615 House Price Prediction Report

Fall 2022

Xueqi Lu, Yihan Hong

Introduction

In this project, our group utilized a data set provided by Dean De Cock at Truman State University to complete two analyses. In the first analysis, we try to produce an estimate of the sale price of a house given its square footage and location within either the Brookside, Edwards, or North Ames neighborhood. In the second analysis, we built the predictive model for house prices in all of Ames, Iowa utilizing many other explanatory variables in addition to square footage. Throughout these analyses, multiple models were tested, diagnostics were checked, and data was inspected to ensure accuracy and limit bias when reasonable. The subsequent final models are what we find as the most appropriate predictive fits for each analysis question.

Data Description

The data used for the analysis in this report was sourced from Kaggle, but compiled by Dean De Cock, a professor of statistics at Truman State University. All of the data was collected on homes located in the city of Ames, Iowa and consists of 2,919 observations on 81 variables (1 being an ID, 79 explanatory, and 1 response) that represent various amenities and features of a house. Kaggle presented this data through two separate files named train which includes 1460 observations and test which includes 1459 observations respectively.

Data Cleaning and Feature Engineering

In order to clean the data, our group first assessed where data was missing. Next, we read the data description file provided by Dean De Cock and Kaggle to determine, on a case-by-case basis, the best method to either replace or remove data points.

Absent Home Features

We found that the overwhelming majority of missing data, or data coded as NA, actually represented the absence of a home feature. For example, >99% of the PoolQC column consisted of NAs; this was because those houses simply did not have pools. Thus, for the columns where missing data meant the home feature was missing, a default value replaced the NAs (e.g. “No Pool”). Table 1 outlines the default values used for each column with missing data as a result of an absent home feature.

Table 1: Data Cleaning for Absent Home Features

Column	Percent of Data Missing in		Imputation Method / Default Value
	Test	Train	

PoolQC	99.8%	99.5%	No Pool
MiscFeature	96.5%	96.3%	None
Alley	92.7%	93.8%	No Alley
Fence	80.1%	80.8%	No Fence
FireplaceQu	50.0%	47.3%	No Fireplace
GarageYrBlt	5.3%	5.5%	Year House Built
GarageFinish	5.3%	5.5%	No Garage
GarageQual	5.3%	5.5%	No Garage
GarageCond	5.3%	5.5%	No Garage
GarageType	5.2%	5.5%	No Garage
BsmtExposure	3.0%	2.6%	No Bsmt
BsmtFinType2	2.9%	2.6%	No Bsmt
BsmtCond	3.1%	2.5%	No Bsmt
BsmtQual	3.0%	2.5%	No Bsmt
BsmtFinType1	2.9%	2.5%	No Bsmt
MasVnrType	1.1%	0.5%	None
MasVnrArea	1.0%	0.5%	0
BsmtFullBath	0.1%	0.0%	0
BsmtHalfBath	0.1%	0.0%	0
BsmtFinSF1	0.1%	0.0%	0
BsmtFinSF2	0.1%	0.0%	0
BsmtUnfSF	0.1%	0.0%	0
TotalBsmtSF	0.1%	0.0%	0
GarageCars	0.1%	0.0%	0
GarageArea	0.1%	0.0%	0

Missing Data

Some of the missing data was not the result of absent home features. For example, Exterior1st describes the material used on the exterior of a home; although a home cannot exist without having external material (e.g. brick), there was data missing. For these columns, we either replaced the missing values with the mode (if the mode was an extremely large majority of the data) or we used KNN to impute the data. Table 2 describes the imputation method or values used for each column.

Table 2: Data Cleaning for Missing Data

Column	Percent of Data Missing in		Imputation Method / Default Value
	Test	Train	
LotFrontage	15.6%	17.7%	KNN
MSZoning	0.3%	0.0%	KNN
Exterior1st	0.1%	0.0%	KNN
Exterior2nd	0.1%	0.0%	KNN
KitchenQual	0.1%	0.0%	KNN
Electrical	0.0%	0.1%	SBkr (mode)

Utilities	0.1%	0.0%	AllPub (mode)
Functional	0.1%	0.0%	Typ (mode)
SaleType	0.1%	0.0%	WD (mode)

Analysis Question 1: Estimation of the Sale Price

The Problem

Our group want to get better understand about housing sale prices in relation to a home's square footage within specific communities (Brookside, Edwards, and North Ames) in Ames, Iowa. We utilized multiple linear regression (MLR) to find the associations of sale price, square footage, and neighborhood.

Assumptions

Before proceeding with MLR, the data was investigated to check procedural assumptions. The equal variance assumption had appeared to be met as box plots of the data were fairly uniform, but this assumption could be checked thoroughly through diagnostic plots of the selected fit. The normality assumption was suspect. Sale price seemed to be skewed for a couple of the neighborhoods. We moved forward with a number of fits, including transformations and different data cleansing approaches, of which the diagnostic plots could provide more insight on assumptions. Lastly, the observations were assumed to be independent of each other provided the information surrounding the sourced data set.

Final Models and Interpretation

The final model that appeared to be the most predictive was:

$$\begin{aligned} \ln(\text{SalePrice}) = {} & + 10.79 \\ & + 0.0007382*\text{SqFt} \\ & + 0.2339*\text{Edwards} \\ & + 0.6517*\text{NorthAmes} \\ & - 0.0001996*\text{SqFt}*\text{Edwards} \\ & - 0.0004141*\text{SqFt}*\text{NorthAmes} \end{aligned} \quad (1)$$

SqFt = square footage of home in units of 1 square foot;
Edwards = 1 if house in Edwards, 0 otherwise; NorthAmes = 1 if house in North Ames, 0 otherwise

All the parameters of this model were statistically significant at an alpha level of 0.05. Their p values and confidence intervals can be found in Table 3 below.

Table 3 P-Values and Conf. Intervals of Model Parameters

Parameter	P-Value	95% Confidence Interval
β_0	$< 2*10^{-16}$	[10.6305, 10.9526]
$\beta_{\text{SquareFootage}}$	$< 2*10^{-16}$	[0.0006, 0.0008]
β_{Edwards}	0.0314	[0.0209, 0.4468]
$\beta_{\text{NorthAmes}}$	$7.13*10^{-12}$	[0.4707, 0.8327]
$\beta_{\text{SquareFootageEdwards}}$	0.0186	[-0.0003, -0.00003]
$\beta_{\text{SquareFootageNorthAmes}}$	$1.62*10^{-8}$	[-0.0005, -0.0002]

The p-values of all these parameters indicate there is less than a thousandth of a percent chance of witnessing the data that was observed if their true value is 0, or that the variable that the parameter is associated with is not a regressor of sale price. This finding is consistent with the confidence intervals as 0 is not contained within any of the intervals, or there is 95% confidence that the mean parameter values in repeated sampling are contained in an interval which excludes 0. As the neighborhoods are levels of a categorical variable, model (1) employs indicator variables and can be reduced to three separate models (2), (3), (4), respective of specific neighborhoods.

If a house is within Brookside, the model is:

$$\ln(\text{SalePrice}) = 10.79 + 0.0007*\text{SqFt} \quad (2)$$

If a house is within Edwards, the model is:

$$\ln(\text{SalePrice}) = 11.02 + 0.0005*\text{SqFt} \quad (3)$$

If a house is within North Ames, the model is:

$$\ln(\text{SalePrice}) = 11.44 + 0.0003*\text{SqFt} \quad (4)$$

For model (2), a home in Brookside, an increase of 100 square feet in living area is associated with a multiplicative change of roughly 1.08 in the median sale price. For model (3), a home in Edwards, an increase of 100 square feet in living area is associated with a multiplicative change of roughly 1.06 in the median sale. For model (4), a home in North Ames, an increase of 100 square feet in living area is associated with a multiplicative change of roughly 1.03 in the median sale price.

Diagnostic Plots and Checking for Influential Points

The diagnostic plots for the selected fit confirmed the tentative assumptions of equal variance and normality. Fig. 1 shows small and unstructured residuals with no evidence for unequal variance. Fig. 2 shows limited departures from normality (minimal deviations are expected at the tails).

Figure 1

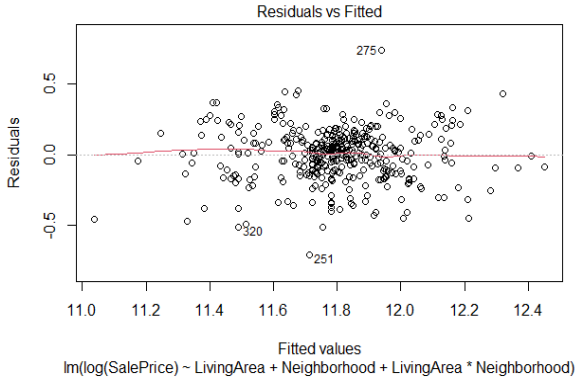
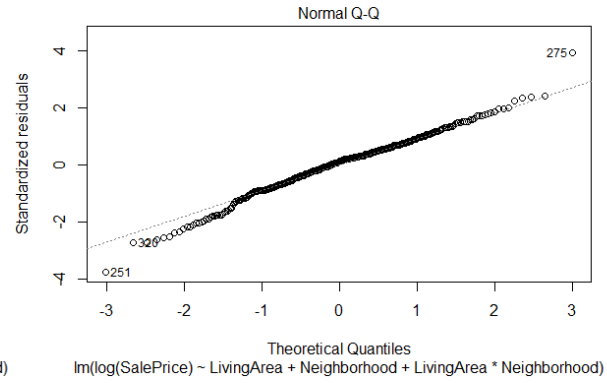


Figure 2



A cook's distance and leverage analysis of the most basic multiple linear regression model indicated four outliers, seen in Fig. 3 and 4. These points appeared to be influential as either true extremes or erroneous. Two of the four data points were especially uncharacteristic given the general knowledge that housing price increases as square footage increases. The final model, as well as competing models, was run on the raw data, data with the 2 most influential points eliminated, and data with all 4 influential points eliminated. The chosen regression fit performed measurably better on data with the two most outlying observations eliminated in comparison to

the raw (nearly 10% increase in adjusted R^2); however, the fits did not change largely between the dataset with four observations removed versus two observations removed. Model (1) is based upon data with 2 influential observations removed to incorporate as much data in the prediction as possible while eliminating bias as well.

Figure 3

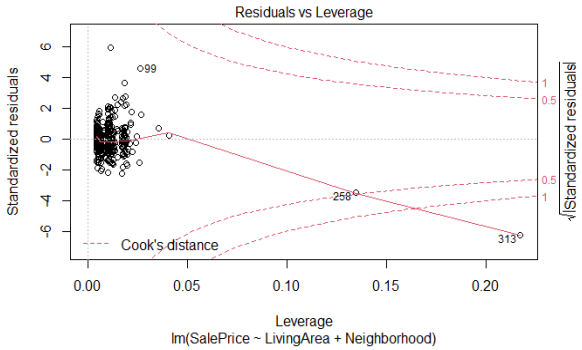
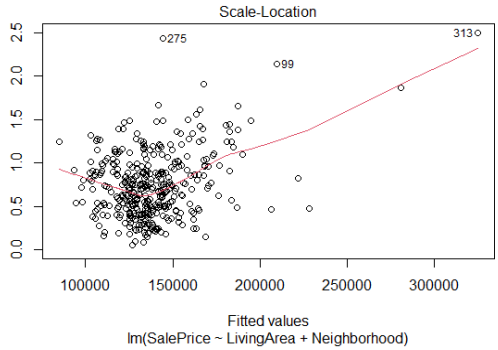


Figure 4



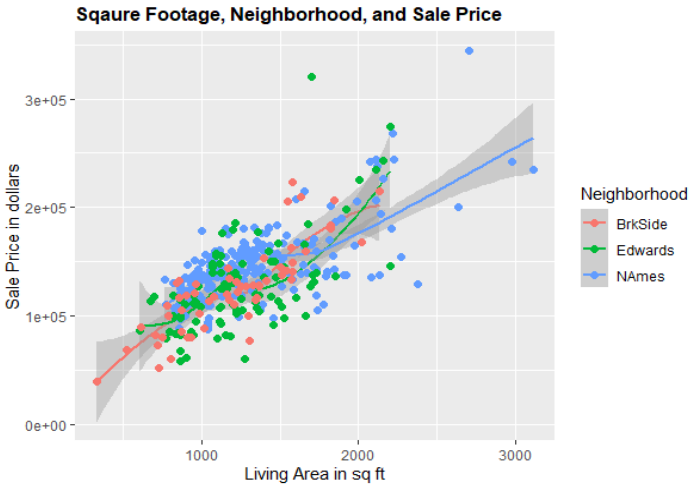
Competing Models' Performance and Final Model Summary

As mentioned, model (1) was the most successful of all the tested fits. Tested fits included various combinations of untransformed variables (sale price and/or square footage) and interactions. The adjusted R^2 of these fits were all lower than that of model (1), and the models for these fits contained parameters that were statistically insignificant. While these assessments were made with statistical reasoning, it is still important to assess the overall performance of the final model irrespective of others. The adjusted R^2 for this fit was 0.5208 meaning that roughly 52% of the variation in the natural log of sale price was explained by square footage, the neighborhood, and their interaction. This value is not poor nor great.

Conclusion

We can use this model going forward when wanting to predict sale prices of homes in Ames, Iowa that are within the neighborhoods of Brookside, Edwards, and North Ames and of which the square footage is within the interval of [334 sq. ft , 3112 sq, ft] (using this model for prediction outside of these restrictions is highly disregarded). This model confirms the known idea that housing prices increase as home size increases, but it is able to place a number on this relationship that can be helpful when wanting a more accurate idea of the sale price on a home. Fig. 5 displays all the data in which the final model was drawn upon.

Figure 5



This analysis was performed using R software. All relevant code written can be found in Appendix A.

Analysis Question 2: Modeling Sale Price for Homes in Ames, Iowa

Problem and Possible Model Solutions

In this second analysis, our group predicted the sale price of homes in Ames, Iowa using the various explanatory variables in the train data set. We worked to recommend a final model that was selected after assessing all variables and all levels (of the cleaned data) in order to identify appropriate variables to include in the model. Techniques include stepwise selection, the lasso, ridge regression, and the elastic net. The Residual Mean Squared Error (RMSE) of the competing models were compared to select the most suitable and predictive model. Of the techniques explored, we were able to build the most predictive model by using the Elastic Net technique. Our LASSO model produced a successful model as well. Ridge regression performance came next, followed by Stepwise regression, which lagged far behind.

Model Selection

Each of the modeling methods (variable selection and penalized regression) were run on all explanatory variables of the cleaned and transformed data set. This decision was made since we had no domain knowledge indicating the irrelevance of any of the explanatory variables, and the methods would, by nature, select significant variables and exclude those that appeared to be non-predictive. The variable selection methods, stepwise selection, utilized ordinary least square regression meaning fits were made through minimizing residual error. The stepwise method is a combination of forward selection and backwards elimination. Forward Selection selects one explanatory variable at a time and adds it to the model if the parameter's p value falls under the significance threshold (should one that was significant become insignificant, it is kicked out of the model). Backwards elimination starts with all the variables included in the model and eliminates the parameter with the highest p value, one at a time, until all parameters fall under the significance threshold. The next three methods we utilized were penalized regression methods. These methods, Ridge, LASSO, and Elastic Net, also employ ordinary least squares regression, but with an associated penalty in attempts to curb the effects of multicollinearity given the large number of regressors. All three PR methods sport different penalties, thus producing different models. Their benefits and increased robustness do not come without drawbacks, unfortunately. Ridge regression introduces bias through shrinkage of parameters, but it also reduces variance which can help in prediction. LASSO's penalty consists of summing the absolute value of betas, as opposed to ridge's sum of betas squared, so that still some shrinkage and bias occurs, although less than ridge. The LASSO also in essence performs variable selection because its penalty can cause betas to shrink to 0, meaning we do not incorporate them in the model. Elastic net is a combination of the former two PR method as its penalty is the sum of squared betas and the sum of the absolute value of betas.

Assumptions and Transformations

Provided the information from Kaggle, we assumed observational independence. Our group then examined the numeric variables to assess normality, equal variance, and whether transformations would be appropriate. It is common with monetary data to perform log transformations, so we plotted each numeric variable against SalePrice with and without this transformation, fig. 6 and fig. 7. The log transformations appeared to show a stronger relationship, greater normality, and more uniform variance, so we used the log transformations for our models.

Figure 6: Pre-Transformed Plots

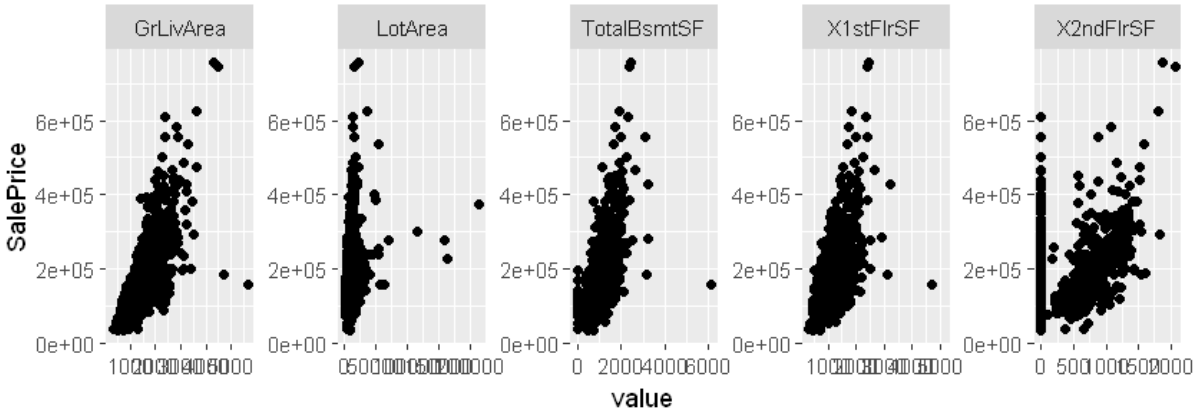
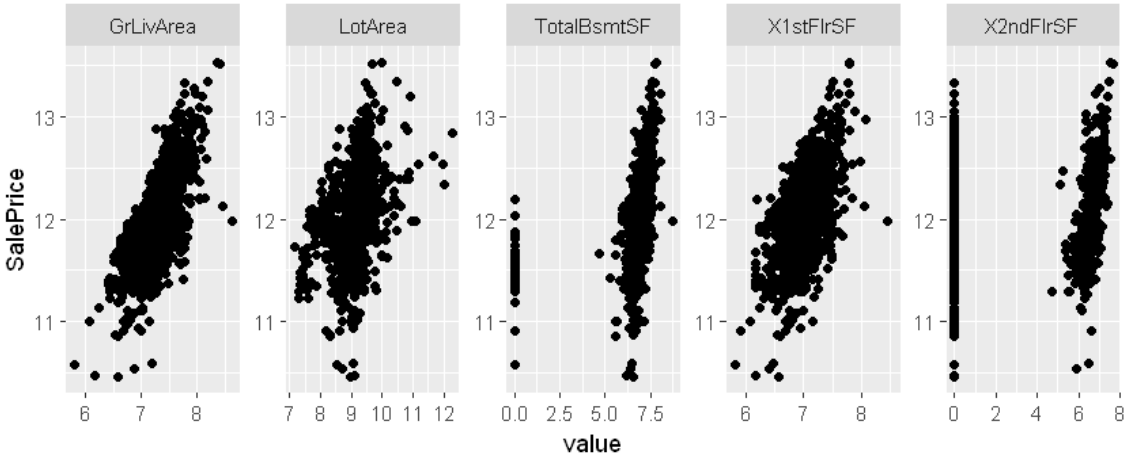


Figure 7: Log-Transformed Plots



Addressing Diagnostics of Final Model: Elastic Net

The diagnostic plots for the final model ran on the preliminary cleaned data set are displayed in Fig. 9. The equal variance assumption stands, but the normality assumption is still suspect. Tails of the Normal Q-Q plot indicate there may be some skew at the bounds of the data. The leverage and cook's distance plot also indicated some influential points. These points were removed for a second fitting of the Elastic Net model, and the resultant diagnostic plots can be seen if Fig. 10. The cook's distance and leverage plot show an elimination of influential points.

Figure 9 Diagnostic Plots of Fit of Raw Cleaned Data

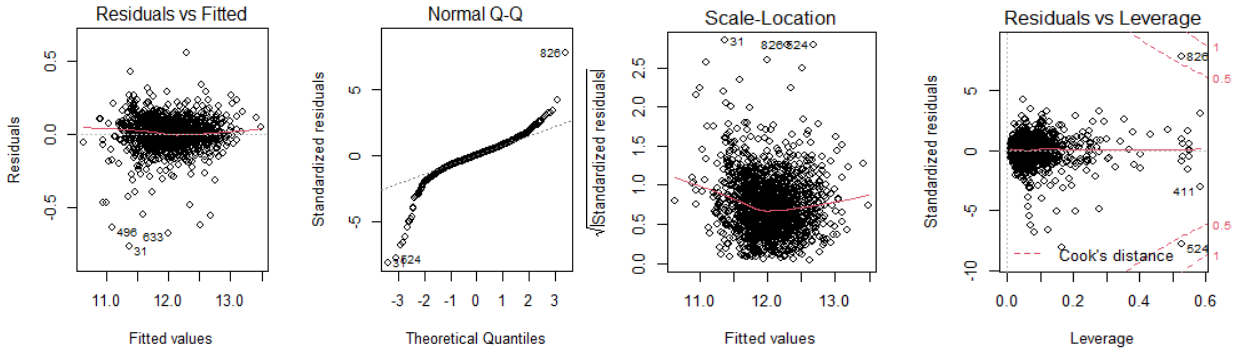
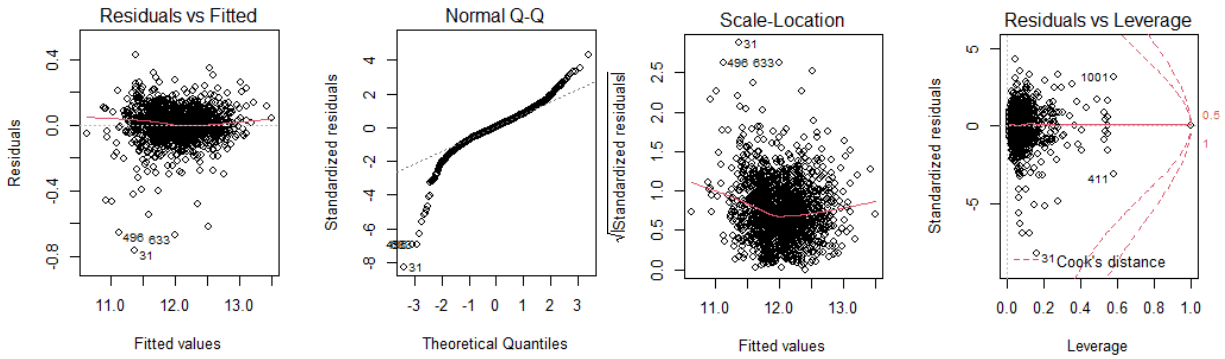


Figure 10 Diagnostic Plots of Fit on Non-Raw Cleaned Data



Comparing Competing Models

The final model our group has agreed upon to predict the sale price of houses in Ames, Iowa is the Elastic Net model (Appendix E). By comparing the RMSEs of all the tested methods, shown in Table 4, we believed the Elastic Net model is the most appropriate model. It should however be noted that the LASSO method did not produce the lowest RMSE. This occurrence is likely due to the fact that the elastic net method generally performs better than the LASSO method at reducing variability when there are correlated variables in the data set. This is because the LASSO method will tend to only keep one variable without heed for the other correlated one. Knowing the drawbacks of penalized regression methods, it is likely that the Ridge method did not perform as good of prediction as the other two penalized regression methods due to the bias introduced from its penalty.

Table 4 RMSE of Competing Models

Model	RMSE
Stepwise	197572.4
Ridge	20282.6
Lasso	20851.63
Elastic Net	19933.83

Conclusion

After multiple tests and intense assessment of the data, our group is able to provide a model that we think successfully predicts the sale price of homes in Ames, Iowa. As always, this model should not be extrapolated nor used on a location other than this city. Through penalize regression methods we found ourselves able to create the best models. However, we concede other models may prove to be more predictive than our final model. Different approaches to data cleaning and different imputation techniques are clear avenues one could take to fit a different, and perhaps more successful, model than our own. Nonetheless, we stand by our statistical analysis and believe this Elastic Net model performance in predicting sale price. This method incorporated a large number of predictors into our model that again reaffirmed general knowledge on housing prices (i.e. greater square footage increase sale price, better condition of a home increases sale price, etc.), but there were some unanticipated predictors such as land contour and alleyways. This model can be utilized for different interests (e.g. setting house

prices, negotiating house prices, starting local businesses, etc.) by different entities (realty companies, interested citizens, urban planners, etc.).

This analysis was performed using R software. All relevant code written can be found in Appendix B.

Self-reflection

We have been preparing for this project since we learned categorical variables. During the thanksgiving break, I have been working on this project every day, including reading "An Introduction to Statistical Learning with Applications in R", "Machine Learning Essentials: Practical Guide in R" and other books. The most troublesome part of coding is that there are too many variables and most of the variables are not numbers, which causes great difficulties in data cleaning and operation. When the code is running, I often encounter problems with incorrect data types or incorrect lengths. So learning variable selection and debugging took up most of my time, but it is precisely because of this that I can learn a lot from this project.

If I do a similar project next time, I may choose a data set that I know better. For example, the other group mentioned about the dataset which includes some variables that have impact on the admission of the master project. I hope to tell future students to start preparing their own projects as early as possible and spend more time on it. It always take more time to complete a project than you imagined.

Appendix A:
Code for Estimation of the Sale Price

```
### Project Question 1 ###

library(ggplot2)
library(olsrr)
library(MASS)

# import dataset, named house
house = read.csv("…/train.csv", header = TRUE, sep = ",")

# subset dataset for desired vars, named prob1
house_1 = house[house$Neighborhood == "NAmes",]
house_2 = house[house$Neighborhood == "Edwards",]
house_3 = house[house$Neighborhood == "BrkSide",]

house_1 = data.frame(Neighborhood = house_1$Neighborhood,LivingArea =
house_1$GrLivArea,SalePrice = house_1$SalePrice)
house_2 = data.frame(Neighborhood = house_2$Neighborhood,LivingArea =
house_2$GrLivArea,SalePrice = house_2$SalePrice)
house_3 = data.frame(Neighborhood = house_3$Neighborhood,LivingArea =
house_3$GrLivArea,SalePrice = house_3$SalePrice)

prob1 = rbind(house_1, house_2, house_3)

# check for NAs
mean(is.na(prob1))

# look for multicollinearity and nonlinearity
plot(prob1, main = "Scatter Plot Matrix of Vars")

# first fit raw data
fit1 = lm(SalePrice~LivingArea+Neighborhood, data = prob1)
plot(fit1)
summary(fit1)
sresid = rstudent(fit1)
pred = fitted(fit1)
plot(pred,sresid, xlab = "fitted values", ylab = "studentized residuals", main = "studentized residual plot")

# log of saleprice raw data
fitlog = lm(log(SalePrice)~LivingArea+Neighborhood, data = prob1)
plot(fitlog)
summary(fitlog)

# fits on take out outliers 99 258 275 313
prob1.1 = prob1[-c(99,258,275,313),]
fit1.1 = lm(SalePrice~LivingArea+Neighborhood, data = prob1.1)
fit1.1.1 = lm(log(SalePrice)~LivingArea+Neighborhood, data = prob1.1)
fit1.1.2 = lm(log(SalePrice)~LivingArea+Neighborhood+LivingArea*Neighborhood, data = prob1.1)
fit1.1.3 = lm(SalePrice~LivingArea+Neighborhood+LivingArea*Neighborhood, data = prob1.1)
```

```
summary(fit1.1)
summary(fit1.1.1)
summary(fit1.1.2)

# fits on take out outliers 258 313
prob1.2 = prob1[-c(258,313),]
fit1.2 = lm(SalePrice~LivingArea+Neighborhood, data = prob1.2)
fit1.2.1 = lm(SalePrice~LivingArea+Neighborhood+LivingArea*Neighborhood, data = prob1.2)
fit1.2.2 = lm(log(SalePrice)~LivingArea+Neighborhood+LivingArea*Neighborhood, data = prob1.2)
fit1.2.3 = lm(log(SalePrice)~LivingArea+Neighborhood, data = prob1.2)

summary(fit1.2.1)
summary(fit1.2.2)
summary(fit1.2.3)


# fits on take out outlier 313
prob1.3 = prob1[-313,]
fit1.3 = lm(SalePrice~LivingArea+Neighborhood, data = prob1.3)
fit1.3.1 = lm(log(SalePrice)~LivingArea+Neighborhood, data = prob1.3)
fit1.3.2 = lm(SalePrice~LivingArea+Neigborhood+LivingArea*Neigborhood, data = prob1.3)
fit1.3.3 = lm(log(SalePrice)~LivingArea+Neigborhood+LivingArea*Neigborhood, data = prob1.3)

summary(fit1.3.1)
summary(fit1.3.2)
summary(fit1.3.3)

# proceed with taking out 258 and 313 only, get stats and diagnostics

summary(fit1.2.2)
confint(fit1.2.2)
plot(fit1.2.2)

# inspect plots for assumptions
ggplot(prob1, aes(x=LivingArea, color = Neighborhood, fill = Neighborhood)) + ggtitle("Histogram of
Living Area by Neighborhood (Raw Data)") + geom_histogram(size = 1.15, position = "identity", alpha =
0.10) + theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1.2, aes(x=LivingArea, color = Neighborhood, fill = Neighborhood)) + ggtitle("Histogram of
Living Area by Neighborhood (Cleaned Data)")+ geom_histogram(size = 1.15, position = "identity",
alpha = 0.10) + theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1, aes(x=SalePrice, color = Neighborhood, fill = Neighborhood)) + ggtitle("Histogram of Sale
Price by Neighborhood (Raw Data)") + geom_histogram(size = 1.15, position = "identity", alpha = 0.10)
+ theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1.2, aes(x=SalePrice, color = Neighborhood, fill = Neighborhood)) + ggtitle("Histogram of
Sale Price by Neighborhood (Cleaned Data)")+ geom_histogram(size = 1.15, position = "identity", alpha
= 0.10) + theme(plot.title=element_text(size=12,face = "bold"))

# hist(prob1$LivingArea)
# hist(prob1$SalePrice)
# hist(prob1.2$LivingArea)
```

```
# hist(prob1.2$SalePrice)

ggplot(prob1, aes(x=LivingArea, color = Neighborhood, fill = Neighborhood)) + ggtitle("Box Plots of Living Area by Neighborhood (Raw Data)") + geom_boxplot(size = 1, position = "identity", alpha = 0.1) + theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1.2, aes(x=LivingArea, color = Neighborhood, fill = Neighborhood)) + ggtitle("Box Plots of Living Area by Neighborhood (Cleaned Data)") + geom_boxplot(size = 1, position = "identity", alpha = 0.1) + theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1, aes(x=SalePrice, color = Neighborhood, fill = Neighborhood)) + ggtitle("Box Plots of Sale Price by Neighborhood (Raw Data)") + geom_boxplot(size = 1, position = "identity", alpha = 0.1) + theme(plot.title=element_text(size=12,face = "bold"))
ggplot(prob1.2, aes(x=SalePrice, color = Neighborhood, fill = Neighborhood)) + ggtitle("Box Plots of Sale Price by Neighborhood (Cleaned Data)") + geom_boxplot(size = 1, position = "identity", alpha = 0.1) + theme(plot.title=element_text(size=12,face = "bold"))

# boxplot(prob1$LivingArea~prob1$Neighborhood)
# boxplot(prob1$SalePrice~prob1$Neighborhood)
# boxplot(prob1.2$LivingArea~prob1.2$Neighborhood)
# boxplot(prob1.2$SalePrice~prob1.2$Neighborhood)

ggplot(prob1.2, aes(x = LivingArea, y = SalePrice, color = Neighborhood)) + geom_smooth(method = loess) + geom_point(size = 2) + xlab("Living Area in sq ft") + ylab("Sale Price in dollars") + ggtitle("Sqaure Footage, Neighborhood, and Sale Price") + theme(plot.title=element_text(size=12,face = "bold"))
```

Appendix B:

```
library(glmnet)
library(MASS)
library(tidyverse)
library(broom)
library(dplyr)
library(VIM)
library(mltools)
library(data.table)
clean = read.csv(file.choose())
######## data cleaning with transformation #############
clean$SalePrice = log(clean$SalePrice)
clean$GrLivArea = log(clean$GrLivArea)
clean$LotArea = log(clean$LotArea)
clean$TotalBsmtSF = log(clean$TotalBsmtSF)
clean$TotalBsmtSF[is.infinite(clean$TotalBsmtSF)] = 0
clean$X1stFlrSF = log(clean$X1stFlrSF)
clean$X2ndFlrSF = log(clean$X2ndFlrSF)
clean$X2ndFlrSF[is.infinite(clean$X2ndFlrSF)] = 0

clean$Alley[is.na(clean$Alley)] <- "No Alley"
clean$BsmtQual[is.na(clean$BsmtQual)] <- "No Bsmt"
clean$BsmtCond[is.na(clean$BsmtCond)] <- "No Bsmt"
clean$BsmtExposure[is.na(clean$BsmtExposure)] <- "No Bsmt"
clean$BsmtFinType1[is.na(clean$BsmtFinType1)] <- "No Bsmt"
clean$BsmtFinType2[is.na(clean$BsmtFinType2)] <- "No Bsmt"
clean$FireplaceQu[is.na(clean$FireplaceQu)] <- "No Fireplace"
clean$GarageType[is.na(clean$GarageType)] <- "No Garage"
```

```
clean$GarageFinish[is.na(clean$GarageFinish)] <- "No Garage"
clean$GarageQual[is.na(clean$GarageQual)] <- "No Garage"
clean$GarageCond[is.na(clean$GarageCond)] <- "No Garage"
clean$PoolQC[is.na(clean$PoolQC)] <- "No Pool"
clean$Fence[is.na(clean$Fence)] <- "No Fence"
clean$MiscFeature[is.na(clean$MiscFeature)] <- "None"
clean$MasVnrType[is.na(clean$MasVnrType)] = "None"
clean$MasVnrArea[clean$MasVnrArea > 0 & clean$MasVnrArea <= 100] = ">0 & <=100"
clean$MasVnrArea[clean$MasVnrArea > 100 & clean$MasVnrArea <= 1000] = ">100 & <=1000"
clean$MasVnrArea[clean$MasVnrArea > 1000] = ">1000"
clean$MasVnrArea[is.na(clean$MasVnrArea)] = "None"

# Converting char to factors
clean = mutate_if(clean, is.character, as.factor)

# NKK imputation for missing values

clean_imputed_data <- kNN(clean, variable = c("LotFrontage", "Electrical", "GarageYrBlt" ))

# Drop imputation logic variables and Id variable

clean_imputed_data <- subset(clean_imputed_data, select = -c(LotFrontage_imp,
                                                Electrical_imp,GarageYrBlt_imp,Id))
###################### For test data set #################
library(glmnet)
library(MASS)
library(tidyverse)
library(broom)
library(dplyr)
library(VIM)
library(mltools)
library(data.table)
###################### test data cleaning and transformation ##############
test = read.csv(file.choose())
test$GrLivArea = log(test$GrLivArea)
test$LotArea = log(test$LotArea)
test$TotalBsmtSF = log(test$TotalBsmtSF)
test$TotalBsmtSF[is.infinite(test$TotalBsmtSF)] = 0
test$X1stFlrSF = log(test$X1stFlrSF)
test$X2ndFlrSF = log(test$X2ndFlrSF)
test$X2ndFlrSF[is.infinite(test$X2ndFlrSF)] = 0

test$Alley[is.na(test$Alley)] <- "No Alley"
test$BsmtQual[is.na(test$BsmtQual)] <- "No Bsmt"
test$BsmtCond[is.na(test$BsmtCond)] <- "No Bsmt"
test$BsmtExposure[is.na(test$BsmtExposure)] <- "No Bsmt"
test$BsmtFinType1[is.na(test$BsmtFinType1)] <- "No Bsmt"
test$BsmtFinType2[is.na(test$BsmtFinType2)] <- "No Bsmt"
test$FireplaceQu[is.na(test$FireplaceQu)] <- "No Fireplace"
test$GarageType[is.na(test$GarageType)] <- "No Garage"
test$GarageFinish[is.na(test$GarageFinish)] <- "No Garage"
test$GarageQual[is.na(test$GarageQual)] <- "No Garage"
test$GarageCond[is.na(test$GarageCond)] <- "No Garage"
test$PoolQC[is.na(test$PoolQC)] <- "No Pool"
test$Fence[is.na(test$Fence)] <- "No Fence"
test$MiscFeature[is.na(test$MiscFeature)] <- "None"
```

```
test$MasVnrType[is.na(test$MasVnrType)] = "None"
test$MasVnrArea[test$MasVnrArea > 0 & test$MasVnrArea <= 100] = ">0 & <=100"
test$MasVnrArea[test$MasVnrArea > 100 & test$MasVnrArea <= 1000] = ">100 & <=1000"
test$MasVnrArea[test$MasVnrArea > 1000] = ">1000"
test$MasVnrArea[is.na(test$MasVnrArea)] = "None"

colnames(test)[colSums(is.na(test))>0]
############## combine the train and test then do better estimation ###########
test$SalePrice <- NA
test <- subset(test, select = -c(Id))
combine_data <- rbind(clean_imputed_data, test)

# Converting char to factors
combine_data = mutate_if(combine_data, is.character, as.factor)

combined_imputed_data <- kNN(combine_data, variable =
c("MSZoning","LotFrontage","Utilities","Exterior1st","Exterior2nd","BsmtFinSF1","BsmtFinSF2",
                                  "BsmtUnfSF","TotalBsmtSF","BsmtFullBath","BsmtHalfBath","KitchenQual","Functio
nal","GarageYrBlt",
                                  "GarageCars","GarageArea","SaleType", "Electrical"))

combined_imputed_data <- subset(combined_imputed_data, select = -c(MSZoning_imp, LotFrontage_imp,
Utilities_imp, Exterior1st_imp, Exterior2nd_imp,
BsmtFinSF1_imp,BsmtFinSF2_imp,BsmtUnfSF_imp,TotalBsmtSF_imp,BsmtFullBath_imp,BsmtHalfBath_imp,Kit
chenQual_imp,Functional_imp,GarageYrBlt_imp,GarageCars_imp,GarageArea_imp,SaleType_imp,
Electrical_imp))

################## subsetting actual test data set from combined #######
test_imputed_data <- subset(combined_imputed_data, is.na(SalePrice))
test_imputed_data <- subset(test_imputed_data, select = -c(SalePrice))


#################### ONE HOT ENCODING for combined data ####################


library(caret)
set.seed(555)
combine_hot_data = one_hot(as.data.table(combined_imputed_data))

############## subsetting imputed and encoded test data ################################

one_hot_imputed_test <- subset(combine_hot_data, is.na(SalePrice))
one_hot_imputed_test <- subset(one_hot_imputed_test, select = -c(SalePrice))

############ ONE HOT ENCODING FOR IMPUTED CLEAN SET ####################
library(caret)
set.seed(555)
clean_onehot_imputed_data = one_hot(as.data.table(clean_imputed_data))
#########LASSO MODEL ############

clean_onehot_imputed_data=mutate_if(clean_onehot_imputed_data,is.factor,as.numeric)

colnames(clean_onehot_imputed_data)[colSums(is.na(clean_onehot_imputed_data))>0]

Lassonet <- cv.glmnet(as.matrix(clean_onehot_imputed_data[,1:305]), clean_onehot_imputed_data[,306], alpha=1)
```

```
##Optimal tuning parameter
best.lambda <- lassonet$lambda.min

##Check parameter estimates for the optimal model
coef(lassonet, s=best.lambda)


SalePrice=predict(lassonet,newx=as.matrix(one_hot_imputed_test),s=best.lambda)
SalePrice = exp(SalePrice)


###################### RMSE lasso ###################
lasso.pred_train <- predict(lassonet,newx=as.matrix(clean_onehot_imputed_data[,1:305]),s=best.lambda)

lasso.pred_train <- exp(lasso.pred_train)
lasso.rmse <- sqrt(mean((lasso.pred_train - exp(clean_imputed_data[,80]))^2))
lasso.rmse


##Elastic Net##

##Note to use the elastic net we must tune both alpha and lambda
##The easiest way to do this is to utilize the caret package

tcontrol <- trainControl(method="repeatedcv", number=10, repeats=5)

elastics.glmnet3 <- train(as.matrix(clean_onehot_imputed_data[,1:305]), clean_onehot_imputed_data[,306],
trControl=tcontrol,
                  method="glmnet", tuneLength=10)
attributes(elastics.glmnet3)
elastics.glmnet3$results
elastics.glmnet3$bestTune
elastics.glmnet4 <- elastics.glmnet3$finalModel
coef(elastics.glmnet4, s=elastics.glmnet3$bestTune$lambda)

en.pred <- predict(elastics.glmnet4, as.matrix(one_hot_imputed_test), s=elastics.glmnet3$bestTune$lambda)
en.pred <- exp(en.pred)


###################### RMSE Elastic Net ###################
en.pred_train <- predict(elastics.glmnet4, as.matrix(clean_onehot_imputed_data[,1:305]),
s=elastics.glmnet3$bestTune$lambda)
en.pred_train <- exp(en.pred_train)
en.rmse <- sqrt(mean((en.pred_train - exp(clean_imputed_data[,80]))^2))
en.rmse

##Ridge Regression##

##Note that cv.glmnet by default does 10-fold cross-validation
ridge.glmnet <- cv.glmnet(as.matrix(clean_onehot_imputed_data[,1:305]), clean_onehot_imputed_data[,306],
alpha=0)
attributes(ridge.glmnet)
ridge_best.lambda <- ridge.glmnet$lambda.min
coef(ridge.glmnet, s=ridge_best.lambda)

ridge.pred <- predict(ridge.glmnet, as.matrix(one_hot_imputed_test[,1:305]), s=ridge_best.lambda)
```

```
ridge.pred <- exp(ridge.pred)

###################### RMSE Ridge ###################
ridge.pred_train <- predict(ridge.glmnet, as.matrix(clean_onehot_imputed_data[,1:305]), s=ridge_best.lambda)
ridge.pred_train <- exp(ridge.pred_train)

ridge.rmse <- sqrt(mean((ridge.pred_train - exp(clean_imputed_data[,80]))^2))
ridge.rmse


library(leaps)
library(caret)

#############Stepwise variable selection##############
train.control <- trainControl(method = "cv", number = 10)

step.model <- train(SalePrice ~ ., data = clean_imputed_data[,],
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax = 1:60),
                    trControl = train.control
)

step.model$results
step.model$bestTune
summary(step.model$finalModel)

lm.sale <- lm(SalePrice ~  ExterCond + Foundation  + BsmtQual  +BsmtCond +BsmtExposure
                        +BsmtFinType1 +BsmtFinType1 +BsmtUnfSF +TotalBsmtSF +Heating+ CentralAir+
GrLivArea
                        + BsmtFullBath + KitchenAbvGr + KitchenQual + Functional  + Fireplaces
                        +FireplaceQu + GarageCars + GarageCond + WoodDeckSF + ScreenPorch + PoolArea +
PoolQC
                        +MiscFeature + SaleType, data=clean_imputed_data[,1:80])
stepwise.pred <- predict(lm.sale, test_imputed_data[,])

stepwise.pred <- exp(stepwise.pred)

stepwise.pred_train <- predict(lm.sale, clean_imputed_data[,])

stepwise.rmse <- sqrt(mean((stepwise.pred_train - exp(clean_imputed_data[,80]))^2))
stepwise.rmse


library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)
library(glmnet)
library(MASS)
library(tidyverse)
library(broom)
library(VIM)
library(mltools)
library(data.table)
library(glmnetUtils)
```

```
options(warn=-1)

set.seed(1)
#########First, load the train and test data. Label them accordingly and combined##############
train <- read.csv(file.choose())
test <- read.csv(file.choose())

train$DataType = "Train"
test$DataType = "Test"

test$SalePrice = NA

df <- rbind(train, test)

head(df)
##########Missing data analysis #############
{sum(is.na(x))}, na.action = NULL), "missing_data.csv")

##### clean the data ##############
clean <- df %>%
  mutate_if(is.factor, as.character) %>%
  tidyr::replace_na(list(
    Alley = "No Alley",
    BsmtQual = "No Bsmt",
    BsmtCond = "No Bsmt",
    BsmtExposure = "No Bsmt",
    BsmtFinType1 = "No Bsmt",
    BsmtFinSF1 = 0,
    BsmtFinType2 = "No Bsmt",
    BsmtFinSF2 = 0,
    BsmtUnfSF = 0,
    TotalBsmtSF = 0,
    BsmtFullBath = 0,
    BsmtHalfBath = 0,
    GarageCars = 0,
    FireplaceQu = "No Fireplace",
    GarageType = "No Garage",
    GarageFinish = "No Garage",
    GarageQual = "No Garage",
    GarageCond = "No Garage",
    GarageArea = 0,
    PoolQC = "No Pool",
    Fence = "No Fence",
    MiscFeature = "None",
    MasVnrType = "None",
    MasVnrArea = 0,
    Utilities = "AllPub",
    Functional = "Typ",
    Electrical = "SBrkr",
    SaleType = "WD"
  )) %>%
  mutate_if(is.character, as.factor)

clean$GarageYrBlt[is.na(clean$GarageYrBlt)] <- clean$YearBuilt[is.na(clean$GarageYrBlt)]

clean$MSSubClass = as.factor(clean$MSSubClass)
```

```
# Converting char to factors
clean = mutate_if(clean, is.character, as.factor)

################## Log transform necessary variables ###########

clean.gathered <- clean %>% dplyr::select(SalePrice, GrLivArea, LotArea, TotalBsmtSF, X1stFlrSF, X2ndFlrSF)
%>% gather(key = "variable", value = "value", -SalePrice)

ggplot(clean.gathered, aes(x = value, y = SalePrice)) +
   geom_point() +
   facet_wrap(~ variable, scales = "free", ncol = 5) +
   theme(aspect.ratio=2)

clean$SalePrice = log(clean$SalePrice)
clean$GrLivArea = log(clean$GrLivArea)
clean$LotArea = log(clean$LotArea)
clean$TotalBsmtSF = log(clean$TotalBsmtSF)
clean$TotalBsmtSF[is.infinite(clean$TotalBsmtSF)] = 0
clean$X1stFlrSF = log(clean$X1stFlrSF)
clean$X2ndFlrSF = log(clean$X2ndFlrSF)
clean$X2ndFlrSF[is.infinite(clean$X2ndFlrSF)] = 0

clean.gathered <- clean %>% dplyr::select(SalePrice, GrLivArea, LotArea, TotalBsmtSF, X1stFlrSF, X2ndFlrSF)
%>% gather(key = "variable", value = "value", -SalePrice)

ggplot(clean.gathered, aes(x = value, y = SalePrice)) +
   geom_point() +
   facet_wrap(~ variable, scales = "free", ncol = 5) +
   theme(aspect.ratio=2)

clean.onehot = one_hot(as.data.table(clean.imputed))


final.train <- clean.onehot %>% filter(DataType_Train == 1) %>% dplyr::select(-c(Id, DataType_Train,
DataType_Test))
final.test <- clean.onehot %>% filter(DataType_Test == 1) %>% dplyr::select(-c(SalePrice, DataType_Train,
DataType_Test))

##############Regression models¶###############

train.control <- trainControl(method = "cv", number = 10)

############ Lasso ###############

X <- as.matrix(final.train %>% dplyr::select(-SalePrice))
Y <- as.matrix(final.train %>% dplyr::select(SalePrice))

lassonet <- glmnetUtils::cv.glmnet(X, Y, alpha=1)

coef(lassonet, s=lassonet$lambda.min)

SalePrice = predict(lassonet, newx=as.matrix(final.test %>% dplyr::select(-Id)), s = lassonet$lambda.min)
SalePrice = exp(SalePrice)
```

```
####Forward Selection#######


forward.model <- train(
  SalePrice ~ .,
  data = final.train,
  method = "leapForward",
  trControl = train.control,
  tuneGrid = data.frame(nvmax = 1:251)
)

metrics <- forward.model$results %>% dplyr::select(nvmax, Rsquared)
metrics$AdjRsquared = 1 - ((1 - metrics$Rsquared) * (nrow(clean) - 1)) / (nrow(clean) - metrics$nvmax - 1)

ggplot(data = metrics, mapping = aes(x = nvmax, y = AdjRsquared)) +
  geom_point() +
  ggtitle("Forward Selection")

predictions <- data.frame(predict(
    forward.model,
    final.test
)) %>% tibble::rownames_to_column("Id")

colnames(predictions) <- c("Id", "SalePrice")
predictions$Id = 1461:2919
predictions$SalePrice = exp(predictions$SalePrice)



#########Backwards Selection######
backward.model <- train(
  SalePrice ~ .,
  data = final.train,
  method = "leapBackward",
  trControl = train.control,
  tuneGrid = data.frame(nvmax = 1:251),
  metric = "RMSE"
)

metrics <- backward.model$results %>% dplyr::select(nvmax, Rsquared)
metrics$AdjRsquared = 1 - ((1 - metrics$Rsquared) * (nrow(clean) - 1)) / (nrow(clean) - metrics$nvmax - 1)

ggplot(data = metrics, mapping = aes(x = nvmax, y = AdjRsquared)) +
  geom_point() +
  ggtitle("Backwards Selection")

predictions <- data.frame(predict(
    backward.model,
    final.test %>% dplyr::select(-c(Id))
)) %>% tibble::rownames_to_column("Id")

colnames(predictions) <- c("Id", "SalePrice")
predictions$Id = 1461:2919
predictions$SalePrice = exp(predictions$SalePrice)
```

```
##########Stepwise Selection#########

stepwise.model <- train(
  SalePrice ~ .,
  data = final.train,
  method = "leapSeq",
  trControl = train.control,
  tuneGrid = data.frame(nvmax = 1:30)
)

metrics <- stepwise.model$results %>% dplyr::select(nvmax, Rsquared)
metrics$AdjRsquared = 1 - ((1 - metrics$Rsquared) * (nrow(clean) - 1)) / (nrow(clean) - metrics$nvmax - 1)

ggplot(data = metrics, mapping = aes(x = nvmax, y = AdjRsquared)) +
  geom_point() +
  ggtitle("Stepwise Selection")

predictions <- data.frame(predict(
    stepwise.model,
    final.test %>% dplyr::select(-Id)
)) %>% tibble::rownames_to_column("Id")

colnames(predictions) <- c("Id", "SalePrice")
predictions$Id = 1461:2919
predictions$SalePrice = exp(predictions$SalePrice)
```

Appendix C
Residual plot for final model

```
names(final.train) <- gsub(" ","_",names(final.train))

elesticnet.lm <- lm(SalePrice ~
            MSSubClass_30+      MSSubClass_90+       MSSubClass_160+ MSZoning_FV+   MSZoning_RM+
            LotArea+    Street_Grvl+ Street_Pave+ Alley_Grvl+     Alley_Pave+
            LotShape_IR2+     LandContour_Bnk       +   Utilities_AllPub+ LotConfig_Corner+LotConfig_CulDSac+
            LotConfig_FR2+   LandSlope_Mod+LandSlope_Sev+  Neighborhood_BrkSide+
            Neighborhood_ClearCr+     Neighborhood_Crawfor+
            Neighborhood_Edwards+    Neighborhood_MeadowV+
            Neighborhood_Mitchel+  Neighborhood_NoRidge+  Neighborhood_NridgHt+
            Neighborhood_NWAmes+Neighborhood_OldTown+   Neighborhood_Somerst+
            Neighborhood_StoneBr+    Condition1_Artery+   Condition1_Norm+
            Condition1_RRAe+  Condition2_PosA+Condition2_PosN+   BldgType_1Fam+      BldgType_Duplex+
            OverallQual+OverallCond+YearBuilt+YearRemodAdd+  RoofMatl_ClyTile+   RoofMatl_Membran+
            RoofMatl_WdShngl+    Exterior1st_BrkComm+Exterior1st_BrkFace+  Exterior1st_HdBoard+
            Exterior1st_MetalSd+
Exterior1st_Wd_Sdng+  MasVnrType_BrkCmn+    MasVnrArea+ExterQual_Ex+
            ExterQual_TA+ExterCond_Ex+ExterCond_Fa+     ExterCond_TA  +Foundation_BrkTil+
            Foundation_PConc+Foundation_Stone+Foundation_Wood+BsmtQual_Ex+     BsmtQual_TA+
            BsmtCond_Fa+ BsmtExposure_Gd+       BsmtExposure_No+ BsmtFinType1_GLQ+
            BsmtFinType1_Unf+BsmtFinSF1+BsmtFinType2_ALQ+
            BsmtFinType2_BLQ+ BsmtFinSF2+     TotalBsmtSF+   Heating_GasW+
            Heating_Grav+Heating_OthW+       HeatingQC_Ex+     HeatingQC_TA+CentralAir_N+
```

```
        X1stFlrSF+    GrLivArea+BsmtFullBath+     FullBath+HalfBath+       KitchenAbvGr+
        KitchenQual_Ex+    KitchenQual_TA+   Functional_Maj2+    Functional_Mod+Functional_Sev+
        Functional_Typ+Fireplaces+     FireplaceQu_Gd+FireplaceQu_No_Fireplace
+    GarageType_2Types+
        GarageType_Basment+      GarageType_CarPort+   GarageCars+
        GarageArea+GarageQual_Ex+GarageQual_Fa+GarageQual_Gd+   GarageCond_Fa+
        PavedDrive_Y+WoodDeckSF+OpenPorchSF+        ScreenPorch+PoolArea+
        PoolQC_Ex+    Fence_GdWo+      MiscFeature_Othr+   SaleType_Con+SaleType_ConLD+
        SaleType_New+   SaleType_WD+SaleCondition_Abnorml+    SaleCondition_Family   , data =
final.train)

par(mfrow=c(2,2))
plot(elesticnet.lm)
```

Appendix D

Final Elastic Net Model Parameters (with outliers)

Ln(SalePrice) =

(Intercept)	2.467995e+00
MSSubClass_30	-1.959866e-02
MSSubClass_90	-1.128721e-02
MSSubClass_160	-2.773189e-02
MSZoning_C (all)	-3.514181e-01
MSZoning_FV	1.770702e-02
MSZoning_RM	-3.817699e-02
LotArea	6.972495e-02
Street_Grvl	-2.213341e-02
Street_Pave	2.272434e-13
Alley_Grvl	-2.380771e-03
Alley_Pave	5.499329e-03
LotShape_IR2	8.018220e-04
LandContour_Bnk	-9.543339e-03
Utilities_AllPub	3.125022e-02
LotConfig_Corner	1.103019e-03
LotConfig_CulDSac	1.854121e-02
LotConfig_FR2	-1.073192e-02
LandSlope_Mod	4.297371e-03
LandSlope_Sev	-3.200342e-03
Neighborhood_BrkSide	2.761410e-02
Neighborhood_ClearCr	1.850171e-02
Neighborhood_Crawfor	1.045909e-01
Neighborhood_Edwards	-4.064740e-02
Neighborhood_MeadowV	-2.696587e-02
Neighborhood_Mitchel	-1.870896e-02
Neighborhood_NoRidge	7.169308e-02
Neighborhood_NridgHt	5.425305e-02
Neighborhood_NWAmes	-7.030672e-03
Neighborhood_OldTown	-7.073000e-03

Neighborhood_Somerst	2.838542e-02
Neighborhood_StoneBr	1.019548e-01
Condition1_Artery	-3.531361e-02
Condition1_Norm	3.082964e-02
Condition1_RRAe	-5.609350e-02
Condition2_PosA	1.176925e-01
Condition2_PosN	-5.784089e-01
BldgType_1Fam	4.223849e-03
BldgType_Duplex	-1.047269e-04
OverallQual	5.139810e-02
OverallCond	3.386813e-02
YearBuilt	1.650690e-03
YearRemodAdd	8.225704e-04
RoofMatl_ClyTile	-1.665438e+00
RoofMatl_Membran	5.421716e-03
RoofMatl_WdShngl	8.027969e-02
Exterior1st_BrkComm	-1.161625e-01
Exterior1st_BrkFace	6.351341e-02
Exterior1st_HdBoard	-1.565826e-03
Exterior1st_MetalSd	3.175074e-03
Exterior1st_Wd Sdng	-1.206862e-02
MasVnrType_BrkCmn	-1.850684e-02
MasVnrArea	2.380841e-05
ExterQual_Ex	1.401279e-02
ExterQual_TA	-1.237096e-02
ExterCond_Ex	4.621793e-03
ExterCond_Fa	-2.089533e-02
ExterCond_TA	5.830882e-03
Foundation_BrkTil	-3.798789e-03
Foundation_PConc	1.842084e-02
Foundation_Stone	4.122043e-02
Foundation_Wood	-4.152916e-02
BsmtQual_Ex	4.704593e-02
BsmtQual_TA	-2.594669e-03
BsmtCond_Fa	-1.554424e-02
BsmtExposure_Gd	3.713350e-02
BsmtExposure_No	-1.050262e-02
BsmtFinType1_GLQ	3.425392e-03
BsmtFinType1_Unf	-1.335438e-02
BsmtFinSF1	5.685118e-05
BsmtFinType2_ALQ	1.334784e-02
BsmtFinType2_BLQ	-5.936410e-03
BsmtFinSF2	1.134470e-05
TotalBsmtSF	1.670748e-02
Heating_GasW	3.411951e-02
Heating_Grav	-9.844486e-02
Heating_OthW	-7.012744e-03
HeatingQC_Ex	2.434897e-02
HeatingQC_TA	-5.209819e-03
CentralAir_N	-3.408670e-02
X1stFlrSF	6.523085e-02
GrLivArea	3.746197e-01
BsmtFullBath	1.957327e-02
FullBath	1.235946e-02
HalfBath	3.402584e-03
KitchenAbvGr	-5.489073e-02

KitchenQual_Ex	6.578285e-02
KitchenQual_TA	-4.802735e-03
Functional_Maj2	-1.528964e-01
Functional_Mod	-6.965478e-03
Functional_Sev	-1.969989e-01
Functional_Typ	6.113198e-02
Fireplaces	1.859045e-02
FireplaceQu_Gd	2.914865e-03
FireplaceQu_No Fireplace	-1.445146e-03
GarageType_2Types	-4.781206e-02
GarageType_Basment	-2.802517e-03
GarageType_CarPort	-1.198580e-03
GarageCars	3.192550e-02
GarageArea	8.300926e-05
GarageQual_Ex	6.967168e-02
GarageQual_Fa	-2.124709e-03
GarageQual_Gd	5.770117e-03
GarageCond_Fa	-3.141172e-02
PavedDrive_Y	5.616107e-03
WoodDeckSF	6.506392e-05
OpenPorchSF	1.906913e-05
ScreenPorch	1.503068e-04
PoolArea	8.069897e-05
PoolQC_Ex	4.616803e-02
Fence_GdWo	-1.530234e-02
MiscFeature_Othr	-3.078158e-02
SaleType_Con	7.252894e-04
SaleType_ConLD	2.749498e-02
SaleType_New	3.683787e-02
SaleType_WD	-7.268857e-03
SaleCondition_Abnorml	-5.480935e-02
SaleCondition_Family	-3.638685e-02

Appendix E

Final Elastic Net Model Parameters (without outliers)

Ln(SalePrice) =

(Intercept)	2.397693e+00
MSSubClass_30	-1.809295e-02
MSSubClass_90	-1.764378e-02
MSSubClass_160	-2.626593e-02
MSZoning_C_(all)	-3.461694e-01
MSZoning_FV	2.198580e-02
MSZoning_RM	-3.785700e-02
LotArea	7.335658e-02
Street_Grvl	-2.633640e-02
Street_Pave	3.036134e-13
Alley_Grvl	-4.181273e-03
Alley_Pave	4.210664e-03
Utilities_AllPub	3.008598e-02
LotConfig_Corner	5.037646e-04
LotConfig_CulDSac	1.667724e-02
LotConfig_FR2	-1.236009e-02
LandSlope_Sev	-7.953312e-03

Neighborhood_BrkSide	3.106942e-02
Neighborhood_ClearCr	1.663605e-02
Neighborhood_Crawfor	1.053321e-01
Neighborhood_Edwards	-3.156753e-02
Neighborhood_MeadowV	-2.341713e-02
Neighborhood_Mitchel	-1.721336e-02
Neighborhood_NoRidge	6.710995e-02
Neighborhood_NridgHt	3.844208e-02
Neighborhood_NWAmes	-9.934724e-03
Neighborhood_OldTown	-4.439375e-03
Neighborhood_Somerst	2.063796e-02
Neighborhood_StoneBr	9.585158e-02
Condition1_Artery	-3.518322e-02
Condition1_Norm	2.988016e-02
Condition1_PosN	7.431478e-04
Condition1_RRAe	-5.709834e-02
Condition2_PosA	9.184483e-02
BldgType_1Fam	1.537833e-04
HouseStyle_1.5Unf	1.068534e-03
OverallQual	5.119272e-02
OverallCond	3.402457e-02
YearBuilt	1.668510e-03
YearRemodAdd	8.066964e-04
RoofMatl_ClyTile	-1.735986e+00
RoofMatl_Membran	1.602551e-02
RoofMatl_WdShngl	7.108940e-02
Exterior1st_BrkComm	-1.186983e-01
Exterior1st_BrkFace	6.315637e-02
Exterior1st_HdBoard	-5.601010e-04
Exterior1st_MetalSd	3.518939e-03
Exterior1st_Wd_Sdng	-1.285160e-02
Exterior2nd_Plywood	-5.878257e-04
MasVnrType_BrkCmn	-1.962641e-02
MasVnrType_Stone	6.454447e-03
MasVnrArea	2.616613e-05
ExterQual_Ex	1.828036e-02
ExterQual_TA	-1.262716e-02
ExterCond_Ex	4.277385e-04
ExterCond_Fa	-1.898026e-02
ExterCond_TA	6.373923e-03
Foundation_BrkTil	-3.794820e-03
Foundation_PConc	1.771444e-02
Foundation_Stone	4.186911e-02
Foundation_Wood	-4.344206e-02
BsmtQual_Ex	4.568046e-02
BsmtQual_TA	-1.428655e-03
BsmtCond_Fa	-1.599411e-02
BsmtExposure_Gd	4.268888e-02
BsmtExposure_No	-9.479463e-03
BsmtFinType1_GLQ	4.123082e-03
BsmtFinType1_Unf	-1.028318e-02
BsmtFinSF1	6.328161e-05
BsmtFinType2_ALQ	1.282707e-02
BsmtFinType2_BLQ	-4.357892e-03
BsmtFinSF2	1.506045e-05
TotalBsmtSF	1.594390e-02

Heating_GasW	3.024691e-02
Heating_Grav	-9.852389e-02
Heating_OthW	-4.410270e-03
HeatingQC_Ex	2.360456e-02
HeatingQC_TA	-5.430433e-03
CentralAir_N	-3.503254e-02
CentralAir_Y	1.172427e-12
X1stFlrSF	6.943390e-02
GrLivArea	3.754099e-01
BsmtFullBath	1.737712e-02
FullBath	1.431361e-02
HalfBath	5.560900e-03
KitchenAbvGr	-5.692243e-02
KitchenQual_Ex	6.701510e-02
KitchenQual_TA	-4.789931e-03
Functional_Maj2	-1.460251e-01
Functional_Mod	-8.216490e-03
Functional_Sev	-2.083696e-01
Functional_Typ	6.302455e-02
Fireplaces	1.543717e-02
FireplaceQu_Gd	3.991085e-03
FireplaceQu_No_Fireplace	-2.473641e-03
GarageType_2Types	-4.646142e-02
GarageType_Basment	-6.817852e-03
GarageType_BuiltIn	2.363594e-03
GarageCars	3.410383e-02
GarageArea	7.082603e-05
GarageQual_Ex	7.567624e-02
GarageQual_Fa	-1.904565e-03
GarageQual_Gd	6.073878e-03
GarageCond_Fa	-2.909088e-02
GarageCond_TA	1.456917e-03
PavedDrive_Y	7.634426e-03
WoodDeckSF	6.377688e-05
OpenPorchSF	6.394638e-05
ScreenPorch	1.418465e-04
PoolArea	7.552771e-05
PoolQC_Ex	4.637901e-02
Fence_GdWo	-1.377113e-02
MiscFeature_Othr	-2.695481e-02
SaleType_Con	1.409528e-03
SaleType_ConLD	2.395426e-02
SaleType_New	3.820194e-02
SaleType_WD	-7.152694e-03
SaleCondition_Abnorml	-5.483267e-02
SaleCondition_Family	-3.667644e-02