Less Money for Better Home

Group Member Bingying Feng

Group Member Dingzhe Leng

Group Member He Zhang

April 26th, 2019

Summary

Residential homes are hedonic pricing goods, whose prices are determined both by internal characteristics of the good being sold and external factors affecting it. Suppose we control the external factors, what internal features are strongly related to housing price? How good can we predict the housing price using the home features? These are the three questions that drive this project. The answers will be useful for

1. Homebuyers: For them to choose their dream house given a budget.

2. Real estate developers => choose the product to deliver

3. Real estate agents => provide valuable insights and identify the needs of customers

Data

The original data is retrieved from a Kaggle open competition. (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data) It covers data of Housing data that describes features of residential homes in Ames, Iowa sold between 2006 and 2010. There are 80 explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) describing the homes in different aspects such as space, amenities, conditions, zoning, etc (almost every aspect of residential homes).

Method

In this project, we will build models to explain what features are related to home prices as well as models to predict home prices in Ames, Iowa.We will first conduct data cleaning, EDA and feature analysis. Then we will build Elastic net model to investigate the features that are related to home prices using methods. For the predictive models, we will incorporate more variable engineering to enhance model performance. Modern machine learning models such as Elastic net, Random Forest, Gradient Boosting and Neural Net Models will be employed in predictive modeling.

Data Analysis

EDA (Part I before data processing)

Overlook

The response variable is `SalePrice`. There are 80 explanatory variables.

We created a histogram base of SalePrice (Appendix: Histogram of Property Sales Price) and find the property sales price is right-skewed. Most properties are below $200k.Given its distribution, we will later take the log of it to increase variability and approximate normality.

Data processing

Missing value

There are 34 variables with missing values. I dealt with these columns based on the data description.

We find some missing values means no such feature present such as missing value is PoolQC means no pool, missing value in MiscFeature means no such feature. So we imputed missing value in PoolQC, MiscFeature, Alley, Fence, FireplaceQu, GarageType, GarageFinish, GarageQual, GarageCond, BsmtCond, BsmtExposure, BsmtQual, BsmtFinType2, BsmtFinType1, MasVnrType and MasVnrArea as “None” or 0 meaning no such feature.

Some missing value could be imputed by geographical interpolation- taking the median value of the neighborhood such as The LotFrontage (Appendix: Box plot- lot frontage grouped by neighborhood).

Some missing value can be replaced by other values, which is reasonable to an extent. `GarageYrBlt` is `NA` if there is no garage with this property. As `YearRemodAdd` equals to `YearBuilt` if there is no remodeling, we replace the `NA`s in `GarageYrBlt` with the `YearBuilt` of that property.

After taking care of missing values in the previous variables, we now have only 17 variables with missing values. And these variables have only 1 or 2 missing values.

Now we impute the rest of the missing values.

`MSZoning` There are 4 missing values. As `RL` (Residential Low density) is the most common in this data, we assign `RL` to the missing values.

`Utilities` Only 1 property does not have all public utilities, so this is a variable of no use. We drop it.

`BsmtHalfBath` 0 is the most common value, so we assign 0 to the two missing values.

`Exterior1st`, `Exterior2nd`. one observation has missing values in both cells. We assign the most common value `VinylSd`, `VinylSd` to them.

`Electrical`,`KitchenQual`,`SaleType`. The most common values are `SBrkr`, `TA`, `WD`respectively. We assign these values to the missing value.

`BsmtFullBath`We assign 0 to the missing value. We assign 0 to all other basement variables with NA as well.

`Functional` The predominant value is `Typ` typical functionality, so we assign it to the missing values.

`GarageCars`,`GarageArea` We assign 0 to the missing values.

So far, all missing values have been imputed.

Variable structures

We now need to make sure that all categorical and numeric variables have the correct structures. We turned MSSubClass, MoSold, YrSold into factors. OverallQual and OverallCond can be wither factor or numeric, as they are ordinal. Here we keep them as numeric for now.

Months & years

The gragh in Appendix: “Box plot- sales price by month sold”" visualized how sales price varies across months. We see Janurary and April have relatively low median of sale price. July has lots of outliers. Janurary and July have the highest outlier values. Overall, the sale price does not vary much during month

The graph in Appendix:“Box plot- sales price by Year sold” shows how sales price varies across years. From the diagram, we see the price does not vary much during years. 2007 has a couple of high outlier values.

MSZoning Analysis

From the graph and table below in Appendix:“Distribution of MSZoning”, it is obvious that most of houses in this dataset are built in the area of Residential Low Density, and follows by Residential Medium Density(460 houses). Few houes are built in Commercial, Floating Village and Residential High Density.Since a large amount of houses comes to the categoreis of Residential Low Density and Residential Medium Density, these two areas should be paid more attention for housing price analysis.

From the boxplot in Appendix:“Box plot- sales price by MSZoning”, we can tell that sale price in Residential Low Density zone has a lot of outliers, and the range is wide too. Commercial zone has relatively low sale price

SalePrice vs Numerical Values

We then visualized the relationship of saleprice between 4 numerical values:

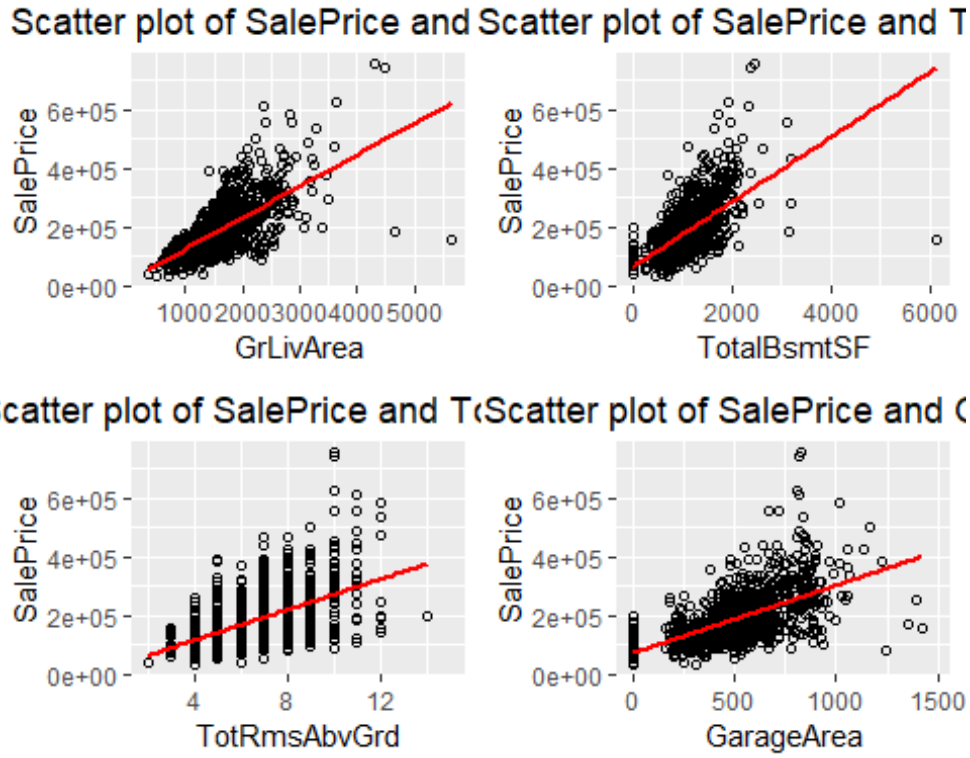
GrLivArea (Above grade (ground) living area square feet),

TotalBsmtSF (Total square feet of basement area),

TotRmsAbvGrd (Total rooms above grade (does not include bathrooms)),

GarageArea (Size of garage in square feet).

All 4 variables have positive relationship with the sale price. GrLivArea and TotalBsmtSF have larger positive relationship then TotRmsAbvGrd and GarageArea



Feature Engineering

In order to get a normalized dataset, we log the saleprice and deleted the original sale price column. The LogSalePrice is our response variable.

ID is not useful, but we still restore it for reporting the results. Then we take ID out of our data set.

We can also get the age of a house by taking `Yearsold-Yearbuilt`. We can also add 3 new features by the age of house. Isnew represents if a house is a new house. If the house age is 0, then mark 1 in Isnew to represent the house is a brand new one. If a house's age is not new, but the age of the house is less then 16 years, then we mark 1 in IsRecent to represent the house is recently built. If the house is more than 50 years old, then we mark 1 in IsOld to represent the house is old.

About the neighborhood, we not only want Iowa-specific results, but also generally interpretable results. So we are generating neighborhood feature data. A next step can be adding neighborhood data from census. (Appendix: Neighborhood)

Total square footage is important by intuitive. So we add a column named Totalsqft by adding GrLivArea and TotalBsmtSF.

The porch variables are not providing much variability. So we consolidate it by adding OpenPorchSF, EnclosedPorch, X3SsnPorch, ScreenPorch together. We deleted all the porch variable and only consider the consolidated one "PorchArea"

The bathroom numbers. Now only the number of full bath ranks No.19 in the important features. A number of total bathrooms could be more helpful. Intuitively, we count full bath as 1 and half bath as 0.5. By using the following equation, we get TotalBath= BsmtFullBath+ 0.5*BsmtHalfBath+ FullBath+ 0.5*HalfBath. we deleted all other bathroom variables and will only onsider the TotalBath.

Final preparation

We make two boxplot for Totalsqft and LogSalePrice to have an overview of the data. In the Totalsqft boxplot, there seems to be two very large houses, and one very small house. LogSalePrice looks fine. (Appendix: Final Preparation)

Since there are many missing value in the response variable (LogSalePrice), we filtered out useful data and get our final dataset. We have 1460 observations and 79 variables and 1 response variable.

We reserve the testing data. 70% of the dataset is seperated to be training set, and the rest 30% is testing set. Now we have 1021 observations in the training set and 439 observations in the testing set.

EDA (Part II after data processing)

Correlations

Both the correlation matrixs and heat map shows similar relationships. The top continuous variables correlated with Sale Price:

OverallQual 0.817184418

Totalsqft 0.773276841

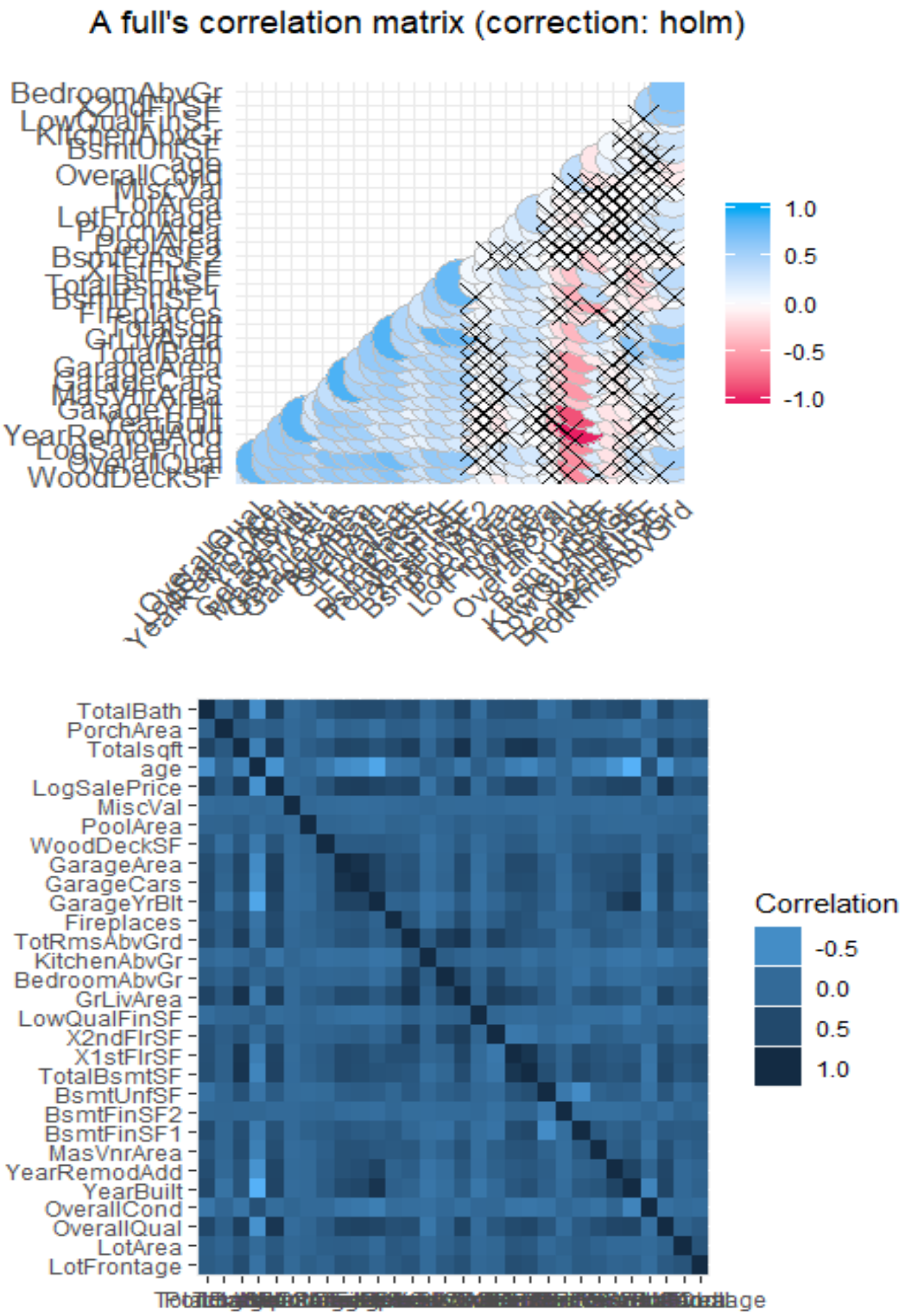
GrLivArea 0.700926653

GarageCars 0.680624807

TotalBath 0.673010594

GarageArea 0.650887556

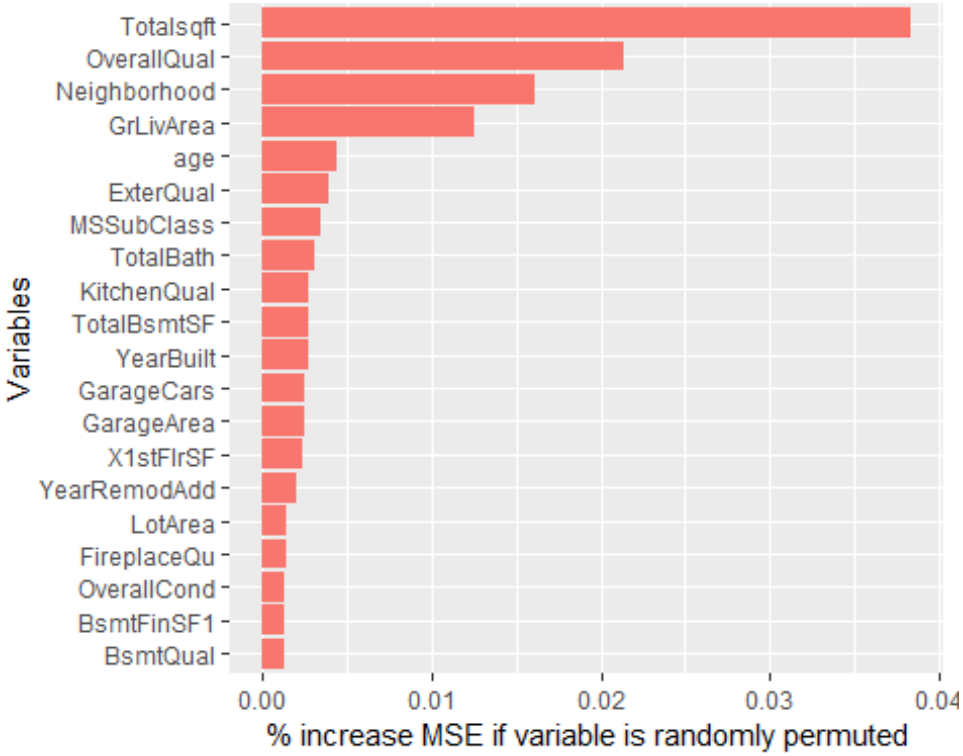
TotalBsmtSF 0.612133975



Models

Random Forest Modeling

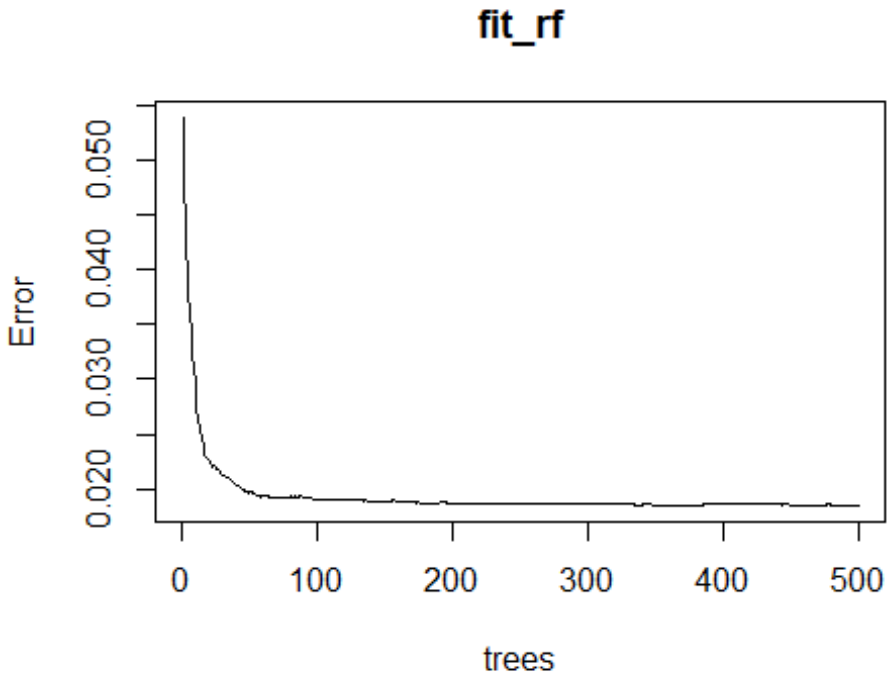
Before building the random forest model, we used the importance of random forest to see the important variables. This complements correlation analysis shows the top five important variables are `Totalsqft`, `OverallQual`, `GrLivArea`, `Neighborhood`,`age` which is similar to the elastic net model analysis.



We used the training data set to build our random forest model using randomForest() functiom. We tune ntree and mtry, the two parameters of random forest.From the error ntree plot, we may need at least 100 trees to settle the OOB testing errors, so 500 trees are enough here. Now we fix ntree=500, We only want to compare the OOB mse[500] to see the mtry effects. Here we loopmtry from 1 to 30 and return the testing OOB errors. (Appendix: ntree plot)

The recommended mtry for reg trees are mtry=p/3=76/3 about 25 or 26. We run a loop around this recommended value and found smallest OOB mse at mtry= 25. We take mtry=25. (Appendix: mtry plot)

The OOB error is 0.003064511 and the testing error is 0.0219367 Testing error is smaller than the training errir, but the testing error is also relatively small. So, this model predicts the testing dataset well. The following plot shows the error based on tree numbers.



Boosting tree

After trying different tuning parameter, we get n.trees = 20000, interaction.depth = 2, cv.folds = 5 for minimizing the training error.

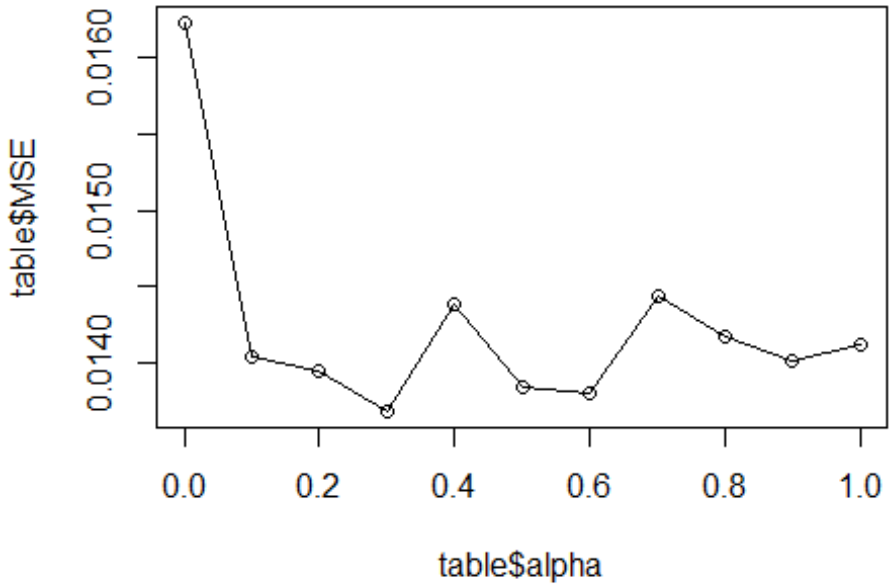
```
fit_boosting <- gbm(
  formula = LogSalePrice~.,
  distribution = "gaussian",
  data = data_train,
  n.trees = 20000,
  interaction.depth = 2,
  shrinkage = 0.001,
  cv.folds = 5,
  n.cores = NULL, # will use all cores by default
  verbose = FALSE
  )
pred.train <- predict(fit_boosting, n.trees = fit_boosting$n.trees, dat
a_train)
caret::RMSE(pred.train, data_train$LogSalePrice)
```

The boosting tree model gives us the cross-validation error to be 0.02045268, the training error to be 0.250696 and the testing error tobe 0.3419559 The training and testing errors are very similar, which indicates the model estimates the testing dataset well.

LASSO/Elastic Net

After modifying the data set, a elastic net was used to reduce dimensionality.

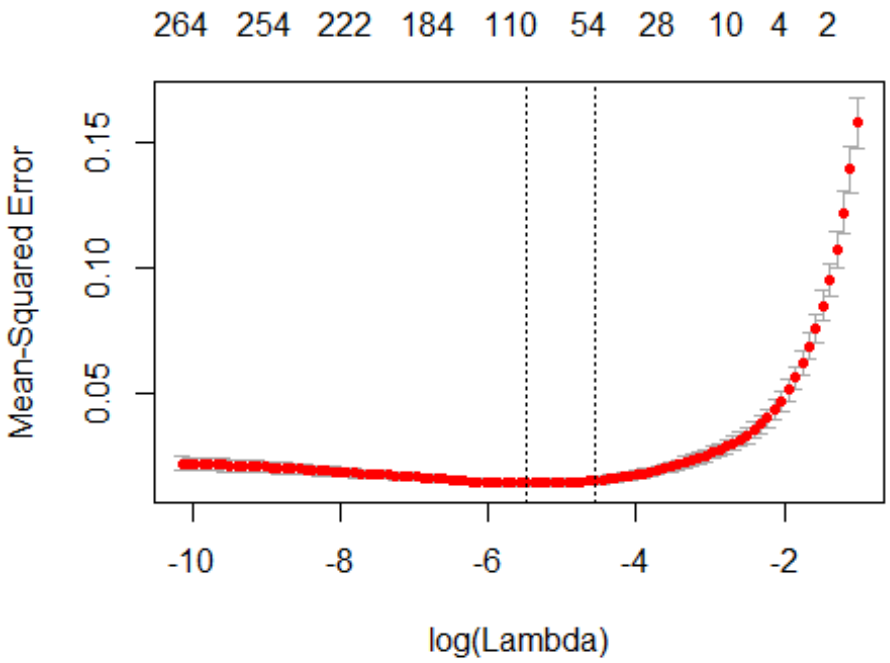
Ridge regression doesn't cut any variable, but it gives us unique solutions. LASSO estimation can give us a smaller model for the ease of interpretaion. Elastic Net combines Ridge Regression and LASSO, by choosing alpha between 0 and 1. so that it will do feature selection, yet still benefit from Ridge Regression. First, we use the following plot to choose the alpha. The plot describes how mse changes with alpha. However, the plot is random here because mse depends on the split of nfolds. We run it several times and find that mse is always low at alpha=0.9. What's more, there is no big differences of mse for different alpha, so we stick with it in the following



analysis.

We use cross validation to select the lambda. From this plot we show that as lambda increases, the impact of the shrinkage penalty grows, and the coefficient estimates will approach zero. To have a parsimonious model, we decide to use lambda.1se in our elastic net, which give us 54 variales in our final model. (We also try lambda.min

and discuss the results in the appendix.)



Here are the non-zero coefficients and variables. After cleaning the results and sorting different levels of the same categorical variables, the elastic net returns 41 variables. We conclude that there are six types pf main factors with most effect on home price.
- Area: lot size, shape, and configuration - Location: neighborhood locations, proximity to main road - Garage: size, age, quality, and area - Add-on features: street pave, material, basement, heating or AC, porch area, fireplace - Age: original construction date, type of dwelling - Zoning: the building class, the general zoning classification

As a house buyer, it's easy to notice the relationship between house price and house area, age and locatioin. Since cars are necessary to many families, the condition of garage is also taken into account when chooing houses. However, the majority people may not pay too much attention to add-on features and zoning. From our elastic net results, we can see that these add-on features are also determinants to the house price. It's not surprising that the quality of heating system, central air condition, and basement are key factors, but house buyer should know that they also pay for the porch area, fireplace, and street pave! Don't complain the narrow porch while enjoying the lower price of the house. Interestingly, the house in medium density has lower price. The possible reason is that the residental low density represents house and the residental high density represents luxury departments in the downtown, so the prices are both higher than the medium density.

```
##          (Intercept)         MSSubClass30        MSSubClass160
##         7.673182e+00        -1.755463e-02        -2.728172e-02
##           MSZoningRM          LotFrontage               LotArea
##        -3.954713e-02         1.743101e-04         1.401263e-06
##           StreetPave          LotShapeReg     LotConfigCulDSac
##         6.607452e-03        -3.820401e-03         8.566733e-03
##  NeighborhoodClearCr  NeighborhoodCrawfor  NeighborhoodEdwards
##         1.397941e-02         6.258993e-02        -4.490593e-03
##  NeighborhoodOldTown  NeighborhoodVeenker       Condition1Norm
##        -2.851915e-02         1.722838e-02         1.482421e-02
##        BldgTypeTwnhs          OverallQual          OverallCond
##        -1.862132e-02         7.093263e-02         3.021772e-02
##            YearBuilt         YearRemodAdd   Exterior1stBrkComm
##         2.978691e-04         1.154693e-03        -2.940607e-01
##   Exterior1stBrkFace      FoundationPConc       BsmtExposureGd
##         2.149185e-02         2.321765e-02         2.912716e-02
##       BsmtExposureNo           BsmtFinSF1          HeatingGrav
##        -5.478978e-04         6.013055e-05        -8.981592e-02
##          HeatingQCTA          CentralAirY             GrLivArea
##        -1.636682e-02         4.581362e-02         1.041790e-04
##          KitchenAbvGr        KitchenQualTA       FunctionalMaj2
##        -3.980158e-02        -4.065831e-03        -5.350507e-02
##        FunctionalMod        FunctionalSev        FunctionalTyp
##        -7.818873e-03        -9.988601e-02         2.109388e-02
##           Fireplaces        FireplaceQuGd      FireplaceQuNone
##         2.133120e-02         1.725584e-03        -7.359474e-03
##          GarageYrBlt           GarageCars           GarageArea
##         7.622285e-05         2.850357e-02         7.334998e-05
##         GarageQualTA           WoodDeckSF  SaleConditionNormal
##         9.896721e-03         3.303305e-05         1.974291e-02
## SaleConditionPartial                  age              Isnew1
##         7.246234e-02        -1.045379e-03         4.793261e-03
##              IsPoor1              IsRich1            Totalsqft
##        -6.886251e-02         4.192627e-02         1.209935e-04
##            PorchArea            TotalBath
##         1.076294e-04         2.725731e-02
```

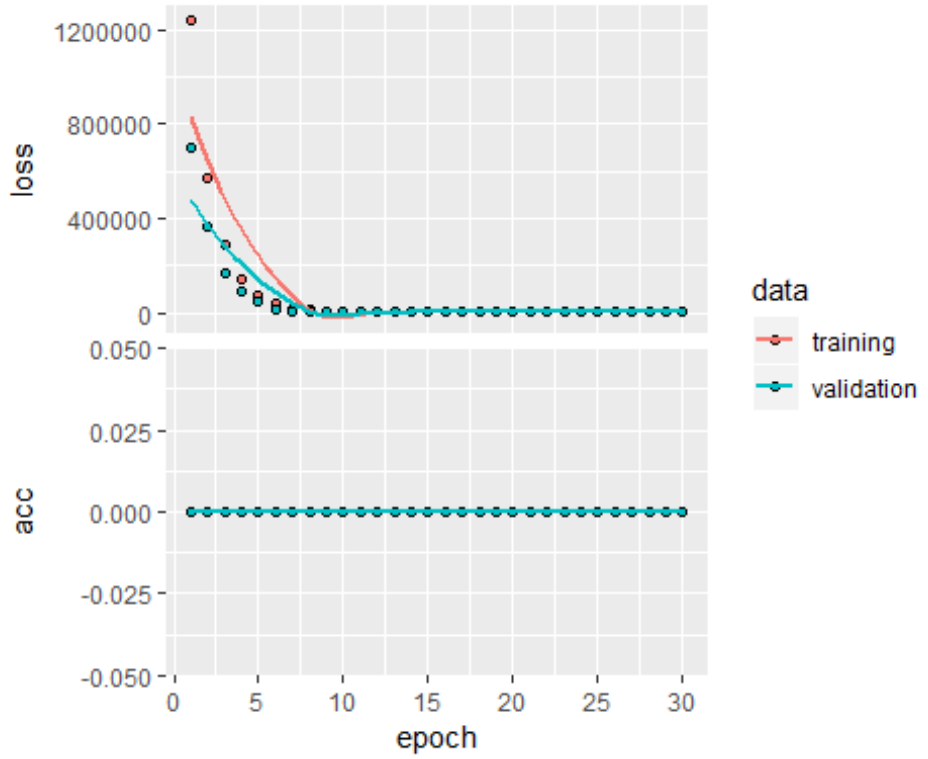
The training error of this elastic net is 0.01275193, while the testing error is 0.03670769.

The LASSO estimators are biased, so we use the same set of variables to refit our model using lm() function. The training error of this relaxed elastic net is 0.06878601, while the testing error is 0.01075683.

Neural Net

We also use neural net to address this problem. The neural net we learn in the class can deal with classification problems. We really want to if it can be used to solve problems with continues response variables. After searching for more information, we modify the original neural net. To be specific, we change the activation function

in the output layer to the “linear” rather than “sigmoid”, and also change the loss function to “mean_squared_error” instead of “categorical_crossentropy”. Then, we train this neural net and choose epoch=22 according to the plot.




```
## [1] "The training error of neural net is 544.451918761054"

## [1] "The testing error of neural net is 850.94697331322 Neural net g
ives us a reasonal result."
```

Model comparison

```
##                 model training_error testing_error
## 1         Elastic Net   1.303331e-02    0.03535397
## 2 Relaxed Elastic Net   1.108830e-02    0.06771490
## 3       Random Forest   3.064511e-03    0.02181422
## 4            Boosting   8.634368e-02    0.13944302
## 5          Neural Net   5.444519e+02  850.94697331
```

Overall, all the models give us the reasonal well results. From this table, we can see that elastic net and random forest give us relatively low training error and testing error. So both models can be our final models.

Conclusion

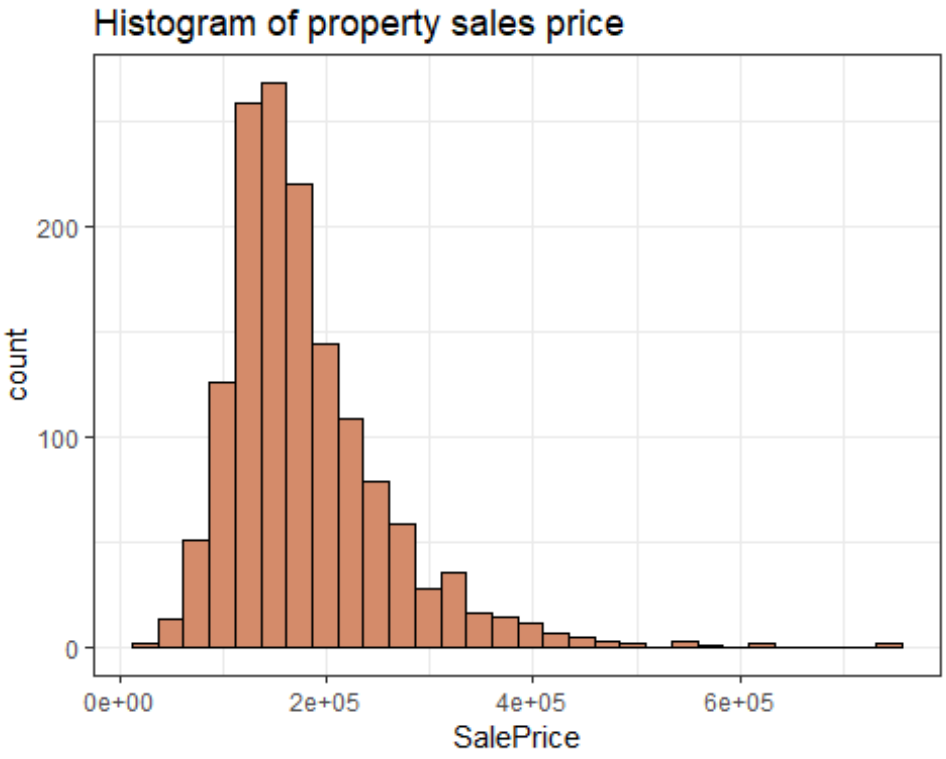
1. We use elastic net, random forest, boosting and neural net to predict the house price based on internal characteristics, like lot area, material, heating quality, and so on, and external factors, such as zoning, proximity to main road, and so on.

2. We find that the house price mainly depends on six types of factors: Area, Location, Garage, Add-on features, Age and Zoning. It's easy to notice the relationship between house price and house area, age, locatioin and garage. It's also not surprising that the quality of heating system, central air condition, and basement are key factors, but house buyer should know that they also pay for the porch area, fireplace, and street pave!
3. All of our models do pretty good job, which means we can predict the house price accurately based on physical characteristics. As we know, homes are hedonic pricing goods through intrinstic features which is all the information used to price the home, while controlling the exterior factors, like sales year fixed effect and local market. So we think the price well captures the information of the houses in this market. Few people in this area treat the house as the investment and hope to make profits by playing the market.
4. Overall, we believe our models work well to predict house price, and they should work better for a healthy housing market.

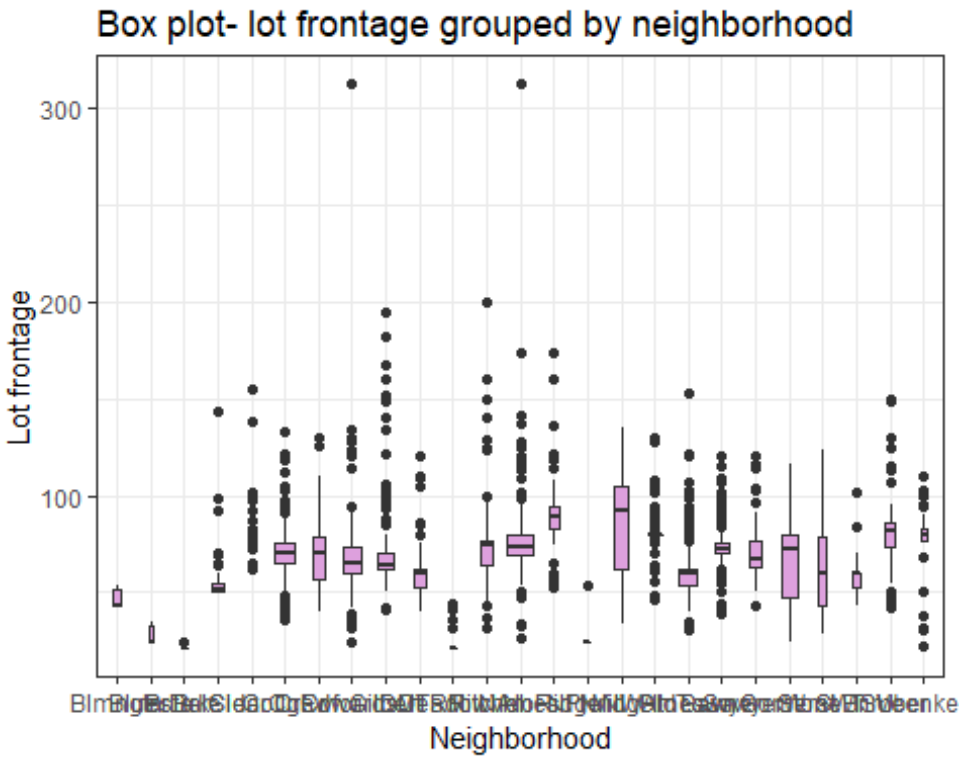
Appendix

Histogram of Property Sales Price

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

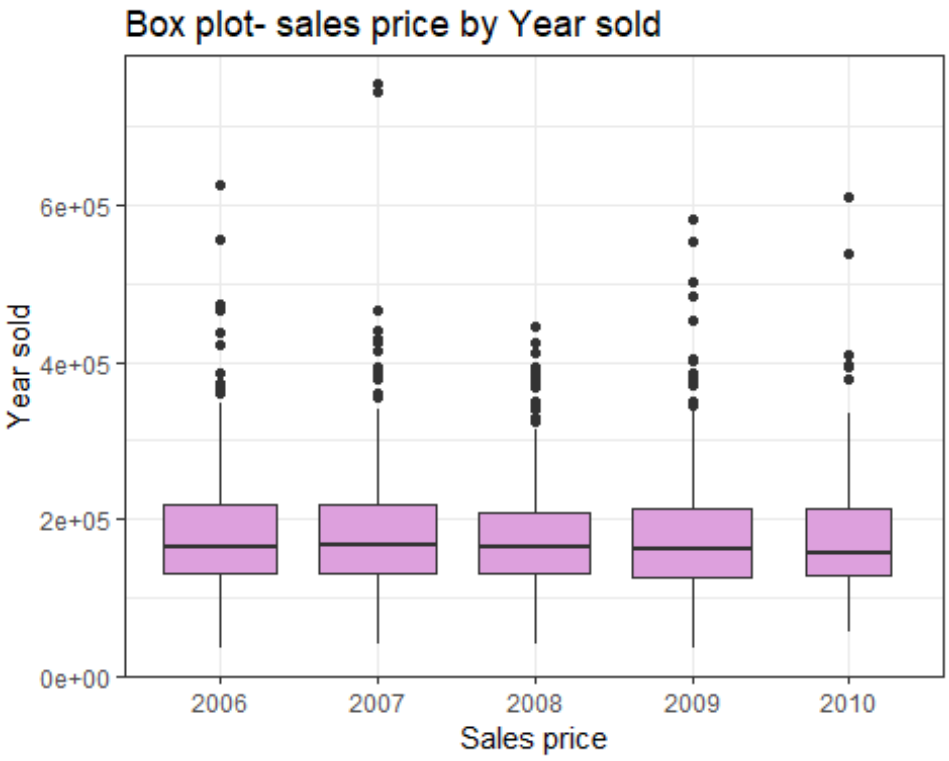


Box plot- lot frontage grouped by neighborhood

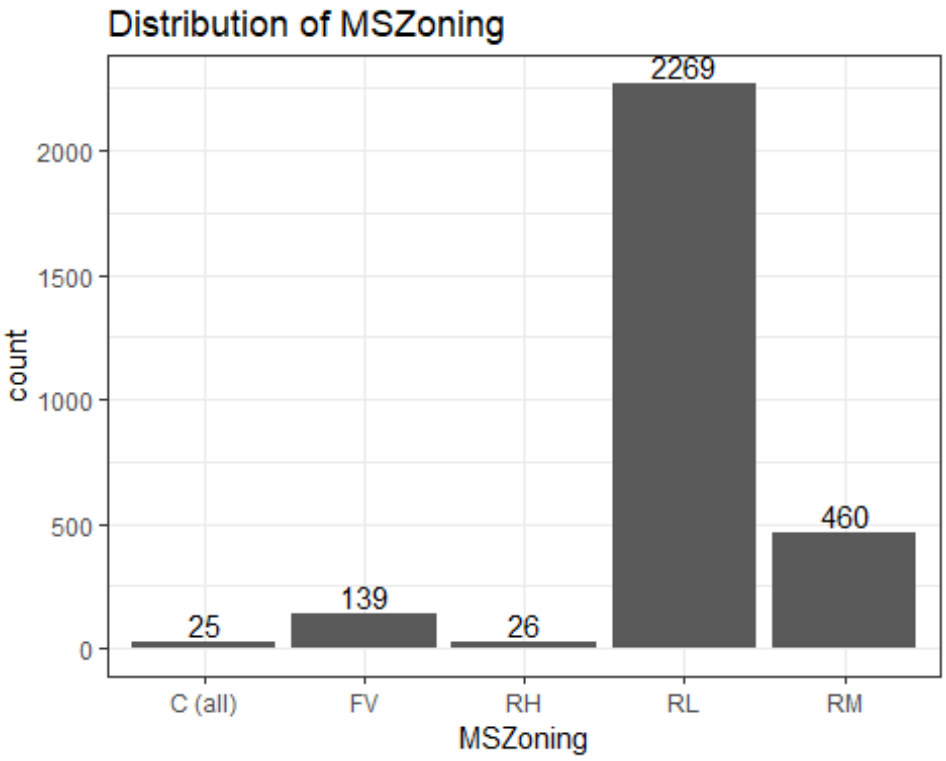


Box plot- sales price by month sold

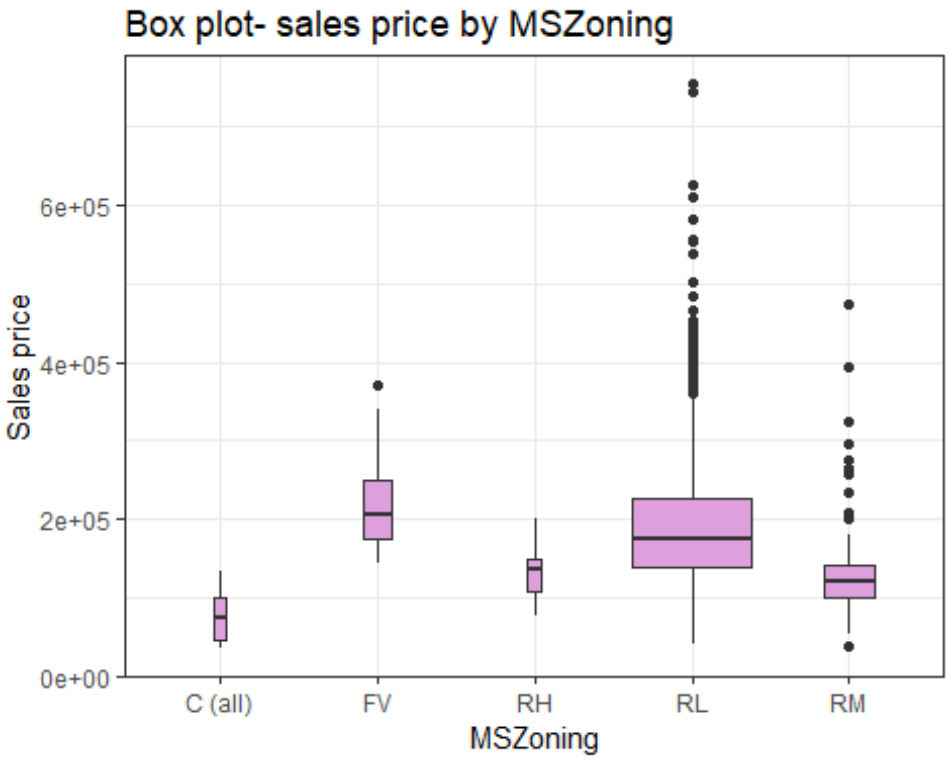
Box plot- sales price by Year sold



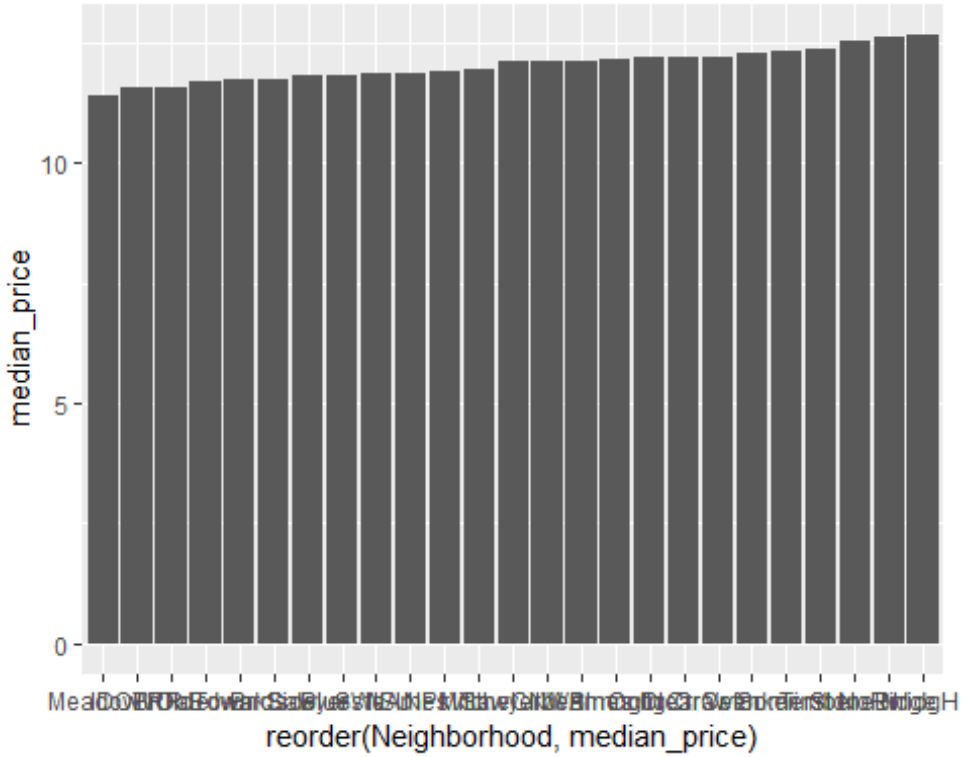
Distribution of MSZoning



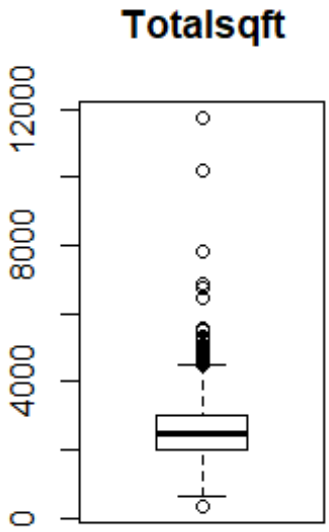
Box plot- sales price by MSZoning

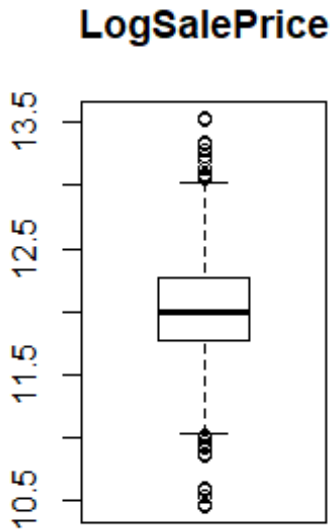


Neighborhood

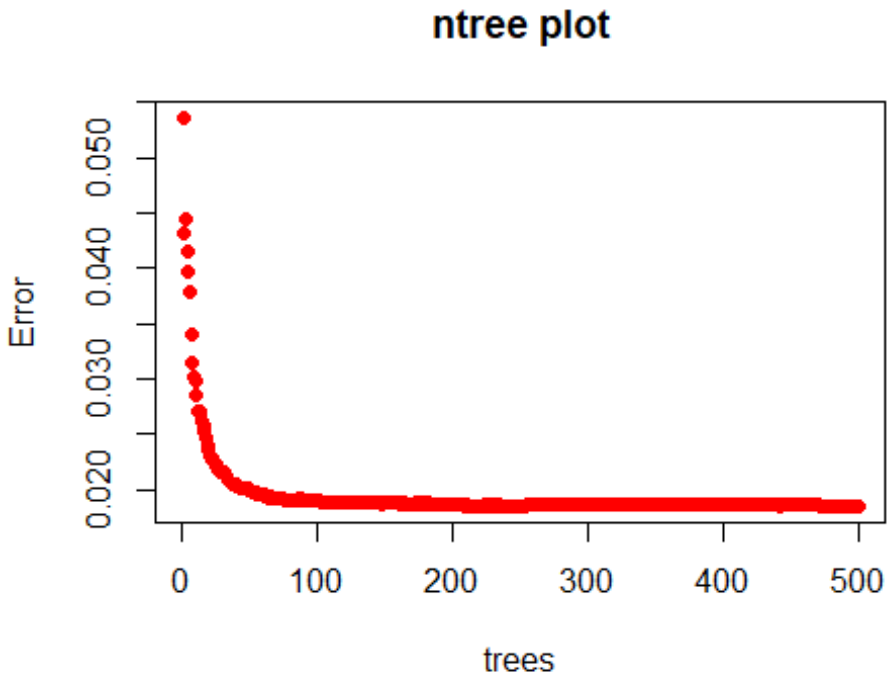


Final Preparation





ntree plot


ntree plot
Error
0.020
0.030
0.040
0.050
0
100
200
300
400
500
trees

mtry plot