Factors Affecting House Price

Tianying Xu

Abstract

This report analyzes factors affecting house sale price using Ames housing data since house price differs a lot. With linear regression, correlation analysis, LASSO and principal component analysis, features that affect house price most are analyzed. Using Multivariate Analysis of Variance (MANOVA), difference across neighborhoods on house price is tested. How houses tend to clustered is mentioned using cluster analysis. In the end, whether house price in a neighborhood is overpriced, underpriced or with fair price is discussed with prediction of linear regression and XGBoost. What I found is that both size and geographical location affect housing price a lot.

Introduction

1. Background

House price differs a lot. There are plenty of factors that may affect house price, such as size of a house, amount of rooms, quality etc. Therefore, it is hard for people to predict house price. Thus the goal of this report is to detect factors that affect house price significantly and predict price for each house so that we can get a general sense of whether the house is overpriced or underpriced or with fair price.

2. Solution

Factors are split into four categories: continuous numerical feature, discrete numerical feature, ordered categorical feature and unordered categorical feature. Continuous numerical features, most of which are related to house size, are analyzed first since house size tends to be the most significant factor to price. Then, importance of categorical features are assessed after adjusting size effect. Finally, location information like neighborhood, zoning and type of dwelling ("MSSubClass") are discussed since these are also tend to be significant to house price.

Methods

1. Data source

The data set is Ames housing data, which describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 1460 observations and a large number of features such as house size, amount of rooms, neighborhood information, year of sold, sale price etc.

The data set was cleaned before use. First, features that contain more than 25% missing value were deleted. Then, in terms of continuous numerical features, I filled the missing values with the mean of other values in this feature. Finally, for discrete numerical features and categorical features, observation containing missing values were removed. After all these, there are altogether 1338 observations with 76 variables.

2. Methods in the analysis

In the first part of analysis, I used linear regression with scaled continuous numerical features as predictors and scaled sale price as outcome at first. After deleting three observations shown as outliers in the first regression, I did linear regression again. Then I checked the correlation between features. After calculating correlation matrix, I plotted ridge, LASSO and elastic net method, which are all feature selection methods, to choose one method to do the feature selection. Next, I chose LASSO with lamda 0.03 and stepwise selection with direction “both” to choose features. Finally, I did Principal Component Analysis (PCA) to get a lower dimensional representation of continuous features.

In the second part of analysis, I assessed importance of categorical features after adjusting size effect. With residuals of the linear regression from the first part as outcome, I did linear regression of residuals on each categorical feature, one at a time, and used adjust square as the importance of that categorical feature.

In the third part, I tried to detect whether these is significant difference of house size and price across neighborhoods. I chose lot area, total basement size, first floor size and second floor size as the size features and sale price as price feature. I first created Exploratory Data Analysis (EDA) plots to visualize the situation. Then, I did MANOVA on these features across the neighborhoods.

In the fourth part, I clustered houses with all numerical features. At first, I did hierarchical cluster with “average” method and I figured out outliers in the clustering. After deleting those outliers, I re-clustered houses using hierarchical cluster with “ward.D2” method. According to the result, I created heatmap to visualize the similarity within and among clusters. Neighborhood, zoning and type of dwelling (“MSSubClass”) proportion within each cluster are also plotted to see if clustering annotate information from these three features. Then, I did K-means clustering. Although both elbow methods and gap statistic method indicate that 8 should be the best k value, I preferred 3 since visualization will be much clearer. Similarly, I created plot with only information of K-means clustering and information of both clustering and neighborhood. Therefore, it is convenient to tell whether there is similarity between clustering and information like neighborhood, zoning and type of dwelling.

For the last part, I aimed to figure out whether a house is with fair price, or is overpriced and underpriced. I fitted linear regression model including all features and used the fitted value as the prediction for each house. Then I took the median among absolute value of residuals as threshold. House price lower than prediction subtract threshold was identified as “underprice”, house price higher than sum of prediction and threshold was identified as “overprice”, and house price other than previous two cases, which means it is within prediction minus threshold and prediction add threshold, was identified as “fair price”. Therefore, after visualization, it was evident to see overpriced, underpriced and fair priced neighborhoods. In the end, I also tried XGBoost to predict the house price. After tuning parameters with grid search, I ended up with Mean Absolute Error as 113.62, which is great compared to the mean housing price as 186762. Visualization was created to show the price situation of each neighborhood.

Results

1. Regression and correlation

In the first part, the regression results are shown as below:

```
Call:
lm(formula = SalePrice ~ ., data = D1)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.3333 -0.2259 -0.0052  0.2216  3.6062 

Coefficients: (2 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -9.711e-17  1.587e-02   0.000 1.000000    
LotFrontage   -1.832e-02  1.840e-02  -0.995 0.319687    
LotArea        2.016e-02  1.728e-02   1.167 0.243483    
BsmtFinSF1     3.559e-01  4.269e-02   8.337  < 2e-16 ***
BsmtFinSF2     8.127e-02  2.213e-02   3.673 0.000249 ***
BsmtUnfSF      2.703e-01  4.105e-02   6.585 6.55e-11 ***
TotalBsmtSF           NA         NA      NA       NA    
X1stFlrSF      3.068e-01  3.666e-02   8.368  < 2e-16 ***
X2ndFlrSF      4.127e-01  1.833e-02  22.513  < 2e-16 ***
LowQualFinSF  -3.394e-04  1.605e-02  -0.021 0.983134    
GrLivArea             NA         NA      NA       NA    
GarageArea     2.278e-01  1.998e-02  11.404  < 2e-16 ***
WoodDeckSF     8.017e-02  1.692e-02   4.738 2.39e-06 ***
OpenPorchSF    2.955e-02  1.729e-02   1.709 0.087638 .  
EnclosedPorch -5.795e-02  1.639e-02  -3.535 0.000422 ***
PoolArea      -4.374e-02  1.650e-02  -2.652 0.008106 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5806 on 1324 degrees of freedom
Multiple R-squared:  0.6662,	Adjusted R-squared:  0.663 
F-statistic: 203.3 on 13 and 1324 DF,  p-value: < 2.2e-16
```

Figure 1-1 Regression Table before Removing Outliers

```
Call:
lm(formula = SalePrice ~ ., data = D12)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.00227 -0.22269  0.01307  0.24794  2.91089 

Coefficients: (2 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.011274   0.013348   0.845 0.398488    
LotFrontage    0.035957   0.015732   2.286 0.022431 *  
LotArea        0.030175   0.014529   2.077 0.038001 *  
BsmtFinSF1     0.606719   0.037952  15.986  < 2e-16 ***
BsmtFinSF2     0.143003   0.018832   7.593 5.86e-14 ***
BsmtUnfSF      0.453758   0.035653  12.727  < 2e-16 ***
TotalBsmtSF          NA         NA      NA       NA    
X1stFlrSF      0.225161   0.031164   7.225 8.44e-13 ***
X2ndFlrSF      0.454782   0.015712  28.945  < 2e-16 ***
LowQualFinSF   0.003810   0.013482   0.283 0.777555    
GrLivArea            NA         NA      NA       NA    
GarageArea     0.185318   0.016909  10.959  < 2e-16 ***
WoodDeckSF     0.059047   0.014265   4.139 3.70e-05 ***
OpenPorchSF    0.046888   0.014608   3.210 0.001361 ** 
EnclosedPorch -0.053662   0.013775  -3.896 0.000103 ***
PoolArea      -0.001126   0.015379  -0.073 0.941627    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4874 on 1321 degrees of freedom
Multiple R-squared:  0.7561,	Adjusted R-squared:  0.7537 
F-statistic:   315 on 13 and 1321 DF,  p-value: < 2.2e-16
```

Figure 1-2 Regression Table after Removing Outliers

The regression tables indicate evidently that adjusted R square increases from 0.663 to 0.753 after removing outliers. The right table also shows that “BsmtFinSF1”, which means Type 1 finished square feet, tends to be the most important feature among all the continuous features. The coefficient 0.61 means that when "BsmtFinSF1" increases with amount of its standard deviation, sale price will increase 0.61 of its standard deviation. The NA of “TotalBsmtSF” and “GrLivArea” indicate that may be they are highly correlated with other variables. Then residual plots were created as below:

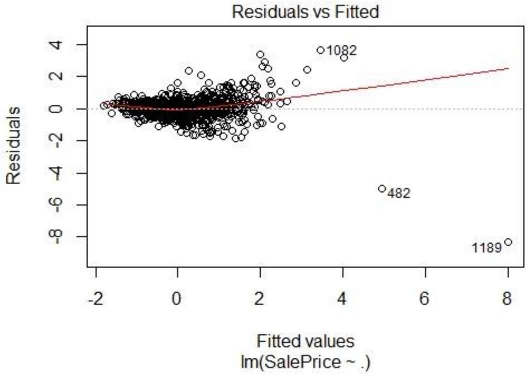


Figure 1-3 Residual Plot before Removing Outliers

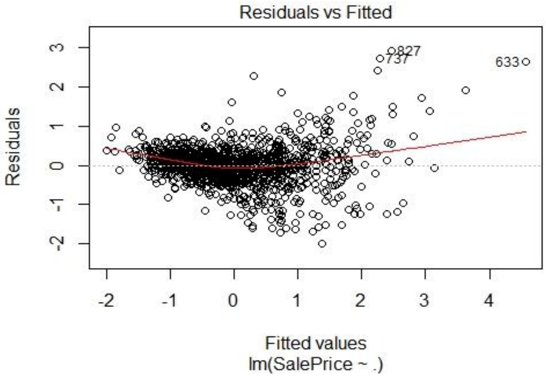


Figure 1-4 Residual Plot after Removing Outliers

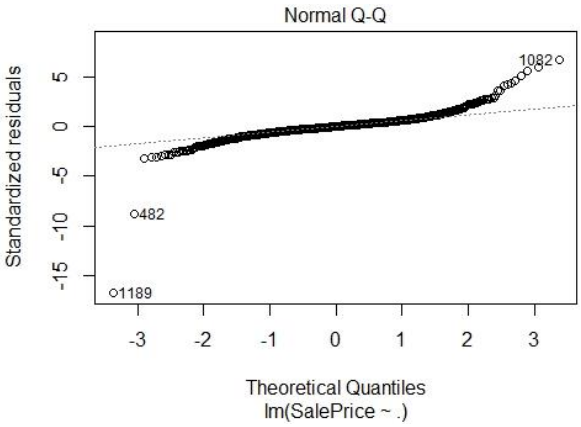


Figure 1-5 QQ-Plot before Removing Outliers

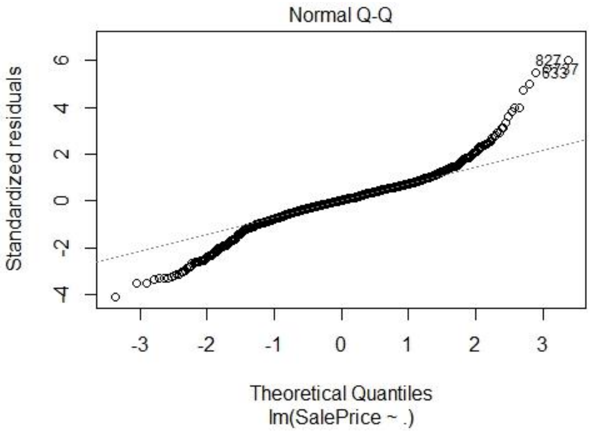


Figure 1-6 QQ-Plot after Removing Outliers

The residuals plots indicate that there is still nonlinear trend in the residuals, which means that there are trends that continuous predictors can not explain. QQ-plots show that although there is some tail issue, the residuals are approximately follow normal distribution, which validate the assumption of linear regression.

Then the correlation plots are as follow:

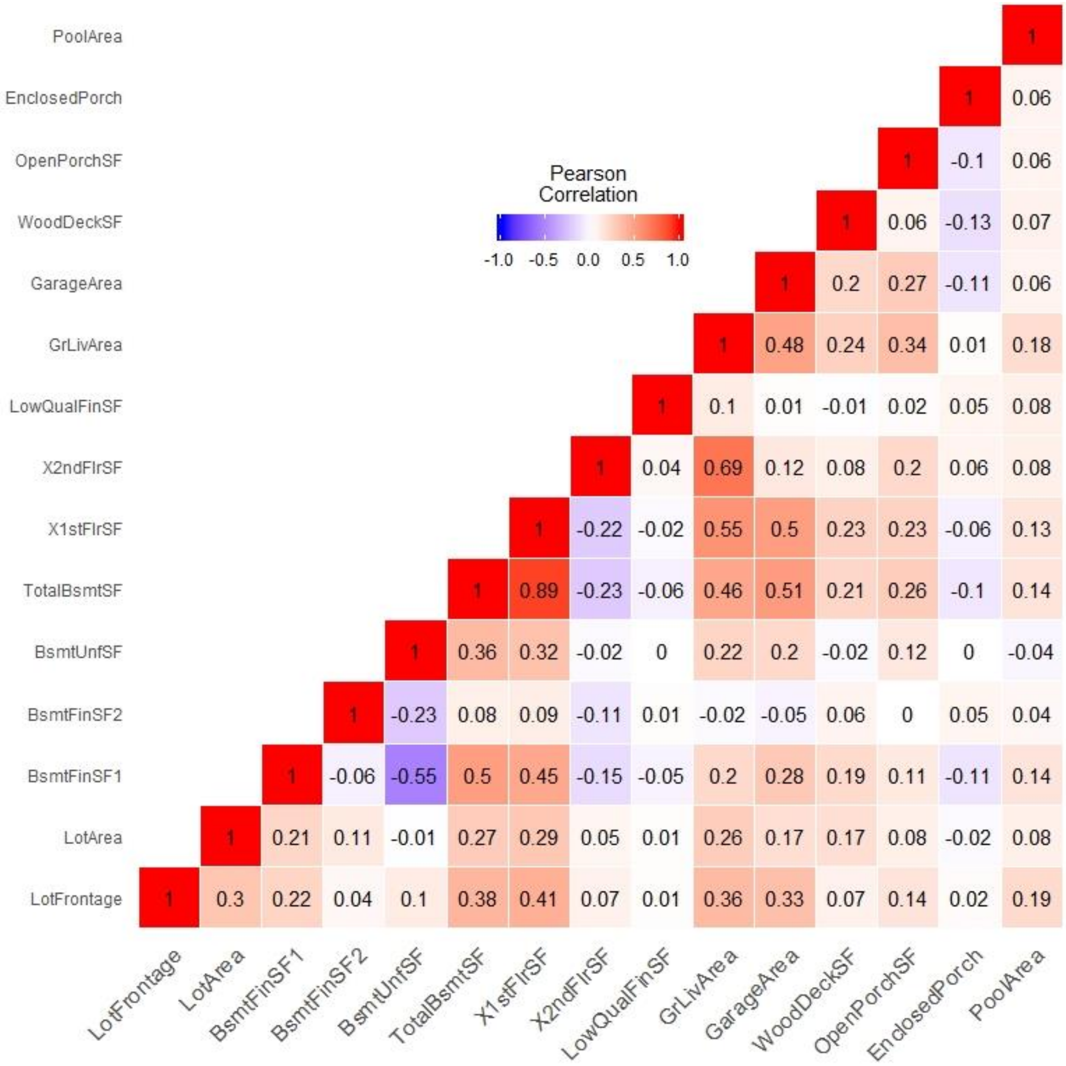


Figure 1-7 Correlation Plot with all Features

I first created a correlation plot with all the continuous features. The whole correlation plot shows that there are several features that are highly correlated, which are shown as red with their correlation. I set threshold to 0.5, therefore correlation higher than 0.5 is identified as "highly correlated". For instance, "TotalBsmtSF" is highly correlated with "BstFinSF1"; "X1stFlrSF" is highly correlated with "TotalBsmtSF"; "GrLivArea" is highly correlated with "X2ndFlrSF", "X1stFlrSF"; "GarageArea" is highly correlated with "X1stFlrSF" and "TotalBsmtSF". For those highly correlated features, I created a correlation plot for them to deep dive into their relationship.

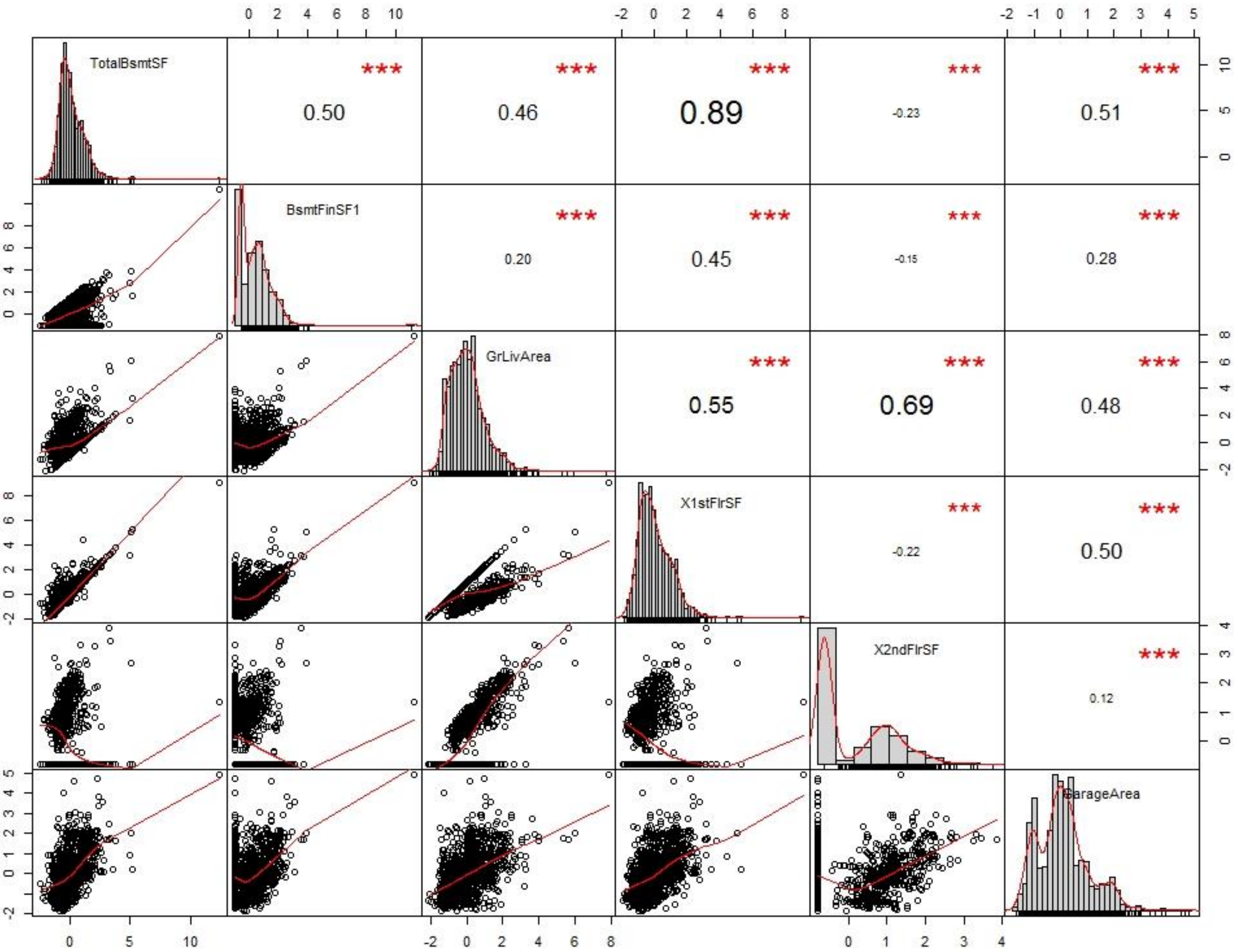


Figure 1-8 Correlation Plot with Highly Correlated Features

This correlation plot shows that some of these features have linear relationship and some of them have nonlinear relationship, and most of these features are right skewed. I also checked the relationship between features and I found out “GrLivArea” is the sum of “X1stFlrSF” and “X2ndFlrSF”, and “TotalBsmtSF” is the sum of “BsmtFinSF1”, “BsmtFinSF2” and “BsmtUnfSF”. These can explain the NA of the coefficients of “GrLivArea” and “TotalBsmtSF” in the regression table, and these findings can be intuitively understood. Therefore, I decided to remove “X1stFlrSF”, “X2ndFlrSF”, “BsmtFinSF1”, “BsmtFinSF2” and “BsmtUnfSF”.

Since there are many features and some of them are highly correlated, I tried several methods to choose features next. Plot of ridge regression, LASSO regression and elastic net regression is shown as follow:

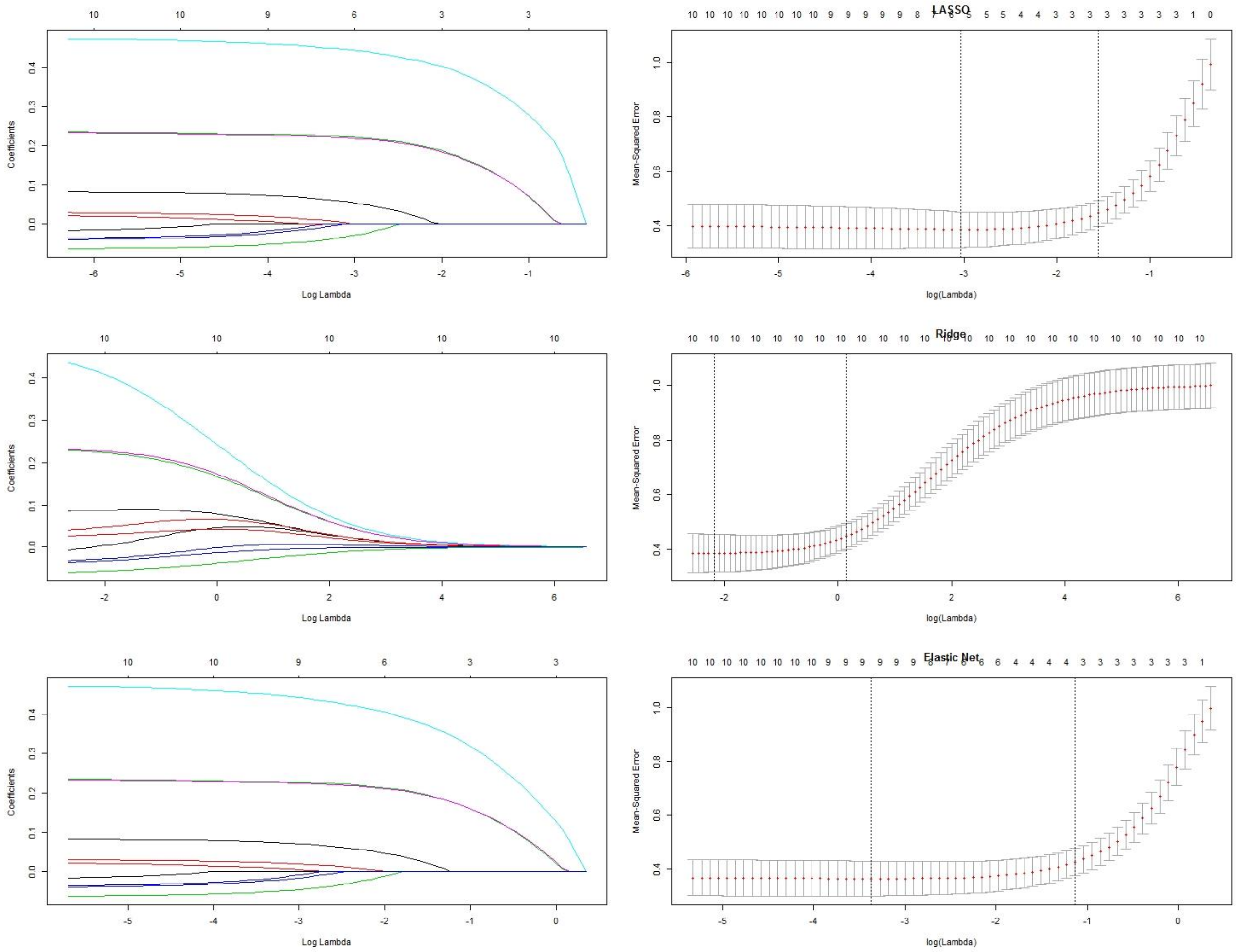


Figure 1-9 Plot of Ridge, LASSO and Elastic Net Regression

All these three methods include penalty term in regression, which sets restrict on the parameters, therefore some parameters will be 0 so that we can select several features among all. The plot indicates that ridge regression might not be a great choice in this case. Ridge regression set a circle in the parameter space so that parameters will all on the circle, therefore, sometimes it is hard to set parameters to zero as shown here. LASSO and elastic net perform similarly, and mean square error tends to increase significantly when there are less than 6 features. Hence, I decided to use LASSO to choose at least 6 features. The Lasso() function choose the lamda based on cross validation, and it chose lamda as 0.033. As a result, it kept 8 features: “TotalBsmtSF”, “LowQualFinSF”, “GrLivArea”, “GarageArea”, “WoodDeckSF”, “OpenPorchSF”, “EnclosedPorch” and “PoolArea”.

I also tried stepwise selection with “both” method. Surprisingly, it shows the same result as LASSO!

```
Call:
lm(formula = SalePrice ~ TotalBsmtSF + LowQualFinSF + GrLivArea +
    GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch + PoolArea,
    data = D2)

Coefficients:
  (Intercept)    TotalBsmtSF   LowQualFinSF      GrLivArea     GarageArea     WoodDeckSF    OpenPorchSF  EnclosedPorch       PoolArea
   -2.074e-17      2.358e-01     -4.074e-02      4.735e-01      2.307e-01      8.639e-02      3.016e-02     -6.440e-02     -3.916e-02
```

Figure 1-10 Stepwise Selection Result

Finally, I did PCA analysis. The result:

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
Standard deviation	0.6653	0.3300	0.3201	0.28819	0.27844	0.25994	0.19920	0.17935	0.1473	0.07487	5.672e-18
Proportion of Variance	0.4511	0.1110	0.1044	0.08463	0.07901	0.06886	0.04043	0.03278	0.0221	0.00571	0.000e+00
Cumulative Proportion	0.4511	0.5621	0.6665	0.75111	0.83012	0.89898	0.93941	0.97219	0.9943	1.00000	1.000e+00

Figure 1-11 Principal Component Analysis Table

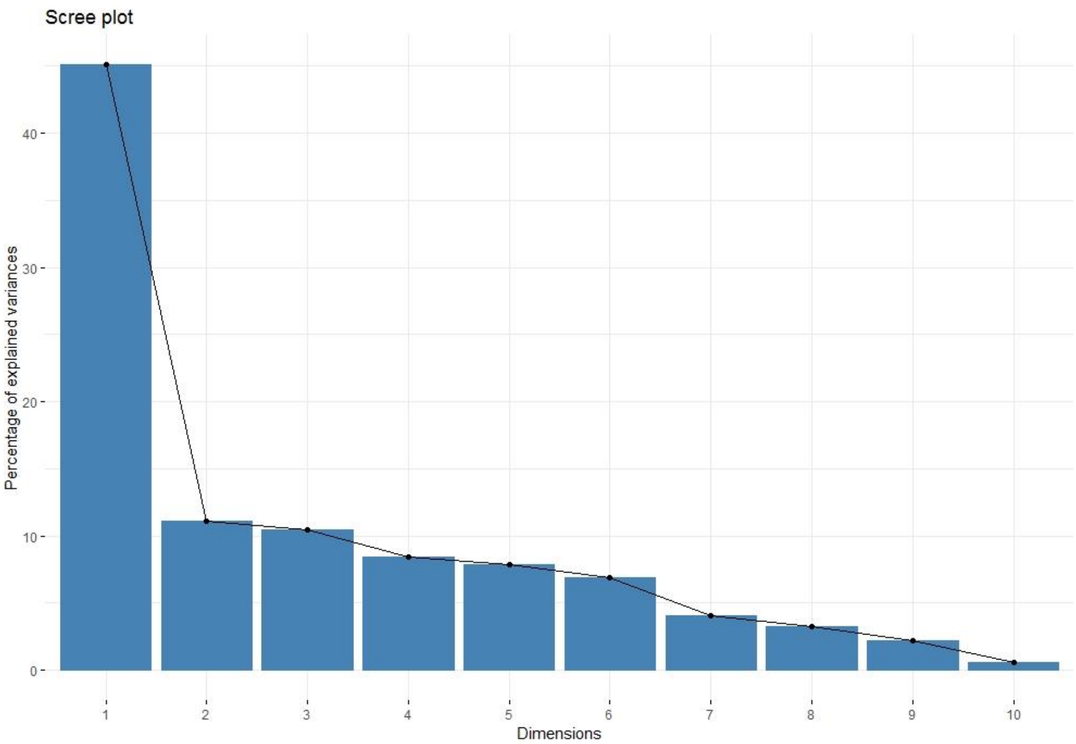


Figure 1-12 Scree Plot

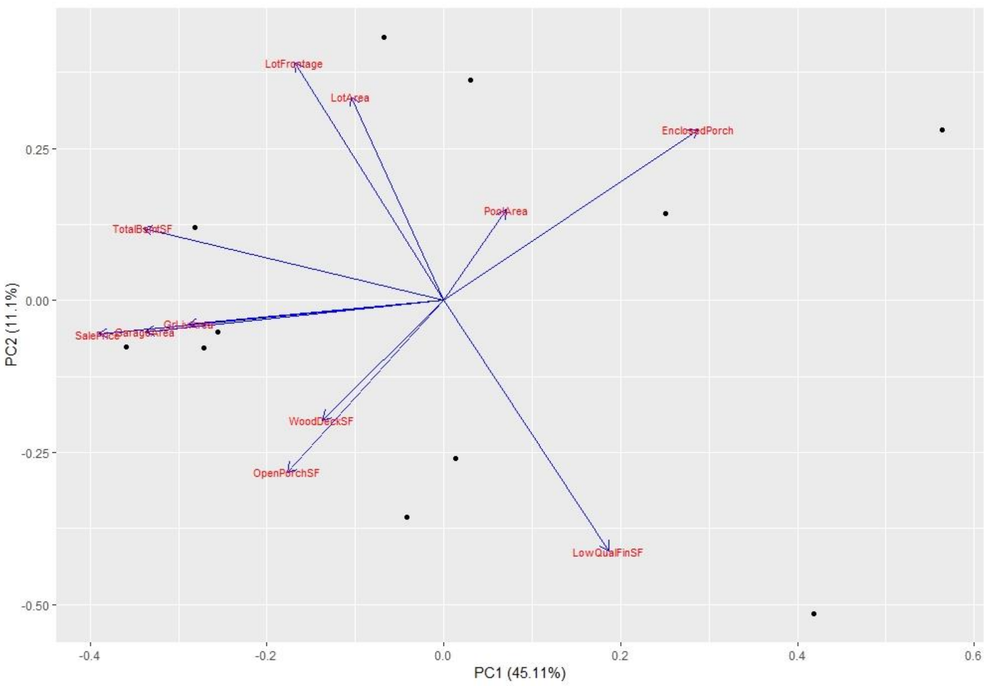


Figure 1-13 Bi-plot of PCA

These tables and plots show that 6 principle components can explain 85% of the whole variance and 8 components can explain 90% of the whole variance. Also there is a sudden decrease when dimension increase from 6 to 7, thus, I set 85% threshold and chose 6 components to represent all the continuous features.

2. Importance of categorical features

The result of final regression after feature selection in the last part looks like this:

```
Call:
lm(formula = SalePrice ~ ., data = D13)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0506 -0.2363  0.0011  0.2636  3.2269 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.0093369  0.0138829   0.673 0.501353    
TotalBsmtSF    0.3338037  0.0179394  18.607  < 2e-16 ***
LowQualFinSF  -0.0422460  0.0141015  -2.996 0.002788 ** 
GrLivArea      0.5028224  0.0177708  28.295  < 2e-16 ***
GarageArea     0.2032348  0.0172866  11.757  < 2e-16 ***
WoodDeckSF     0.0690004  0.0146275   4.717 2.64e-06 ***
OpenPorchSF    0.0500252  0.0151045   3.312 0.000951 ***
EnclosedPorch -0.0632314  0.0142644  -4.433 1.01e-05 ***
PoolArea      -0.0005512  0.0159433  -0.035 0.972426    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.507 on 1326 degrees of freedom
Multiple R-squared:  0.7351,	Adjusted R-squared:  0.7335 
F-statistic: 459.9 on 8 and 1326 DF,  p-value: < 2.2e-16
```

Figure 2-1 Regression Table of Final Regression

Although the adjusted R square is less than 0.75 as before, but there is only 8 features now, which is much simpler than before. Then I used the residuals of this model as outcome to regression on each categorical feature, the adjust R square was identified as importance for each of them. In this way, importance of categorical features after adjusting size effect can be assessed. The result is ordered from high importance to low importance as follow:

Name	Importance
Neighborhood	0.21
ExterQual	0.18
KitchenQual	0.17
BsmtQual	0.14
BldgType	0.11
SaleType	0.1
SaleCondition	0.1
HeatingQC	0.08
Foundation	0.08
Exterior1st	0.07
Exterior2nd	0.07
GarageType	0.07
BsmtFinType1	0.06
GarageFinish	0.06

Functional	0.05
MasVnrType	0.05
BsmtExposure	0.04
MSZoning	0.03
Condition1	0.03
CentralAir	0.02
HouseStyle	0.02
MasVnrArea	0.02
BsmtCond	0.01
BsmtFinType2	0.01
GarageQual	0.01
PavedDrive	0.01
Condition2	0.01
Street	0.01
RoofStyle	0.01
RoofMatl	0.01
Heating	0.01
Electrical	0.01
ExterCond	0
GarageCond	0
MSSubClass	0
MiscVal	0
Utilities	0

Table 2-1 Importance of Categorical Features

The table shows that neighborhood is the most important feature among all the categorical features. Therefore, in the later parts, more analysis for neighborhoods is discussed.

3. Difference of house size and price across neighborhoods

Since neighborhood plays an important role in affecting sale price, I created plots about house size and sale price in different neighborhoods. I chose “LotArea”, “TotalBsmtSF”, “GrLivArea” as indicators of house size and “SalePrice” as price indicator. They seem different across the neighborhoods:

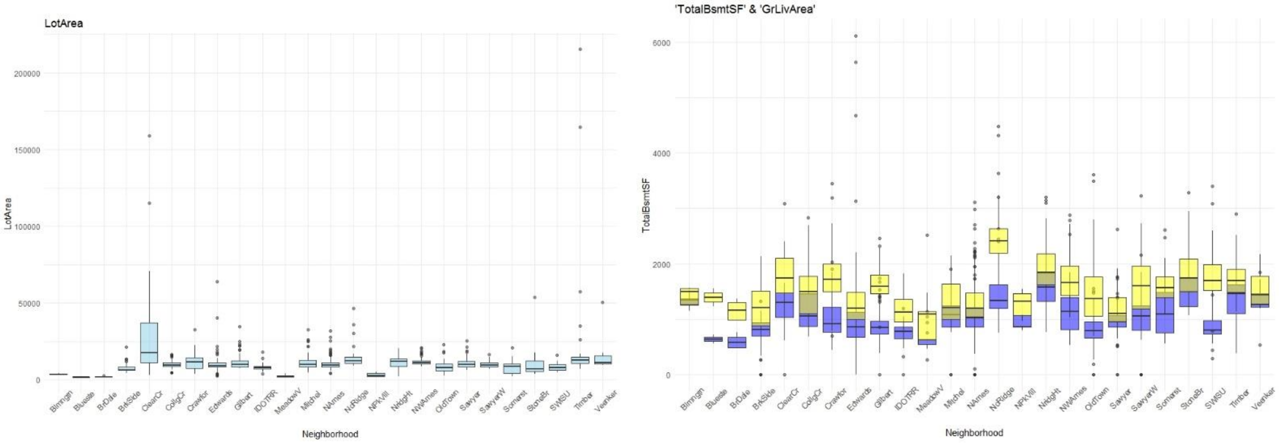


Figure 3-1 House Size across Neighborhoods

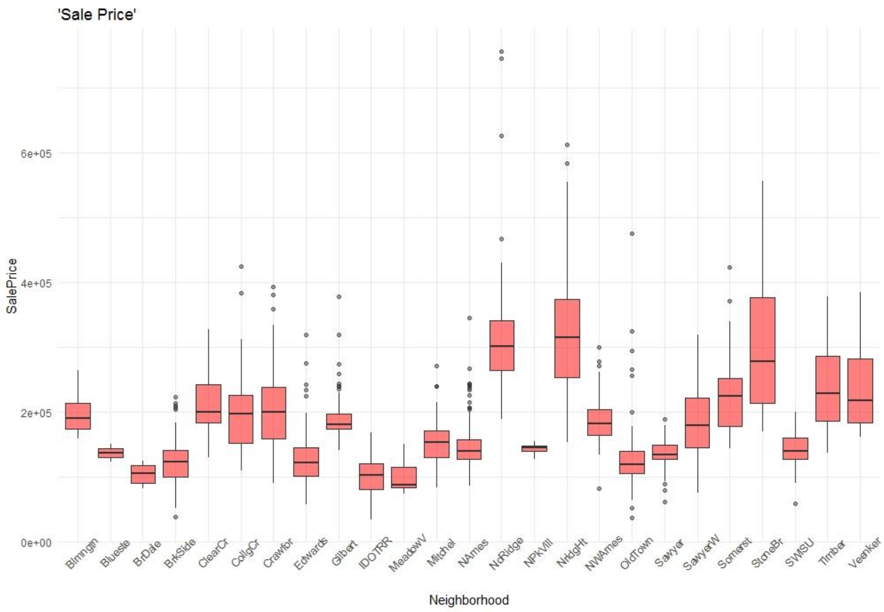


Figure 3-2 Sale Price across Neighborhoods

The MANOVA result is the same as what we expected.

```
 Response LotArea :
                Df     Sum Sq    Mean Sq F value    Pr(>F)    
Neighborhood    24 2.4884e+10 1036822347   11.54 < 2.2e-16 ***
Residuals     1313 1.1797e+11   89846925                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response TotalBsmtSF :
                Df    Sum Sq Mean Sq F value    Pr(>F)    
Neighborhood    24  60438999 2518292  20.735 < 2.2e-16 ***
Residuals     1313 159463273  121450                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response X1stFlrSF :
                Df    Sum Sq Mean Sq F value    Pr(>F)    
Neighborhood    24  50655524 2110647  18.572 < 2.2e-16 ***
Residuals     1313 149218415  113647                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response X2ndFlrSF :
                Df    Sum Sq Mean Sq F value    Pr(>F)    
Neighborhood    24  53594143 2233089  14.259 < 2.2e-16 ***
Residuals     1313 205631559  156612                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response SalePrice :
                Df     Sum Sq    Mean Sq F value    Pr(>F)    
Neighborhood    24 4.3856e+12 1.8273e+11   60.89 < 2.2e-16 ***
Residuals     1313 3.9404e+12 3.0011e+09                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 3-3 MANOVA Result

All P-values are less than 0.05, which means that both house size and house price are statistically significant different across all neighborhoods.

4. Houses clustering

When first clustering 1338 houses using hierarchical cluster with “average” method, the result looks like this:

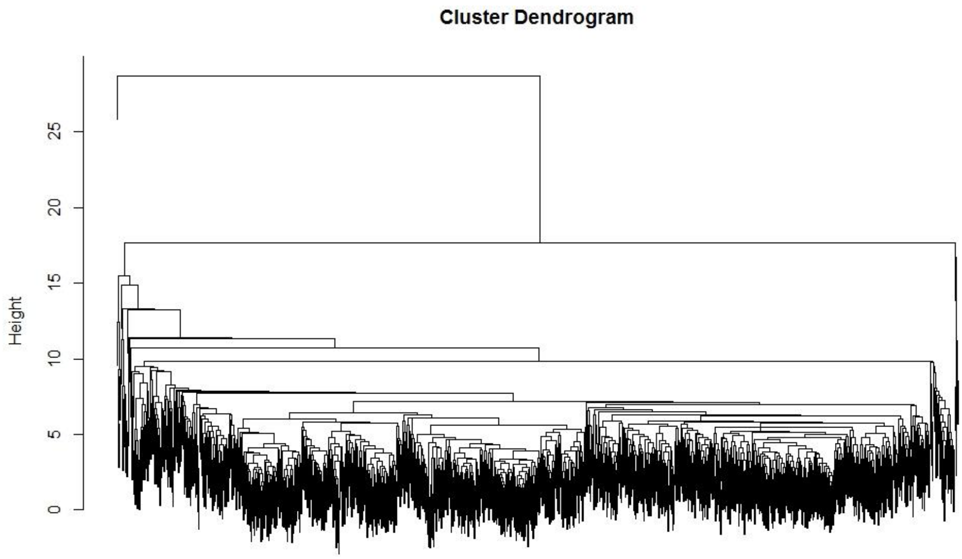


Figure 4-1 Dendrogram for Hierarchical Clustering – "average"

The plot indicates that there are several outliers in these houses, thus I cut the clustering into 10 groups and removed 9 groups with small numbers. Thus, I got 1312 houses and I re-clustered them using hierarchical cluster with "ward.D2" method. The result looks really balanced:

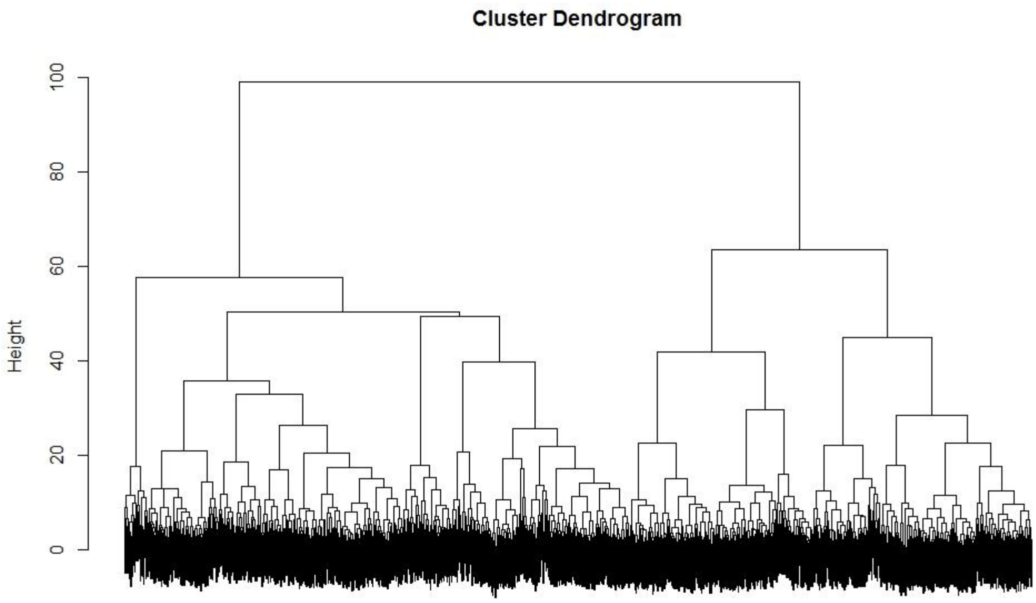


Figure 4-2 Dendrogram for Hierarchical Clustering – "ward.D2"

I also created heatmap based on this clustering:

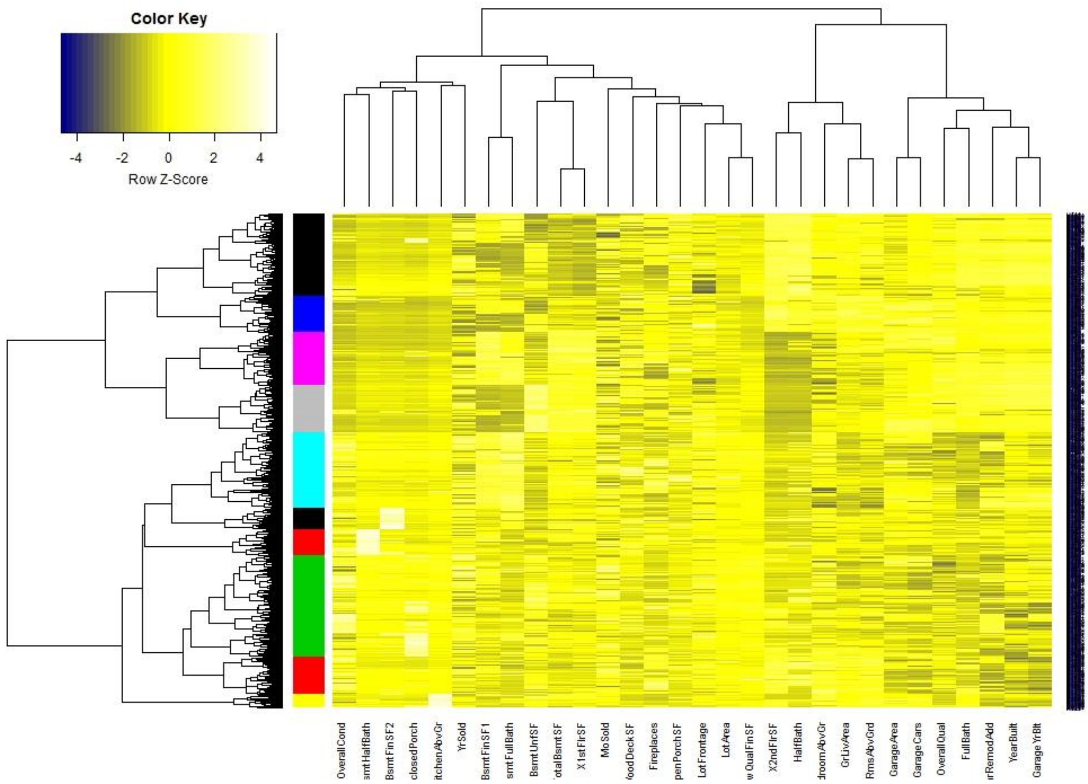


Figure 4-3 HeatMap

I cut the clustering result into 10 groups, and used different color to indicate them as shown in the heatmap. Also, there seems to be some similarity within a group. For instance, in left top of the heatmap, it seems to be a dark blue area, which means that top clusters all have negative values in the first four or five features, while clusters below in the left bottom area tend to have positive values on these features. Another instance is that in the bottom right area, clusters below tend to have negative values in the last five features, while top clusters tend to have positive values in these features.

Then, plots of "Neighborhood", "MSZoning" and "MSSubClass" proportion within clusters indicate that clustering may annotate some information from these features.

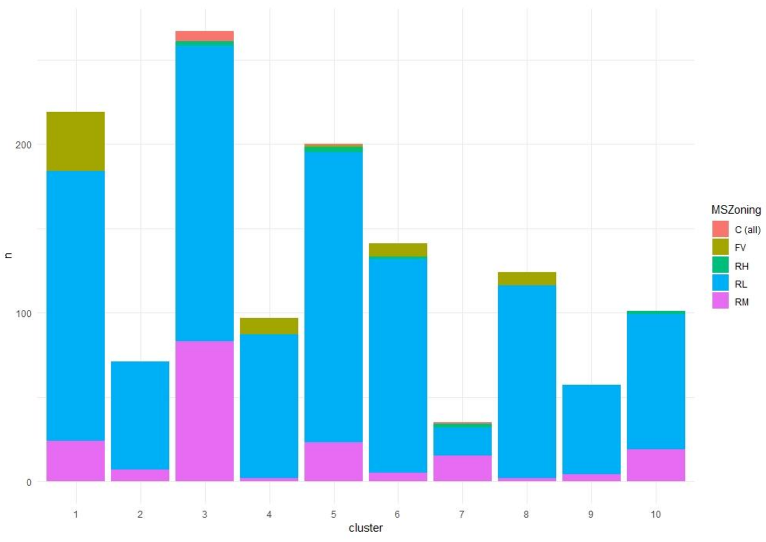
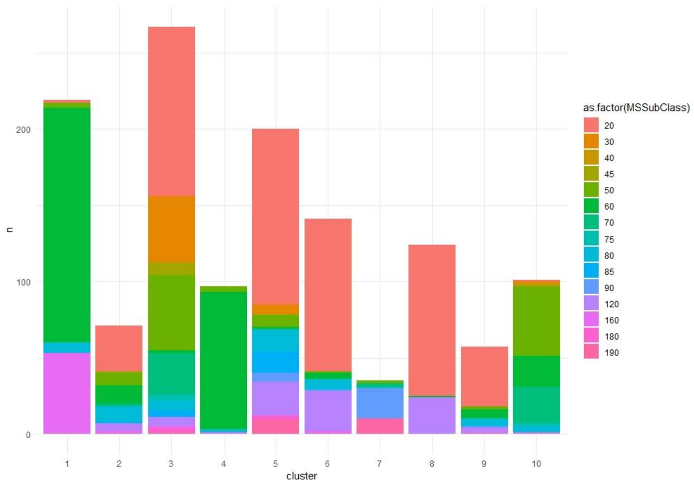


Figure 4-4 "MSZoning" proportion within clusters

This plot shows that although it is really unbalanced in the levels of “MSZoning”, there is still pattern here. For instance, the red area, corresponding to Commercial -- “C” in MSZoning information, mostly appears in cluster 3 and a little bit in cluster 5. Also, most Floating Village Residential houses, which is “FV” in MSZoning information are in cluster 1, and few in cluster 4, 5, 6 and 8. For Residential Medium Density, which is “RM” in MSZoning, it shows in all clusters but in cluster 3, it has a large proportion while in cluster 4, 6, 8 and 9, it has tiny proportion.



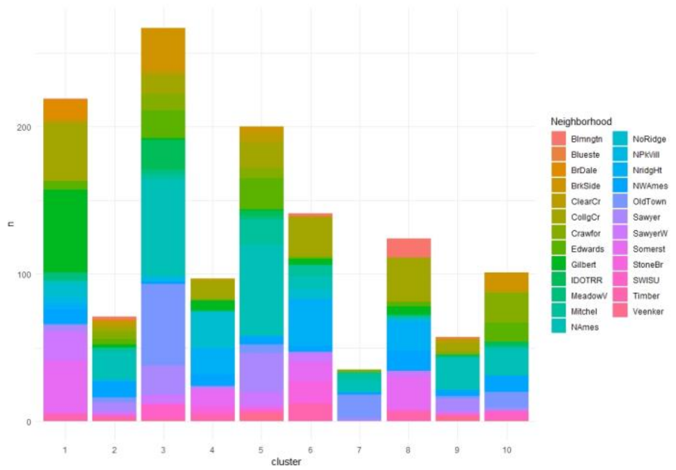
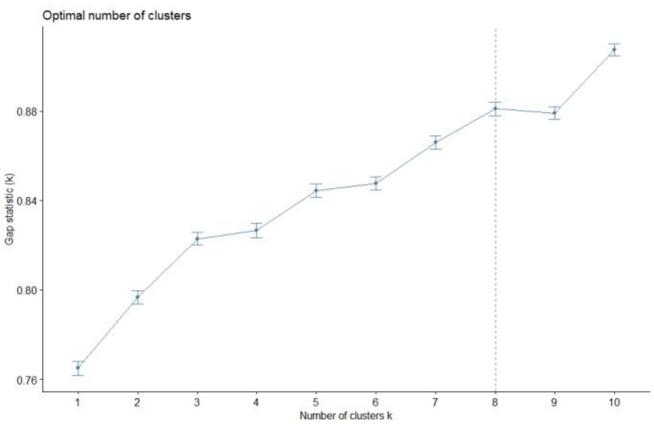


Figure 4-5 “MSSubClass” proportion within cluster Figure 4-6 “Neighborhood” proportion within clusters

Plots above show similar result for “MSSubClass” and “Neighborhood”, therefore, clustering result can annotate information from “MSZoning”, “MSSubClass” and “Neighborhood”.

Next, I tried K-means clustering to cluster houses. To choose k, I use both gap statistic and elbow method.



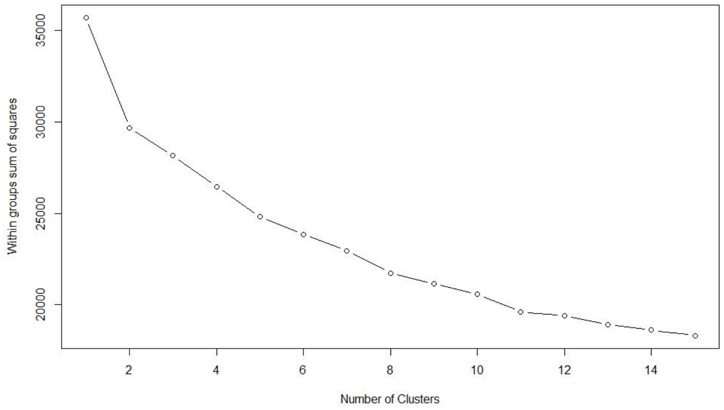


Figure 4-7 Gap Statistic

Figure 4-8 Elbow Method

Both two methods indicate that 8 is the best value for K, however, the plot is kind of a mess when k is 8:

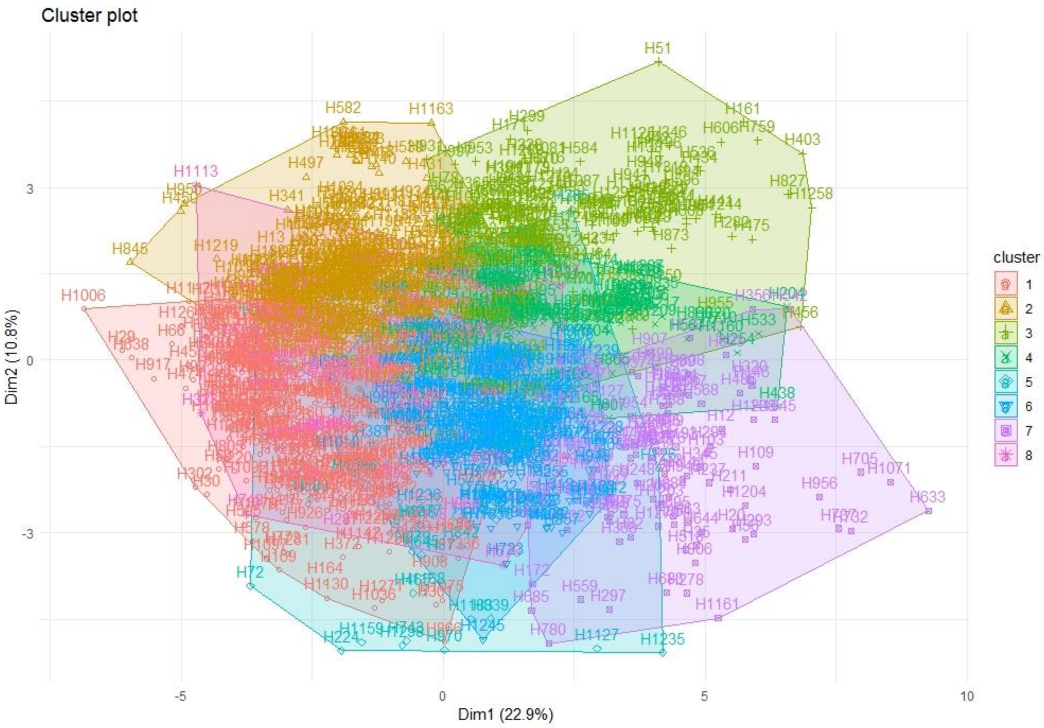


Figure 4-9 K-means with 8 groups

However, when K is 3, it is pretty clear!

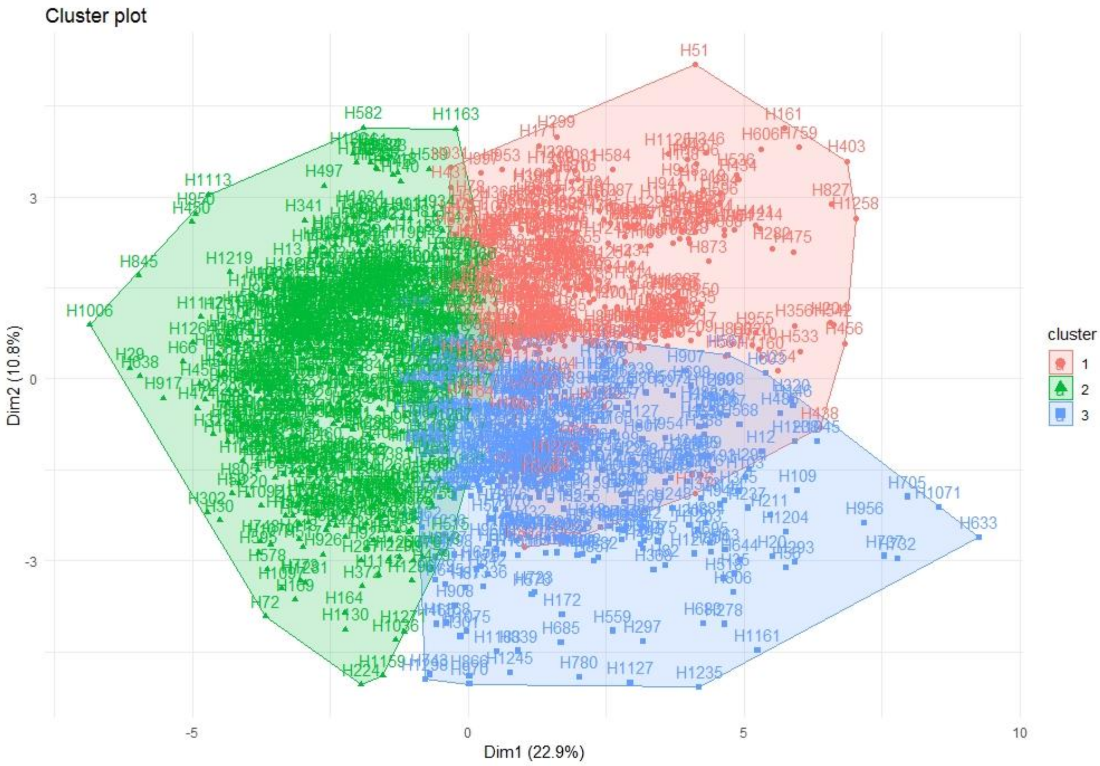


Figure 4-10 K-means with 3 groups

Finally, plots of "Neighborhood", "MSZoning" and "MSSubClass" and clusters indicate that clustering may annotate some information from these features. The color of points indicates the Zoning, neighborhood or subclass information and the shape of points indicates the cluster it is in. We can see patterns here.

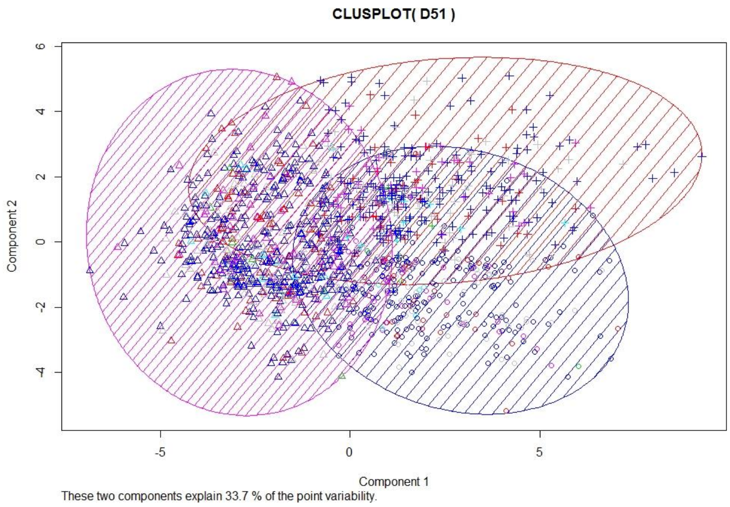


Figure 4-11 K-means with "MSZoning"

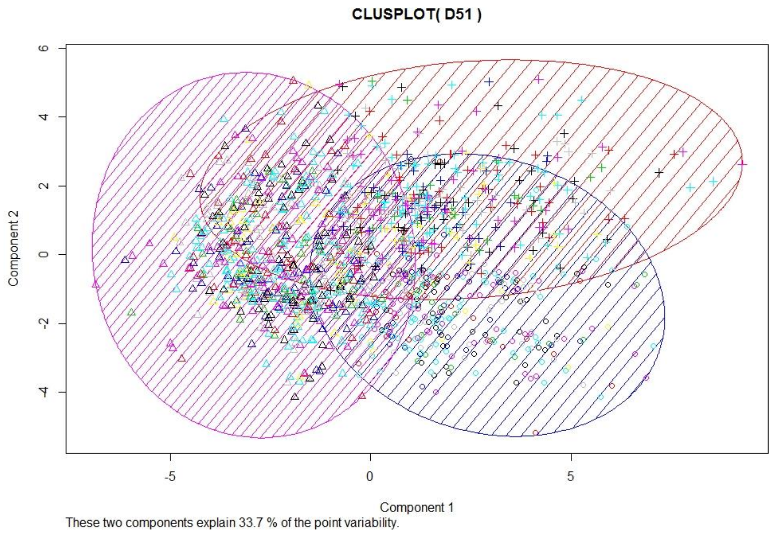


Figure 4-12 K-means with "MSSubClass"

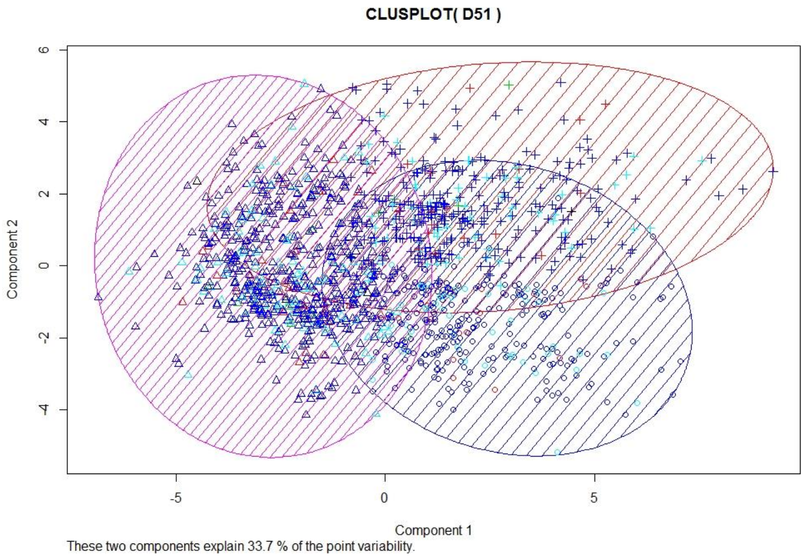


Figure 4-13 K-means with "Neighborhood"

Therefore, clustering result can annotate information from “MSZoning”, “MSSubClass” and “Neighborhood”.

5. Overprice, underprice and fair price

For the linear regression, the median of absolute residuals is 9500, and based on this, the price situation in each neighborhood is shown in the following plot ordered:

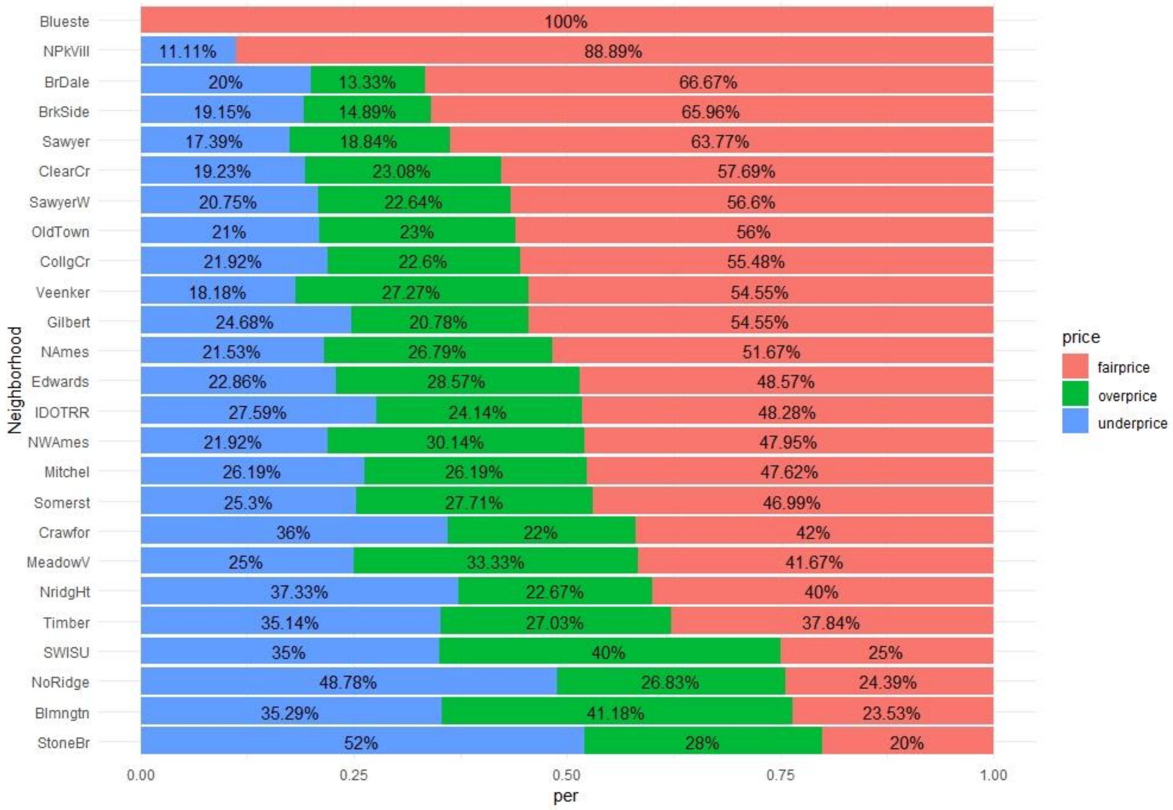


Figure 5-1 Price Situation in Neighborhoods --- Linear regression

I also used XGBoost method. I tuned parameters such as iterations, depth of decision tree and shrinkage with grid search. In the end, I came up with train Mean Absolute Error (MAE) 113.62. The median of residuals is 45.4, which is much better than linear regression.

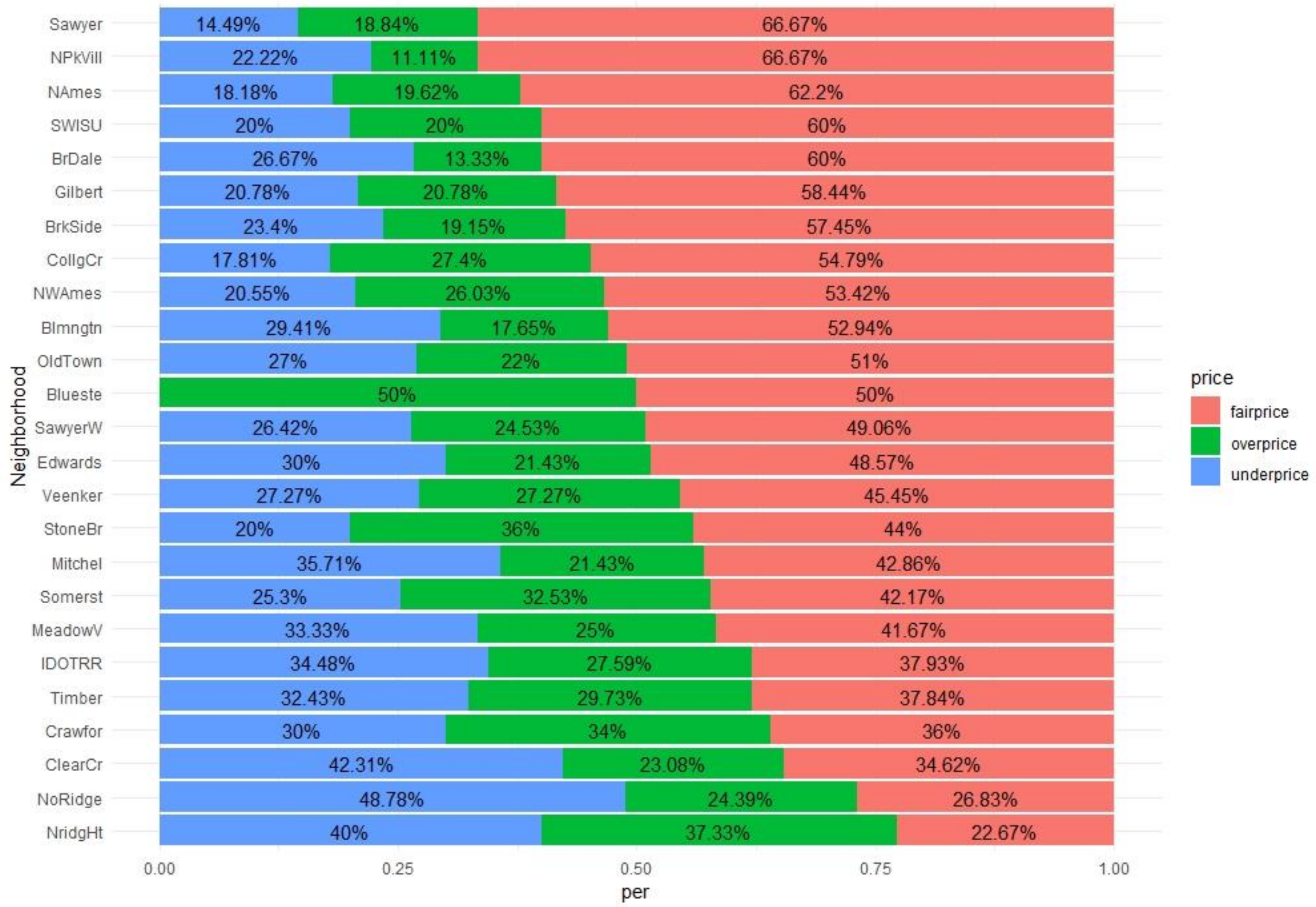


Figure 5-2 Price Situation in Neighborhoods --- XGBoost

I set the threshold that: if there is more than 50% houses in a neighborhood is with fair price, then the neighborhood is identified as “fair price”; else if there is more “overprice” houses than “underprice” in a neighborhood, then the neighborhood is identified as “overprice”; else if there is more “underprice” houses than “overprice”, it is identified as “underprice”; otherwise if there is any equal amount of houses of price situation, it is identified as “fair price”.

The result of two methods are as follow:

Neighborhood	**Linear Regression**	**XGBoost**	**Overall**
Blueste	Fair Price	Fair Price	Fair Price
NPkVill	Fair Price	Fair Price	Fair Price
BrDate	Fair Price	Fair Price	Fair Price
BrkSide	Fair Price	Fair Price	Fair Price
Sawyer	Fair Price	Fair Price	Fair Price
ClearCr	Fair Price	Underprice	?
SawyerW	Fair Price	Underprice	?
OldTown	Fair Price	Fair Price	Fair Price
CollgCr	Fair Price	Fair Price	Fair Price
Veenker	Fair Price	Fair Price	Fair Price

Gilbert	Fair Price	Fair Price	Fair Price
NAmes	Fair Price	Fair Price	Fair Price
Edwards	Overprice	Underprice	?
IDOTRR	Underprice	Underprice	Underprice
NWAmes	Overprice	Fair Price	?
Mitchel	Fair Price	Underprice	?
Somerst	Overprice	Overprice	Overprice
Crawfor	Underprice	Overprice	?
MeadowV	Overprice	Underprice	?
NridgeHt	Underprice	Underprice	Underprice
Timber	Underprice	Underprice	Underprice
SWISU	Overprice	Fair Price	?
NoRidge	Underprice	Underprice	Underprice
Blmngtn	Overprice	Fair Price	?
StoneBr	Underprice	Overprice	?

Table 5-1 Results of Linear Regression and XGBoost

Most of the results are the same, thus I can identify the price situation. However, there are still 10 neighborhoods where results are different, thus I am not sure what exactly their prick situations are.

Discussion

This report can give people a general sense about what may affect house price, how price across neighborhoods may differ and which neighborhood may be overpriced or underpriced. However, the analysis is only based on the Ames housing data during 2006 and 2010. Therefore, this dataset may not be representative to all houses across countries. As a result, the result is not referable when applying to other situation.

Acknowledgement

I would like to express my special thanks of gratitude to Professor Hyonho Chun who assisted me in gaining academic knowledge and helped me when I came into problems. Also, I would like to thank my classmates who shared about this project with me and gave me help.

Appendix: Code

```
pacman::p_load(corrplot,factoextra,ggfortify,ggplot2,reshape2,glmnet,tidyverse,cluster,HDCI,gplots,xgboost
,caret)

#I. Data
Train <- read.csv("train.csv",header=T)

#remove feature with many NA
Train_1 <- Train[,apply(Train,2,function(x){sum(is.na(x))<365})]#Alley, FirePlaceQu, PoolQC, Fence,
MiscFeature

#continuous numeric feature
#fill NA in numeric feature with mean of the rest in that feature: LotFrontage
num <- c("LotFrontage", "LotArea", "BsmtFinSF1", "BsmtFinSF2", "BsmtUnfSF", "TotalBsmtSF",
"X1stFlrSF", "X2ndFlrSF","LowQualFinSF", "GrLivArea", "GarageArea", "WoodDeckSF", "OpenPorchSF",
"EnclosedPorch", "PoolArea")
train_num <- Train_1[,num]; train_numna <-
as.data.frame(train_num[,apply(train_num,2,function(x){sum(is.na(x))>0})])
names(train_numna) <- names(train_num)[apply(train_num,2,function(x){sum(is.na(x))>0})]
Train_1$LotFrontage[is.na(Train_1$LotFrontage)] <- mean(Train_1$LotFrontage, na.rm = TRUE)

#For other feature, remove NA
Train_1 <- Train_1 %>% na.omit()


#II. Q1&Q2: Regression Analysis & Correlation
#1. Regression Analysis
# Data
D <- Train_1 %>%
  select(LotFrontage, LotArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, X1stFlrSF,
X2ndFlrSF,LowQualFinSF, GrLivArea, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, PoolArea,
SalePrice)

# Scale variables
D1 <- as.data.frame(apply(D, 2, function(col) { scale(col) }))

# first Regression
LM1 <- lm(SalePrice~.,data=D1); summary(LM1); plot(LM1,c(1,2))

#Remove Outlier
D12 <- D1[-c(482,1082,1189),]; LM12 <- lm(SalePrice~.,data=D12); summary(LM12); plot(LM12,c(1,2))

#2. Correlation
# Correlation between continuous features
D2 <- D1[,-16]; Cor <- round(cor(D2),2); print(Cor)
Cor[lower.tri(Cor)]<- NA; melted_cor <- melt(Cor,na.rm = T)
ggplot(melted_cor, aes(Var2, Var1, fill = value))+ geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space = "Lab",
name="Pearson\nCorrelation") +theme_minimal()+ theme(axis.text.x = element_text(angle = 45, vjust = 1,
size = 12, hjust = 1))+
  coord_fixed()+ geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
  theme(axis.title.x = element_blank(),axis.title.y = element_blank(),panel.grid.major =
element_blank(),panel.border = element_blank(),panel.background = element_blank(),axis.ticks =
element_blank(),legend.justification = c(1, 0),legend.position = c(0.6, 0.7),legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1, title.position = "top", title.hjust = 0.5))

#highly correlated features
PerformanceAnalytics::chart.Correlation(D1[,c("TotalBsmtSF","BsmtFinSF1","GrLivArea","X1stFlrSF","X
2ndFlrSF","GarageArea")], histogram=TRUE, pch=10)

# relationship between numeric feature
```

```
sum(Train_1$GrLivArea==Train_1$X1stFlrSF + Train_1$X2ndFlrSF)/1338
sum(Train_1$TotalBsmtSF==Train_1$BsmtFinSF1 + Train_1$BsmtFinSF2 + Train_1$BsmtUnfSF)/1338
#"GrLivArea" should be the sum of "1stFlrSF" & "2ndFlrSF";
#"TotalBsmtSF" should be sum of "BsmtFinSF1" & "BsmtFinSF2" & "BsmtUnfSF"
#Remove "1stFlrSF", "2ndFlrSF", "BsmtFinSF1", "BsmtFinSF2" & "BsmtUnfSF"
D2 <- D1 %>% select(-X1stFlrSF,-X2ndFlrSF,-BsmtFinSF1,-BsmtFinSF2,-BsmtUnfSF)

#3. Variables Selection
LM2 <- lm(SalePrice~.,data=D2); summary(LM2)

# Ridge/LASSO/Elastic Net
y <- as.matrix(D1$SalePrice); fit.lasso <- glmnet(as.matrix(D2[,-11]), y, family="gaussian", alpha=1)
fit.ridge <- glmnet(as.matrix(D2[,-11]), y, family="gaussian", alpha=0);
fit.elnet <- glmnet(as.matrix(D2[,-11]), y, family="gaussian", alpha=.5)

for (i in 0:10) {assign(paste("fit", i, sep=""), cv.glmnet(as.matrix(D2[,-
11]),y,type.measure="mse",alpha=i/10,family="gaussian"))}

par(mfrow=c(3,2)); plot(fit.lasso, xvar="lambda"); plot(fit10, main="LASSO"); plot(fit.ridge, xvar="lambda")
plot(fit0, main="Ridge"); plot(fit.elnet, xvar="lambda"); plot(fit5, main="Elastic Net")
# LASSO, choose >=6 features

#Lasso
set.seed(0); obj <- Lasso(as.matrix(D2[,-11]), y, fix.lambda = FALSE); obj$lambda; obj$beta; D_f1 <- D2[,-
11][,obj$beta!=0]

# stepwise
step(LM2, trace = 1,direction="both",steps=1000)

#4. Principal Component Analysis
# PCA
D2.pca <- prcomp(cor(D2), center = TRUE); summary(D2.pca)

#eda
fviz_eig(D2.pca); autoplot(prcomp(cor(D2)), data = cor(D2),loadings = TRUE, loadings.colour = 'blue',
loadings.label = TRUE, loadings.label.size = 3)

#Final Linear Regression
D13 <- D12 %>%
  select(TotalBsmtSF, LowQualFinSF, GrLivArea, GarageArea, WoodDeckSF, OpenPorchSF,
EnclosedPorch, PoolArea, SalePrice)
LM_f <- lm(SalePrice~., data=D13); summary(LM_f); plot(LM_f,c(1,2))


#III. Q3: Importance of Categorical Feature
D31 <- as.data.frame(Train_1) %>% select(ExterQual,ExterCond, BsmtQual, BsmtCond, BsmtExposure,
BsmtFinType1, BsmtFinType2, HeatingQC, CentralAir, KitchenQual, Functional, GarageFinish,
GarageQual,GarageCond, PavedDrive, MSSubClass, MSZoning, Condition1, Condition2, Street,
Neighborhood, BldgType, HouseStyle, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType,
MasVnrArea, Foundation, Heating, Electrical, GarageType, MiscVal, SaleType, SaleCondition, Utilities)
D31 <- D31[apply(D,1,function(x){!anyNA(x)}),]; D31 <- D31[-c(482,1082,1189),];
D331 <- D31[,apply(D31,2,function(x){length(unique(x))>1})]

y <- residuals(LM12); name=names(D331); adj_R2=c()
for(i in 1:37){lm <- lm(y~D331[,i]); adj_R2[i] <- round(summary(lm)$adj.r.squared,2)}
Importance <- as.data.frame(cbind(name,adj_R2)) %>% arrange(desc(adj_R2));
names(Importance)=c("name","importance")


#IV. Q4: House size and sale Price Different Across Neighborhoods?
#1. EDA
```

```
# data
D4 <- Train %>% select(LotArea,TotalBsmtSF,GrLivArea,SalePrice, Neighborhood)

# eda
ggplot(data=D4)+geom_boxplot(mapping=aes(x=Neighborhood,y=LotArea),alpha=0.5,fill="skyblue")+theme_minimal()+
ggtitle("LotArea")+theme(axis.text.x=element_text(angle = 45),title = element_text(hjust = 0.5))

ggplot(data=D4)+geom_boxplot(mapping=aes(x=Neighborhood,y=TotalBsmtSF),alpha=0.5,fill="blue")+

geom_boxplot(mapping=aes(x=Neighborhood,y=GrLivArea),alpha=0.5,fill="yellow")+theme_minimal()+
   ggtitle("'TotalBsmtSF' & 'GrLivArea'")+theme(axis.text.x=element_text(angle = 45),title = element_text(hjust = 0.5))

ggplot(data=D4)+geom_boxplot(mapping=aes(x=Neighborhood,y=SalePrice),alpha=0.5,fill="red")+theme_minimal()+
   ggtitle("'Sale Price'")+theme(axis.text.x=element_text(angle = 45),title = element_text(hjust = 0.5))

#2. Test
neighbor_ma <- manova(cbind(LotArea,TotalBsmtSF,X1stFlrSF,X2ndFlrSF,SalePrice)~ Neighborhood, data=Train_1)
summary.aov(neighbor_ma)


#V.   Q5: Houses Clustering
#1. Data
D5 <- Train_1 %>% select(LotFrontage, LotArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, X1stFlrSF, X2ndFlrSF,LowQualFinSF, GrLivArea, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, PoolArea, OverallQual, OverallCond, YearBuilt, YearRemodAdd, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageCars, MoSold, YrSold,MSZoning,MSSubClass,Neighborhood) %>% na.omit()
D5 <- D5[,apply(D5,2,function(x){length(unique(x))>1})]
D5[,-c(32,33,34)] <- as.data.frame(apply(D5[,-c(32,33,34)],2,function(x){scale(as.numeric(x))}))
rownames(D5) <- c(paste("H",1:1338,sep="")); D5_x <- D5[,-c(32,33,34)]

#2. Clustering
dist <- dist(D5_x,method="euclidean"); cluster1 <- hclust(dist,method="average"); cut <- cutree(cluster1, k=10); table(cut)
par(mar=c(0, 4, 4, 2)); plot(cluster1, labels=FALSE)

#Re-cluster
D51 <- D5_x[cut==1,]; D51 <- D51[,apply(D51,2,function(x){length(unique(x))>1})]; dist <- dist(D51,method="euclidean")

#ward.D2
hr <- hclust(dist,method="ward.D2"); mycl <- cutree(hr, k=10); table(mycl); plot(hr,labels=FALSE)
hc <-   hclust(dist(t(D51),method="euclidean"),method="ward.D2")
mycol <- colorpanel(40, "darkblue", "yellow", "white")
heatmap.2(as.matrix(D51), Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col=mycol,
             scale="row", density.info="none", trace="none", RowSideColors=as.character(mycl))

#3. ~Neighborhood/MSZoning/MSSubClass
D55 <- D5[cut==1,]; D55$cluster <- as.factor(as.matrix(mycl)); N = table(mycl)
D55_1 <- D55 %>% group_by(cluster,Neighborhood) %>% summarise(n=n())
D55_2 <- D55 %>% group_by(cluster,MSZoning) %>% summarise(n=n())
D55_3 <- D55 %>% group_by(cluster,MSSubClass) %>% summarise(n=n())

ggplot(D55_1)+geom_bar(aes(y = n, x = cluster, fill = Neighborhood), stat="identity")+theme_minimal()
ggplot(D55_2)+geom_bar(aes(y = n, x = cluster, fill = MSZoning), stat="identity")+theme_minimal()
ggplot(D55_3) + geom_bar(aes(y = n, x = cluster, fill = as.factor(MSSubClass)), stat="identity")+theme_minimal()
```

```
#4. K-Means Clustering
#determine K
#fviz_nbclust(D51, kmeans, method = "gap_stat",k.max=10,iter.max=20)    #8
#elbow
wss <- (nrow(D51)-1)*sum(apply(D51,2,var)); for (i in 2:15) wss[i] <- sum(kmeans(D51,centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",ylab="Within groups sum of squares")
fit <- kmeans(D51, 8); fviz_cluster(fit, data = D51, frame.type = "convex")+theme_minimal()
fit <- kmeans(D51, 3); fviz_cluster(fit, data = D51, frame.type = "convex")+theme_minimal()

#~MSZoning
fit <- kmeans(D51, 3); clusplot(D51,fit$cluster,color=T,shade=T,col.p = D5$MSZoning,col.txt=col.p)
clusplot(D51,fit$cluster,color=T,shade=T,col.p = D5$MSSubClass,col.txt=col.p)
clusplot(D51,fit$cluster,color=T,shade=T,col.p = D5$Neighborhood,col.txt=col.p)


#VI. Over/Under Price
#1. Linear Regression
LM_6 <- lm(SalePrice~., data=Train_1); summary(LM_6); plot(LM_6); q=quantile(abs(resid(LM_6)),0.5)
TT <- Train_1; TT$pre <- fitted(LM_6)
TT <- TT %>% mutate(price=ifelse(SalePrice>pre+q,"overprice",ifelse(SalePrice<pre-
q,"underprice","fairprice"))) %>%
   group_by(Neighborhood,price) %>% summarise(n=n())
N <- Train_1 %>% group_by(Neighborhood) %>% summarise(N=n())
TT1 <- left_join(TT,N,"Neighborhood") %>% mutate(per=round(n/N,4)) %>%
   mutate(pos = cumsum(per) - (0.5 * per)) %>% arrange(Neighborhood,per)
TT2 <- TT1 %>% filter(price=="fairprice") %>% arrange(per)
TT1$Neighborhood <- factor(TT1$Neighborhood, levels = TT2$Neighborhood)
ggplot(TT1,aes(y= per, x = Neighborhood, fill = price,label = paste0(per*100,"%"))) +
geom_bar(stat="identity",position="stack")+
    geom_text(position = position_stack(vjust = 0.5)) +    coord_flip()+theme_minimal()

#2. XGBoost
#tune parameter
cv_control = trainControl(method = "repeatedcv", number = 5L, repeats = 2L)
xgb_grid = expand.grid(nrounds = c(100,150),max_depth = c(20,25,30),eta = c(0.1,0.15),gamma = 0,
   colsample_bytree = 1.0,subsample = 1.0,min_child_weight = 10L)
xgb_grid1 <- xgb_grid[1:5,]; xgb_grid2 <- xgb_grid[6:10,]; set.seed(1)
# model = train(SalePrice ~ ., data = Train_1, method = "xgbTree",metric = "rmse",trControl = cv_control,
#                               tuneGrid = xgb_grid1,verbose = FALSE); model$results
# model1 = train(SalePrice ~ ., data = Train_1, method = "xgbTree",metric = "rmse",trControl = cv_control,
#                               tuneGrid = xgb_grid2,verbose = FALSE); model1$results
#eta=0.1, max_depth=30, nrounds=150,
set.seed(0)
xgb <- xgboost(data = data.matrix(Train_1),label = Train_1$SalePrice, eta = 0.1, max_depth = 30,nround=150,
                     subsample = 0.5,colsample_bytree = 0.5,seed = 1,eval_metric = "mae",nthread = 3)
#train-mae: 113.62
y_pred <- predict(xgb, data.matrix(Train_1)); dd <- as.data.frame(Train_1$SalePrice);
dd$nbr <- Train_1$Neighborhood; dd$prediction <- y_pred; names(dd) <-
c("SalePrice","Neighborhood","prediction")
res <- y_pred-Train_1$SalePrice; q1 <- quantile(abs(res),0.5)
dd <- dd %>%
   mutate(price=ifelse(SalePrice>prediction+q1,"overprice",ifelse(SalePrice<prediction-
q1,"underprice","fairprice"))) %>%
   group_by(Neighborhood,price) %>% summarise(n=n())
dd1 <- left_join(dd,N,"Neighborhood") %>% mutate(per=round(n/N,4)) %>% arrange(Neighborhood,per)
dd2 <- dd1 %>% filter(price=="fairprice") %>% arrange(per)
dd1$Neighborhood <- factor(dd1$Neighborhood, levels = dd2$Neighborhood)
ggplot(dd1,aes(y = per, x = Neighborhood, fill = price,label = paste0(per*100,"%"))) +
geom_bar(stat="identity",position="stack")+
  geom_text(position = position_stack(vjust = 0.5)) +coord_flip()+theme_minimal()
```