Predicting House Price in Ames, Iowa Using Advanced Modeling Techniques

Samantha Benedict
Amanda Mari
Brittany Schimmenti

Professor Chun Pan
STAT 707
City University of New York, Hunter College

1. Introduction

The goal of this report is to find the model that most accurately predicts the sale price of houses in Ames, Iowa. This will be explored through the use of advanced modeling techniques.

1.1 Data Cleaning

When looking at our data we focused on the NA's throughout the training set, multicollinearity, insignificant predictors, skewness, and different variable scales. Within the data set, most NA's were true NA's, with the exception of lot frontage. We then decided to use median imputation by neighborhood to fill in those missing values. Next, we looked at multicollinearity. We dropped the variables that were highly correlated with each other ($r > .8$), and out of each pair we kept the variable that was most correlated with sale price.

We also one-hot encoded the variables prior to modeling, which means creating dummy variables for each level of a factor. From here on, when "variables" are referred to in this paper, it refers to the dummy coded factor and numeric variables, of which there are 171. When "predictors" are referred to in this paper, it signifies the total number of factor (26) and numeric (53) variables. There are 79 predictor variables when "ID" and "SalePrice" are removed from the dataset. We removed dummy variables with less than ten 1's in our training set.

We also employed feature engineering and binning. We combined multiple variables related to one another to create new variables that could be used for prediction. The variables created were TotalPorchSF, TotalSqFeet, and TotBathrooms as shown in **Equation's 1, 2, and 3.** Lastly, we binned all 25 neighborhoods into three categories by wealth to create the variable NeighRich.

$$TotalPorchSF = OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch \quad \textit{Eq. 1}$$

$$TotalSqFeet = GrLivArea + TotalBsmtSF \quad \textit{Eq. 2}$$

$$TotBathrooms = FullBath + (HalfBath*0.5) + BsmtFullBath + (BsmtHalfBath*0.5) \quad \textit{Eq. 3}$$

1.2 Background

After doing some background exploration on housing price fluctuation, we chose several variables that we suspected to be correlated with sale price. According to research, these variables included total square footage, garage size, and location.[1] We created scatterplots for total square footage and garage area against sale price shown in **Figures 1 and 2**. Both **Figures 1 and 2** show a positive correlation with some outliers within the graph. We then created a boxplot with neighborhood and sale price as seen in **Figure 3**. We saw there are outliers as well, but sale price varies greatly between neighborhoods in Ames, Iowa.

Figure 1 Sale Price vs. Total Sq. Footage

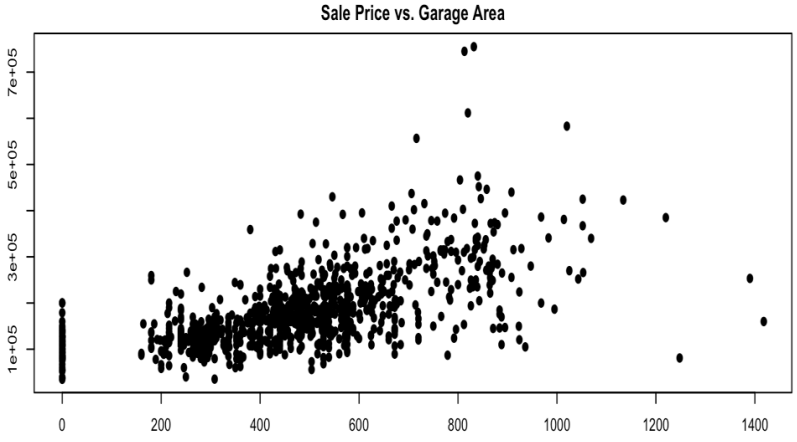

Figure 2 Sale Price vs. Garage Area

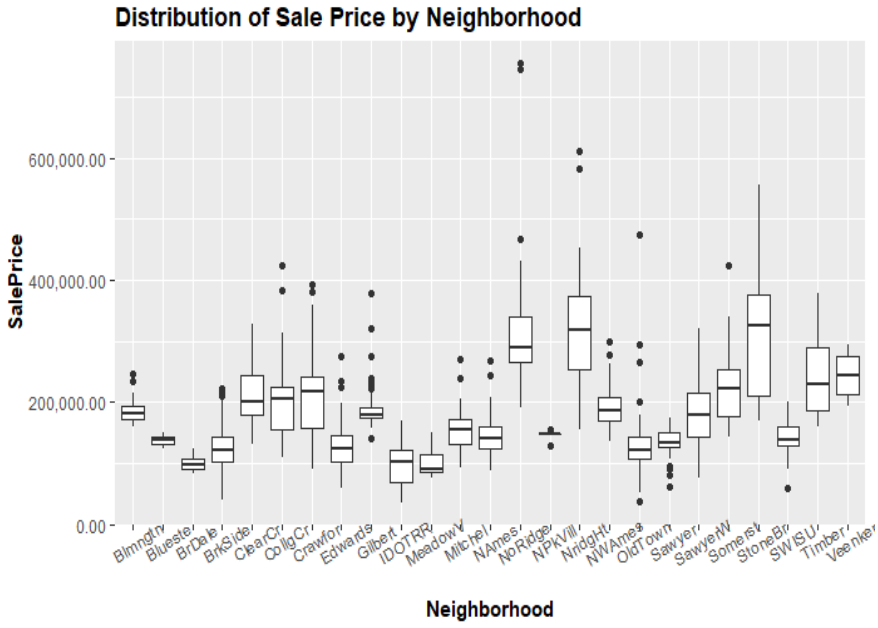

Figure 3 Sale Price vs Neighborhood

1.3 Data Exploration and Visualization

Before we could fit a model, we first looked at the distribution of our response variable, sale price. It can be seen in **Figure 4** that sale price is right skewed so we decided to transform it to see if we could remove this skewness. We chose to take the log of sale price. This then normalized the variable as shown in **Figure 5.** For all of the prediction models later tested, we used the log transformed response variable.

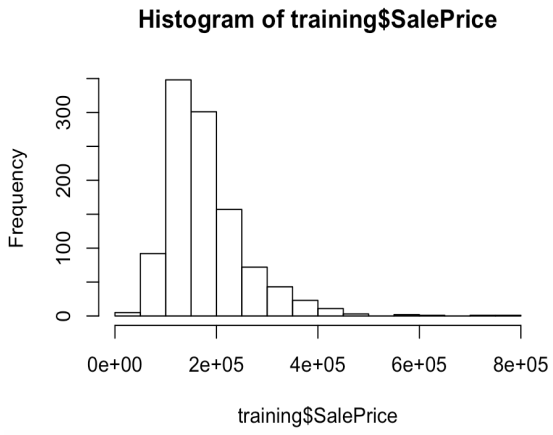

Figure 4 Histogram of Sale Price

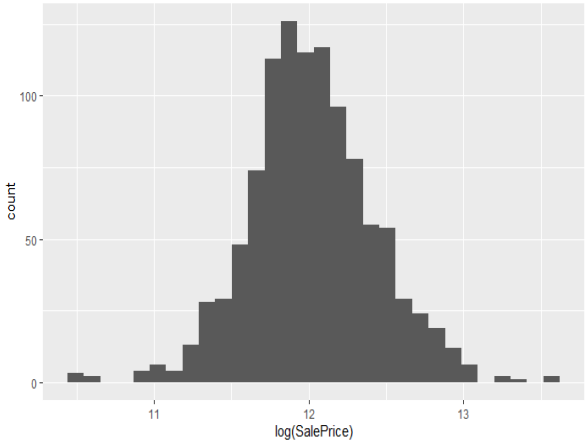

Figure 5 Histogram of Log(Sale Price)

In order to determine what other variables might be significant in predicting sale price, we've plotted all the correlations with the response variable that had a correlation value greater than 0.5 in either direction. This can be seen in **Figure 6**. These correlations have been listed in descending order for easy comparison. It can be seen that some of our original research variables are confirmed in the plot, i.e. total square footage and garage cars (similar to garage area which we originally explored).

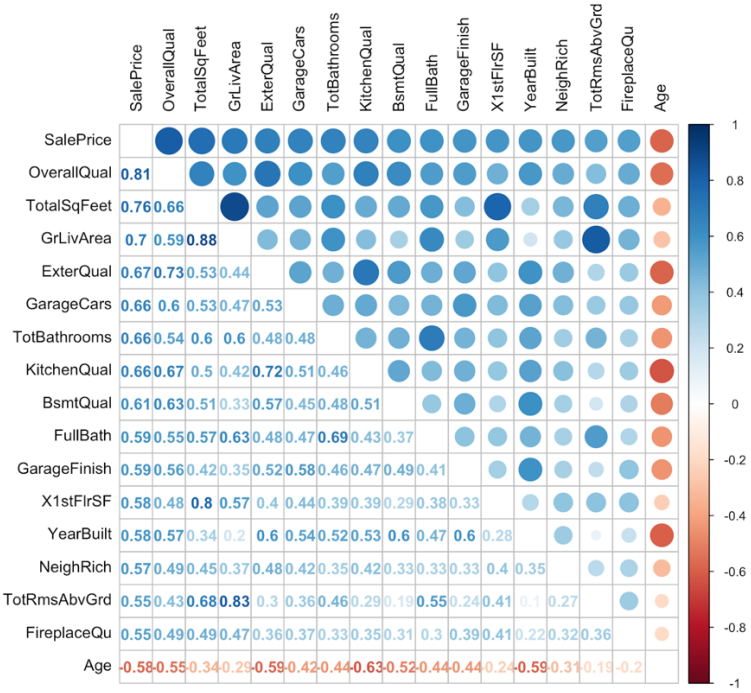

Figure 6 Correlation Plot with r > .5 in Either Direction

Figure 6 also suggests some potentially important variables that did not appear in our background research and therefore we have not yet visualized. These include overall quality and total number of bathrooms. These variables can be seen in **Figures 7 and 8** plotted against sale price. Both **Figures 7 and 8** indicate positive trends – as each variable increases, sale price increases. The only exception to this is in **Figure 7.** While we do see a positive correlation with total number of bathrooms, the majority of homes in the dataset have four or less bathrooms. With minimal data beyond four bathrooms, our trend seems to drop off.

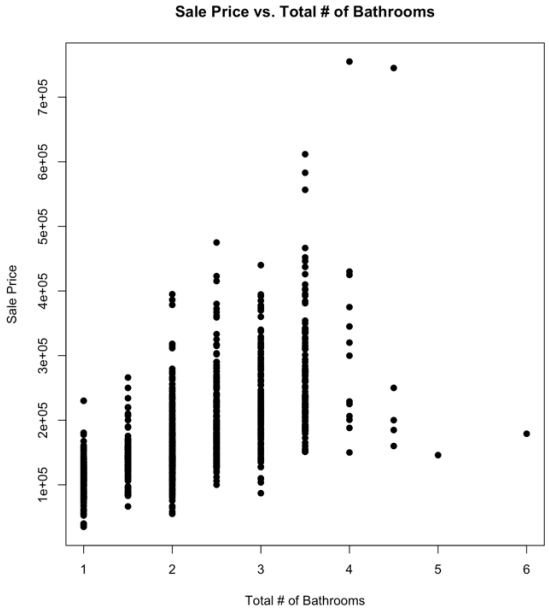

Figure 7 Sale Price vs. Total No. of Bathrooms

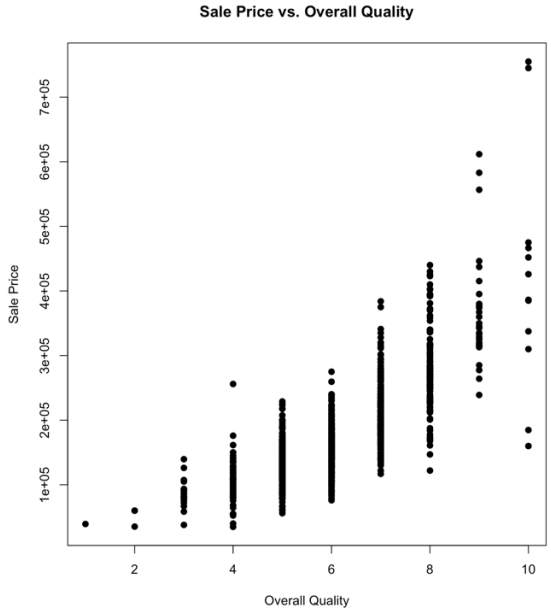

Figure 8 Sale Price vs. Overall Quality

2. Methods

2.1 Model Specification

In order to fit a strong prediction model, we've looked at four types of models and compared each based on certain criteria. The modeling methods we've explored include general linear regression, random forest regression, LASSO regression, and XGboost. In order to test the significance of our research variables, we ran the first two models, general linear regression and random forest regression, with only our research variables. Next, in effort to see if we were missing any important predictors, we fed both the LASSO regression model and the XGboost model all of the variables in our training set and allowed the model to determine the most important features. From here we compared the variable selection in each model to determine variable significance. All models were run with the log of sale price.

After looking at these four models, we set our model selection criteria based on several factors which were not limited to variable selection. Model selection was also determined based on prediction validation. To do this, we created a 70-30 split of our training data in order to do a within-dataset prediction validation. Additionally, predictions were made from the full training set and compared against the test dataset. The best model was chosen based on the lowest RMSE.

2.2 Model Fitting

2.2.1 General Linear Regression

The first model tested was a general linear regression model using the aforementioned research variables. These variables include, grand living area, total number of bathrooms, garage cars, overall quality, and neighborhood. It is important to note that after removing insignificant variables, we were only left with 22 neighborhoods This led to a 25-variable model that tested our 5 predictor variables. The regression equation can be seen in **Equation 4** and the regression output is visible in **Figure 9.**

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \ldots \beta_{25} X_{i,25} + \varepsilon_i \qquad \textit{Eq. 4}$$

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       11.377734   0.052033 218.662  < 2e-16 ***
GrLivArea          0.110474   0.007847  14.079  < 2e-16 ***
TotBathrooms       0.054573   0.006979   7.819 1.30e-14 ***
GarageCars         0.057275   0.006849   8.362  < 2e-16 ***
NeighborhoodBrDale  -0.305139   0.058239  -5.239 1.95e-07 ***
NeighborhoodBrkSide -0.043752   0.042609  -1.027 0.304751
NeighborhoodClearCr  0.186475   0.049248   3.786 0.000162 ***
NeighborhoodCollgCr  0.068141   0.036673   1.858 0.063440 .
NeighborhoodCrawfor  0.136551   0.042772   3.193 0.001453 **
NeighborhoodEdwards -0.089397   0.039382  -2.270 0.023412 *
NeighborhoodGilbert -0.003153   0.040235  -0.078 0.937545
NeighborhoodIDOTRR  -0.251158   0.046924  -5.352 1.07e-07 ***
NeighborhoodMeadowV -0.150365   0.057473  -2.616 0.009020 **
NeighborhoodMitchel  0.023472   0.044386   0.529 0.597049
NeighborhoodNAmes    0.015683   0.036661   0.428 0.668888
NeighborhoodNoRidge  0.158635   0.044716   3.548 0.000406 ***
NeighborhoodNridgHt  0.190035   0.040119   4.737 2.47e-06 ***
NeighborhoodNWAmes   0.016441   0.040040   0.411 0.681440
NeighborhoodOldTown -0.157501   0.038981  -4.040 5.73e-05 ***
NeighborhoodSawyer   0.021895   0.040920   0.535 0.592719
NeighborhoodSawyerW -0.006324   0.040514  -0.156 0.875992
NeighborhoodSomerst  0.049263   0.039482   1.248 0.212415
NeighborhoodStoneBr  0.186366   0.059669   3.123 0.001838 **
NeighborhoodSWISU   -0.086031   0.053668  -1.603 0.109232
NeighborhoodTimber   0.120117   0.044641   2.691 0.007244 **
OverallQual          0.103814   0.006099  17.020  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1624 on 1034 degrees of freedom
Multiple R-squared:  0.8449,    Adjusted R-squared:  0.8411
F-statistic: 225.2 on 25 and 1034 DF,  p-value: < 2.2e-16
```

Figure 9 General Linear Regression Output

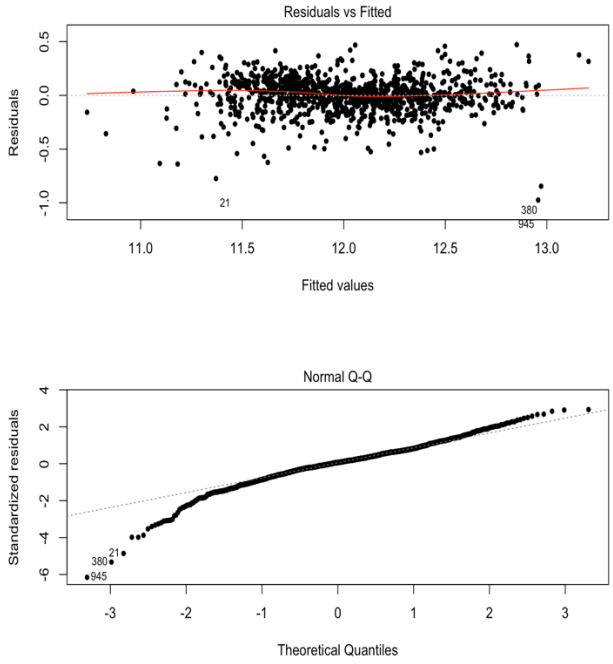

Figure 10 General Linear Regression Model Diagnostics

With an adjusted R^2 of about 84%, we can conclude that the model accounts for a fair amount of the variation in our response variable. In order to assess whether or not the model assumptions were satisfied, we have plotted the residuals against the fitted values and created a normal probability plot. **(Figure 10)** While the residuals show no distinct pattern, the normality plot indicates that the residuals violate the normality assumption, so we turn to another modeling technique in order to test our research variables.

2.2.2 Random Forest Regression

To further test our hypothesis on the most important predictors, we input our research variables into a random forest model. A random forest creates several decision trees that split the data into subsamples by using the most distinguishing features.[2] The trees begin with the most important feature used for classification, then continue to separate the sample by the other features into branches with "nodes." Each node contains the subset of the sample that falls into that category, as well as the predicted value for sale price using that criteria. For this model, we used 500 trees and 8 variables at each split. The random forest model can be seen below in **Figure 11**.

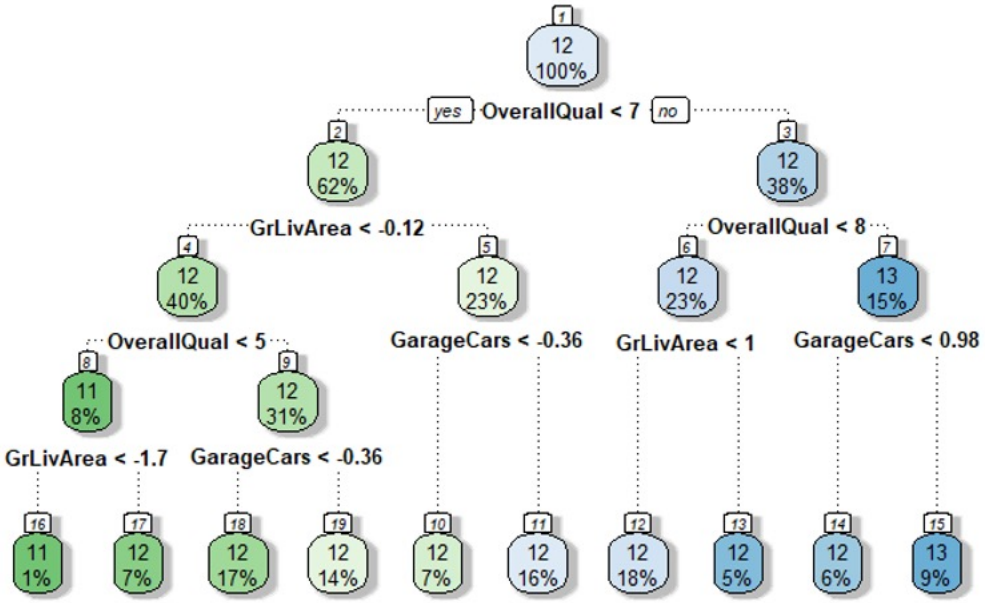

Figure 11 Random Forest Regression Model

This model explains 81% of the variance in sale price. It begins by splitting up the data by houses with overall quality less than 7. It then branches off into subsequent questions about the overall quality, above ground living area, and garage car capacity depending on the previous classification. The final predictions for the log of sale price for each subsample of data are in the last row of nodes in **Figure 11**.

2.2.3 LASSO Regression

To ensure that we are not missing any influential predictors in our model, we tried a LASSO (Least Absolute Shrinkage and Selection Operator) regression that incorporates all the variables. The formula for LASSO regression can be seen in **Equation 5**.

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j| \quad \text{where} \quad RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{i=1}^{p} \beta_j x_{ij} \right)^2 \qquad \textit{Eq. 5}$$

LASSO regression takes the residual sum of squares and adds it to the product of a tuning parameter, λ, times the sum of the absolute value of the predictor coefficients. LASSO does variable selection by forcing the coefficients of several variables to zero, as opposed to the similar method of ridge regression, which forces some coefficients toward zero. To find the optimal λ, we tried values from λ=.001 to λ=.1 in increments of .0005. In **Figure 12**, each value of λ is compared through its cross-validated RMSE on the training set.

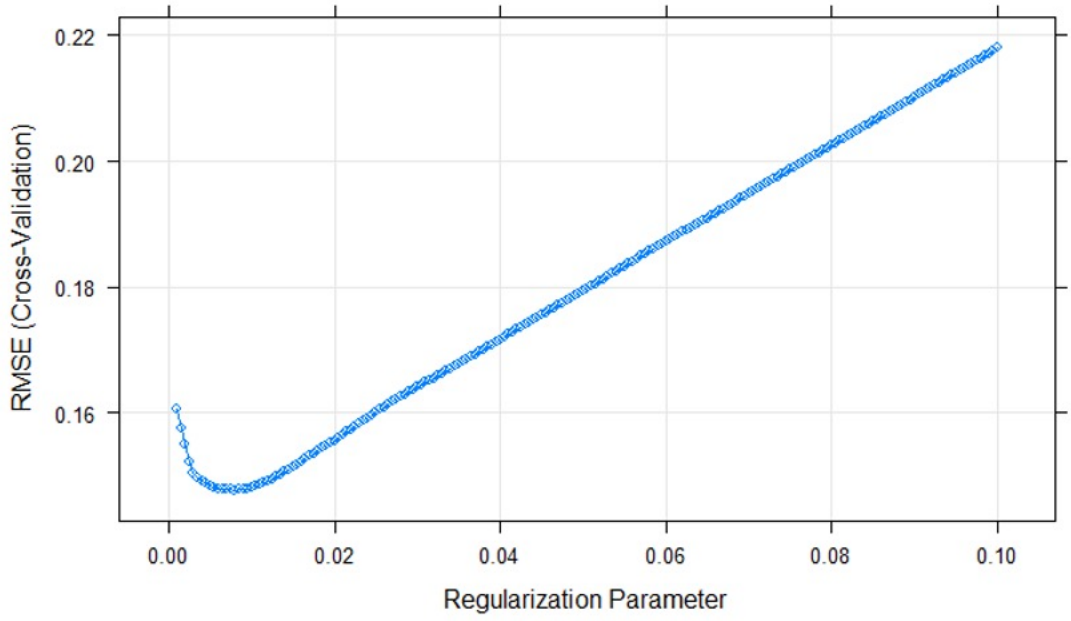

Figure 12 Choice of Tuning Parameter by Lowest RMSE

In **Figure 12**, the lowest RMSE is approximately .1478 which corresponds to λ=.008. Using .008 as the tuning parameter, the LASSO model has an R^2 of approximately .87. After fitting the model, LASSO retained 51 variables and did not choose the remaining 120. This translates to 41 predictors chosen and 38 removed. The first 30 variables selected have been plotted in **Figure 13**.

LASSO ranked these 30 variables as the most relevant in explaining sale price. The 21 variables not shown in **Figure 13** are:

"FoundationPConc","X1stFlrSF","KitchenAbvGr","MoSold10","FireplaceQu", "Functional","PoolQC","ScreenPorch","Fireplaces","Exterior2ndVinylSd", "GarageFinish","SaleConditionNormal","Street","ExterQual","LotConfigCulDSac", "SaleConditionPartial","YrSold2009","HeatingQC","LandSlope","BsmtUnfSF", and "TotalPorchSF"

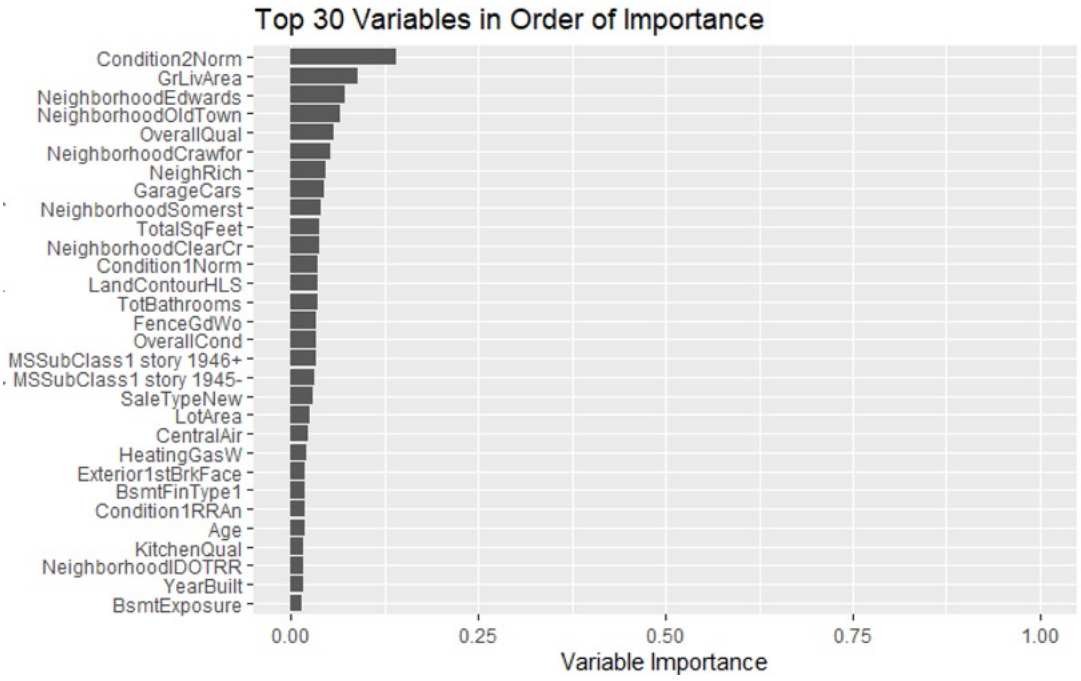

Figure 13 The Top 30 Variables Chosen by LASSO in Order of Importance

2.2.4 XGBoost

XGBoost is a machine learning algorithm that builds decision trees sequentially. It minimizes the error from previous models, while increasing the influence of high performing models.[3] XGBoost also uses boosting techniques to adjust each observation's weight based on classifications from prior models.

The XGBoost has default parameters, as well as three parameters that need to be set which include eta, maximum depth, and minimum child weight. Eta shrinks each subsequent feature's weight to prevent overfitting and ranges from 0 to 1, we chose eta=.05. Next, maximum depth refers to the maximum depth of the tree; we chose a maximum depth of 3 to avoid creating a more complex model. Finally, the parameter of minimum child weight refers to the minimum number of instances needed to be in each node. The default is 1, but we chose a value of 4 to make the algorithm more conservative. Then, using 5-fold cross validation and our parameters, we determined that the optimal value for "nrounds" (the number of trees to build) is 330. XGBoost's ranking of feature importance by relative contribution to prediction of sale price, or "gain", is depicted in **Figure 14**.

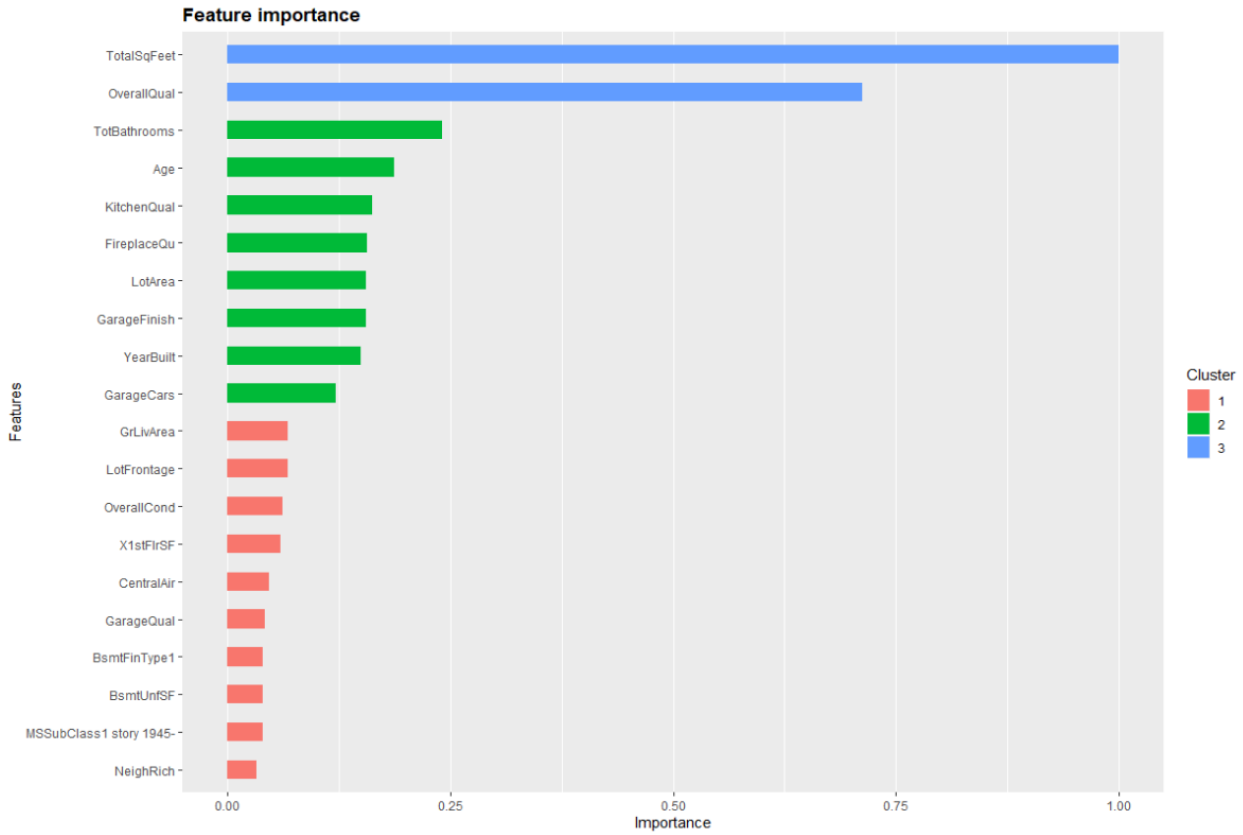

Figure 14 XGBoost Feature Importance

XGBoost ranked these variables in order of relative importance in prediction of sale price. The three clusters represent the groupings of importance. Total square feet and overall quality are the most important variables, followed by the 8 variables in cluster 2, and then the 10 less important variables in cluster 3. In total the XGBoost used 20 predictors in its ensemble.

2.3 Results

Table 1 displays our overall findings for each of the four models tested. The validation RMSE column indicates the RMSE scores from our 70-30 validation split, in dollars, and the test RMSE indicates the RMSE scores from the full training set prediction compared against the test set of data. While all of the models tested have comparable RMSE scores, it is clear that LASSO regression and XGBoost predicted the best. LASSO regression performed better in the 70-30 validation but XGBoost had a better test RMSE and a superior overall average prediction. The complete findings for the XGBoost model are listed in **Table 2**.

MODEL	MODEL QUALITY	VALIDATION RMSE (in $)	TEST RMSE (in $)
General Linear Model	Adjusted R-squared: 84.11%, F-Statistic: 225.2	26,485.58	29,014.85
Random Forest	IncNodePurity: Overall Quality, GrLiveArea, Garage Cars	29,049.08	30,580.66
Lasso Regression	Variable Retention: 53 used, 118 dropped	20,787.54	22,566.25
XGBoost	Number of Clusters: 3	22,515.56	19,283.85

Table 1 Model Fitting Results

MODEL	MODEL QUALITY	VALIDATION RMSE (in $)	TEST RMSE (in $)	TESTING MSE (in $)	BIAS (in $)	MAX DEVIATION (in $)	MEAN ABSOLUTE DEVIATION (in $)
XGBoost	Number of Clusters: 3	22,515.56	19,283.85	371,866,847	-660.8157	104,234.9	13,678.63

Table 2 Final Model: XGBoost Results

2.4 Final Model Diagnostics

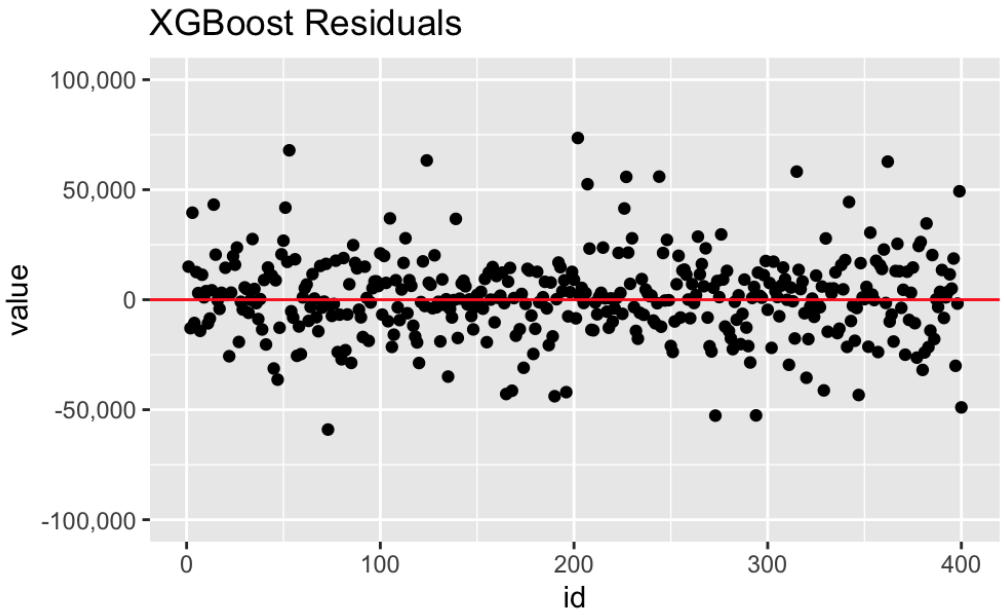

Figure 15 Residual Plot for XGBoost Model

After some research, it showed that there was no consensus on diagnostics for XGBoost. We decided to plot the residuals for the XGBoost model. This can be seen in **Figure 15**. When looking at the graph it shows some outliers on the top part of the graph with a few below as well. Overall there is no clear pattern seen throughout the graph.

3. Prediction

We plotted the predicted sale prices from our XGBoost model against the actual sale prices from our test data set to see visually how accurate our prediction model was. This is seen in **Figure 16**. On this graph, most of the predicted values are close to the actual sale price. The higher priced houses have greater error and have less consistent prediction. There are a few houses within the lower price range that have greater error as well, but most values were close to the actual sale price.

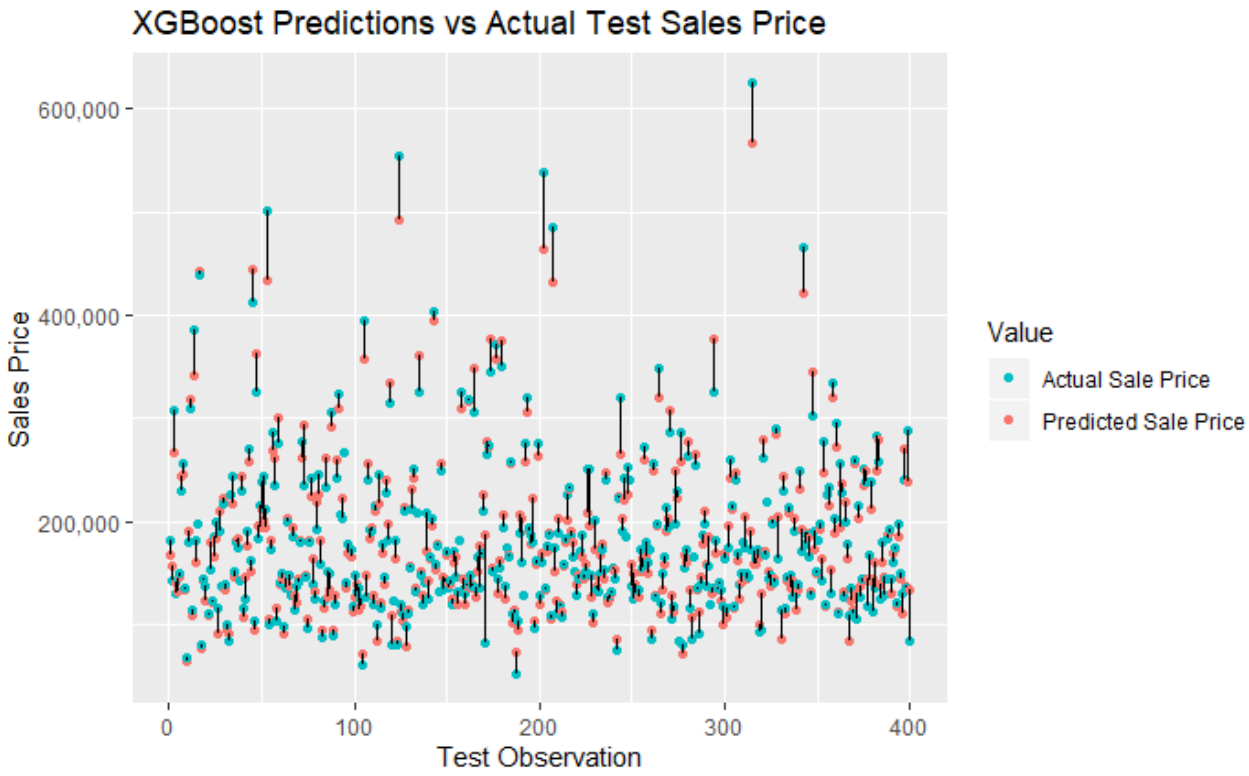

Figure 16 XGBoost Predictions vs Actual Test Sale Price

4. Conclusion and Limitations

Our background research suggested that the predictors that would influence sale price the most were total number of bathrooms, neighborhood, grand living area, garage car capacity, and overall quality. We tested our hypothesis using the GLM and Random Forest models. These models' adjusted R^2 values indicated that the research predictors explained a large amount of variation in sale price.

When we input all 79 predictors in our LASSO and XGBoost models, they both agreed that total square feet, overall quality, overall condition, and neighborhood were some important features in prediction. Both the LASSO and XGBoost were more accurate than the GLM and RF, but they also used more variables.

Based on our ultimate goal of accuracy, the XGBoost is our final model. Its predictions tend to be off by $20,000, which is better than the average RMSE of the other models. However, one negative to the use of the XGBoost algorithm is that it is harder to interpret than the other models. It focuses less on optimal variable selection and more on predictive accuracy. But, as seen in **Figure 14**, it does tell us which variables are most important in the prediction of price, such as total square feet and overall quality. If our goal was to have a more interpretable model as opposed to a more accurate one, we would have chosen the LASSO or GLM as our final model.

A problem we ran into in modeling process was that certain levels of our factor predictors were significant while other levels of that same predictor were not. For instance, in our GLM, only some neighborhoods were significant in explaining the variation in price, while other neighborhoods were not. This indicates that the "significant" neighborhoods influence price differently than the neighborhoods that have insignificant coefficients. In theory, the insignificant neighborhoods could be merged into an "other" category. However, in practice, researchers typically retain all levels of the factor regardless of their significance, so long as some of the levels influence the response.[4] It may introduce bias into the model to collapse categories with insignificant coefficients. Therefore, whenever some levels of a factor were considered important in our models, we deemed all levels of that predictor to be important in explaining and predicting sale price.

A limitation to our analysis is the small number of training observations. Furthermore, there were some outliers in our response variable. In future analyses, we could remove sale price outliers and fit models without them.

5. References

[1] Mburugu, Charles. "10 Factors That Affect Property Value (#7 Is Surprising): Mashvisor." Investment Property Tips | Mashvisor Real Estate Blog, July 17, 2019. https://www.mashvisor.com/blog/factors-that-affect-property-value/.

[2] Koehrson, Will. "An Implementation and Explanation of the Random Forest in Python," August 30, 2018. https://towardsdatascience.com/an-implementation-and-explanation-of the-random-forest-in-python-77bf308a9b76

[3] Morde,Vishal. April 7, 2019. "XGBoost Algorithm: Long May She Reign!" https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long she-may-rein-edd9f99be63d

[4] "XGBoost Parameters." XGBoost Parameters - xgboost 1.1.0-SNAPSHOT documentation. Accessed May 10, 2020. https://xgboost.readthedocs.io/en/latest/parameter.html.