ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

MSc Data Science & Marketing Analytics

Real Estate Modeling:
Traditional or Machine Learning?

Name student: van Es, Julian
Student ID number: 545416
Supervisor: Franses, PHBF
Second assessor: Karpienko, R

Date final version: 07-09-2024

Abstract

In this paper, I explore the prediction of house sale prices using machine learning techniques and Hedonic linear regression. I apply Ordinary Least Squares (OLS) regression, LASSO regression, Principal Component Regression (PCR), Principal Component LASSO Regression (PCLR), the Random Forest (RF), and Extreme Gradient Boosting (XGBoost). I find that OLS, LASSO, and XGBoost are the most accurate models in terms of out-of-sample predictive accuracy, measured by the Root Mean Square Error (RMSE). I combine model predictions and find that if the optimal model is unknown beforehand, combining models reduces the risk of picking bad models: combinations of predictions closely resemble those of the optimal model. Lastly, I compare general relationships between house features and house prices found in the three most accurate models LASSO, OLS, and XGBoost. The sign of relationships generally matches between models, however, each of the three models presents counterintuitive relationships between house features and price where I use coefficients for OLS and LASSO, and Individual Conditional Expectation (ICE) plots for XGBoost to uncover patterns.

Contents

1.0 Introduction

Mass appraising is the practice of estimating the value of many houses at the same time. The use of Machine Learning methods has recently become more popular when mass appraising due to access to large amounts of data and Machine Learning methods scaling well. In this paper, I apply several Machine Learning (ML) methods to predict house prices in Ames, Iowa between 2006 and 2010 using house features. I implement LASSO Regression, and two Ensemble Machine Learning models: the Random Forest (RF) and Extreme Gradient Boosting (XGBoost). Ensemble models use small models, in this paper Regression Trees, to make better predictions in a final model.

I compare the predictive accuracy, measured by the Root Mean Square Error (RMSE) and Root Mean Square Error of Logged Sale Prices ($RMSE_{log}$) of ML models to the more traditional Hedonic Regression Model. The Hedonic Regression Model uses Ordinary Least Squares (OLS) and regresses the price of a house on its features. Furthermore, I apply Principal Component Analysis on size-related features to reduce the number of features in the dataset and include the Principal Component Regression (PCR) and the Principal Component LASSO Regression (PCLR) in the analysis.

Additionally, I combine model predictions to improve upon the best individual predictor model. I apply three methods: Equal Weights, Root Mean Square Error-inverse weights, and Linear Regression weights in a forecast combination model. Equal Weights is the average of predictions whilst the RMSE-Inverse weights assign a weight inversely proportional to every model's RMSE in a Validation dataset. The Linear Regression Weights are determined by regressing the observed Sale Price of houses over each model's predictions of the Validation dataset.

Furthermore, I evaluate how each model describes the relationship between the price of a house and its features. For OLS regression I evaluate coefficient signs and significance. For LASSO I evaluate non-zero coefficient signs. Finally, for XGBoost I apply Individual Conditional Expectation (ICE) plots which graph how each prediction made by a Machine Learning model changes with varying values of one variable of interest.

The main research question is as follows:

To what extent can Machine Learning models replace traditional Hedonic OLS models in predicting house sale prices in terms of accuracy and interpretability?

Correctly handling information is key in the housing market. Banks need to estimate the value of collateral when handing out mortgages and loans. Real estate agents need to estimate the price a potential buyer might be willing to pay for a house. Home buyers want to know what they should realistically bid on the house without overpaying. The estimation of House prices is traditionally done using a Hedonic Model. Hedonic modeling assumes a product's price is the sum of prices of all its characteristics (Rosen, 1974). Similarly, a Hedonic model assumes a house's price is the sum of all the prices of its features. Hedonic modeling is popular within the housing market because it makes use of Ordinary Least Squares regression which yields highly interpretable coefficients. These coefficients estimate the sign, the strength, and the significance of the relationship of a house feature to the price.

Banks, however, need to update the value of collateral more frequently, leading to the use of mass appraisals (Hong et al. 2020). In the context of mass appraisals, Machine Learning models can handle large amounts of data efficiently and successfully: Machine Learning methods can accurately predict house prices (Madhuri et al., 2019 & Lu et al., 2017). Unfortunately, Machine Learning models oftentimes lack interpretability, which has led to several complex ML models being labeled Black Box models: The user can only see the input and output of the model (Kenton, 2024). Methods such as Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) plots have been developed to graph relationships uncovered by a Black Box model. These methods shine a light on how a Black Box model makes its predictions using independent variables. This paper fits into the growing body of literature that displays the utility of Machine Learning models in Real Estate by comparing machine learning models to the traditional linear regression method. In this paper, I evaluate 3 hypotheses:

- Hypothesis 1: The accuracy of Machine Learning models will be superior to that of Hedonic OLS models, measured by out-of-sample *RMSE* and $RMSE_{log}$
- Hypothesis 2: A combination of models improves upon the best individual model's predictions, measured by out-of-sample *RMSE* and $RMSE_{log}$
- Hypothesis 3: Relationships between features and price found in the Hedonic OLS model are more intuitive or resemble the literature more closely than Machine Learning Models.

Data is obtained from Kaggle's *House Prices - Advanced Regression Techniques* competition, prepared by Dean de Cock. The dataset includes the sale prices of 1460 houses in Ames, Iowa between 2006 and 2010. Furthermore, the data has characteristics of each house. Features include size-related characteristics such as the number of rooms, living area on each floor, and lot size, but also features relating to which neighborhood the house is in. The dataset also has variables that describe the quality

of building materials and the condition of the house. I handle missing values in the dataset by setting these to the median found within the Training dataset. Furthermore, I reduce the effect of extreme house prices in the data by setting prices exceeding 500k$ to 500k$.

In this paper, I find that LASSO, XGBoost, and OLS work the best when estimating out-of-sample. The Random Forest, PCR, and PCLR perform considerably worse than these models. Furthermore, I find that combining model predictions leads to predictions that closely resemble the best individual predictor model. I do not find evidence that combining models leads to better predictions, as there is only one occasion where a combination outperforms the best individual model. Lastly, I find that each model suffers from counter-intuitive relationships in terms of what features drive the price of a house.

The rest of this paper is set up as follows: I first provide an overview of the literature regarding modeling for real estate appraisals. I then move on to a description of the dataset. In the methodology section, I discuss the models used, hyperparameter tuning, and evaluation metrics. The last two sections are dedicated to the results and the conclusion.

2.0 Literature Review

Estimating the price of a house can be a challenging task. An appraiser needs to consider influences determining the supply as well as the demand for the house to predict a future sales price.

2.1 House Price Modeling

Lancaster (1966) argues that it is not the good that provides utility to the consumer, but the set of characteristics present in the good. In his framework, each characteristic provides the consumer with an attributable portion of utility which the consumer considers when choosing a product. Rosen (1974) loses Lancaster's (1966) assumption that a good's characteristics are strictly indivisible. Rosen (1974) instead, focuses more on how a good's price is the result of an equilibrium between supply and demand where characteristics are priced in heterogeneous goods.

Hedonic pricing theory uses the idea that a price is attributable to the combination of characteristics. This way of thinking is crucial for the traditional appraisal techniques of real estate. Two identical houses can differ in price because one has a pool, and the other does not. Traditionally, Hedonic real-estate pricing models use Ordinary Least Squares (OLS) regressions. A general Hedonic OLS regression can be seen in Equation (1),

Equation 1: $$P_i = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + e_i$$

Where

- P_i is the Sale Price of house *i*,
- β_0 is the intercept,
- β_j represents the marginal value attributed to house-characteristic *j*,
- *J* is the total number of house characteristics,
- x_{ij} is the observed house-characteristic *j* for house *i,*
- e_i is the error term to house *i.*

One major downside of Hedonic OLS models in real estate is that the model fails to capture people's behavior. Newsome & Zietz (1992) argue that homebuyers who can afford high-priced houses attribute different values to house features than people who can only afford low-priced houses. Dubin (1998) argues that a house's sale price is the result of the perceived value of the buyer and seller as well as the circumstances of the sale: A seller who needs cash may sell a house for less money to facilitate a sale.

Such circumstances can lead to differing sales prices of identical properties despite the house's features being constant. As a result, all Hedonic models are influenced by measurement errors: Deviations from the true value of a house due to people's behavior.

Selim (2009) applies a Hedonic model to evaluate the effect housing characteristics have on house prices in Turkey. For this research, the author uses the 2004 Household Budget Survey Data for Turkey. The author finds that the water system, the type and size of the house, the number of rooms, local characteristics, and the type of building are among the most important features. Furthermore, larger houses and houses that have a pool tend to be more expensive. Lastly, Selim (2009) finds that older houses are generally cheaper. More specifically, the results indicate that on average, houses between 5 and 10 years old are 8% cheaper than those aged 0-5 years. This percentage becomes 12% for rural areas, and 5.8% for urban areas.

Zietz et al. (2008) apply a different approach than the standard OLS regression. The authors use a quantile regression to model the different relationships of low-priced and high(er)-priced houses. The model is applied to 1,366 home sales from mid-1999 to mid-2000 in Utah. The authors find evidence suggesting higher-priced homes tend to have a lower age-related premium than lower-priced houses. Additionally, the number of bedrooms found within lower to mid-priced houses is found to be significant, in contrast to high-priced houses. The authors argue that lower-priced houses generally have fewer bedrooms, driving up the marginal value in this segment of houses. Conversely, the number of bathrooms, living area, lot size, and floor type are more important for high-end houses.

Fan et al. (2006) use a Decision Tree model to estimate house sale prices. The Decision Tree is a model that splits a dataset using splitting rules applied to features. The authors use the Decision Tree because it can find non-linear relationships without many harsh assumptions. Using Singapore apartment sales, the authors find that people who seek a 2-, 3-, or 4-room apartment care more about the model type, floor area, and flat age. People who seek 5-room apartments care more about the floor level. Lastly, people who buy executive apartments (>5 rooms) care less about basic characteristics and focus more on quality, the environment, and nearby facilities. These results fall in line with Zietz et al. (1992) and Zietz et al. (2008), who had the idea of heterogeneous preferences in different market segments. The authors, however, do not find neighborhood-related variables to be of significance. The authors argue the low frequency of neighborhood-related variables prevents the Decision Tree from making meaningful splits in the data using these features. The results by Fan et al. (2006) do not mean that location is unimportant since the literature finds a premium to parks near a house (Crompton & Nicholls, 2020).

2.2 Machine Learning in Real Estate

Machine Learning can be defined as "the process of computers improving their own ability to carry out tasks by analysing new data, without a human needing to give instructions in the form of a program, or the study of creating and using computer systems that can do this" (Cambridge University Press, n.d.) Due to the increase in data availability and computational power, machine learning is becoming a great asset to house price modeling. There is a growing body of literature that investigates the benefits of Machine Learning models in a real estate setting.

Madhuri et al. (2019) evaluate the predictive accuracy of a variety of machine learning models on house prices in Vijayawada, India. The authors compare the RMSE of Multiple Linear Regression, Ridge, LASSO & Elastic Net Regression, as well as Gradient Boosting and AdaBoost. Boosting is a method that uses a base learner, which is a simpler model, and iteratively adds models to the base learner to improve upon the mispredictions, whilst Ridge, LASSO & Elastic Net regressions are forms of penalization of regression coefficients. Madhuri et al. (2019) find that the Gradient Boosting model has the lowest RMSE among all models considered. The paper provides evidence that Boosting models can be a useful tool when predicting house prices. For this reason, I incorporate a model similar to Gradient Boosting in this paper: Extreme Gradient Boosting (XGBoost).

Hong et al. (2020) also see the value in applying a machine learning model to predict house prices. The authors use 39,564 apartment transactions during the 2006-2017 period in Gangnam, South Korea. The authors then compare the traditional Hedonic OLS model to the Random Forest algorithm and find the Random Forest has better prediction accuracy. The RF has an average deviation of 5.5% to the true market price, whilst the OLS model's deviation is nearly 20%. The authors argue that the OLS model might be too simplistic: the model assumes that housing characteristics have a constant effect on house prices. One critique of the paper by Hong et al. (2020) is the fact that all apartments within the sample are close to one another. The Random Forest model works great for the Gangnam apartments; it remains unclear how the results will hold when the sample is extended to other districts. The paper by Hong et al. (2020) highlights the potential benefits of the Random Forest algorithm in real estate modeling.

Lu et al. (2017) use the same Ames dataset as this paper. The authors propose a hybrid LASSO & Gradient Boosting model. This hybrid model is a linear combination of the predictions made by a LASSO regression model and those of an Extreme Gradient Boosting model. The authors find the optimal weight of the LASSO model to be 65%, with the remaining 35% going to XGBoost. This paper contributes to the idea that Boosting models are a useful tool in real estate modeling. At the same time,

the paper showcases how XGBoost is not the single best method as combining the model with LASSO yields improvements in predictive accuracy. The authors end with potential improvements for future research. One suggestion by the authors that I will apply in this paper is the implementation of the Random Forest algorithm.

Fan et al. (2018) also use the Ames dataset and do implement the Random Forest algorithm. The authors find that the Random Forest has relatively worse predictive accuracy than LASSO & Ridge regression, and XGBoost. The authors also apply a linear forecast combination of a Lasso, Ridge, and XGBoost model. These three models perform best independently and are subsequently chosen to be in the combination that leads to an improvement in predictive accuracy.

Complex Machine Learning models such as the Random Forest and XGBoost are oftentimes referred to as Black Boxes: The user does not see the inner workings of the model. Researchers have developed tools that allow us to peek inside the box. One such method is the Partial Dependence Plot (PDP). The PDP is a global method: it illustrates the average effect of changing a variable on predictions. The PDP was introduced by Jerome H. Friedman in 2001 to visualize the inner workings of models, particularly Gradient Boosting models. The PDP has one major assumption: Independent variables may not be too correlated with each other. PDP marginalizes the effects of other variables. When the variable we want to plot PDP for is correlated with other variables, the PDP might extrapolate impossible values (Molnar, 2020). Molnar (2020) gives an example of how weight and height are correlated. The Partial Dependence Plot might try to create datapoint of when a person is two meters tall and 50kg. This data point in the graph would be (near) impossible in reality and might skew results.

Individual Conditional Expectation (ICE) Plots display the effect of a variable on each prediction made by the machine learning model. Each line within the ICE plot graphs how the predicted value of the dependent variable changes with varying levels of one independent variable of interest. The average of all lines within the ICE plot is the Partial Dependence Plot. ICE is used when heterogeneous relationships are present in the data but not captured by the PDP (Molnar, 2020). In the scenario with two independent variables and one dependent variable, this means that the effect of the first independent variable on the dependent variable might be different for varying levels of the second independent variable. Similarly to Partial Dependence Plots, ICE Plots can struggle when the independent variable of interest is highly correlated with other features: ICE Plots can make use of invalid data points (Molnar, 2020). For these cases, it is important to make sure all coordinates in the ICE plot are realistic. One way this can be done is by omitting the extremes of a dependent variable in the creation of an ICE plot.

2.3 Forecast Combinations

The paper by Lu et al. (2017) illustrates the benefits of combining forecasts. The combination of forecasts leading to better forecasts than the best individual predictor is a pattern often found in empirical studies (Timmermann, 2006). Simple averaging of the forecasts can already lead to substantial improvements relative to the best individual prediction (Makridakis et al, 1982, Makridakis and Winkler, 1983, as cited in Timmermann, 2006). In the case of one or several forecasters performing significantly worse, it is beneficial to omit these from the combination (Makridakis and Winkler, 1983, as cited in Timmermann, 2006). Other common methods include using linear regression to estimate a model's weight or using the Mean Square Error to determine weights. In practice, the latter method works well when using MSE-inverse weights (Timmermann, 2006). In this paper, I apply the Linear Regression weights, RMSE-inverse weights, as well as simple averaging of forecasts to improve upon the best individual model.

3.0 Data

The data used in this paper is obtained from Kaggle. The dataset is part of the website's machine learning competition *House Prices - Advanced Regression Techniques.* The website hosts competitions where users obtain the same training data to optimize out-of-sample predictions. This house prices dataset has been prepared by Dean de Cock.

The data consists of two distinct parts, the Ames dataset and the Competition Dataset. These two datasets differ in that the Competition Dataset does not include the Sale Prices of houses. The Sale Prices of the Competition Dataset are hidden by Kaggle. Predictions of Sale Prices within this dataset must be handed into Kaggle. The website then tells the user the accuracy of these predictions.

The Ames dataset consists of 1460 house sales in Ames, Iowa. Sales occur between 2006 and 2010. The data consists of variables relating to features of the house such as building year, remodel year, house type, utilities present, overall condition & quality of building materials, foundation type, functionality of the house, and the number of bathrooms. Furthermore, the data includes variables describing the size of the first and second floor, the total above-ground living area, the number of rooms, the number of kitchens (and kitchen quality), the number of fireplaces (and fireplace quality), the heating in the house, and the presence of central air conditioning.

The dataset also has features relating to outside the house such as the type of driveway, alley type, lot shape & area, lot configuration, lot frontage[1], land contour, slope of the land, size of the porch, quality of the fence, presence of a pool, roof type and materials, and exterior materials used (including potential masonry veneer walls[2]), and condition & quality of the roof

Furthermore, various variables relating to the garage and basement are present in the data such as garage type, the condition of the garage, garage size, building year, and quality of materials used in the garage. For the basement, features include basement height, overall condition, total area of the rooms, room ratings, and the basement's exposure to the outside.

[1] Lot Frontage: Size of street adjacent to the house, in linear feet.

[2] Masonry Veneer: Decorative Outer layer of the Exterior

Lastly, the dataset includes features describing the general surroundings of the house including the neighborhood, the type of housing zone (e.g. agricultural or high density), and what type of condition is close. Conditions include railways, parks, and roads.

The dataset has missing values for Masonry Veneer Area and Lot Frontage. The data has 259 observations where Lot Frontage has a missing value (NA). Masonry Veneer Area (and Veneer Type) has 8 NA's. The missing values for Lot Frontage indicate that these houses are either not in a street, or the dataset has not recorded this variable for the house. For the Masonry variables, missing values either indicate no Masonry Veneer at all, or the data is unrecorded. For both variables, I assume the data is not recorded and I will impute them in the Methodology Section. Missing values in other variables are more meaningful: Having no Pool Quality is due to not having a pool for example.

Table 1 presents the most correlated independent variables with the Sale Prices of the houses in the Ames dataset. How the Sale Price correlates to other variables indicates potential relationships. For example, the Sale Price is positively correlated with the Quality of materials used in the building (.79). This positive correlation indicates that using higher quality materials might increase the value of a house.

Sale Price is also positively correlated with the size of the Above Ground Living Area (.71), Garage Area (.62), and Basement Area (Bsmnt Area, .61). The correlation between the Total Rooms Above Ground and Sale Price is similarly positive (.53). This positive correlation indicates that larger houses might be more expensive on average.

Furthermore, the Sale Price is negatively correlated with the age of the house (-.52) as well as the Remodel Age (-.30). These correlations indicate that older houses might be relatively cheaper.

Table 1

Most correlated variables to Sale Price

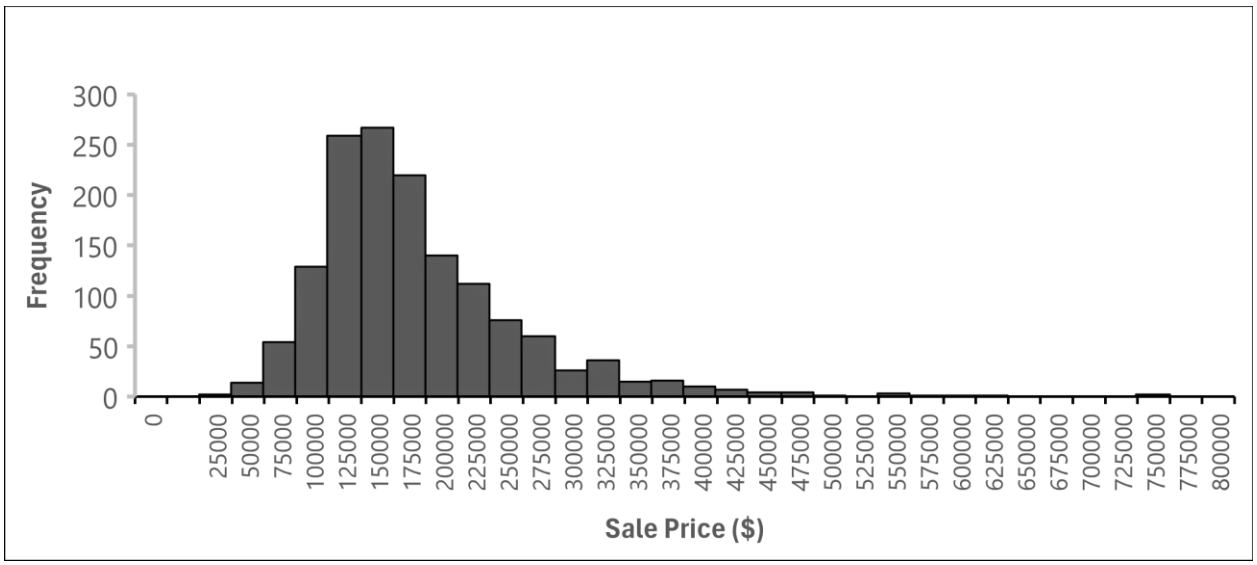
Variable	Correlation	Variable	Correlation
Overall Quality	.79	Second Floor Area	.32
Above Ground Living Area	.71	Open Porch Area	.32
Garage Car Spots	.64	Exterior Vinyl Siding	.30
Garage Area	.62	Half Bathrooms	.28
Bsmnt Area	.61	Lot Area	.26
First Floor Area	.61	Has Garage	.24
Full Bathrooms	.56	Bsmnt Full Bathrooms	.23
Total Rooms Above Ground	.53	Basement Unfinished Area	.21
Masonry Veneer Area	.47	Bedrooms Above Ground	.17
Fireplaces	.47	Exterior Metal Siding	-.17
Bsmnt Area Finished Room 1	.39	House Remodel Age	-.30
Lot Frontage	.34	Garage Age	-.39
Wood Deck Area	.32	House Age	-.52

Note: This table depicts the variables most correlated with the Sale Price in the training dataset, computed using Pearson correlation. Only correlations exceeding an absolute value of 0.15 are displayed. This table presents potential relationships between variables and the Sale Price of houses in Ames. Bsmnt is short for Basement.

Figure 1 plots the distribution of house Sale Prices in the Ames dataset. The figure demonstrates how the distribution has extreme values on the right side of the figure. Models can struggle with outliers due to their uncommon nature. These extreme values, however, are true Sale Prices. Omitting these observations from the dataset could lead to a loss of information regarding the most valuable houses. On the contrary, including these extreme values could lead to poorer model performance. I therefore set all house prices exceeding 500,000$ to 500,000$. This method reduces the effect extreme sale prices have on the model, while at the same time retaining information on what makes a house expensive. To make the data even more normal, I take the natural logarithm when training the models.

Figure 1

Distribution of House Prices Ames (2006-2010)



Note: This figure plots the distribution of Sale Prices in Ames between 2006 and 2010 in the Ames dataset.

4.0 Methodology

I split the Ames data randomly into three parts: a Training Set (60%), a Validation Set (20%), and a Test Set (20%). I use the Test set and the Competition dataset to evaluate the out-of-sample accuracy of each model.

4.1 Imputation

The Ames dataset has missing values for Lot Frontage and the Masonry Veneer Area. I impute missing values in the Training data using the median for these variables in the Training data. For each out-of-sample dataset, I replace missing values using the same median computed from the Training data. Using the median to impute missing values is a relatively simple, but common imputation method. The median is robust to extreme values, as a middle value is chosen regardless of the maximum and minimum values. Lastly, using the median means imputed values are part of the middle of the variable's distribution. As a result, a variable's distribution remains nearly the same.

4.2 Squared Variables

For size and age-related house features, I include squared terms to help Linear Regression Models find non-linear patterns. To reduce collinearity between house features I first subtract each variable's mean from an observation and then square it. The resulting polynomial describes an observation's deviation from the mean. The squared term becomes:

Equation 2: $Squared\ x_i = (x_i - \bar{x})^2$

Where

- x_i is observation *i* of feature *x*,
- $\bar{x}$ is the mean for feature x in the Training dataset.

I apply this method to the following variables: Lot Area, Lot Frontage, all Basement size variables, First Floor Area, Second Floor Area, Living Area Above Ground, Wood Deck Area, Open Porch Area, Enclosed Porch Area, Screen Porch Area, 3-Season Porch[3] Area, Pool Area, Masonry Veneer Area, Garage Area, House Age, and Garage Age.

[3] 3-Season porch: Porch between walls and under a roof; sunroom

4.3 Variable Grouping

Machine learning models can struggle when categorical variables have rare values. The Ames dataset has such variables. I group infrequent values of categorical variables. Variables such as Exterior Condition, which assign a rating, infrequently take on the value "Poor." I combine this rating with the second lowest rating "Fair." Similarly, I combine the highest category "Excellent" with the second highest rating "Good." Furthermore, the Functionality variable contains information on whether a house has major or minor livability flaws. I combine the two categories of major flaws and the two categories of minor flaws as these categories mean approximately the same. Furthermore, I combine neighborhoods that have small frequencies and similar means of Sale Prices. I combine neighborhoods Bluestem (137,500$) and NorthPark Villa (142,694$), Timber (242,247$) & Veenker (238,773$), and Bloomington Heights (194,4871$) & Northwest Ames (189,505$). Table 2 describes the grouping of variables in more detail.

Table 2

Variable grouping and original frequencies Ames Dataset

Variable	Categories
Neighborhood	Bluestem (2), NorthPark Villa (9)
Neighborhood	Timber (38), Veenker (11)
Neighborhood	Bloomington Heights (17), Northwest Ames (73)
Lot Shape	Moderately Irregular (43), Irregular (10)
Lot Configuration	Two-sides Frontage (47), Three-sides Frontage (4)
Housing Zone	Medium Density (218), High Density (16)
Land Slope	Moderate (65), Severe (13)
Roof Style	Flat (13), Gable (1141), Gambrel (11), Mansard (7)
Roof Style	Hip (286), Shed (2)
Fence Type	Wood (54), Wire (11)
Garage Type	2-Types (6), Basement (19), Carport (9)
Foundation	Concrete (647), Wood (3), Stone (6)
Electrical	Fuse Fair (27), Fuse Poor (3), Mix (1)
Heating	Floor (1)-, Gravity (7)- & Wall-Furnace (4), Gas steam heat (18)
Heating Quality	Fair (49), Poor (1)
Exterior Condition	Excellent (3), Good (146)
Exterior Condition	Fair (29), Poor (1)
Functionality	MajorFlaws1 (14), MajorFlaws2 (5), Severe Flaws (1)
Functionality	MinorFlaws1 (31), MinorFlaws2 (34)
Basement Condition	Fair (29), Poor (2)

Note: This table presents the grouping of categorical variables. The name of the variable is in the column Variable. Which categories are grouped is in the column Categories. Between brackets is the frequency of the category occurring in the Training dataset.

4.4 Ordinary Least Squares Regression

For the Hedonic OLS regression categorical variables need to be one-hot encoded. One-hot encoding is the process in which each unique value of a categorical variable is transformed into a dummy variable. This expands the total number of independent variables to 141. The reference category is a house in Clear Creek, without an alley, with a regular lot shape, a level land contour, no fence, normal surroundings/condition, an exterior of vinyl, no paved driveway, typical exterior and heating quality, and an attached garage in typical condition with normal exposure to the outside.

This regression can be prone to overfitting due to the large number of predictors (GeeksforGeeks, 2024a). The regression follows Equation 3. Within the regression, I take the natural logarithm of the house Sale Prices as it leads to higher accuracy.

Equation 3: $$ln(P_i) = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + e_i$$

Where

- P_i is the sale price of house i,
- β_j represents the marginal value attributed to house-characteristic j,
- J is the total number of house characteristics,
- β_0 is the intercept,
- e_i is the error term of house i.

4.5 Least Absolute Shrinkage and Selection Operator

The large number of variables in the multiple linear regression means dimension reduction is needed. Least Absolute Shrinkage and Selection Operator (LASSO) regularization, selects and shrinks regression coefficients towards zero. The dimension reduction LASSO applies, relates to setting coefficients to zero; variables that LASSO deems important variables retain non-zero coefficients whilst unimportant variables are set to 0. LASSO regularization aims to omit redundant variables from the regression.

LASSO regularization consists of two parts. The first part of LASSO regularization is the computation of the Residual Sum of Squares (RSS) in Equation 4. RSS is a measure of how well a model fits the data.

Equation 4: $$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{J}\beta_j x_{ij})^2$$

Where

- y_i is the observed Sale Price for house *i* in the Training data,
- β_j is the regression coefficient for variable *j*,
- $\beta_0 + \sum_{j=1}^{J}\beta_j x_{ij}$ is the predicted Sale Price for house *i* in the Training data,
- *n* is the total number of observations the regression model is fitted on.

The second part of LASSO regularization is the minimization problem in Equation 5 including the RSS, penalty term λ, and the sum of the absolute values of regression coefficients. The regularization occurs through the penalization of the sum of absolute coefficient sizes using the λ term. Larger values of λ lead to stronger penalization of coefficients. This stronger penalization leads to more coefficients being set to 0. Lower values of λ lead to less regularization, and coefficients will resemble those of the OLS model more.

Equation 5: $$min(\beta)\ RSS\ +\ \lambda\sum_{j=1}^{J}|\beta_j|$$

4.5.1 Hyperparameter Tuning

To choose the optimal value for λ , I apply 10-fold cross-validation optimizing the Mean Square Error using a Grid Search. 10-Fold cross-validation randomly splits the training data into 10 equally sized portions, called folds. For each combination of hyperparameters in the Grid, the LASSO regression is fitted 10 times. Each time the model is fitted, 9 of the 10 folds of the dataset are used to train the model. The tenth subset, the test fold, is used to evaluate the Mean Square Error of the model. Each subset of the data is used as the evaluation subset once, all other times the subset is used as training data. The accuracy of each combination of hyperparameters within the Grid is determined by the average of the Mean Square Errors that are computed from each test-fold. The combination of hyperparameters that has the lowest value for this average Mean Square Error is used to fit the model on the whole Training dataset. For the LASSO regression, the Grid consists of λ values ranging from 0.0 to 1 with steps of 0.000001.

4.6 Random Forest

The Random Forest (Breiman, 2001) is part of a larger group of machine-learning models named ensemble models. Ensemble models use the predictions of multiple small models, called learners, to come to a final prediction.

4.6.1 Regression Tree

The Random Forest in this paper uses Regression Trees as learners. The Regression Tree is a tree-based model that falls into the category of supervised machine learning. The model starts in the Root Node, which holds all observations within a dataset. The model then seeks to split the data into a Left Child Node, and a Right Child Node, each holding a subset of the data after the split. The Regression Tree seeks to split the data into two groups with identical characteristics: homogeneous groups. The splits are determined by evaluating all possible splits of all independent variables and picking the split that leads to the lowest variance (Equation 6) in the Child Nodes.

The Variance of a split is computed by taking the squared difference between the mean of the Sale Price in the Left Child Node to each observation's Sale Price in the Left Child Node. When no further split is made from a Child Node, the node is called a Leaf Node.

Equation 6: $Variance = \frac{\sum_{r=1}^{R}(x_r - \mu)^2}{R}$

Where

- x_r is the observed Sale Price of house *r*,
- μ is the mean Sale Price of all houses in the Node,
- R is the total number of observations in the Node.

4.6.2 Random Forest Algorithm

Each Regression Tree within the Random Forest is created using a bootstrapped dataset. Bootstrapping randomly samples observations from the dataset and allows for repeated observations to be sampled. As a result, not all observations are present in each bootstrapped dataset. Furthermore, within the Random Forest, each Regression Tree uses a random subset of independent variables to split the bootstrapped data at each node. Using bootstrapped datasets and random subsets of variables in each

Regression Tree leads to a more diverse set of Regression Trees within the forest. A more diverse set of Regression Trees within a Random Forest means the Trees are less correlated. Generally, "better Random Forests have lower correlation between classifiers and higher strength" (Breiman, 2001)

For an observation, the final prediction by the Random Forest is the result of averaging each Regression Tree's prediction for this observation.

Equation 7: $\hat{f}_{rf}^{B}(i) = \frac{1}{B}\sum_{b=1}^{B} T_b(i)$

Where

- B is the total number of Regression Trees within the Random Forest,
- $\hat{f}_{rf}^{B}(i)$ is the Random Forest's prediction made for observation *i* using *B* Regression Trees,
- $T_b(i)$ is the prediction made for observation *i* by Regression Tree *b*.

4.6.3 Hyperparameters

The Random Forest has several parameters that need to be tuned to optimize the model. These parameters are called hyperparameters. Hyperparameters include the Number of Trees, Mtry, Node Size, and Max Nodes in the Random Forest.

The Number of Trees parameter determines the number of Regression Trees that are grown in the Random Forest. More trees lead to better accuracy, however there is a diminishing effect (GeeksforGeeks, 2024b).

Mtry controls the number of randomly selected variables that are considered at each split within each Regression Tree in the Forest. Setting Mtry to 8 for example, means that at each split within each Regression Tree 8 independent variables are randomly picked. From these 8 independent variables, the most optimal split is determined using the Regression Tree algorithm. When Mtry is set to a low value, trees within the forest are less correlated. Low values of Mtry however, lead to worse-performing trees but do allow for variables with 'moderate' effects to be included more in the predictions. Higher values of Mtry tend to select the 'moderate' effect variables less frequently because an independent variable that is generally better at splitting the data can be used (Probst et al., 2019). Having higher values of Mtry means this 'moderate' effect variable is less often the best splitting variable in the subset.

Regression Trees tend to fit the training data too well. This generally leads to overfitting and poor generalizability (Nadar, 2023). To reduce overfitting, the complexity of Regression Trees needs to be

reduced. How complex each Regression Tree is within the Random Forest is controlled by Node Size and Max Nodes. Node Size controls the minimum number of observations within each Leaf Node in a regression tree. When the minimum number of observations assigned by Node Size is not met, the split is removed from the Regression Tree. As a result, the Regression Tree becomes smaller and less complex. The Max Nodes parameter controls the maximum number of Leaf Nodes in a tree. Similarly to Node Size, lower values for this parameter lead to simpler Regression Trees.

I apply 10-fold cross-validation optimizing the Mean Square Error using a Grid Search. Table 3 presents the Grid Search as well as the optimal value found for each hyperparameter.

Table 3

Random Forest Hyperparameter Tuning

Parameter	Grid	Optimal Value
Number of Trees	400, 500, 600, 750, 1000	600
Mtry	12, 24, 36, 48	36
Node Size	5, 8, 10, 12, 14, 16	5
Max. Nodes	10, 25, 50, 100, 200, 500	500

Note: This table describes the Grid Search and the resulting Optimal Hyperparameters used in the Random Forest algorithm. Tuning is performed using 10-fold cross-validation optimizing the Root Mean Square Error. The final Random Forest model is fitted using the values in the Optimal Value column.

4.6.5 Variable Importance

To determine which variables are the most important predictors of house prices in the Random Forest, I use the Average Decrease in Node Impurity. The Node Impurity is a measure that indicates how alike observations within a Node are. For each Node in each Regression Tree within the Random Forest, the Node Impurity is defined as:

Equation 8: $$RSS_v = \sum_{r=1}^{R}(P_{r,v} - \overline{P}_v)^2$$

Where

- RSS_v is the Residual Sum of Squares in Node *v*,
- $P_{r,v}$ is the Sale Price of house r in Node *v*,
- R is the total number of houses within Node *v*,
- $\overline{P}_v$ is the average Sale Price of all houses in Node *v*.

The Decrease in Node Impurity is the difference between the sum of Child-Node Impurities and the Parent Node Impurity. The Decrease in Node impurity is formally defined as:

Equation 9: $$Decrease\ in\ Node\ impurity\ = RSS_{Parent}\ -\ (RSS_{Left} + RSS_{Right})$$

Where

- RSS_{Parent} is the Residual Sum of Squares of the Parent Node, before splitting,
- RSS_{Right} and RSS_{Left} are two Child Nodes after splitting.

Finally, the Decrease in Node impurities are averaged each time a variable is used to split data in a Regression Tree. This average is the variable's importance within the Random Forest. The bigger the Decrease in Node impurity, the more important the variable is.

4.7 Extreme Gradient Boosting

4.7.1 Algorithm

Extreme Gradient Boosting (XGBoost) is another ensemble machine learning model that (usually) makes use of tree-based predictors. Boosting is the practice of combining models that have low accuracy, called weak learners, to create one model that has high accuracy, a strong learner. Extreme Gradient Boosting is an implementation of the Gradient Boosting Machine (GBM) designed by Friedman (2001).

Extreme Gradient Boosting aims to minimize the difference between observed values and predicted values through a loss function. For continuous dependent variables, the loss function generally follows loss function *L* in Equation 10. For each observation *i* in the Training Data, the loss is:

Equation 10: $$\mathrm{L}(P_i\ , \hat{P}_i) = \frac{1}{2}(P_i\ - \hat{P}_i)^2$$

Where

- P_i is the observed price of house *i* in the training data,
- $\hat{P}_i$ is the predicted price of house *i* by the model.

Extreme Gradient Boosting works by iteratively adding a weak regression tree to the previous model that improves upon the previous model's mispredictions. For iteration *t*, the dependent variable of the Regression Tree is the residuals of the predictions made by the full model in its previous iteration *(t-1)*. For each house *i*, the residual is defined as: $P_i - \hat{P}_i^{(t-1)}$. The prediction of the model at iteration *t* is the sum of the previous prediction and the output by Regression Tree *t* multiplied by a learning rate (Equation 11). Lower values make the algorithm learn slower and more conservatively (Chen et al., 2019).

Equation 11: $$\hat{P}_i^{(t)} = \hat{P}_i^{(t-1)} + \eta \emptyset_i^{(t)}$$

Where

- $\emptyset_i^{(t)}$ is the output of the weak learner at iteration *t* for house *i,*
- η is the learning rate, the contribution of each weak learner to the model,
- $\hat{P}_i^{(t-1)}$ is the predicted value of house *i* in boosting iteration *t-1*.

XGBoost has built-in regularization. This regularization occurs within the loss function in Equation 12 and Equation 13. The output by Leaves ($\emptyset$) through λ, as well as the number of number of Leaves in a weak learner (T) through γ.

Equation 12: $\text{Penalized Loss} = \sum_{i=1}^{n} L(P_i, \hat{P}_i) + \Omega(f_t)$

Equation 13: $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \emptyset_j^2$

Where

- T is the number of Leaves in the Regression Tree,
- γ (Gamma) is the penalization for the number of Leaves,
- $\emptyset_j$ is the output of Leaf *j* in the Regression Tree,
- λ (lambda) penalizes the squared output of each Leaf,
- n is the total number of observations in the dataset.

4.7.2 Hyperparameters

Similarly to the LASSO and Random Forest models, I use a Grid Search and 10-fold cross-validation to optimize the RMSE. Table 4 presents all checked parameters within the grid. Hyperparameters include the Learning Rate η, γ as well as the Number of Trees (Iterations), the Maximum Depth of each Tree, and the Minimum Child Weight. Furthermore, I set the Sub Sample and Column Sample

parameters to 0.8. Tuning these variables alongside the previously mentioned parameters would drive up training time excessively. I use the default value of 1 for λ, for the same reason.

The Maximum Tree Depth controls the maximum number of splits within a decision tree. Lower values for this parameter help prevent overfitting by limiting tree complexity.

The Minimum Child Weight controls the number of observations required in each node after a split. If the requirement is not met, no split is created. This parameter works the same as the Random Forest's Node Size.

The Sub Sample parameter controls what portion of the data is randomly subsampled to train a weak learner in each boosting iteration.

The Column Sample controls how many (randomly chosen) columns are used for splitting data within each tree. This parameter works the same as the Random Forest's Mtry parameter.

Table 4

XGBoost parameter tuning

Parameter	Grid	Optimal Value
η	0.001, 0.01, 0.05, 0.08, 0.1, 0.13, 0.15, 0.2	0.1
γ	0, 0.01, 0.1, 1	0.1
λ	1	1
Number of Trees	100, 250, 500, 750, 1000	250
Maximum Tree depth	5, 8, 10, 15	5
Minimum Child Weight	1, 5, 10, 15, 20	1
Sub Sample	0.8	0.8
Column Sample	0.8	0.8

Note: This table presents the Grid Search used as well as the optimal hyperparameters of the XGBoost algorithm. Hyperparameter tuning is done using 10-fold cross-validation optimizing the RMSE.

4.7.3 Variable Importance

To determine which features, have the biggest influence on XGBoost's predictions of house prices, I use the Gain measure. Gain indicates how the training error decreases at a split in the data (Equation 14). The Gain is the difference between the sum of Similarity Scores (Equation 15) of the two Child Nodes and their Parent Node:

Equation 14: $Gain = Similarity_{Left} + Similarity_{Right} - Similarity_{Parent}$

Equation 15: Similarity = $\frac{(\sum_{q=1}^{Q} g_q)^2}{Q+\lambda}$

Where

- Q is the number of Residuals in a Node,
- g_q is the residual in the Node for observation q,

The importance of a variable is the average Gain for all splits made using that variable.

4.5 Principal Component Analysis

Table 5 showcases the most correlated dependent variables in the Training dataset. Correlations within the dataset are oftentimes intuitive: The number of cars that fit inside a garage is directly related to the size of the garage (.88) because larger garages can fit more cars inside. Similarly, the number of rooms above ground is directly related to the total living area above ground (.83) as larger houses have more rooms. Houses with a larger living area above ground tend to have more bathrooms (.63). The age of the garage and house are correlated since both are oftentimes built in the same period. Apart from the garage age and house age being correlated, all variables in Table 5 are related under a common theme: size.

Table 5

Correlated Independent Variables

Variable 1	Variable 2	Correlation
Garage Area	Garage Cars	.88
Total Rooms above ground	Living Area Above Ground	.83
Area First Floor	Total Basement Area	.82
Garage Age	House Age	.69
Living Area Above Ground	Area Second Floor	.69
Total Rooms Above Ground	Bedrooms Above Ground	.68
Full Bathrooms Basement	Basement Area Finished Room 1	.65
Full Bathrooms	Living Area Above Ground	.63
Total Rooms Above Ground	Area Second Floor	.62
Half Bathrooms	Area Second Floor	.61

Note: This table presents correlated variables in the Ames dataset using Pearson Correlation. The table indicates how size-related variables tend to be correlated.

To reduce dimensionality, I apply Principal Component Analysis (PCA) to these size-related continuous variables. PCA is a method that approximates data by extracting components that capture a majority of the variance. For PCA to work, variables first need to be standardized. For the Training data, I standardize the continuous variables by subtracting each variable's mean from the observed values and dividing the result by the variable's standard deviation. For the Test dataset, Validation dataset, and Competition dataset I standardize the variables using the mean and standard deviation of the Training data.

PCA then consists of computing eigenvalues and eigenvectors using the sample correlation matrix of the size-related continuous variables. The principal components are estimated based on these eigenvectors, with each component representing a linear combination of the original variables. The first component captures most of the variation. Each further component explains less variation and less variation on its own. The components are uncorrelated with one another.

I apply PCA using the correlation matrix due to varying scales in the continuous variables. I then run a Principal Component Regression and Principal Component LASSO regression where I insert the principal components into the dataset, the size-related variables are thus replaced. The dimensionality reduction comes at the cost of less interpretability and a loss of information.

To determine the optimal number of Components I use two methods. The first method is called the Elbow method. In the Elbow method the optimal number of components is derived from finding an elbow-shape in the amount of variance each component explains. The second method sets the optimal number of components to the final component that has an eigenvalue exceeding 1.

Finally, I replace the size-related variables and run the OLS and LASSO regression again. These models are respectively called Principal Component Regression (PCR) and Principal Component LASSO regression.

4.8 Performance Metric: RMSE

In this paper, I use the Root Mean Square Error (RMSE) to evaluate a model's accuracy (Equation 7). The RMSE is an easily interpretable method to evaluate model performance. Lower values of RMSE indicate better predictive accuracy, as predicted values closely match observed values.

Equation 7: RMSE $\sqrt{\sum_{i=1}^{N} \frac{(\hat{P}_i - P_i)^2}{N}}$

Where

- $\hat{P}_i$ is the predicted Sale Price of observation *i*,
- P_i is the observed Sale Price of observation *i*,
- N is the total number of predictions.

The RMSE is scale-sensitive, meaning that the size of the dependent variable influences the size of the RMSE. A drawback of using the RMSE is that it is sensitive to outliers. The squared term in the numerator emphasizes predictions that are far off the observed value. The RMSE of a model might therefore seem relatively worse in the presence of a few large mispredictions.

To make sure expensive houses and cheap houses influence the RMSE equally, Kaggle takes the logarithm of the predicted and observed house prices to evaluate the Competition dataset. The $RMSE_{log}$is given in Equation 7:

Equation 7: $RMSE_{log} = \sqrt{\sum_{i=1}^{N} \frac{(log(\hat{P}_i) - log(P_i))^2}{N}}$

Where

- $\hat{P}_i$ is the predicted Sale Price of observation *i*,
- P_i is the observed Sale Price of observation *i*,
- *N* is the total number of predictions.

4.9 Inside the black box: Individual Conditional Expectations

Partial Dependence Plots (PDP) showcase the average effect a variable has on a Machine Learning model's prediction. To highlight potential interaction effects that cannot be found using Partial Dependence Plots, I use the case-specific Individual Conditional Expectation (ICE) plots (Goldstein et al., 2015). The ICE method plots how a given variable influences a machine-learning model's prediction of the dependent variable. For the Ames dataset, each line within the ICE plot shows the changes in

predicted Sale Price with varying values of a variable of interest. Formally, using observations $\{(x_S^{(i)}, x_C^{(i)})\}_{i=1}^{Q}$, ICE-curve $\hat{f}_S^{(i)}$, is plotted against varying values of $x_S^{(i)}$ whilst holding $x_C^{(i)}$ constant (Goldstein et al., 2015).

Where

- x_S is the variable of interest,
- $x_C^{(i)}$ is the remaining subset of features in the data for observation *i*,
- *Q* is the number of predictions made by the Machine Learning model,
- $\hat{f}_S^{(i)}$ is the response function determined by the Machine Learning model for observation *i*.

I apply ICE plots to the variables that are deemed most important by the Random Forest and XGBoost models. I then bundle ICE plots together based on quintiles of Lot Area. The resulting ICE plots depict the average effect a variable has on the Machine Learning's predictions for each quintile of Lot Size. I use Lot Size as a general proxy for house size. By bundling together quintiles of house size, the ICE plots can display interaction effects between a variable of interest and the size of a house. These resulting ICE plots can be considered the Partial Dependence Plots for each quintile. To enhance interpretability, I use the centered ICE plot. This version of the method anchors each plot to start at the lowest possible value of the variable of interest. The interpretation then becomes a comparison to the anchor point. Formally the centered ICE plot becomes:

Equation 8: $$\hat{f}_{S,centered}^{(i)} = \hat{f}_S^{(i)} - \mathbf{1}\hat{f}(x_S^*, x_C^{(i)})$$

Where

- $\hat{f}_{S,centered}^{(p)}$ is the centered ICE plot of variable S for observation *p*,
- x_S^* is the anchor point of the variable of interest,
- $\mathbf{1}$ is a vector of 1's,
- $\hat{f}$ is the fitted machine-learning model.

For each one-hot-encoded categorical variable, ICE plots are computed by first setting the variable to 0. The Individual Conditional Expectation is computed by taking the difference in predicted Sale Price if the variable would be set to 1. The ICE plot indicates the extra value of a house if a house has the characteristic compared to not having the characteristic. The ICE plot in this case becomes a coefficient.

4.10 Model Combinations

A combination of models often leads to better predictions (Timmermann, 2006). I use three weighting schemes: Equal Weights, Root Mean Square Error-inverse (RMSE-inverse) weights, and Linear Regression weights. Each combination of models is evaluated on the Test dataset and the Competition dataset so that each combination can be directly compared to the individual models.

The Equal weighting scheme assigns the same weight to each individual model and is thus the average. Formally, each model's weight is 1/T, with T being the number of included models. With 6 models, each model has 1/6 weight.

The RMSE-inverse weights assign each prediction by a model a weight inversely proportional to the Root Mean Square error of the model in the Validation set. Each model's weight can be given by the following formula:

Equation 9: $$\widehat{\mathrm{P}}_{\mathrm{final,k}} = \sum_{\mathrm{t=1}}^{\mathrm{T}}\left(\left(\frac{1/\mathrm{RMSE}_{\mathrm{val,t}}}{\sum_{\mathrm{t=1}}^{\mathrm{T}} \mathrm{RMSE}_{\mathrm{val,t}}}\right) \widehat{\mathrm{P}}_{\mathrm{k,t}}\right)$$

Where

- $\hat{P}_{final,k}$ is the final prediction of house *k*
- $RMSE_{val,t}$ is the Root Mean Square Error of Sale Prices of model *t* applied to the Validation data,
- $\hat{P}_{k,t}$ is the predicted Sale Price of house *k* by model *t*,
- *T* is the number of models included in the combination.

The Linear Regression weights are determined using the OLS regression in Equation 10. The observed Sale Prices of house prices in the Validation dataset are a linear combination of each model's predicted Sale Prices for houses in the Validation dataset.

Equation 10**:** $$P_{val,k} = \sum_{\mathrm{t=1}}^{\mathrm{T}}\left(\beta_t \hat{P}_{val,t,k}\right) + e_k$$

Where

- $P_{val,k}$ is the true Sale Price of house *k* in the Validation dataset,
- β_t is the weight assigned to model *t,*
- $\hat{P}_{val,t,k}$ is the predicted value of house *k* in the Validation dataset by model *t*,
- e_k is the error term of house *k* in the Validation dataset.

The predictions in an out-of-sample dataset are thus defined as follows:

Equation 11: $\widehat{\mathrm{P}}_{\mathrm{final,k}} = \sum_{\mathrm{t=1}}^{\mathrm{T}}(\beta_t \widehat{P}_{k,t})$

Where

- $\widehat{P}_{k,t}$ is the prediction made by model *t* for house *k*,
- $\widehat{P}_{final,k}$ is the final prediction for house *k*,
- β_t is the weight assigned to model *t.*

5.0 Results

This section will present the results of the analysis. For the substitution of size-related continuous variables using Principal Component Analysis, I chose the optimal number of components to be 10. I present an in-depth explanation for this choice in the appendix. For the PCLR regression optimal λ 0.0001898069.

5.1 Model Performance

Table 6 describes each model's predictive accuracy on the Train data, Validation data, Test data and Competition data.

The results in Table 6 suggest that the OLS regression model, LASSO regression model, and the XGBoost model are the best predictors of house prices out-of-sample. The LASSO regression model has the best Competition $RMSE_{log}$ (0.123) out of all models, indicating that the model generalizes well when the mispredictions of expensive houses and cheaper houses influence the RMSE similarly. The OLS regression has the second-best Competition $RMSE_{log}$ (0.129), followed by XGBoost (0.135). XGBoost has the lowest RMSE for the Test dataset (22,250.58) and Validation dataset (26,534.49) out of all models. The OLS and LASSO regressions have worse Test RMSE (24,514.57 & 23,458.11 respectively) and Validation RMSE (27,350.82 & 27,717.88 respectively).

Secondly, I find evidence suggesting house price models' performance differs with subsets of data. The three most successful models, OLS, LASSO, and XGBoost, all have at least one dataset where the model has better accuracy than another model. XGBoost performs best for the Test and Validation datasets. LASSO performs best for the Competition dataset. The OLS model is not the best for any dataset but has a better RMSE for the Validation data than LASSO and a better Competition RMSE than XGBoost. These results indicate that the perceived accuracy of models depends on the data the model is evaluated on. In other words, each model can have a subset of houses for which the model is the best choice to predict the houses.

Thirdly, the Random Forest model generalizes relatively poorly. The Random Forest Algorithm has the lowest RMSE for its Training data (11,054.88) out of all models, followed by OLS (16,468.91), but has worse Test RMSE (25,324.06), Validation RMSE (30,070.03) and Competition $RMSE_{log}$ (0.159) than the OLS, LASSO and XGBoost models. These results indicate that the Random Forest in this

paper overfits on the Train data too much which prevents the model from accurately predicting out-of-sample, compared to OLS, LASSO, and XGBoost.

Lastly, I find evidence that applying PCA to size-related variables leads to worse predictive accuracy. Both regression models that incorporate PCA perform worse relative to their respective non-PCA regression models for all out-of-sample datasets. The Test RMSE of the PCR and PCLR models are the worst out of all models (25,746.83 and 27,2536.7 respectively). Furthermore, the Validation RMSEs are also the worst out of all models (33,060.04 & 32,912.05 respectively), and the Competition $RMSE_{log}$ are worse than their non-PCA counterparts (0.144 & 0.143 vs 0.129 & 0.123 respectively). The loss in predictive performance could be caused by the information loss inherent to PCA's dimension reduction.

Table 6

Model Performance

Model	Train RMSE	Validation RMSE	Test RMSE	Competition $RMSE_{log}$
OLS	16,468.91	27,350.82	24,514.57	0.129
LASSO	21,409.03	27,717.88	23,458.11	0.123
Random Forest	11,054.88	30,070.03	25,324.06	0.159
XGBoost	17,905.68	26,534.49	22,250.58	0.135
PCR	20,402.47	33,060.04	25,746.83	0.144
PCLR	20,427.42	32,912.05	25,565.53	0.143

Note: This table presents each model's RMSE on the Training dataset (Train RMSE), Validation dataset (Validation RMSE), and Test dataset (Test RMSE). Furthermore, the table presents each model's $RMSE_{log}$ for the Competition dataset.

5.2 Forecast Combinations

Table 7 describes exactly what weight each model gets within each combination. I omit the PCR and PCLR models from Combination 3 because the PCR weight is 7.4636 and the PCLR weight is -7.4800. These weights indicate that the models have very similar predictions which cause extreme weights when using the Regression Weights.

Table 7

Combination Weighting

Combination	Scheme	OLS	LASSO	Random Forest	XGBoost	PCR	PCLR
1	1/N	0.166	0.166	0.166	0.166	0.166	0.166
2	RMSE-Inverse	0.179	0.177	0.163	0.185	0.148	0.149
3	Regression	1.654	-1.361	0.018	0.680	0	0
4	1/N	0.333	0.333	0	0.333	0	0
5	RMSE-Inverse	0.332	0.327	0	0.342	0	0
6	OLS	-1.344	1.641	0	0.695	0	0

Note: This table presents the weights given to each model for each combination model. Combinations 1 & 4 use equal weights. Combinations 2 and 5 use RMSE-inverse weights (Equation 9). Combinations 3 and 6 use linear regression weights (Equation 10). Weight adds up to 1.

Table 8 describes each model combination's out-of-sample accuracy in the Test data and Competition dataset.

Combining individual models leads to out-of-sample accuracies that are comparable to that of the best-performing individual model. The combinations, however, fail to consistently outperform the best individual model. In this paper combining weights yields an improvement to the best individual model in only one case: Combination 6 has a lower RMSE (21,818.24) for the Test dataset than XGBoost (22,250.58). All other combinations yield a lower RMSE. For the Competition dataset, no combination outperforms LASSO's $RMSE_{log}$ (0.123).

All combinations, however, have RMSE and $RMSE_{log}$ close to the optimal model. The worst model with the worst Test RMSE, Combination 4, has a Test RMSE that is only 208.93 lower than that of the best individual model (XGBoost). Conversely, the difference in Test RMSE between XGBoost and the worst individual model (PCR) is 3,496.25. A similar pattern holds for the Competition $RMSE_{log}$. Here

the difference in the $RMSE_{log}$ between the worst combination, Combination 3, and the best individual model, LASSO, is 0.00948 with the difference between the best and worst individual models being 0.02092. These results indicate that, in situations where the choice of model is unclear a priori, it is beneficial to combine model predictions as this leads to predictions comparable to the best individual model. In other words, to prevent choosing the worst model combining models is beneficial.

Table 8

Combination Performance

Combination	Weighting	Test $RMSE$	Competition $RMSE_{log}$
1	1/N	22,389.38	0.1279
2	RMSE-Inverse	22,318.00	0.1274
3	Linear Regression	22,291.01	0.1329
4	1/N	22,459.51	0.1238
5	RMSE-Inverse	22,428.62	0.1239
6	Linear Regression	21,818.24	0.1291

Note: This table presents each combination model's RMSE on the Test dataset (Test RMSE) and each combination's $RMSE_{log}$ for the Competition dataset. An in-depth description of how prediction weights are presented in Table 7.

5.3 Regression Evaluation

This section will describe relationships found in the Linear Regression and LASSO models. I only evaluate the OLS and LASSO regressions as these two models strictly outperformed their respective PCR and PCLR regression models in terms of out-of-sample RMSE. Table 9 presents all significant ($p<.05$) coefficients in the Hedonic OLS regression. Table 10 presents the LASSO coefficients of all variables present in Table 9.

The Hedonic linear regression in Table 9 provides evidence suggesting that House Size and Lot Size are positively related to Sale Price. The total size of the premise, captured by the Living Area Above Ground, Lot Area, and Lot Frontage all have significantly positive coefficients in the regression. These results indicate that larger houses tend to be more expensive on average. There is one conflicting result to this pattern however, the coefficient for the Number of Bedrooms Above Ground is significantly ($p<.05$) negative. Although the coefficients from the LASSO regression slightly differ from those of

OLS in terms of coefficient size, the patterns found regarding house size and lot size are the same. Additionally, the coefficient for the Number of Bedrooms Above Ground variable is similarly negative.

Secondly, the linear regression results in Tables 9 and 10 also suggest that the general state of a house is important for its selling price. The positive coefficients for the Overall Condition of a house suggest that well-maintained houses tend to sell for more money on average. Similarly, the positive coefficients for Overall Quality suggest that houses with higher-quality building materials are also more valuable to home buyers in Ames, holding everything else equal. The general state of a house is also captured by the house's Age and Age of Remodeling. An older house that has recently been remodeled tends to be in a better state and oftentimes has new building materials installed. The older the Remodel Age is, the less expensive a house is on average. Lastly, the functionality of the house captures any reductions in how livable a house is. According to Tables 9 and 10, A house that has Major, Minor, or Moderate functionality reductions, a house's livability, is worth less, evident from the significantly negative coefficients compared to No functionality reductions.

Furthermore, I find evidence that luxury has a positive relationship with a house's price. The regression coefficients for the Number of Bathrooms, the Number of Half Bathrooms, the Number of Fireplaces, the Pool Area, and Excellent Kitchen Quality (Compared to a 'typical' kitchen) are all positive ($p<.05$ for OLS). Furthermore, the garage is seemingly an important asset: the size of the garage (Garage Cars) is positive.

Lastly, I find evidence suggesting that location is an important determinant of house prices. The models suggest that only houses in Brook Side, Crawford, North Ridge, North Ridge Heights, and Stone Brook are more expensive than houses in Clear Creek. These coefficients are positive ($p<.05$ for OLS).

Furthermore, in the OLS and LASSO regressions, the Feeder Street[4] & Artery Street[5] coefficients are negative ($p<.05$ for OLS). These coefficients indicate that having either a road parallel to a big road or a main road near the house tends to lead to lower house prices compared to not living near such roads. The Cul de Sac[6] configuration of houses, where a road ends in a circle surrounded by houses, is a house feature positively associated with the Sale Price compared to a house 'locked' inside by other houses and a street.

One odd result from the OLS and LASSO regressions regarding location is the negative coefficients ($p<.05$ for OLS) for the Positive Off-Site variable. This condition indicates proximity near a positive

[4] Feeder Street: Small roads leading to a major road.

[5] Arterial Street: A major road

[6] Cul de Sac: End of a road, usually circular, surrounded by houses.

locational feature, a park for example. The regressions indicate that the presence of such a feature can have a negative effect compared to having a Normal Condition, meaning nothing special close by.

Table 9

OLS coefficients

Variable	Coefficient	Variable	Coefficient
Intercept	10.845***	Feeder Street	-0.038**
Lot Shape: Slightly Irregular	-0.016*	Arterial Street	-0.06**
Lot Config: Cul de Sac	0.046***	Positive Off-Site	-0.065***
Building: Duplex[7]	-0.074*	Exterior: Brick Face	0.082***
Building: Town House[8]	-0.085**	Foundation: Brick & Tile	-0.04*
Bsmnt Exposure: Good	0.063***	Fireplaces	0.017*
Bsmnt Finish Room 2: Low Quality	-0.037*	Garage Cars	0.026*
House Type: 2.5 Story	0.124*	Full Bathrooms	0.031**
House Type: 2 Story	-0.046**	Half Bathrooms	0.024*
2nd Story: Finished	-0.082*	Bedrooms Abv. Gr.	-0.016*
Kitchen Quality: Excellent	0.054*	House Age	-0.004***
Garage Type: Other	-0.093***	House Age^2	2.29E-05**
Garage Type: Detached	0.026*	House Remodel Age	-0.001**
Functionality: Major Flaws	-0.121***	Lot Area	5.85E-06***
Functionality: Minor Flaws	-0.078***	Lot Area ^2	-3.77E-11***
Functionality: Moderate Flaws	-0.164**	Lot Frontage	0.001**
Overall Condition	0.049***	Bsmt Area	6.01E-05**
Overall Quality	0.041***	Bsmt Area^2	-5.63E-08***
Neighborhood: Brookside	0.100*	Bsmt Area Finished Room 1	6.85E-05***
Neighborhood: Crawford	0.183***	Bsmt Area Finished Room 2 ^2	7.69E-08**
Neighborhood: NorthRidge	0.108*	Living Area Abv. Gr.	3.09E-04***
Neighborhood: NorthRidge Heights	0.098*	Living Area Abv. Gr.^2	-6.65E-08***
Neighborhood: StoneBrook	0.133*	Wood Deck Area	1.20E-04**
		Pool Area	1.84E-04*

Note: This table presents significant coefficients from the regression in Equation 2. P-value of the Breusch-Pagan test is .042. I reject H0 of Homoskedasticity. Significance is determined using Heteroskedasticity robust errors with *** = p<.001, ** = p<.01, and * = p<.05. The reference category is a house in Clear Creek, without an alley, with a regular lot shape, a level land contour, no fence, normal surroundings/Condition, an exterior of Vinyl, no paved driveway, typical exterior and heating quality, and an attached garage in typical condition with normal exposure to the outside. "^2" indicates the squared term

[7] Duplex: House split into two separate houses

[8] Townhouse: Single-family home that shares one or more walls with adjacent units in similar style

Table 10

LASSO regression coefficients

Variable	Coefficient	Variable	Coefficient
Intercept	10.837	Feeder Street	-0.038
Lot Shape: Slightly Irregular	-0.013	Arterial Street	-0.060
Lot. Config: Cul de Sac	0.045	Positive Off Site	-0.063
Building: Duplex	-0.063	Exterior: Brick Face	0.077
Building: Town House	-0.081	Foundation: Brick & Tile	-0.037
Bsmnt Exposure: Good	0.062	Fireplaces	0.019
Bsmnt Finish Room 2: Low Quality	-0.033	Garage Cars	0.030
House Type: 2.5 Story	0.122	Full Bathrooms	0.029
House Type: 2 Story	-0.040	Half Bathrooms	0.020
2nd Story: Finished	-0.039	Bedrooms Abv. Gr.	-0.013
Kitchen Quality: Excellent	0.056	House Age	-0.003
Garage Type: Other	-0.088	House Age^2	1.83E-05
Garage Type: Detached	0.021	House Remodel Age	-0.001
Functionality: Major Flaws	-0.113	Lot Area	5.29E-06
Functionality: Minor Flaws	-0.073	Lot Area ^2	-3.42E-11
Functionality: Moderate Flaws	-0.155	Lot Frontage	1.00E-03
Overall Condition	0.049	Bsmt Area	6.31E-05
Overall Quality	0.042	Bsmt Area^2	-5.62E-08
Neighborhood: Brookside	0.084	Bsmt Area Finished Room 1	6.43E-05
Neighborhood: Crawford	0.166	Bsmt Area Finished Room 2 ^2	7.21E-08
Neighborhood: NorthRidge	0.089	Living Area Abv. Gr.	3.02E-04
Neighborhood: NorthRidge Heights	0.088	Living Area Abv. Gr.^2	-6.23E-08
Neighborhood: StoneBrook	0.117	Wood Deck Area	1.06E-04
		Pool Area	1.46E-04

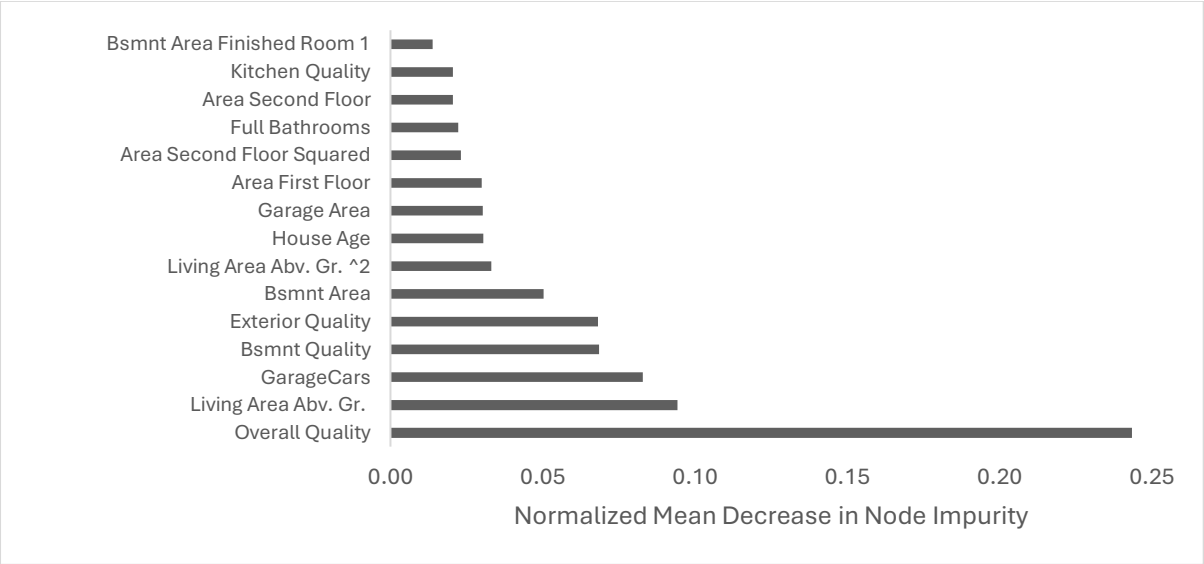
Note: This table presents a subset of coefficients from the LASSO regression. The variables in this table are the same as those in Table 9. Optimal is λ 0.0003227. "^2" indicates the squared term.

5.4 Machine Learning Relationships

Figures 2 and 3 show the importance of the 15 most important variables for the Random Forest and XGBoost models, respectively. For both models, the Overall Quality is deemed most important. Secondly, the Above Ground Living Area is the second most important for both models. Both models have the House Age, Garage Cars, Basement Area, and the Area of the First floor in the top 15. Variable Importances therefore generally match between the two models. Due to XGBoost's superior out-of-sample accuracy relative to that of the Random Forest, I apply ICE plots only to XGBoost's predictions.

Figure 2

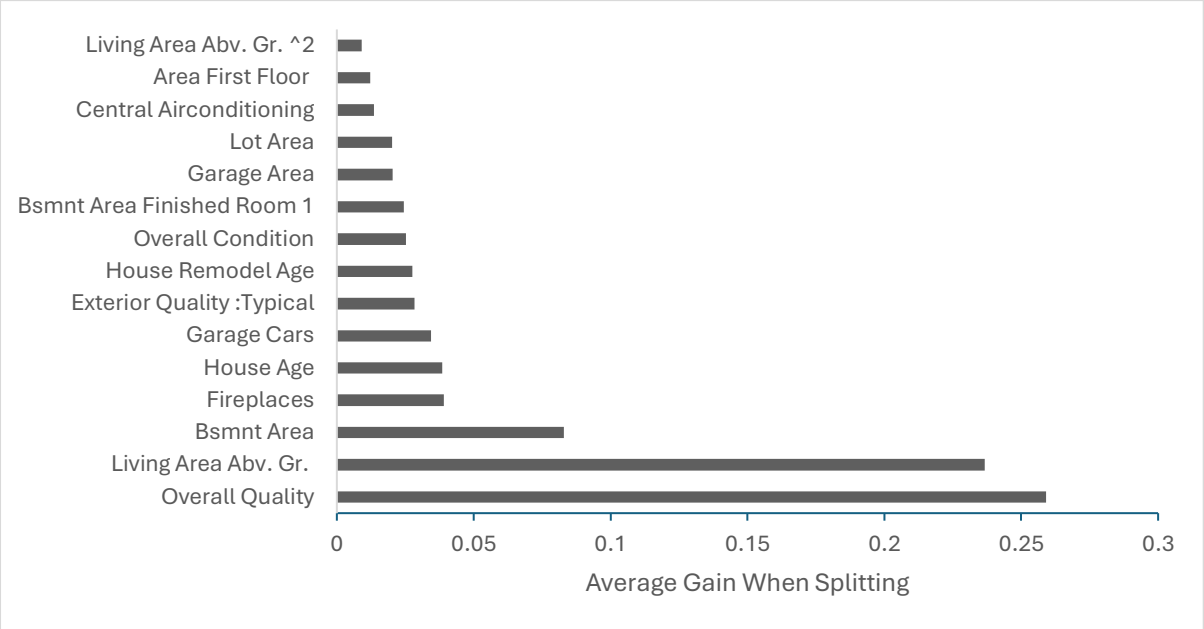
Random Forest Variable Importance



Note: This figure plots the relative variable importance of the 15 most important features according to the Random Forest algorithm. Variable importance is computed using the Mean Decrease in Node impurity when splitting data using a variable. Variable importance is normalized for readability. "^2" indicates the squared term. Abv. is short for Above, Gr. is short for Ground.

Figure 3

XGBoost's most important variables

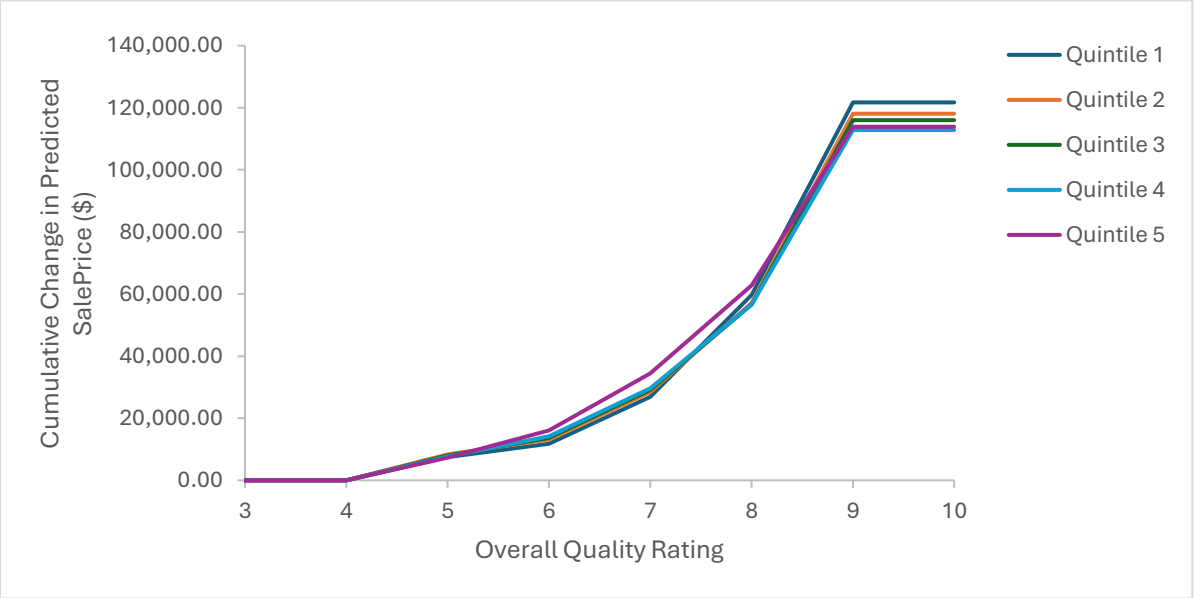


Note: This figure plots the relative variable importance of the 15 most important features according to the XGBoost algorithm. Variable importance is computed using the Average Gain when splitting data using a variable. "^2" indicates a squared term. Abv. is short for Above, Gr. is short for Ground.

Figure 4 depicts the centered-ICE plot for the Overall Quality where observations are clustered by the 5 quintiles of Lot Area. The slopes of all lines in the figure provide evidence that higher quality building materials used in the house are related to higher Sale Prices. Furthermore, the ICE plots on the interval [9,10] display a sudden flattening of the curve. The flattening of the curve indicates that, according to XGBoost, the difference in Sale Prices between houses with the highest two categories of Overall Quality is minimal. Having a high Overall Quality, >8.5, can increase the expected Sale Price by over 100,000$. Lastly, the curves for each quintile of Lot Area are nearly identical to one another. The curves being nearly identical indicates that XGBoost finds no interaction between the Overall Quality and Lot Area. I therefore find no evidence that the Overall Quality of building materials differs with house size.

Figure 4

Relationship Sale Price and Overall Quality clustered by Lot Size Quintiles - XGBoost

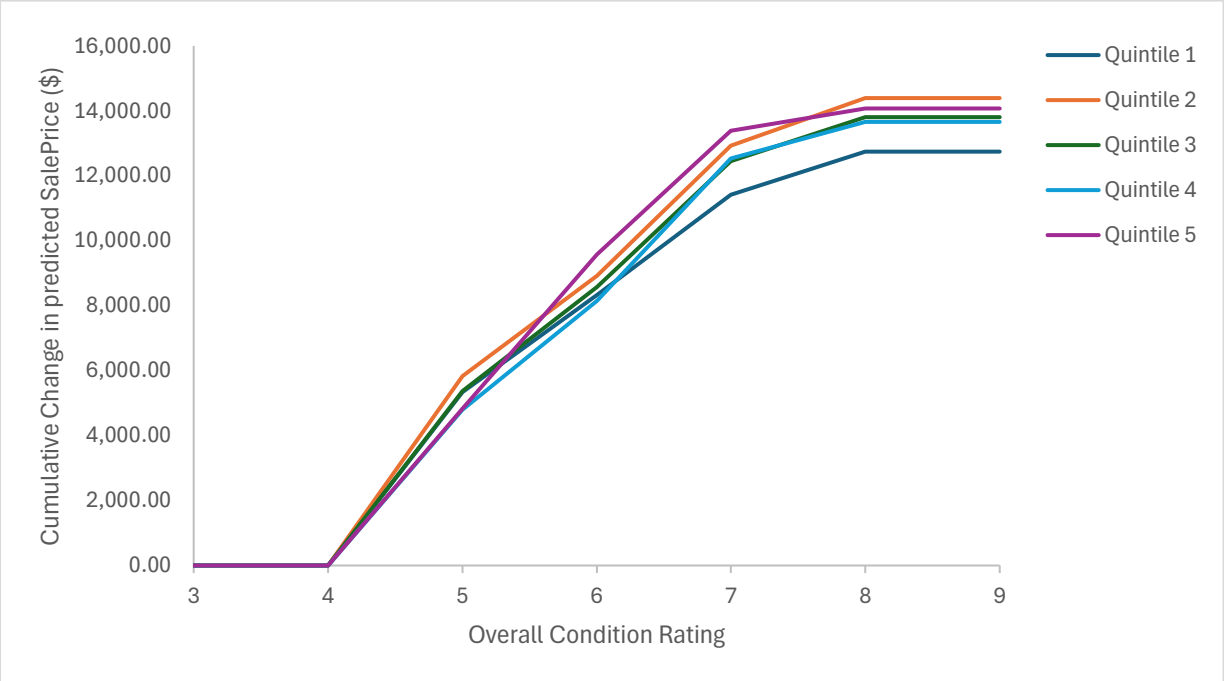


Note: This figure presents the centered-ICE plot for Overall Quality and Sale Price. This c-ICE plot illustrates how the quality of building materials influences the predicted Sale Price of houses by XGBoost. The Individual Conditional Expectations lines are clustered by 5 quintiles of Lot Size.

Figure 5 plots the effect the Overall Condition has on the predicted House Prices. Similarly to Figure 4, the ICE plots have been centered and clustered by quintiles of Lot Area. The relationship between Overall Condition and Sale Price is a decreasingly positive one. The slope for each curve in Figure 5 is positive until it flattens out at ratings of 8 and 9. This result indicates that the best Overall Conditions, 8 & 9, tend to be similar houses in terms of price. Furthermore, the Overall Condition has a smaller effect for the smallest cluster of Lot Area, Quintile 1. This result suggests the Overall Condition of a house matters more to the price for all but the smallest size houses. Lastly, compared to the Overall Quality of a house, the scale of the effect each variable has on the House Price, the Overall Condition has a perceived influence on the predicted Sale Price nearly 10 times smaller.

Figure 5

Relationship Sale Price and Overall Quality clustered by Lot Size Quintiles - XGBoost



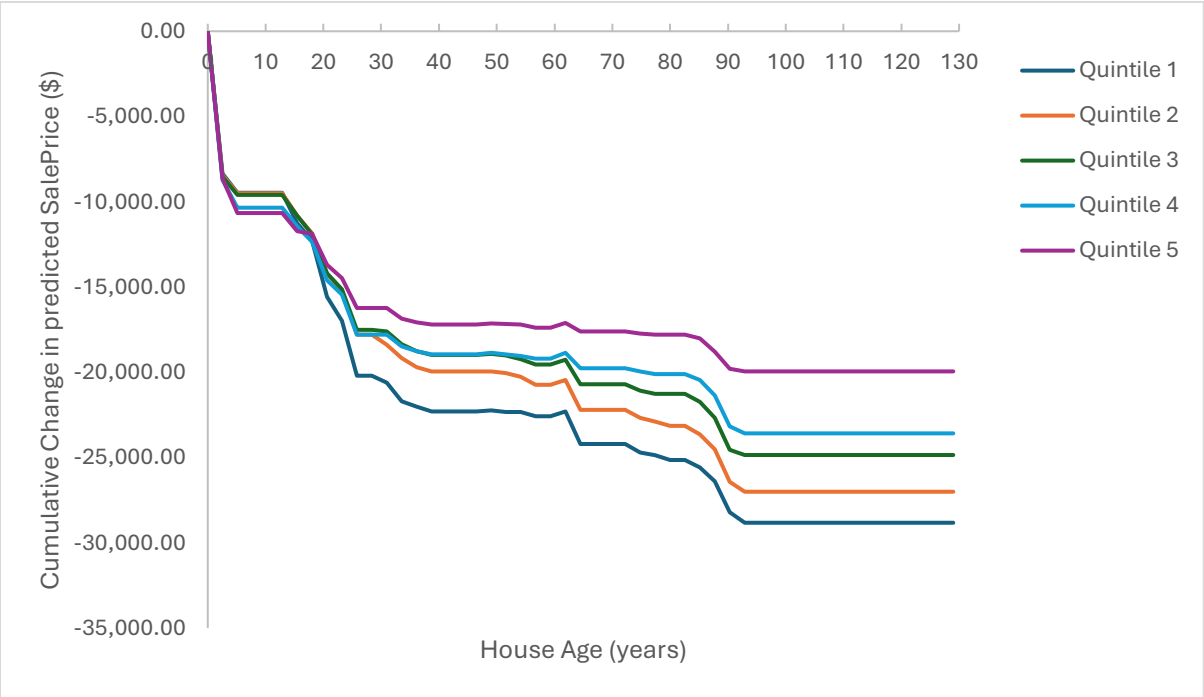
Note: This figure presents the centered-ICE plot for Overall Condition and Sale Price. This c-ICE plot illustrates how the general condition of the house influences the predicted Sale Price of houses by XGBoost. The Individual Conditional Expectations lines are clustered by 5 quintiles of Lot Size.

Figure 6 plots a stepwise negative relationship between Age and Sale Price clustered by quintiles of Lot Area. Figure 6 depicts a sharp decline in the first 5 years after a house is built. Consequently, between ages 5 and 13, the curve falls flat. After the 13 years mark the slopes of the curves become strongly negative until the house is 25 years old. From ages 25 to 60 the curves flatten once more, and generally decrease once more until at 93 years old all curves are completely flat. Figure 6 is evidence that newer houses are more expensive than older houses. Furthermore, the curves in Figure 6 provide evidence that for houses between certain intervals of age, e.g. 93+ years, an extra year matters relatively little to the Sale Price.

Lastly, the curves in Figure 6 differ between the 5 quintiles of Lot Area. Most notably, after the 20 year-mark the five curves stop overlapping. The 5 curves become sorted by Lot Area: Quintile 5 is on top, followed by Quintile 4, 3, 2, and finally 1. This ordering of curves suggests that bigger houses experience a smaller decrease in Sale Prices due to age than smaller houses.

Figure 6

Relationship Age and Sale Price clustered by Lot Size Quintiles -XGBoost

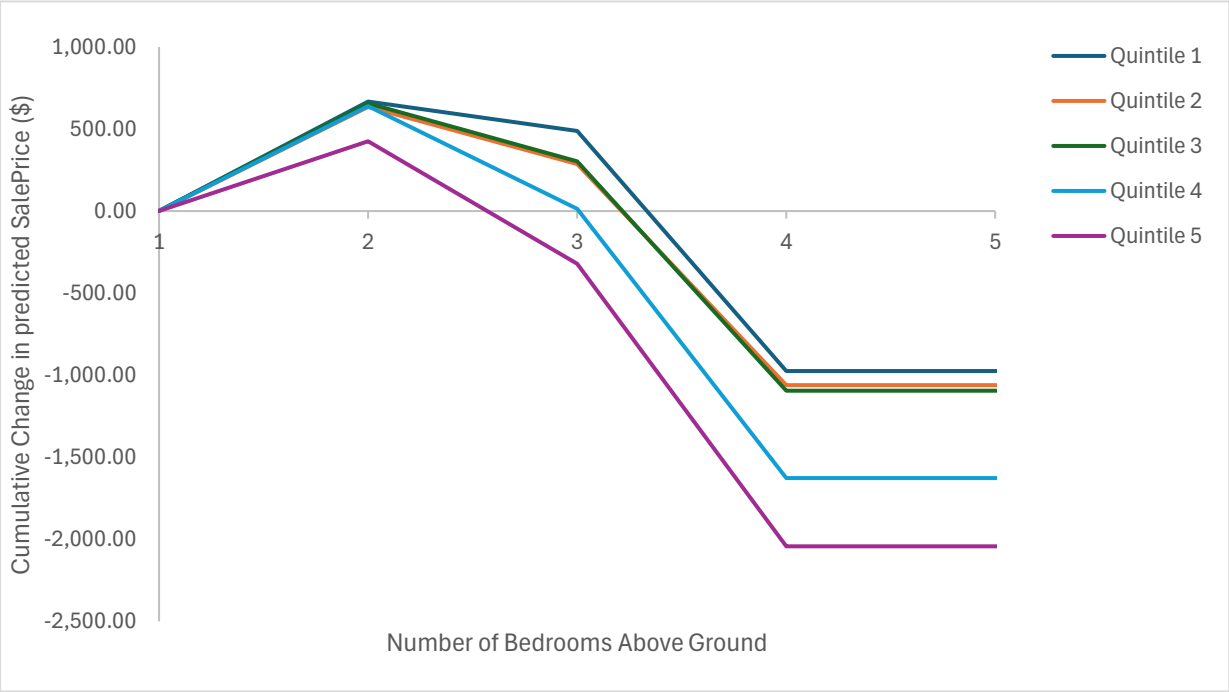


Note: This figure presents the centered-ICE plot for House Age and Sale Price. This c-ICE plot illustrates how the age of a house influences the predicted Sale Price of houses by XGBoost. The Individual Conditional Expectations lines are clustered by 5 quintiles of Lot Size.

The results from the Linear Regression and LASSO Regression indicate that the Number of Rooms Above Ground is negatively related to a house's Sale Price. The c-ICE plot in Figure 7 showcases XGBoost's perceived relationship between the two variables. The curves, clustered by quintiles of Lot Size, show how houses with 2 bedrooms above ground are generally more expensive than houses with any other number of bedrooms above ground. Furthermore, quintile 5 showcases the biggest negative effect. Figure 7 gives the counterintuitive idea that bigger houses with more bedrooms above ground tend to be cheaper on average than smaller houses with more bedrooms above ground (holding everything else equal).

Figure 7

Relationship Bedrooms Above Ground and Sale Price clustered by Lot Size Quintiles -XGBoost

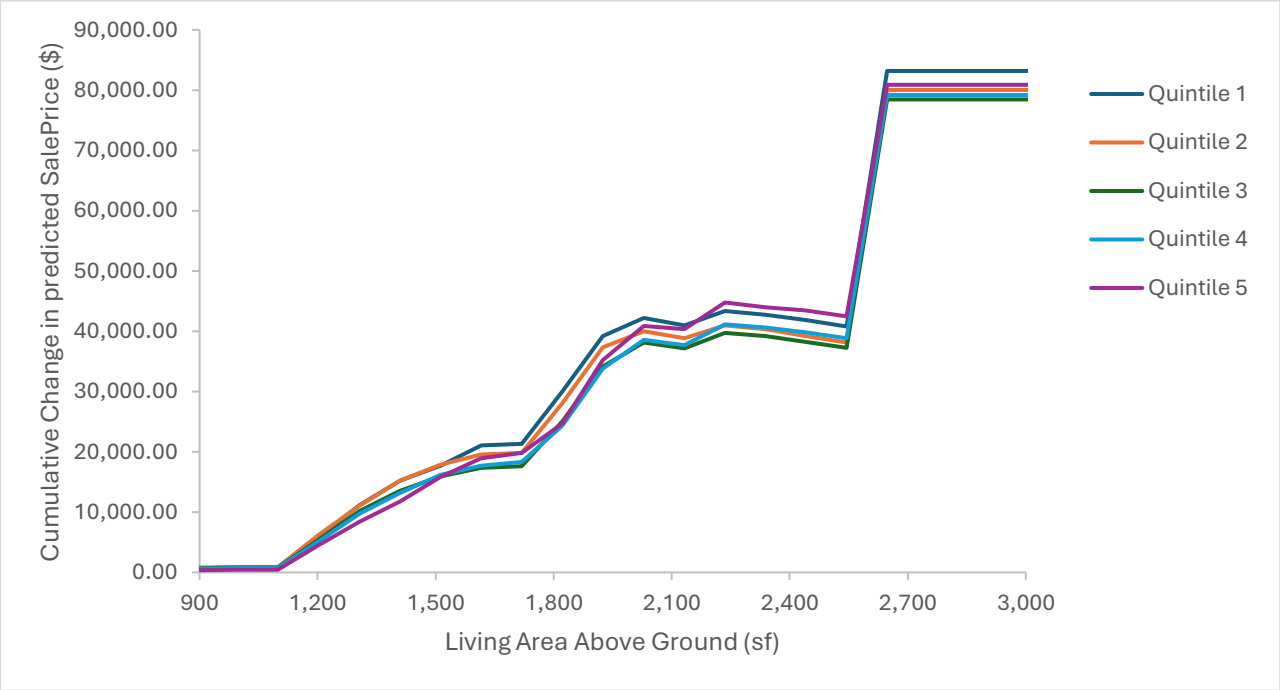


Note: This figure presents the centered-ICE plot for Bedrooms Above Ground and Sale Price. This c-ICE plot illustrates how the number of bedrooms above ground influences the predicted Sale Price of houses by XGBoost. The Individual Conditional Expectations lines are clustered by 5 quintiles of Lot Size.

Figure 8 graphs XGBoost's relationship between the size of the Above Ground Living Area and the Sale Price. The relationship is positive and nearly identical for each cluster of Lot Size. The curves fall flat and decrease slightly between 2,000 square feet and 2500 square feet. From 2500 square feet onwards the rapidly increases until it flattens out completely. Figure 8 provides evidence that larger living areas are positively related to the Sale Price of houses. Furthermore, Figure 8 suggests that houses with living areas exceeding 2,500 square feet tend to be far more expensive on average than houses that don't exceed the 2,500 mark.

Figure 8

Relationship Bedrooms Above Ground and Sale Price clustered by Lot Size Quintiles -XGBoost



Note: This figure presents the centered-ICE plot for Above Ground Living Area and Sale Price. This c-ICE plot illustrates how the size of the living area above ground influences the predicted Sale Price of houses by XGBoost. The Individual Conditional Expectations lines are clustered by 5 quintiles of Lot Size.

The ICE plots for categorical variables can be summarized using coefficient. This coefficient describes the change in the predicted Sale Price for a house when the binary variable of that category becomes 1 instead of 0. Table 11 presents such coefficients for a subset of variables deemed significant by the OLS model, clustered by 5 quintiles of Lot Size.

The XGBoost Individual Conditional Expectations share patterns with the OLS and LASSO regressions regarding the Cul De Sac configuration, and the Brookside, Crawford, and North Ridge Heights neighborhoods being generally more expensive; the changes in predicted Sale Prices are positive. XGBoost also shares the narrative that Feeder Streets and Arterial Streets are negatively associated with lower Sale Prices. These results are also consistent across the 5 quintiles of Lot Size.

Table 11 highlights how XGBoost does not use all categorical variables when making predictions. For example, the result from the OLS and LASSO regression that a Positive Off-Site (e.g., a park close by) is associated with a cheaper house is not shared by XGBoost, all rows are 0, meaning XGBoost finds no effect. Counterintuitively, the variables that indicate Functionality Flaws in a house all have 0 values as well. This result means that XGBoost finds no influence of functionality reductions on house prices.

The 0 values in Table 11 are caused by one of two reasons. The first is that other variables are preferable when splitting data, meaning that the one-hot encoded variable is overlooked. The second reason is that there is indeed no effect. For functionality flaws in a house, the 0 values are likely caused by the former.

Table 11

Relationships categorical variables clustered by Lot Size Quintiles -XGBoost

Variable	Quintile 1	Quintile 2	Quintile 3	Quintile 4	Quintile 5
Positive Off Site	0	0	0	0	0
Cul De Sac	1206.14	1337.07	1381.44	1132.85	1043.88
Minor flaws	0	0	0	0	0
moderate	0	0	0	0	0
Major flaws	0	0	0	0	0
Feeder street	-4437.86	-4850.5	-4910.19	-5059.38	-5464.48
Arterial Street	-3535.05	-3590.7	-3622.00	-3457.08	-3424.36
Brookside	934.56	994.07	883.11	762.73	638.47
Crawford	2578.6	2321.38	2601.56	2552.31	2504.58
North Ridge	0	0	0	0	0
North Ridge Heights	2142.6	2050.39	2082.48	2021.22	1656.56
Stone Brook	0	0	0	0	0
Exterior: Brick Face	0	0	0	0	0

Note: this table presents the Individual Conditional Expectations clustered by 5 quintiles of Lot Size. Each row indicates how the predicted Sale Price by XGBoost changes when the binary categorical variable becomes 1 compared to 0.

6.0 Conclusion

In this paper, I apply several Machine Learning methods in the prediction of house sale prices in Ames, USA, between 2006 and 2010. I compare the Random Forest, XGBoost, LASSO, Principal Component Regression, and Principal Component Lasso regression to the traditional Hedonic OLS regression. Furthermore, I combine predictor models to improve upon the best individual predictor. The main research question is as follows:

To what extent can Machine Learning models replace traditional Hedonic OLS models in predicting house sale prices in terms of accuracy and interpretability?

Hypothesis 1: The accuracy of Machine Learning models will be superior to that of Hedonic OLS models, measured by out-of-sample RMSE and $\text{RMSE}_{\log}$

I find evidence that Machine Learning Methods can outperform OLS regression when predicting house prices. In this paper, OLS, LASSO, and XGBoost have the best out-of-sample accuracy. I find that each of the three models has better performance than at least one other of these three. OLS has a better Validation RMSE than LASSO, but worse Test RMSE and Competition $RMSE_{log}$. OLS also outperforms XGBoost in terms of $RMSE_{log}$, but not in terms of Validation and Test RMSE. Lastly, LASSO has the best Competition $RMSE_{log}$ of the three. I do not find evidence that the Random Forest outperforms OLS, which goes against the findings by Hong et al. (2020).

Furthermore, I find no evidence suggesting that applying Principal Component Analysis on size-related house features improves predictive accuracy. For each out-of-sample dataset, the OLS and LASSO regressions strictly outperform Principal Component Regression and Principal Component LASSO Regression, respectively.

Hypothesis 2: A combination of models improves upon the best individual model's predictions, measured by out-of-sample RMSE and $\text{RMSE}_{\log}$

Regarding forecast combinations, I find no evidence that combining house price models leads to a consistent improvement compared to the best individual predictor. The first reason is that no model is the absolute best, out of OLS, LASSO, and XGBoost. The second reason is that, out of 6 combinations, only one combination of models outperforms the best individual predictor model for the Test dataset. For all other datasets, the combinations yield no improvement. This conclusion does not fall in line with

the literature which finds that combining models generally improves upon the best individual model (Timmermann, 2006)

One upside, however, is that combining predictor models leads to final predictions that are relatively close to the best individual model. In cases where the choice of optimal model is unclear, combining models can be seen as a hedge against choosing the wrong model. The results could differ with other combination methods, as I use relatively simple ones: Equal Weights, RMSE-Inverse Weights, and weights estimated using Linear Regression.

Hypothesis 3: Relationships between features and price found in the Hedonic OLS model are more intuitive or resemble the literature more closely than Machine Learning Models.

Linear regression methods assume a linear relationship between the dependent variable and the independent variables. XGBoost is free from this assumption. This allows the model to find non-linear patterns and relationships. I find that the Individual Conditional Expectation plots derived from the XGBoost model can find complex relationships such as a stepwise negative effect of age. The ICE plots, however, indicate how the ML model makes predictions and can therefore only be seen as a proxy.

Furthermore, I find evidence that the quality of building materials & finish is the most important variable for predicting house prices: the Overall Quality variable has the highest variable importance for both the Random Forest and XGBoost. The ICE plot using XGBoost shows how the predicted house price can increase by over 100k$ compared to a low-quality house.

Lastly, I find that the signs of relationships from the OLS model generally match those of LASSO and XGBoost and are usually intuitive. A higher quality of materials, a better condition, bigger size, and luxury in a house are positive attributes. However, relationships found by both the OLS model and XGBoost model suffer from counterintuitive results. One of which is a negative relationship between the number of bedrooms above ground and Sale Price. For OLS and LASSO specifically, the presence of a park is negatively related to Sale Price. This result goes against the findings of other papers where houses tend to have a premium due to parks (Crompton & Nicholls, 2020) For categorical variables the XGBoost model finds fewer relationships with Sale Price, possibly because other variables are better at splitting data.

To answer the main research question, not one model fits all. Different models perform better on different subsets of data. Furthermore, combining the traditional Hedonic model with LASSO and XGBoost can give more consistent accuracy despite each model's counterintuitive relationships. I therefore do not recommend replacing the Hedonic model, but instead complementing it with more complex models.

Further research could include other machine learning models such as the Artificial Neural Network and explore other weighting schemes for model combinations. I also recommend using different datasets to verify the findings in this paper. The sample is from 2006 to 2010, results might differ in the current housing market. Lastly, further research could use the relationships described by XGBoost's ICE plots and implement these perceived relationships into the OLS and LASSO models to see if this improves accuracy. For example, the step-like pattern found by XGBoost could be modeled in the regression models.

7.0 Bibliography

Montoya, A., DataCanary. (2016). *House Prices - Advanced Regression Techniques.* Kaggle. https://kaggle.com/competitions/house-prices-advanced-regression-techniques

Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

Cambridge University Press. (n.d.). Machine Learning. In *Cambridge dictionary*. Retrieved June 28, 2024 from https://dictionary.cambridge.org/dictionary/english/machine-learning

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

Crompton, J. L., & Nicholls, S. (2020). Impact on property values of distance to parks and open spaces: An update of US studies in the new millennium. *Journal of Leisure Research*, *51*(2), 127-146. https://doi.org/10.1080/00222216.2019.1637704

Dubin, R. A. (1998). Predicting house prices using multiple listings data. *The Journal of Real Estate Finance and Economics, 17*, 35-59. https://doi.org/10.1023/A:1007751112669

Fan, G. Z., Ong, S. E., & Koh, H. C. (2006). Determinants of house price: A decision tree approach. *Urban Studies, 43*(12), 2301-2315. https://doi.org/10.1080/00420980600990928.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

GeeksforGeeks. (2024a, February 29). *The relationship between high dimensionality and overfitting.* https://www.geeksforgeeks.org/the-relationship-between-high-dimensionality-and-overfitting/

GeeksforGeeks. (2024b, April 3). *The Effects of the Depth and Number of Trees in a Random Forest.* https://www.geeksforgeeks.org/the-effects-of-the-depth-and-number-of-trees-in-a-random-forest/

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, *24*(1), 44-65. https://doi.org/10.48550/arXiv.1309.6392

Hong, J., Choi, H., & Kim, W.- sung. (2020). A house price valuation based on the random forest approach: the mass appraisal of residential property in South Korea. *International Journal of Strategic Property Management*, *24*(3), 140-152. https://doi.org/10.3846/ijspm.2020.11544

Kenton, W. (2024, 2 April). What Is a Black Box Model? Definition, Uses, and Examples. *Investopedia*. Retrieved from https://www.investopedia.com/terms/b/blackbox.asp

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., & Team, R. C. (2020). Package 'caret'. *The R Journal*, *223*(7), 48

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Madhuri, C. R., Anuradha, G., & Pujitha, M. V. (2019, March). House price prediction using regression techniques: A comparative study. In *2019 International conference on smart structures and systems (ICSSS)* (pp. 1-5). IEEE. Doi: 10.1109/ICSSS.2019.8882834.

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable* (2nd ed.). https://christophm.github.io/interpretable-ml-book/

Nadar, J. B. (2023, June 23). *Overfitting in decision tree models: Understanding and overcoming the pitfalls.* Medium. https://joelnadarai.medium.com/overfitting-in-decision-tree-models-understanding-and-overcoming-the-pitfalls-980cf7af7d8b

Newsome, B. A., & Zietz, J. (1992). Adjusting comparable sales using multiple regression analysis-The need for segmentation. *Appraisal Journal*, *60*(1), 129-136.

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301. https://doi.org/10.1002/widm.1301

Selim, H. (2009). Determinants of house prices in Turkey: Hedonic regression versus artificial neural network. *Expert systems with Applications*, *36*(2), 2843-2852. https://doi.org/10.1016/j.eswa.2008.01.044

Sirmans, G. S., Macpherson, D. A., & Zietz, E. N. (2005). The Composition of Hedonic Pricing Models. *Journal of Real Estate Literature*, *13*(1), 3–43. http://www.jstor.org/stable/44103506

Sirmans, G.S., Zietz, J. & Zietz, E. N., (2008). Determinants of house prices: a quantile regression approach. *The Journal of Real Estate Finance and Economics*, *37*, 317-333. http://dx.doi.org/10.1007/s11146-007-9053-7

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of political economy*, *74*(2), 132-157. https://doi.org/10.1086/259131

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. http://dx.doi.org/10.1109/IEEM.2017.8289904

Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. *Journal of Political Economy*, *82*(1), 34–55. http://www.jstor.org/stable/1830899

Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting*, *1*, 135-196.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R. (1982). "The accuracy of extrapolation (time series) methods: Results of a forecasting competition". *Journal of Forecasting* 1, 111–153. https://doi.org/10.1002/for.3980010202

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). *Forecasting methods and applications*. John wiley & sons. http://dx.doi.org/10.2307/2581936

8.0 Appendix

Determination of the number of Principal Components

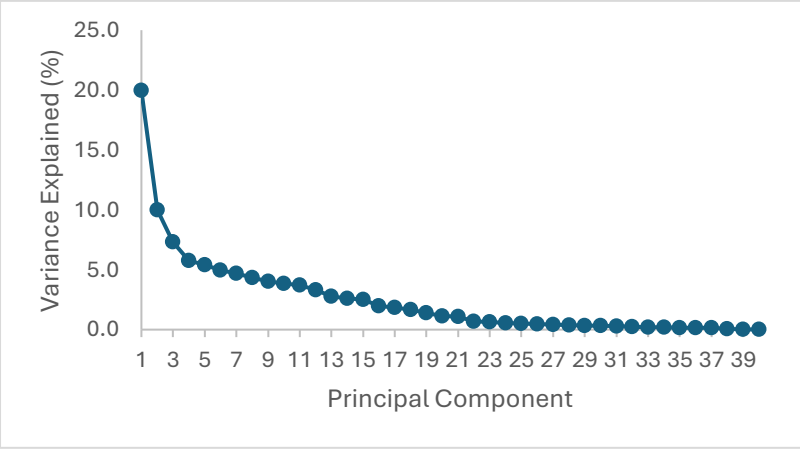
I apply two methods to determine the optimal number of principal components for the PCR and PCLR regressions. Figure A1 plots the portion of variance explained by each individual component. The Elbow Plot method argues that the optimal number of components is chosen when the difference in the portion of variance explained between two consecutive components suddenly shrinks. This process leads to an Elbow Shape within the plot. In Figure A1 the elbow shape occurs at Principal Component 4.

The second method for choosing the number of components consists of picking all components with an eigenvalue exceeding 1. Figure A2 plots each component's eigenvalue. The black line indicates the 1-threshold. For this method, the number of components is 15.

Due to 4 and 15 being far apart, I take the middle ground: 10 components. This number provides a balance between the two methods.

Figure A1

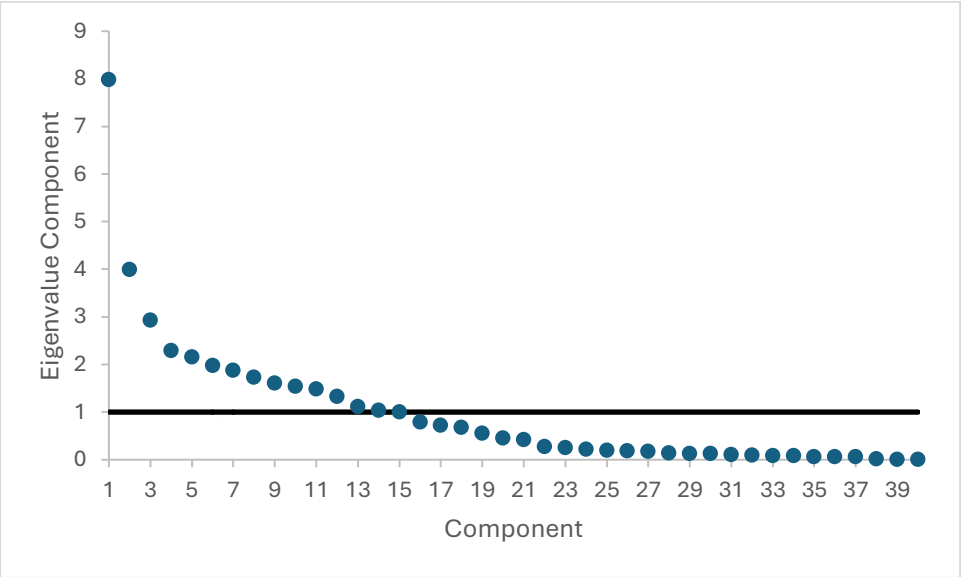
Elbow Plot Component



Note: This graph plots the variance explained by each component. This plot suggests the number of components at the elbow point (4) in the plot must be used.

Figure A2

Component Eigenvalues



Note: This graph plots the eigenvalues of each component. This plot suggests all components above the black line must be used.