Predicting Housing Prices in Ames, Iowa

MATH564 - Final Project

Grant Nikseresht - A20386885 - 1/3 Contribution
Yuqing Zhao - A20392575 - 1/3 Contribution
Yue Ning - A20282614 - 1/3 Contribution

11/30/2018

Abstract

Selling prices of homes, as well as being an important economic indicator, are a value that home buyers and sellers are constantly keeping in mind. Our goal in this project is to predict the selling prices of homes in Ames, Iowa based on data that characterizes those homes using a wide variety of features. The main challenge we found in modeling home prices with such a diverse dataset was selecting significant and independent predictors to maximimize the predictability of home prices. We utilize a variety of techniques including regularization methods and tree regression to search for improvements over a baseline ordinary least squares model in predicting home prices. Our results show that regularization methods and stepwise selection were effective techniques for increasing model performance in the presence of multicollinearity of the input features.

1 Introduction

1.1 Background

Most people understand and accept the risk that stock prices might fall, but may not consider future fluctuations in real estate prices when they buy a house. Many house buyers use recent price performance as a benchmark for what they expect over the next several years. Making purchase for a house can be an important decision. Having an idea of future housing price trend will help customers to decide. However, recent housing market trends are generally not a good prediction of future housing price performance [1]. The housing market is subject to the same forces of supply and demand as any other market. In addition, home itself can be evaluated based on different features. There are more factors that affect the price of a home llike the number of bedrooms or presence of a white picket fence.

1.2 Problem Statement

In this project, we explore how a large variety of home features affect housing price. We performed a variety of regression and variable selection techniques in attempting to model the housing price. The most challenging component in modeling the housing price is feature selection. Our chosen dataset contains 79 explanatory variables describing everything from the amount of square footage on the second floor of a home to whether or not the property has a pool. For each aspect of a house, there are generally several variables given to describe it that are highly correlated. Inference is an important goal of ours as well, as we'd like to determine which housing features are unambiguously associated with changes in housing prices. For this reason, our modeling focused on a variety of variable selection techniques: forward and backward stepwise selection, ridge regression, LASSO regression, and tree regression. We also performed some manual feature engineering in preprocessing using statistical tests and some intuition to reduce the number of features.

1.3 Objective

Our goal is to explore how the different aspects of a residence affect the selling price of homes in Ames, Iowa. In the remaining sections, we'll outline our methods for preprocessing and analyzing the data in such a way to optimize performance of regression techniques in predicting housing prices.

2 Data Background

2.1 Description

The Ames housing dataset is taken from one of the Kaggle challenge: "House Prices: Advanced Regression Techniques" [2]. There are 79 house features and 1460 observations in training dataset. A separate test dataset without target `SalePrice` is also provided. These 79 variables cover almost every aspects of house from the building itself to the outside condition. With these abundant information, this dataset is a good to use for house price predicting.

The raw dataset is available on Kaggle's website and can be downloaded for use. Data is saved in two csv files as training and testing dataset. Detailed explanation for each variable can be found on Kaggle [2]. Generally, the data covers several aspects of a house: house property structure, quality and time length, the basement related features, the garage related features and utilities. Each data field is explained in detail for the feature meaning and meaning for each level if it is categorical variables. Variables are either categorical or numerical. No logical data field in the raw data file. The target is `SalePrice` which is a continuous variable. In reality, not all data may be available for a house. It also happens here where missing values are presented.

2.2 Preprocessing and Exploratory Statistics

First of all, we want to have a look at our target variable: `SalePrice`.

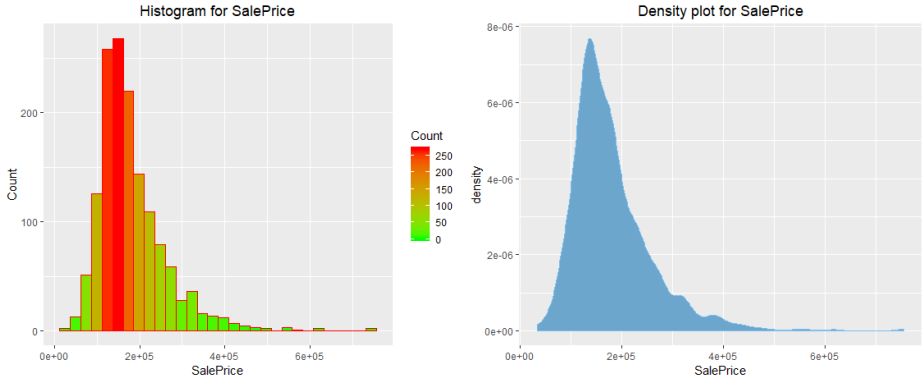


Figure 1: Histogram and density plots of the response variable, `SalePrice`.

We can see from above plots that there is a right skewness in target variable and a little deviate from normal distribution. This is reasonable, because only few people can afford luxury houses. We need to make a log-transformation for `SalePrice`.

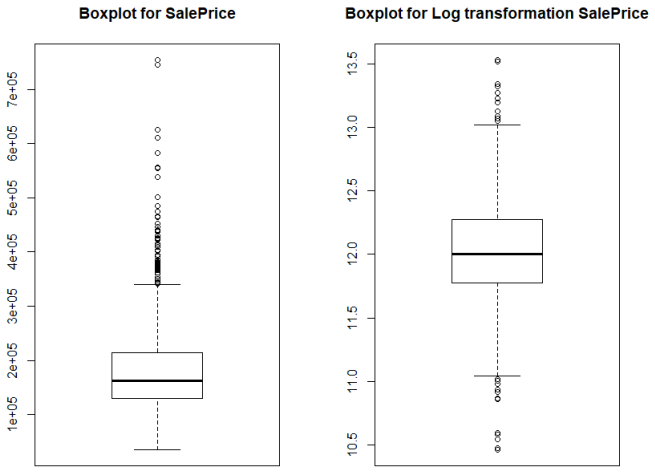


Figure 2: Boxplots before and after applying a log transformation to `SalePrice`.

From above boxplot for `SalePrice`, we can see log transformation makes it more normally distributed.

Second, we did some exploratory analysis of the predictor variables to decide what alterations needed to be made and to look at the data shape and structure of all variables. Visual summaries of the dataset can be seen in the Appendix in Figure 8.

The following is a list of transformations and omissions from the original dataset. A full list of raw variables and descriptions is available on Kaggle [2].

— Many factor variables had an `NaN` value for the case where a feature was not present such as a home not having a Garage. These `NaN` values were replaced with a "NoFeature" type factor for factors relating to garage, fireplace, basement, masonry veneer, alley, lot frontage, and fence.

— Several factor variables were almost entirely homogeneous where all observations had the same value. This provides no new information for modeling, so we discarded them. These variables included `Street`, `Utilities`, `Condition2`, `Heating`, `Pool`, and `Miscellaneous`

— We aggregated component square footage variables for porch size into a signle `TotPorch` variable which is the sum of all component square footages.

— `GarageYrBuilt` is discarded as there's no logical value to impute for homes with no garage.

— `OveralCond`, `MSSubClass`, and `OveralQual` are converted to factor variables.

— Derived variables `BeenRemodeled` and `EraBuilt` are generated from `yearBuilt` and `yearRemodeled` to capture non-linearities in housing trends for construction and remodeling.

— `TotBathrooms` is used as a sum of all of the types of bathrooms in a home including `FullBath`, `HalfBath`, `BsmtFullBath`, and `BsmtHalfBath`.

After these transformations, 66 independent variables remain. We chose to also keep numeric variables with a correlation with `SalePrice` stronger than 0.35. A correlation plot also showed some evidence of multicollinearity between predcitor variables, especially amongst related variables. Several of these highly collinear variables were removed. After all data cleaning and transformation steps, 50 are left variables.

3 Methodology

3.1 Overview

Regression models are used here since the response variable is continuous. Our goal is to establish a baseline using ordinary least squares regression and then try a variety of techniques to improve upon this baseline. One of the key problems from exploratory analysis was collinearity amongst predictor variables, a problem that can derail assumptions in linear regression. In addition to basic OLS linear regression, we also look into methods that can mitigate collinearity to improve the fit.

3.2 Summary of Models

Ordinary least square regression was used as our baseline method. It can represent the linear relationship between the response and predictor variables. The estimated vector of our response variable, `SalePrice`, is given by $\mathbf{y} = \mathbf{X}\beta + \epsilon$, Where $\mathbf{y}$ is the response vector, β is the vector of coefficients, $\mathbf{X}$ is the data matrix and ϵ is the error term assumed to be i.i.d. normal with mean 0 and variance σ^2. Coefficient estimates, β, are selected as to minimize the Root Mean Square of Error (RMSE) for the model.

Coefficient etimates are determined by fitting data. When the model is built, $\hat{\beta}$ that minimize the total error in the model is the optimal. To find optimal coefficients, ordinary least square method is used. In addition, the linear regression follows five key assumptions:

1. There is linear relationship between variables and response.
2. All variables have a multivariate relation.
3. Predictor variables are not collinear.

4. Residuals are independent.

5. Error terms are evenly distributed.

If these assumptions are violated, it could cause problems for our model's predictability and interpretability. Preliminary analysis shows that assumption 3 is violated for the baseline model. Correlated error terms are also a major concern. Methods selected here will seek to make our model conform better to these assumptions.

Stepwise selection can help linear regression by reducing the number of variables. Through a selection process, only significant variables are kept in the model while unnecessary variables will be removed. Three direction can be used in this.

The forward stepwise selection method starts with a null model containing no variables. It builds a model by sequentially adding variables into the model based on which ariable will improve the model the most at each step. The added variable will stay in the model permanently. The improvement of models is evaluated by criteria called AIC or BIC. The process is repeated until the performance of model cannot be improved anymore. On the other hand, backward elimination starts with a full model with all variables. It builds a model by sequentially removing variables based on what removal decision improves the model the most. Bidirectional elimination combines combines previous two method and starts with null model. At each step, forward selection is applied to add new variables into the model and then backward elimination is applied to see if removing any variables can improve the model more.

The family of regularized regression methods include LASSO, Ridge and Elastic Net regression. Based on ordinary linear regression, the model is also subject to a limitation. It can also be represented as adding a penalty to the model to limit the number of variables. The regularization amount is determined by a shrinkage parameter called lambda. This hyperparameter will affect the rate of shrinking coefficient and selection results. The objective function for all three methods, defined by the package we use called `glmnet`[3] , is given by,

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} RMSE + \frac{\lambda[(1-\alpha)\left\|\beta\right\|_2^2}{2} + \alpha\left\|\beta\right\|_1]$$

where the goal is to select the optimal β coefficients given values for the parameters λ and α. Notation $\left\|\cdot\right\|_1$ and $\left\|\cdot\right\|_2$ represent the L1 norm and the L2 norm, respectively. LASSO regression is when $\alpha = 1$ and ridge regression is when $\alpha = 0$. Elastic Net is any value of $\alpha \in (0, 1)$. An optimal model is selected by tuning λ for all three methods and then α for Elastic Net. LASSO will set zero coefficients to unimportant variables, whereas ridge regression will shrink but not entirely eliminate any variables. The optimal coefficients for α and λ determine the optimal ratio of the L1 and L2 norm penalization to include in the model.

Regression tree differs from all previous regression method as a non-linear regression method without limitation on linear regression assumption. It is also a non-parametric regression method which does not have an analytical solution. Tree model is more commonly used in classification problem. The tree is built from a recursive binary partitioning process. The optimal split is done at each partition step and leads to a greedy splitting result, so the final tree can be viewed as basis expansions of simple functions. In detail, one variable is partitioned into segments if it is continuous type of data at each step and these partitions are used as conditions to separate predicted outcomes. The step is repeated for all variables in the training set. Then the mean response in each leaf of the tree will be the estimated response. Optimal regression tree can be found by selecting the values of how to split variable at each step and how the variables are segmented. After each step, the split point will partition the data into two parts. The optimal point will be the one that minimize the sum square of error from both regions after splitting.

3.3 Cross-validation

Cross-validation(CV) helps with avoiding overfitting during the modeling process. Here, 10-fold CV technique is applied on the training dataset. All models are built with cross-validation. For CV, it works in a way that different training and testing splits are selected across k folds. In our case, the data is split into 10 fold and this 10 fold is fixed. At each time, 9 folds of the data will be selected to be training data and used to train the model. 1 fold was left as the testing data without getting in the model building. Then this process is repeated for 10 times. Each

fold will only be selected once as test dataset. The final results(i.e. model metrics) are evaluated and compared based on CV.

3.4 Performance Metrics

Analysis was performed on the preprocessed dataset using each of the techniques described in the previous section. Final performance was measured using root mean squared error (RMSE) averaged over 10 fold cross-validation for each model. The 10 fold cross-validated RMSE for a model is given by,

$$RMSE_{CV} = \frac{1}{10}\sum_{k=1}^{10}\sqrt{MSE_k}$$

A low cross-validated RMSE score indicates better model performance. Since there was no training set available, cross-validation is used to gauge the performance of the models on new data. R^2 is used as a secondary metric to compare models when their RMSE scores are quite close.

4 Analysis

4.1 Ordinary Least Squares Regression

Ordinary least squares (OLS) with all the preprocessed variables included was used as the baseline model. This model gives the best linear fit to the data using all 50 predictor variables with only cursory feature selection. Variable significance was determined using the P-values from tests on of whether β_k is statistically different from 0. Some of the most significant variables with P-values very close to zero were the overall quality of the house, the overall condition, and ground living area, all of which are intuitively considered important in detemrining housing price. The condition of the sale, indicating whether or not a sale was made under normal circumstances or otherwise (such as in a foreclosure), was also a highly significant factor, where a normal sale had a major positive association with sales price. Other critical factors that had positive associations with sales price were the area of the basement, area of the garage, number of bathrooms, and the combined factor representing the total porch size.

This model had an $RMSE_{CV}$ of 0.119 and a cross-validated R^2 value of 0.9089. These values will serve as a baseline for comparison of future models going forward.

In addition to this full model, we also applied OLS to an abridged models with one feature included for each of the intuitive components of a home. The intuition was that a model like this could be more interpretable with fewer collinear predictor variables from naturally separated classes. For instance, out of several predictors assessing the quality and size of a garage, only `GarageArea` was kept given its significance in the earlier model. The same was done for other 'categories' of predictors related to features like the bathroom, garage, kitchen, and others. In this abridged model, only 11 predictors were used including total number of bathrooms, kitchen quality, condition of the sale, zoning classification, and several others with one from each mutually exclusive 'category' of predictors. A full overview of predictors and their coefficients for this model can be seen in Figure 12 in the Appendix.

The final $RMSE_{CV}$ for this abridged model was 0.1337, slightly lower than the full model. However, the R^2 only dropped to 0.8874, indicating that these 11 predictors explain only 2 percent less of the variation in the response variable than the full 50 predictors. This supports the intuition that more careful variable selection should be undertaken to maintain a high R^2, while also improving the accuracy of the model.

There were also several key variables with high VIF scores including the MSZoning dummy variables with a maximum score of 18.6 which were highly significant in the original model and KitchenQuality with a maximum score of 6.84 in the abridged model. This points to a violation of the uncorrelated predictors assumption that we will seek to solve in future models.

4.2 Stepwise Regression

We used both forward and backward selection to try to improve upon our naive baseline OLS models. After expanding the categorical predictor variables, our model can include up to 185 individual variables with dummy variables. Both forward and backward selection attempted to fit models using up to 185 variables. Forward selection found a model with minimum $RMSE_{CV}$ with 137 predictors, whereas backward selection found an optimal model with only 119 predictors. Both were implemented using the `leaps` R package with `caret` [4] for cross-validation. Plots of the $RMSE_{CV}$ against model size for both processes are given in the Appendix in Figure 13.

The best $RMSE_{CV}$ for forward selection was 0.1163 and 0.1191 for backward selection. R^2 was 0.9147 and 0.9108527 for forward and backward selection, respectively. In both criteria, forward selection found a slightly better model than backward selection, and improved $RMSE_{CV}$ over the full baseline model. Backward selection performed nearly identical to the full baseline model, but with a higher R^2 and fewer predictors. In these respects, both models proved to be modest improvements over the baseline model.

Intuitively, these improvements could have come by simplifying the binary factors derived from the categorical variables. For instance, the predictor variable GarageType includes levels 'Attached', '2Types', 'Basement', 'BuiltIn', 'CarPort', 'Detached', and 'NoGarage', but only two of these levels contain more than 100 observations. Both of the selection algorithms tended to not include these rarer factors, thus reducing the dimensionality of the model without reducing the overall score. However, it's likely that predictions on houses with rarer attributes will prove more difficult, as rare attributes can sometimes indicate a different home class (such as a pool).

4.3 Regularized Regression

After stepwise selection, we used several regularized regression techniques to relax the constraint of needing to add or remove variables entirely. LASSO, ridge, and Elastic Net models were constructed using the cross-validation package `caret` and `glmnet`. Cross-validation methods were used to select an optimal value of λ for each method and an optimal norm ratio parameter α. In general, these methods performed quite well due to their ability to minimize the modeling noise generated from collinear variables. Results from each method, including their optimal parameters, are shown in Figure 3.

Method	λ^*	α^*	$RMSE_{CV}$	R^2
LASSO	0.002742	1	0.11566	0.9159
Ridge	0.029836	0	0.11878	0.91096
Elastic Net	0.008909	0.25	0.11614	0.91515

Figure 3: Summary of regularization method results.

Both Elastic Net and LASSO regression improved over the baseline OLS model on $RMSE_{CV}$, while ridge regression was nearly identical. However, all regularization models improved over it in terms of R^2.

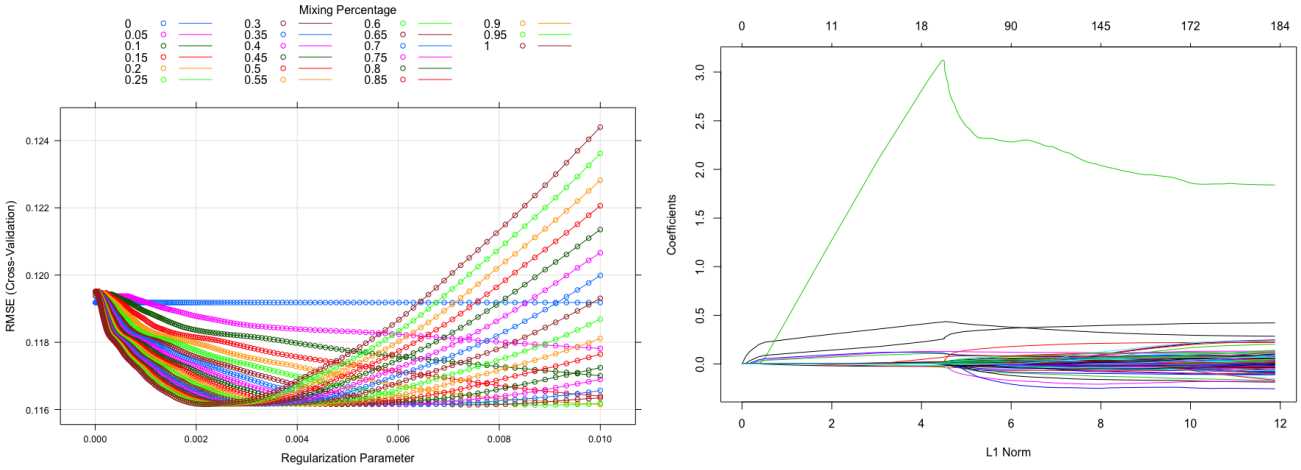


Figure 4: Cross-validated optimization of Elastic Net (left). The mixing percentage is the value of α represented by color with λ on the X axis. This graph shows the changes in dynamics of $RMSE_{CV}$ as the objective function changes from ridge ($\alpha = 0$) to LASSO ($\alpha = 1$) regression. On the right side is a visualization of the value of the unstandardized coefficients as the L1 norm in Elastic Net shrinks for the optimal model. Variable names are omitted for space, but in general it shows that as λ shrinks, so does the value of coefficients deemed less critical to the model. Certain variables are retained more than others by Elastic Net.

4.4 Tree Regression

Tree regression was also used to predict house selling prices. An optimal tree was found via cross-validation by optimizing against the complexity parameter (CP). Due to the large number of factors in our model, a tree seemed like a valuable way to visualize the relationships and importance of various factors. An optimal CP of 0.000302 was selected and produced a tree with an $RMSE_{CV}$ of 0.1848, much lower than the baseline model.

Despite the poor predictive performance of tree regression, the variables selected for the initial splits in the optimal model can shed light on the effects of certain variables on the response. Similar to what was observed in the baseline model, overall house quality, living area, and number of bathrooms are the three earliest splits that explain the highest portion of the deviance in the model.

4.5 Residual Analysis

While linear models were used here, it seems that two key assumptions of linear regression are violated from the beginning of the model. Multicollinearity is present amongst the predictor variables and the residuals appeared to have non-constant variance. In the original model, a Breusch-Pagan test rejected the null hypothesis that there was constant variance among the residuals. To mitigate this, we used a log transformation of the response variable, `SalePrice`, in the models listed here. We've shown that systematic processes for feature selection such as stepwise selection and shrinkage methods are able to improve performance over the original baseline by removing multicollinearity. However, some non-constant variance amongst the residuals remains.

A summary of residuals with overlays for each model is shown in Figure 5 to get a feel for the overall distribution of residuals across models. Despite the log transformation of the response variable, heteroscedasticity is apparent for nearly all models as the variance of residuals at lower fitted values of sales price appears much higher than at central or higher values. In addition, when plotting each model separately, it becomes apparent that the first two OLS models appear to be systematically overestimating the value of the selling price, whereas all other models tend to sysematically underestimate them. Nonetheless, the absolute value of the residuals tends to be higher for lower selling prices across all models.

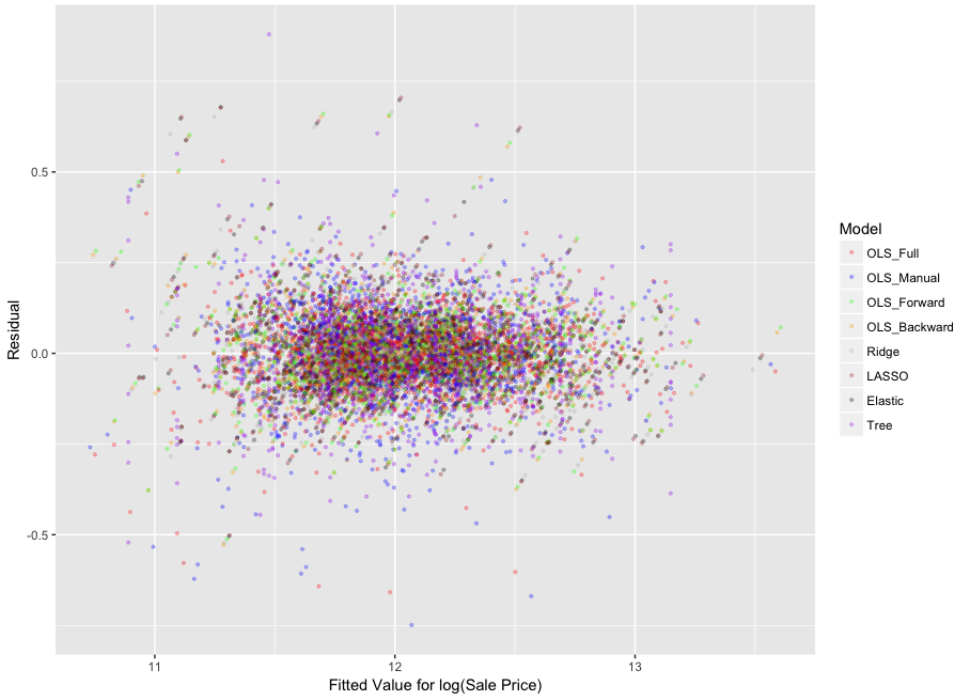


Figure 5: The residuals for each model plotted against the predicted value of the log of sales price is shown above. This overlay reveals some systematic trends in the residuals across groups of models.

4.6 Model Comparison

Amongst the 8 models, LASSO regression achived the lowest $RMSE_{CV}$ score and the highest R^2 making it the strongest model for predicting housing prices according to our analysis. In addition, 4 models defeated the baseline method including the three regularization methods and forward stepwise selection.

Method	$RMSE_{CV}$	R^2
All Predictor OLS (Baseline)	0.119	0.9089
OLS with Manual Feature Selection	0.1337	0.8874
Forward Stepwise Selection	0.1163	0.9147
Backward Stepwise Selection	0.1191	0.9109
LASSO	**0.1157**	**0.9159**
Ridge	0.1188	0.911
Elastic Net	0.11614	0.9152
Tree Regression	0.1848	0.7909

Figure 6: Summary of results across all 8 models. LASSO regression had the lowest cross-validated root mean squared error and the highest R^2 out of all models, followed closely behind by Elastic Net and the model selected via forward stepwise selection.

Despite LASSO regression having the best results amongst our performance metrics, it's worth noting that the standard deviation of its $RMSE_{CV}$ score was particularly high compared to other models. These standard deviations can be shown in the error bars in Figure 7. As cross-validation is used to estimate a model's generalizability to new data, this standard deviation score is an indicator of the reliability of a model across different datasets. For this reason, Elastic Net or forward stepwise regression may prove a more reliable model on a new dataset when factoring in the standard deviation of its $RMSE$.

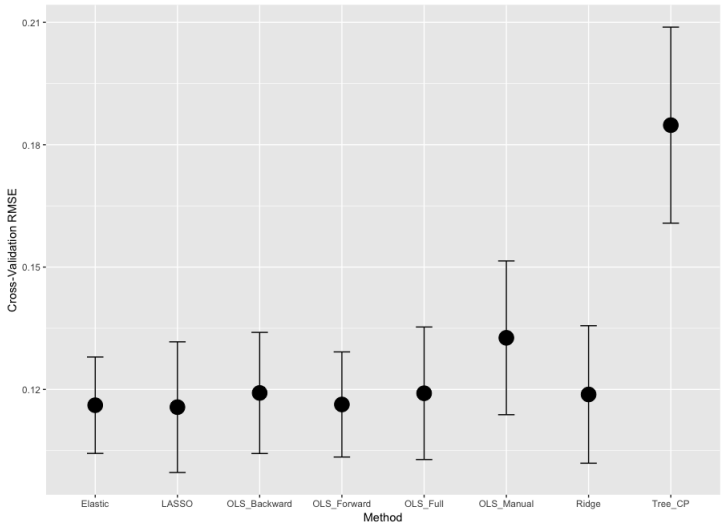


Figure 7: Plot of cross validated $RMSE$ by model accompanied by error bars indicating the standard deviation of the $RMSE$ across 10 fold cross-validation.

5 Conclusion

Across 8 different regression models, we found that regularization and stepwise selection methods were effective at improving model performance in the presence of multicollinearity amongst the predictor variables. LASSO regression, forward stepwise selection, and Elastic Net regression all appear to select stronger models than a baseline model that naively includes all predictors or another OLS model where features have been hand-picked. By using 10 fold cross-validation, we were able to provide an estimate of the applicability of these models to new datasets and could recommend any of these three models as effective tools for predicting the selling price of homes in Ames, Iowa for mid-to-high price houses. Further study is needed to reduce the effects of heteroscedasticity driven by high residuals amonst low price homes.

References

[1] B. Nielsen, "Why Housing Market Bubbles Pop," "https://www.investopedia.com/articles/07/housing_bubble.asp", 2017.

[2] Kaggle, "House Prices: Advanced Regression Techniques," "https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data", 2016.

[3] T. Hastie and J. Qian, "Glmnet Vignette," "https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html", 2014.

[4] M. Kuhn, "Caret," "https://topepo.github.io/caret/model-training-and-tuning.html", 2008.

6 Appendix

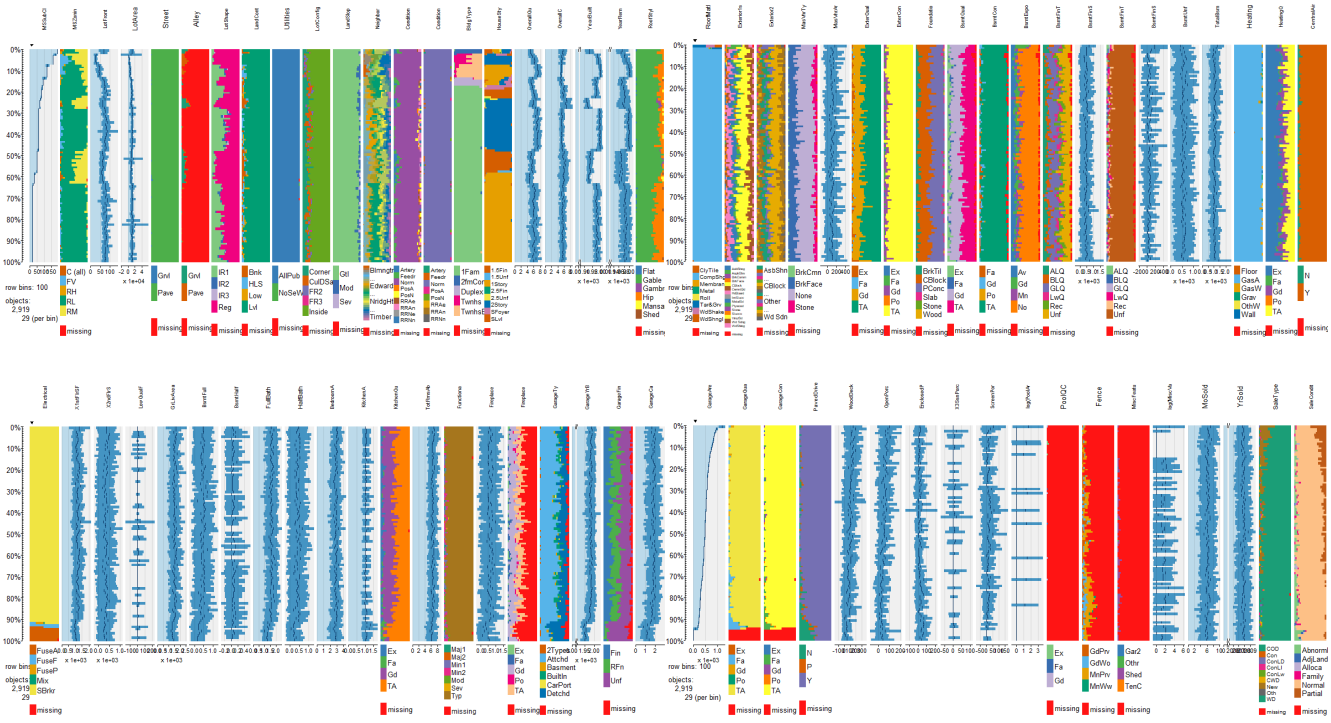


Figure 8: A summary visualization of all the predictor variables provided in the dataset. Histograms represent numeric variables and colored columns represent factor variables. Columns dominated by one color have a largely homogeneous factor distribution. Red bars denote missing or NaN data.

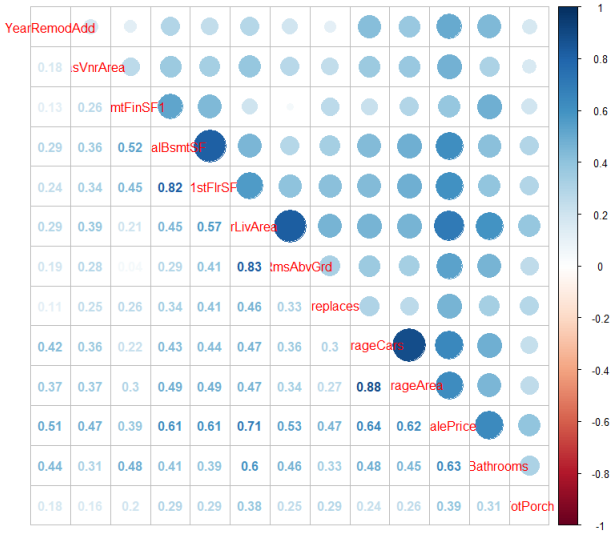


Figure 9: Correlation plot amongst numeric variables.

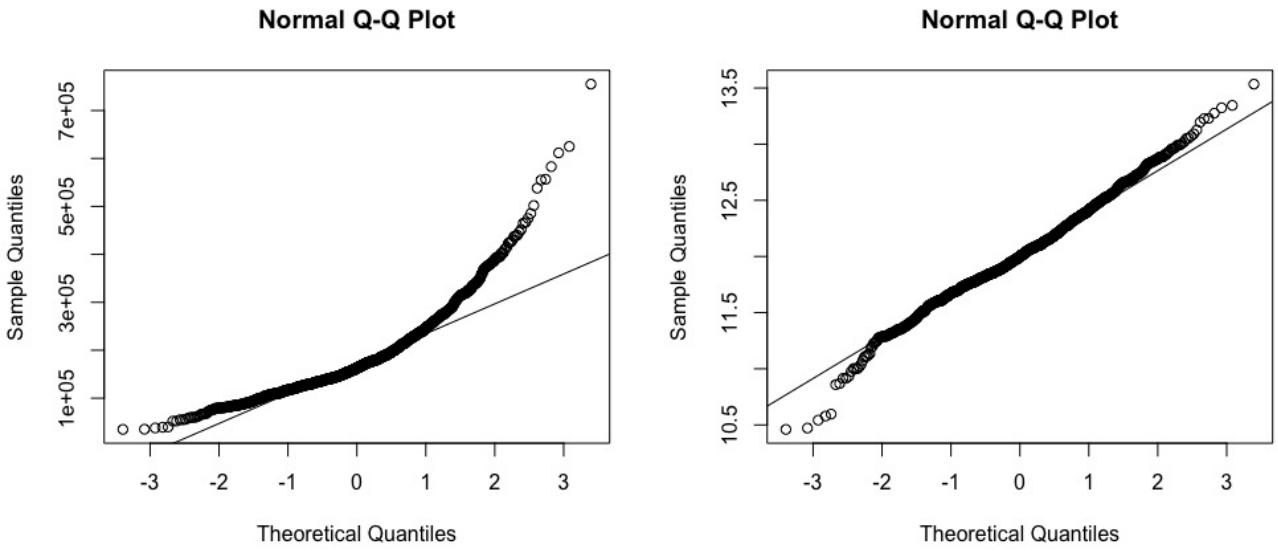


Figure 10: Normal Q-Q Plots for the observed value of the response variable `SalePrice` to see if the data is normally distributed. It seems to be highly skewed in upper quantiles (left) for the untransformed response. The log-transformed (right) response variable mitigates this skew and should improve problems with non-constant variance amongst the residuals.

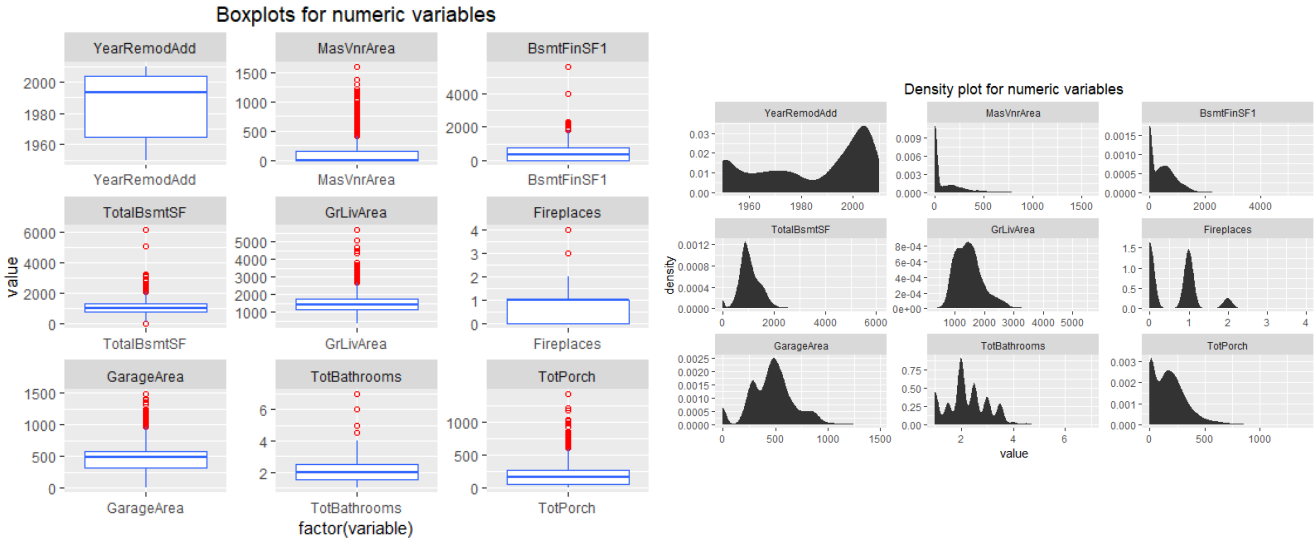
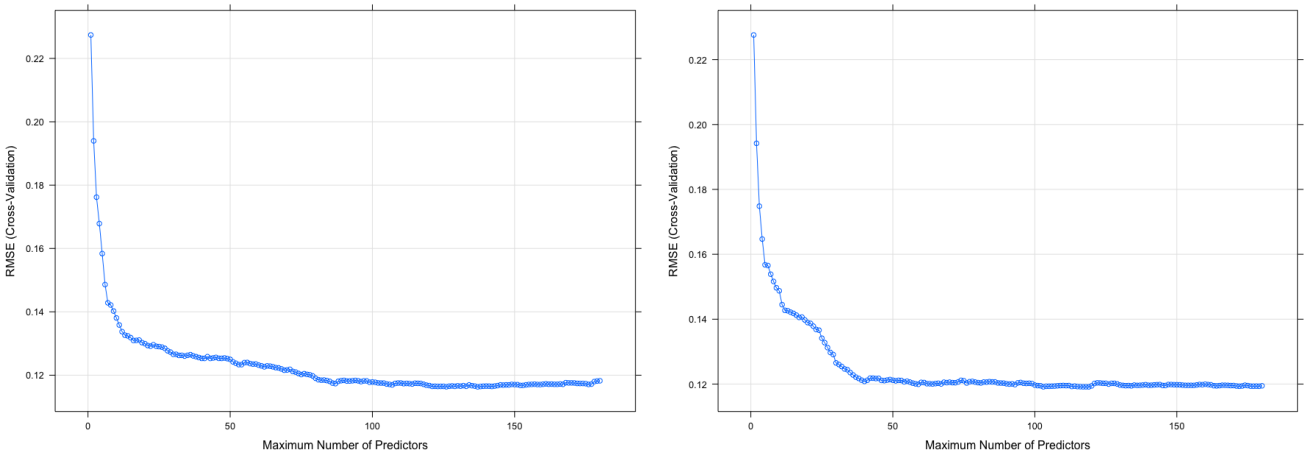


Figure 11: Plots showing the distribution of various numeric predictor variables. In general, many variables are all or nothing resulting in major outliers for some categories. This is likely due to the presence of a few homes with high selling prices and large values for many features like `GarageArea` or `TotPorch` whereas many homes don't have garages or porches.

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         10.10333    0.05431 186.040  < 2e-16 ***
TotBathrooms         0.18235    0.01464  12.455  < 2e-16 ***
SaleConditionAdjLand 0.06301    0.06808   0.926 0.354835
SaleConditionAlloca  0.12556    0.04192   2.995 0.002787 **
SaleConditionFamily  0.03900    0.03237   1.205 0.228364
SaleConditionNormal  0.09739    0.01397   6.971 4.78e-12 ***
SaleConditionPartial 0.16860    0.01898   8.882  < 2e-16 ***
GarageArea           0.12885    0.01106  11.650  < 2e-16 ***
KitchenQualFa       -0.17726    0.02916  -6.080 1.54e-09 ***
KitchenQualGd       -0.06928    0.01561  -4.438 9.79e-06 ***
KitchenQualTA       -0.11482    0.01792  -6.409 1.99e-10 ***
GrLivArea            0.34189    0.01586  21.552  < 2e-16 ***
TotalBsmtSF          0.18903    0.01217  15.532  < 2e-16 ***
OverallCond          0.22295    0.01899  11.743  < 2e-16 ***
OverallQual          0.52833    0.02668  19.802  < 2e-16 ***
BldgTypeZfmCon      -0.07207    0.02455  -2.936 0.003377 **
BldgTypeDuplex      -0.09768    0.02052  -4.760 2.14e-06 ***
BldgTypeTwnhs       -0.08025    0.02178  -3.685 0.000237 ***
BldgTypeTwnhsE       0.01086    0.01417   0.767 0.443447
Condition1Feedr      0.06605    0.02427   2.721 0.006578 **
Condition1Norm       0.11764    0.01975   5.955 3.26e-09 ***
Condition1PosA       0.04285    0.05044   0.850 0.395725
Condition1PosN       0.08984    0.03662   2.453 0.014276 *
Condition1RRAe       0.04589    0.04401   1.043 0.297236
Condition1RRAn       0.10267    0.03222   3.187 0.001471 **
Condition1RRNe       0.09336    0.09464   0.987 0.324033
Condition1RRNn       0.12191    0.06181   1.972 0.048746 *
MSZoningFV           0.32062    0.04000   8.015 2.27e-15 ***
MSZoningRH           0.27224    0.04847   5.616 2.34e-08 ***
MSZoningRL           0.30994    0.03603   8.602  < 2e-16 ***
MSZoningRM           0.18288    0.03697   4.947 8.44e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

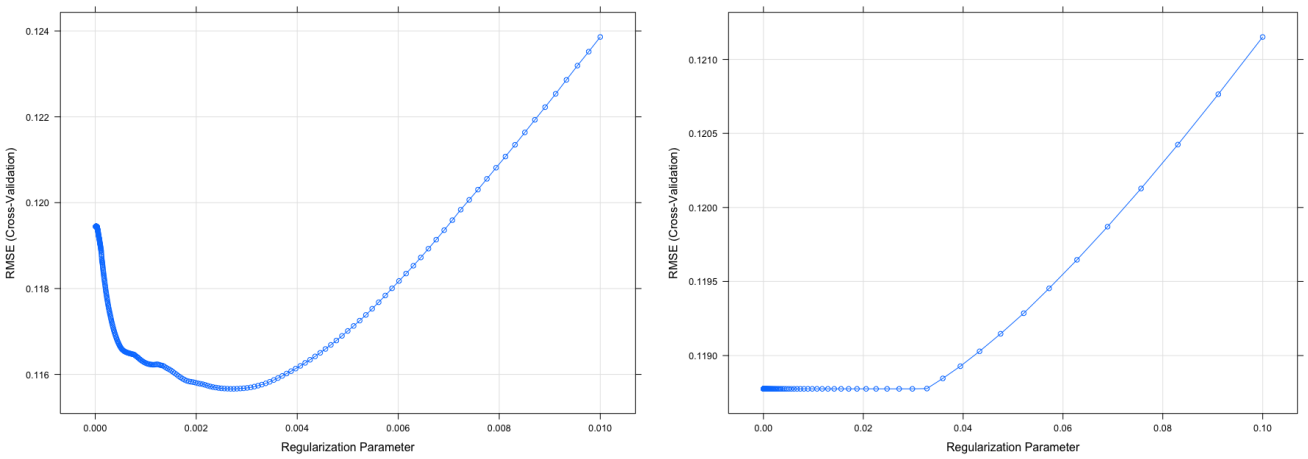
Figure 12: Summary output for each of the predictor variables included in the abridged OLS model.



(a) Forward selection cross-validation

(b) Backward selection cross-validation

Figure 13: $RMSE_{CV}$ plotted against model size generated by forward selection (left) and backward selection (right). Optimal sizes were 137 and 119, respectively. It can be seen here that while forward selection found a model with a lower $RMSE_{CV}$, backwards selection appears to have converged at a lower number of predictors.



(a) LASSO cross-validation

(b) Ridge regression cross-validation

Figure 14: $RMSE_{CV}$ scores across a range of values of λ.