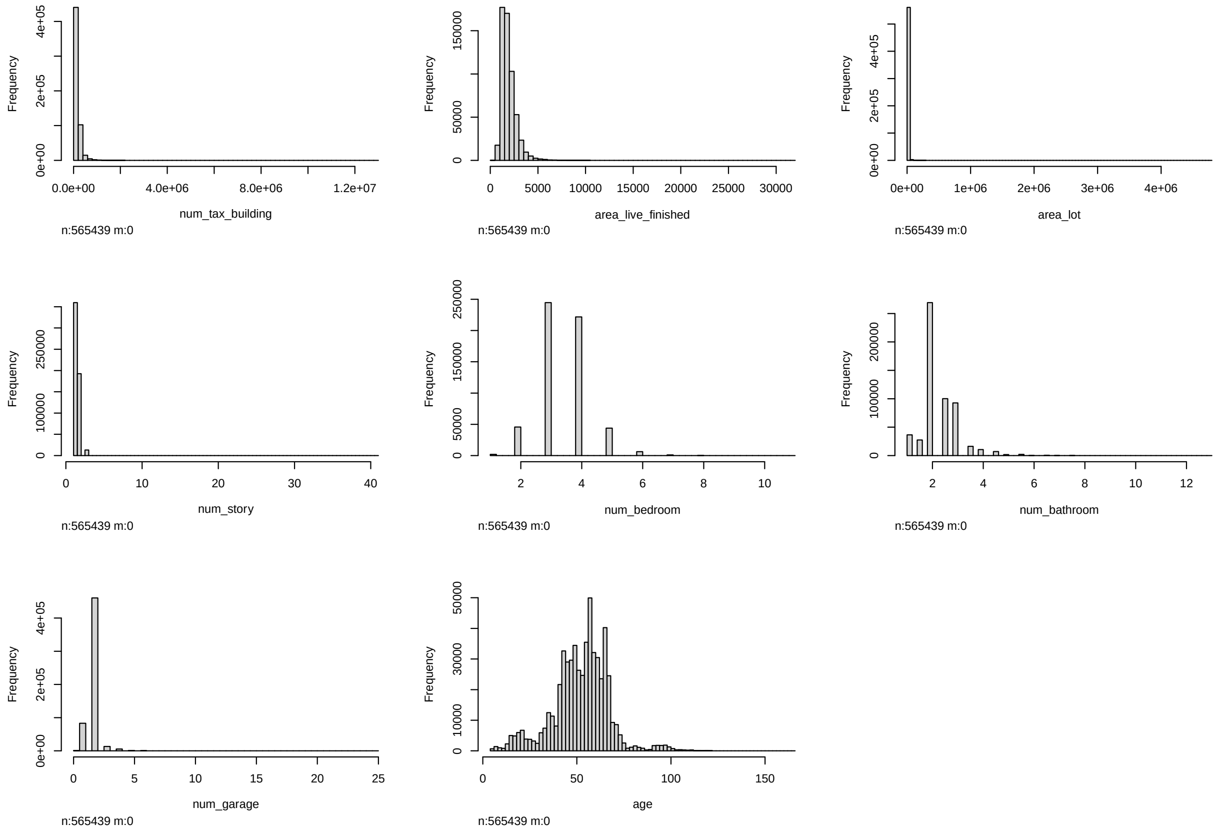
A6: 2016 log House Price vs No. Bedrooms



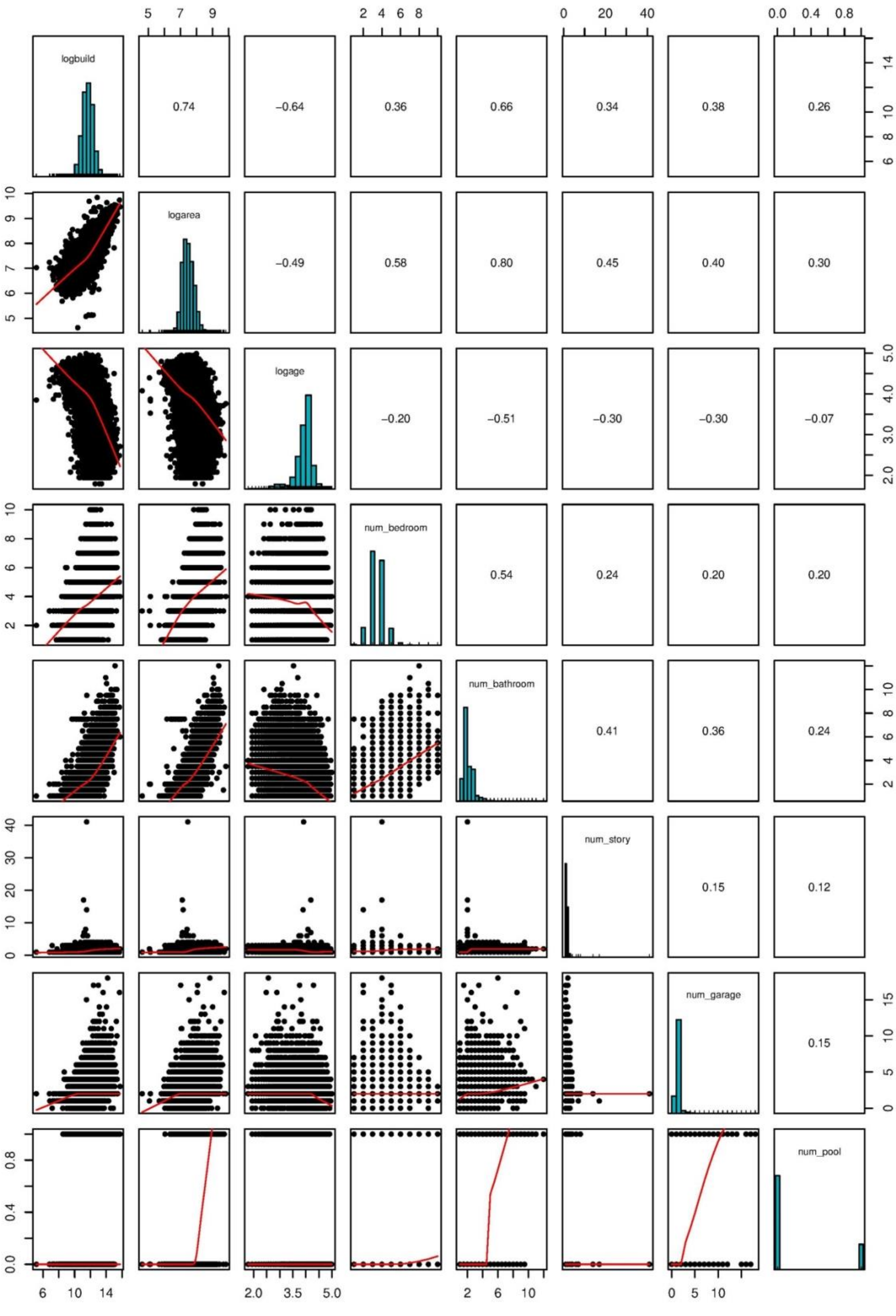
A7: 2016 log House Price vs No. Bathrooms



A8: 2016 Histogram of a Selection of Hedonics



A9: 2016 Correlogram of a Selection of Hedonics



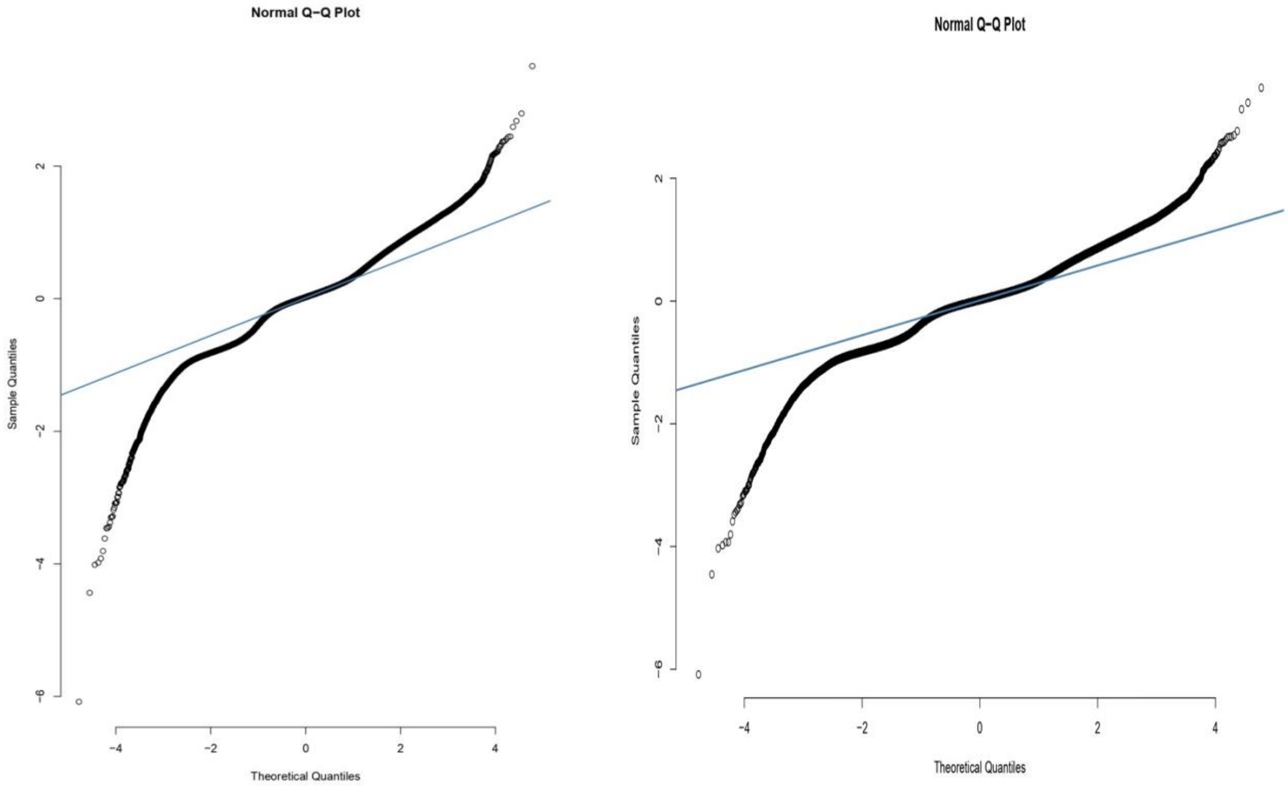
A10: Descriptive Statistics of 2016 Data

		Mean	Std. Deviation	Minimum	Maximum	Range
House Price		148732	132372.43	174	6883066	6882892
	log	11.66	0.69	5.16	15.74	10.59
Living Area (sqft)		1901.37	767.7	102	18708	18606
	log	7.48	0.36	4.62	9.84	5.21
Lot Area		8133.31	6680.41	104	99785	99681
	log	8.84	0.53	4.64	11.51	6.87
	%Building	0.3	0.22	0.01	29.71	29.7
Garage Area		478.4	131.37	0	5596	5596
	log	0.61	0.28	0	2.89	2.89
Age		52.81	14.83	6	147	141
	log	3.92	0.34	1.79	4.99	3.2
No. Bedroom		3.5	0.82	1	10	9
No. Bathroom		2.3	0.73	1	12	11
No. Story		1.39	0.54	1	41	40
No. Garage		1.9	0.5	0	18	18
Area Garage (sqft)		478.4	131.37	0	5596	5596
No. Pool		0.21	0.41	0	1	1
Fireplace		0	0.07	0	1	1
Tub or Spa		0.04	0.2	0	1	1

A11: 2016/17 Variance Inflation Factors a Multicollinearity Measure

	2016		2017	
Variables	Tolerance	VIF	Tolerance	VIF
log Living Area	0.2466055	4.05506	0.2465597	4.055814
log Lot Area	0.6604012	1.514231	0.6599125	1.515352
log Garage Area	0.1035266	9.659353	0.1040813	9.607869
log Age	0.6531019	1.531155	0.659583	1.51611
No. Bedrooms	0.6224354	1.606592	0.6225131	1.606392
No. Bathrooms	0.3288962	3.040473	0.3285457	3.043717
No. Story	0.6820231	1.466226	0.6832137	1.463671
No. Garage	0.1035217	9.659807	0.1039472	9.620267
No. Pools	0.8549785	1.16962	0.8530337	1.172287
Dummy Fireplace	0.9743912	1.026282	0.9744276	1.026243
Dummy Tub or Spa	0.9537501	1.048493	0.9512258	1.051275

A12: 2016 (left) & 2017 (right) QQ Plot of Regression Residuals

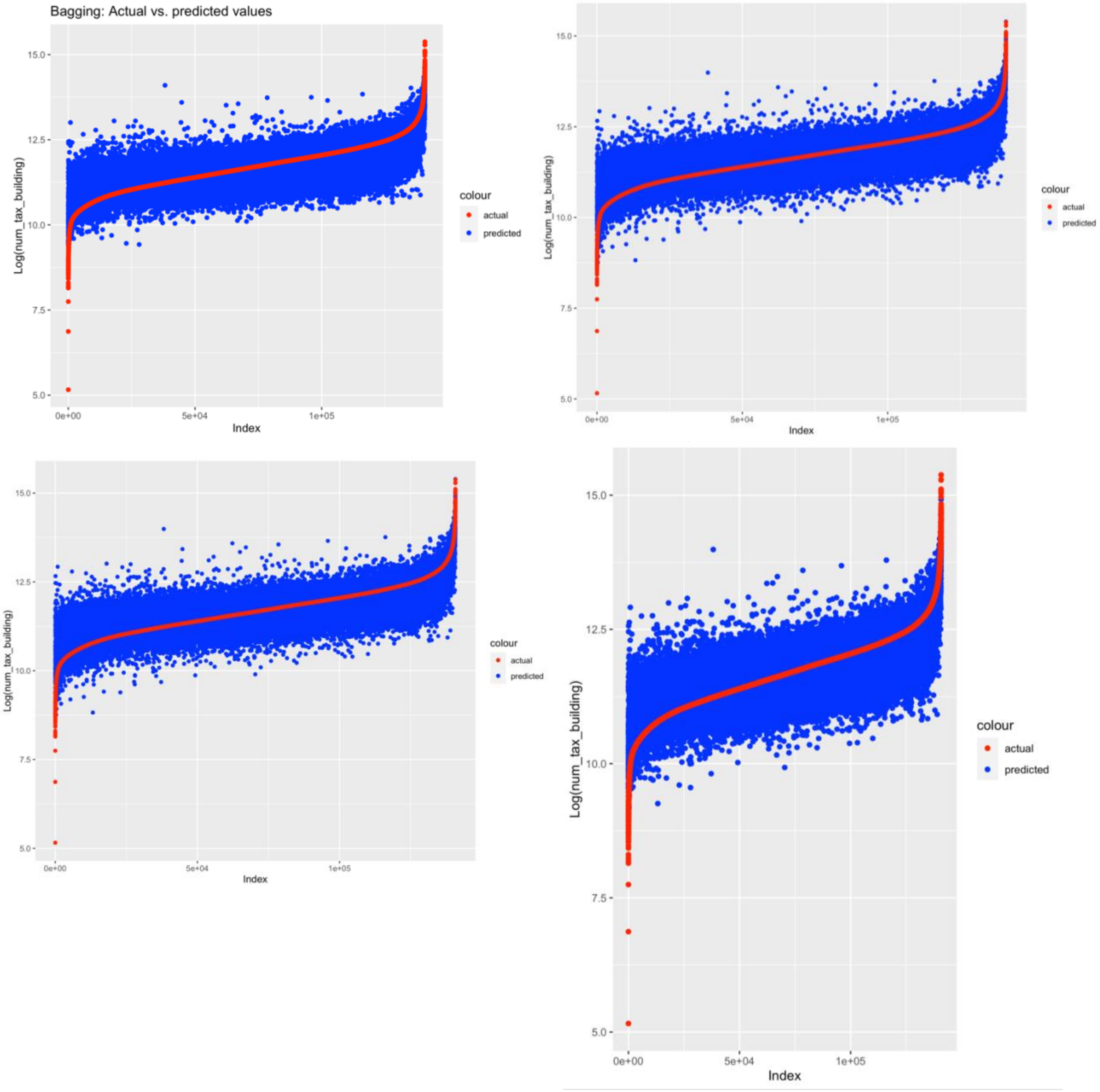


A13: 2016/17 Correlation Contribution of Regression Coefficients

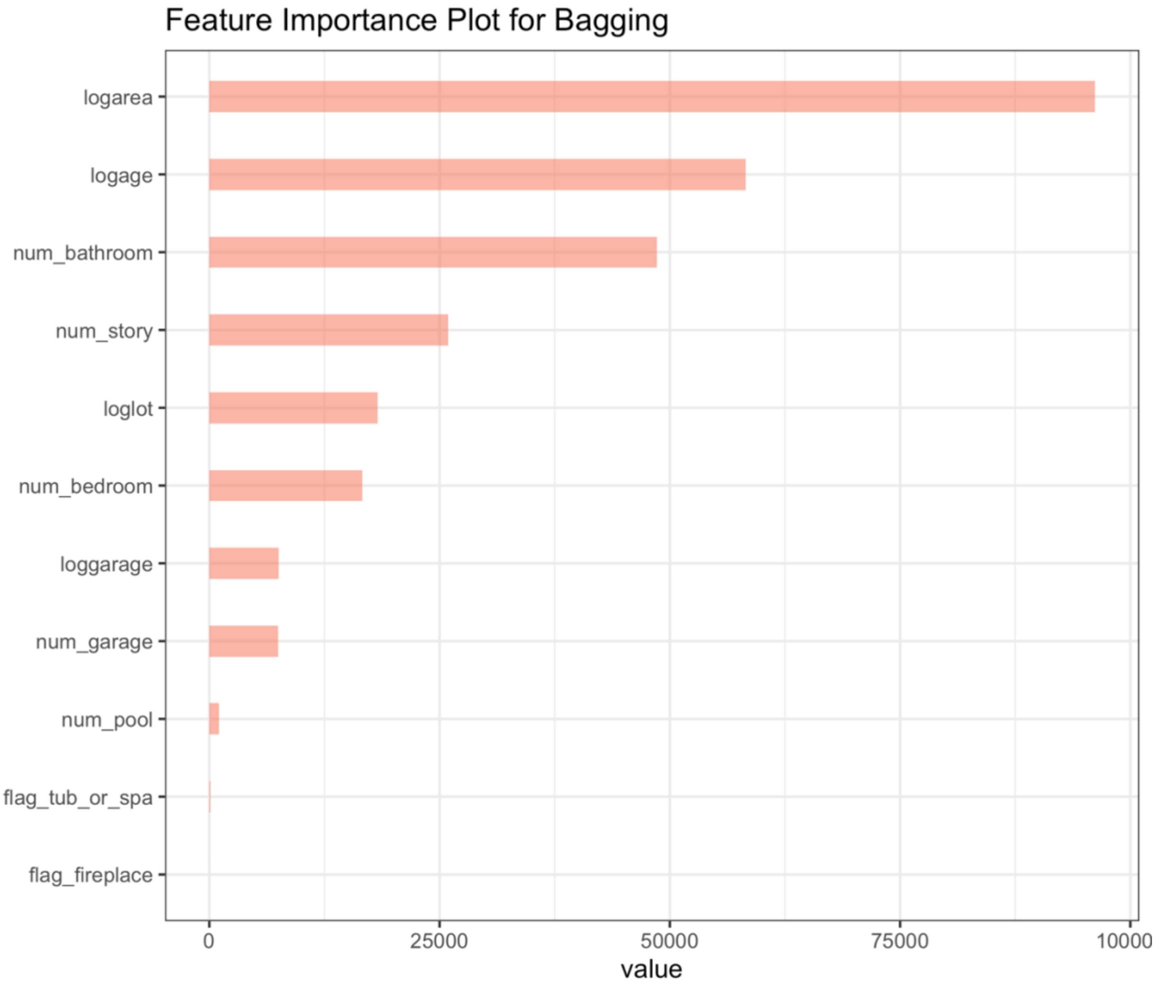
Correlations

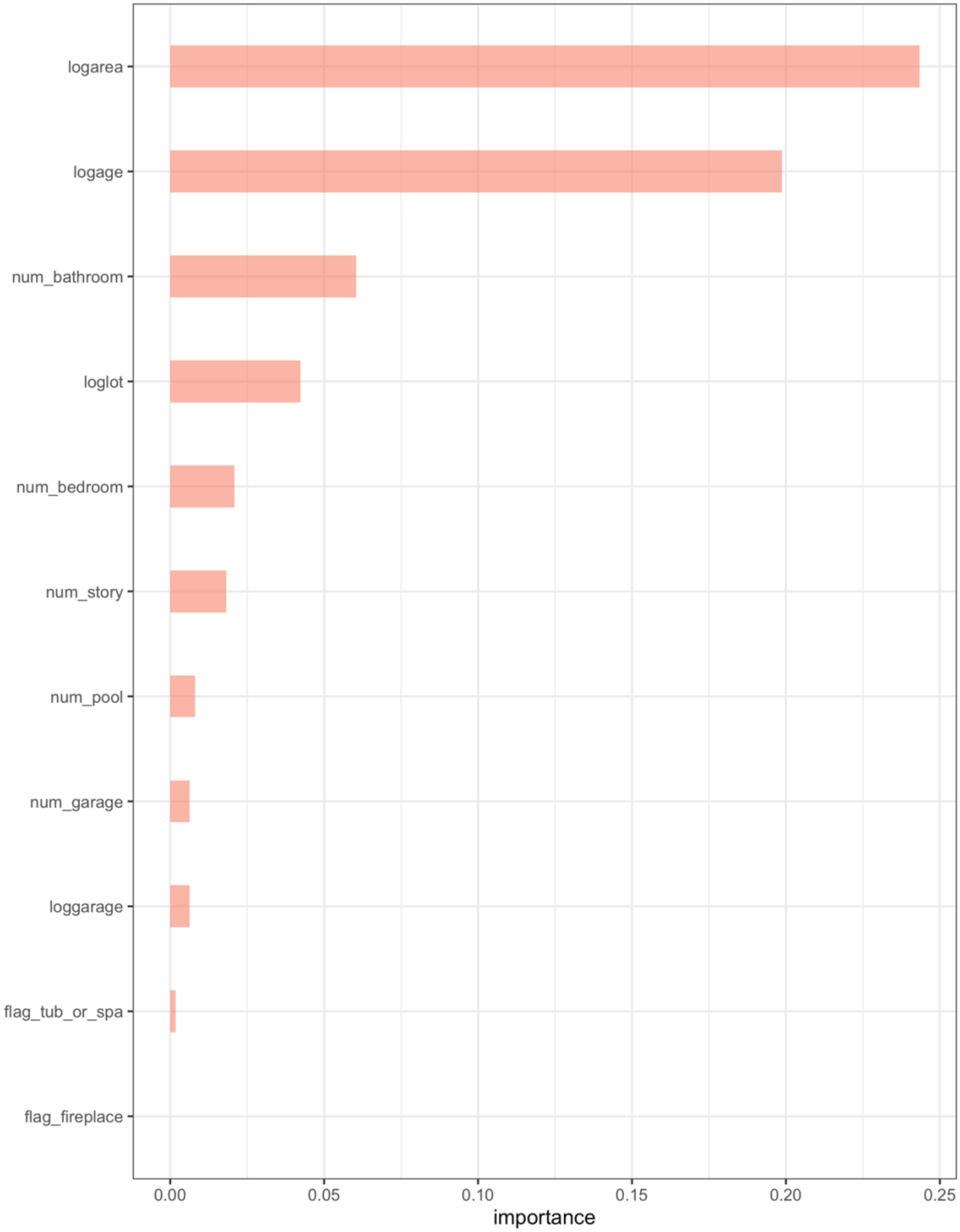
	2016			2017		
Variable	**Zero Order**	**Partial**	**Part**	**Zero Order**	**Partial**	**Part**
log Living Area	0.737	0.366	0.229	0.737	0.369	0.234
log Lot Area	0.269	0.067	0.039	0.272	0.067	0.039
log Garage Area	0.394	0.025	0.015	0.392	0.027	0.016
log Age	-0.638	-0.436	-0.283	-0.628	-0.423	-0.274
No. Bedrooms	0.362	-0.093	-0.054	0.361	-0.098	-0.058
No. Bathrooms	0.66	0.111	0.065	0.66	0.112	0.066
No. Story	0.335	-0.031	-0.018	0.333	-0.031	-0.018
No. Garage	0.384	0.001	0.001	0.383	0.001	0
No. Pools	0.263	0.102	0.06	0.262	0.098	0.058
Dummy Fireplace	-0.009	0.002	0.001	-0.009	0.002	0.001
Dummy Tub or Spa	0.151	0.046	0.027	0.154	0.046	0.027

A14: Actual vs. Predicted plots for Bagging (1), Random Forest (2), XGBoost (3), and Stacked Generalization (4)

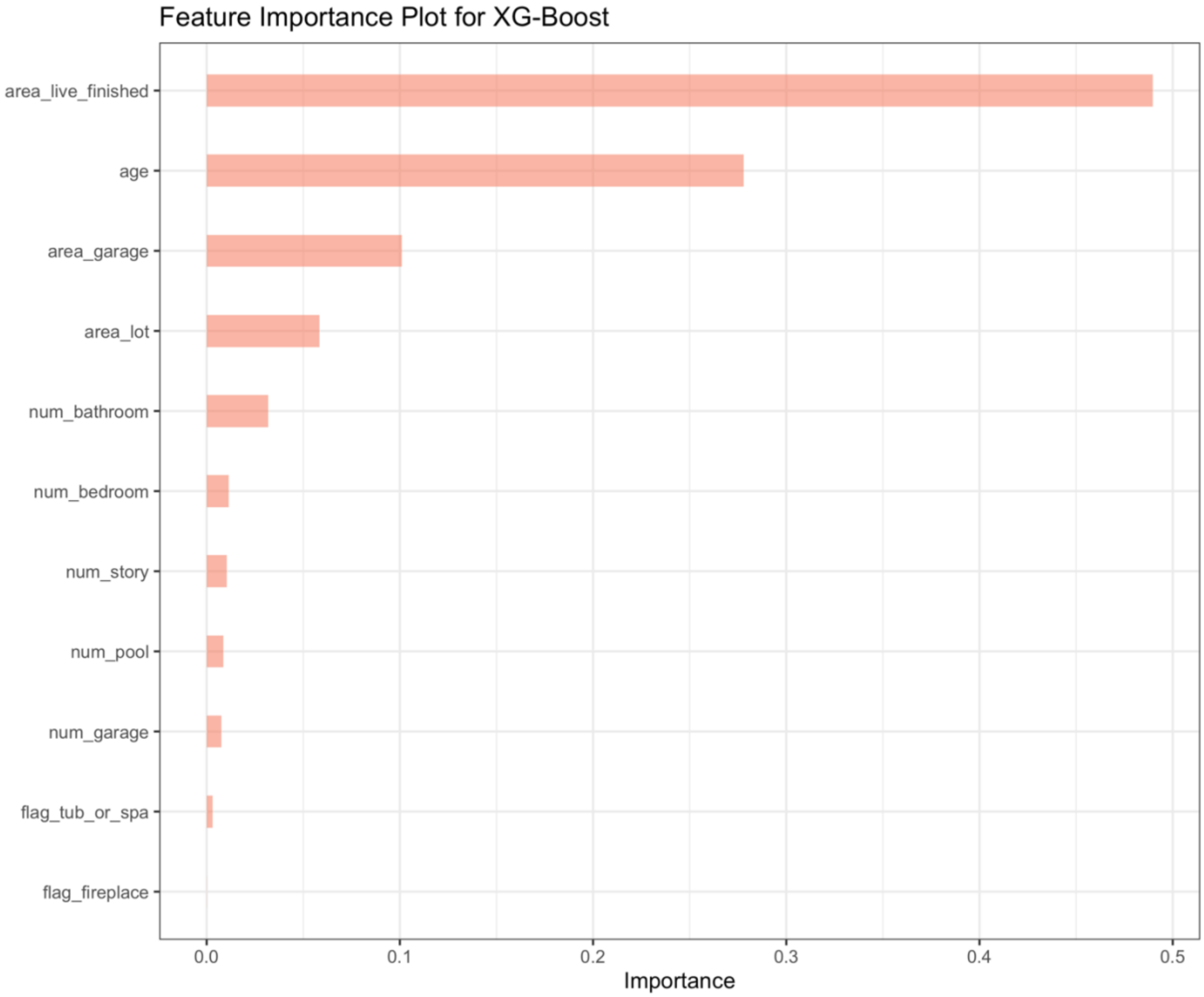


A15: Feature Importance Plot for Bagging (1), Random Forest (2), XGBoost (3)

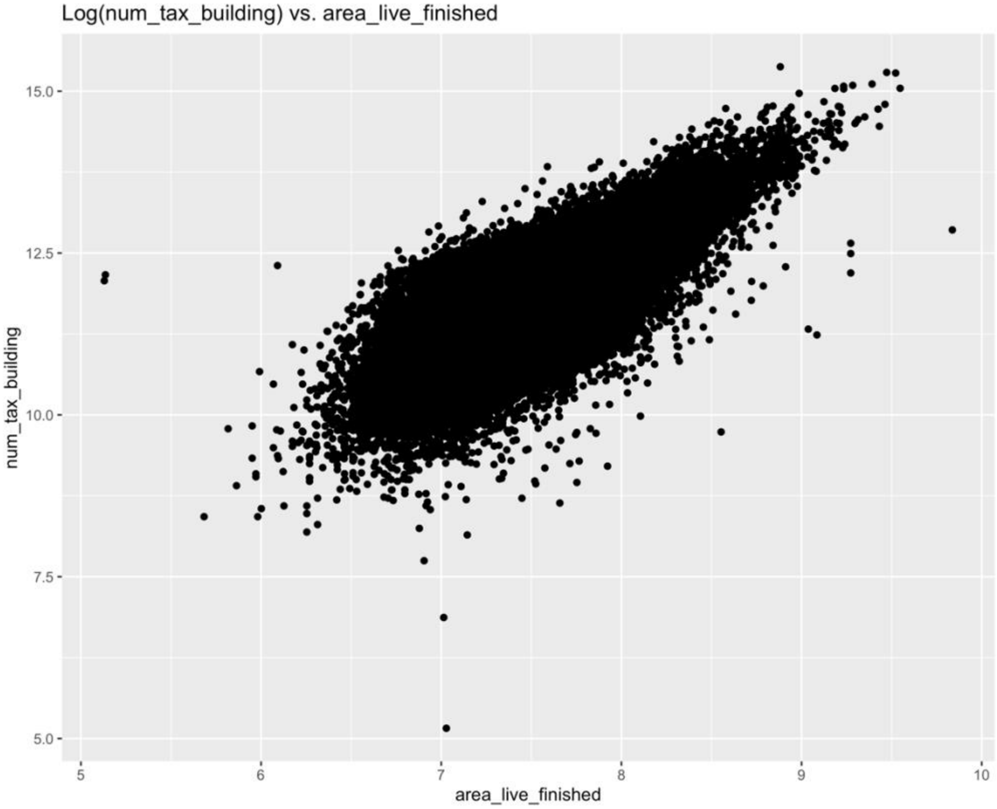


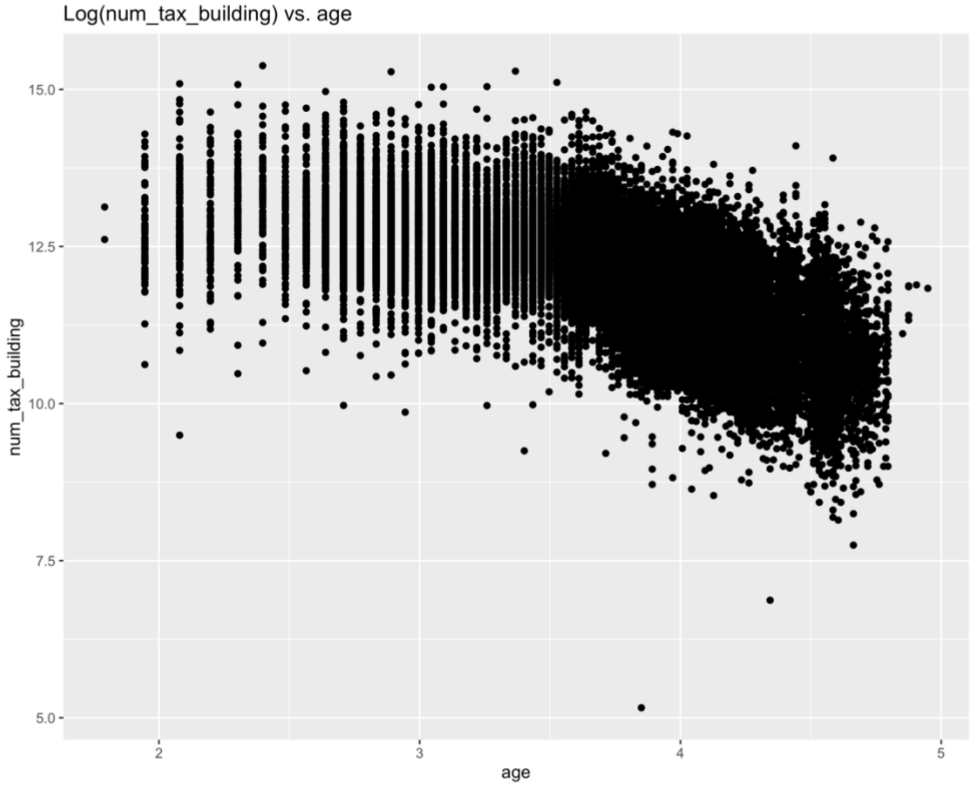


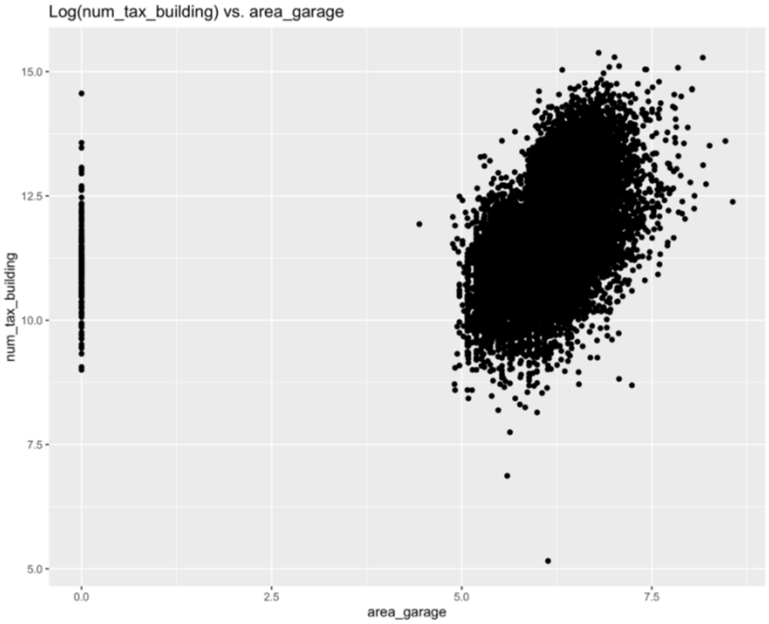
Feature Importance Plot for Random Forest


Feature Importance Plot for XG-Boost
area_live_finished
age
area_garage
area_lot
num_bathroom
num_bedroom
num_story
num_pool
num_garage
flag_tub_or_spa
flag_fireplace
0.0
0.1
0.2
0.3
0.4
0.5
Importance

A16: Scatterplots for Living Area, Age, Area Garage







Appendix 17: SAR Model and linear Hedonic model in comparison

	Area Regression 2016		SAR Model 2016	
Variables	Estimate	Std. Error†	Estimate	Std. Error†
Intercept	*5.332 ***	*-1.834*	*0.8435*	*1.3034*
log Living Area	*1.209 ****	*-0.305*	*1.0668*	*0.2095*
log Lot Area	*-0.053*	*-0.043*	*-0.0424*	*0.0286*
log Garage Area	*-0.804*	*-0.666*	*-1.0032*	*0.4481*
log Age	*-0.713 ****	*-0.157*	*-0.5296*	*0.1012*
No. Bedrooms	*-0.04*	*-0.07*	*0.0054*	*0.0606*
No. Bathrooms	*-0.099*	*-0.171*	*-0.0291*	*0.1034*
No. Story	*-0.029*	*-0.128*	*-0.0709*	*0.0991*
No. Garage	*0.830 **	*-0.352*	*0.8533*	*0.2273*
No. Pools	*-0.209*	*-0.17*	*-0.3079*	*0.1434*
Dummy Fireplace	*-2.631*	*-1.964*	*-1.3874*	*1.5087*
Dummy Tub or Spa	*-0.647*	*-0.354*	*-0.1124*	*0.3041*
N	*114*		*114*	
Rho			*0.38571 ****	
log Likelihood	*42.54*		*66.75*	
AIC	*-59.1*		*-105.5*	

**** p < 0.001; ** p < 0.01; * p < 0.05; †Robust Standard*

Declaration of Authorship

We hereby declare

- that we have written this thesis without any help from others and without the use of documents or aids other than those stated above;
- that we have mentioned all the sources used and that we have cited them correctly according to established academic citation rules;
- that we have acquired any immaterial rights to materials we may have used, such as images or graphs, or that we have produced such materials ourselves;
- that the topic or parts of it are not already the object of any work or examination of another course unless this has been explicitly agreed to with the faculty member in advance and is referred to in the thesis;
- that we will not pass on copies of this work to third parties or publish them without the university's written consent if a direct connection can be established with the University of St. Gallen or its faculty members;
- that we are aware that our work can be electronically checked for plagiarism and that we hereby grant the University of St. Gallen copyright in accordance with the Examination Regulations insofar as this is required for administrative action;
- that we are aware that the university will prosecute any infringement of this declaration of authorship and, in particular, the employment of a ghostwriter, and that any such infringement may result in disciplinary and criminal consequences which may result in our expulsion from the university or us being stripped of our degree.

By uploading this academic term paper, we confirm through our conclusive action that we are submitting the Declaration of Authorship, that we have read and understood it, and that it is true.

T. Graf

Tim Graf

Kilian Gerding

Word Count: 8906, Number of Pages (raw Text): 19, Total Number of Pages: 39