UNIVERSITY OF CALIFORNIA

Los Angeles

Housing Sale Price Prediction Using Machine Learning Algorithms

A thesis submitted in partial satisfaction of the requirements for the degree

Master of Applied Statistics

by

Yichen Zhou

2020

ABSTRACT OF THE THESIS

Housing Sale Price Prediction Using Machine Learning Algorithms

by

Yichen Zhou

Master of Applied Statistics

University of California, Los Angeles, 2020

Professor Yingnian Wu, Chair

In this thesis, I explore how predictive modeling can be applied in housing sale price prediction by analyzing the housing dataset and use machine learning models. Actually, I try four different models, namely, linear regression, lasso regression, randomforest and xgboost. Additionally, as the data have 79 explanatory variables with many missing values, I spend much time dealing with the data. I do explorary data analysis, feature enginnering before model fitting. And then using rmse and R-squared to measure the model performance. After I try four different models, I get some results. As for the first model - linear regression, it doesn't meet the assumption of equality of the variances. Therefore, we can't use the linear model as the candidate of our final model. Then I try lasso regression, but the RMSE and R-squared looks not so good. Then I try Random forest. The R squared in this model of training set is very good, but in the test set the R squared is relatively low, which may show the RF model is a little bit overfitting. Finally I try the

fourth model - xgboost. All of the results of this xgboost model seem very good. Therefore, I will use this xgboost model as my final model to predict the housing price. The xgboost model also shows which variables have important effects on sale price.

The thesis of Yichen Zhou is approved.

Frederic R Paik Schoenberg

Nicolas Christou

Yingnian Wu, Committee Chair

University of California, Los Angeles

2020

Table of Contents

List of Figures

List of Tables

CHAPTER 1

1.Introduction

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence [1]. However, this dataset related to the thesis proves that may have more effects on the housing price than the number of bedrooms or floors. Also, I want to predict the reasonable housing price with these aspects of the houses by using this dataset.

This dataset contains 79 explanatory variables which related to almost every aspect of residential homes in Ames, Iowa. In the following steps, I will explore this dataset, do feature engineering, fit some machine learning models to predict the housing prices and find which aspects of the house influence the housing prices mostly.

Machine learning is closely related to computational statistics, which focus on using mathematical optimization to deliver methods, theory and application domains to solve medical, industry, social and business problems in the real world.

In my thesis, I will try four models: Linear Regression, Lasso Regression, RandomForest and Xgboost to predict the housing sale price. And finally, I will use the Xgboost model as my final model. This model also gives us which aspects have big effects on housing sale price.

CHAPTER 2

2.Explorary Data Analysis

2.1 The response variable – Saleprice

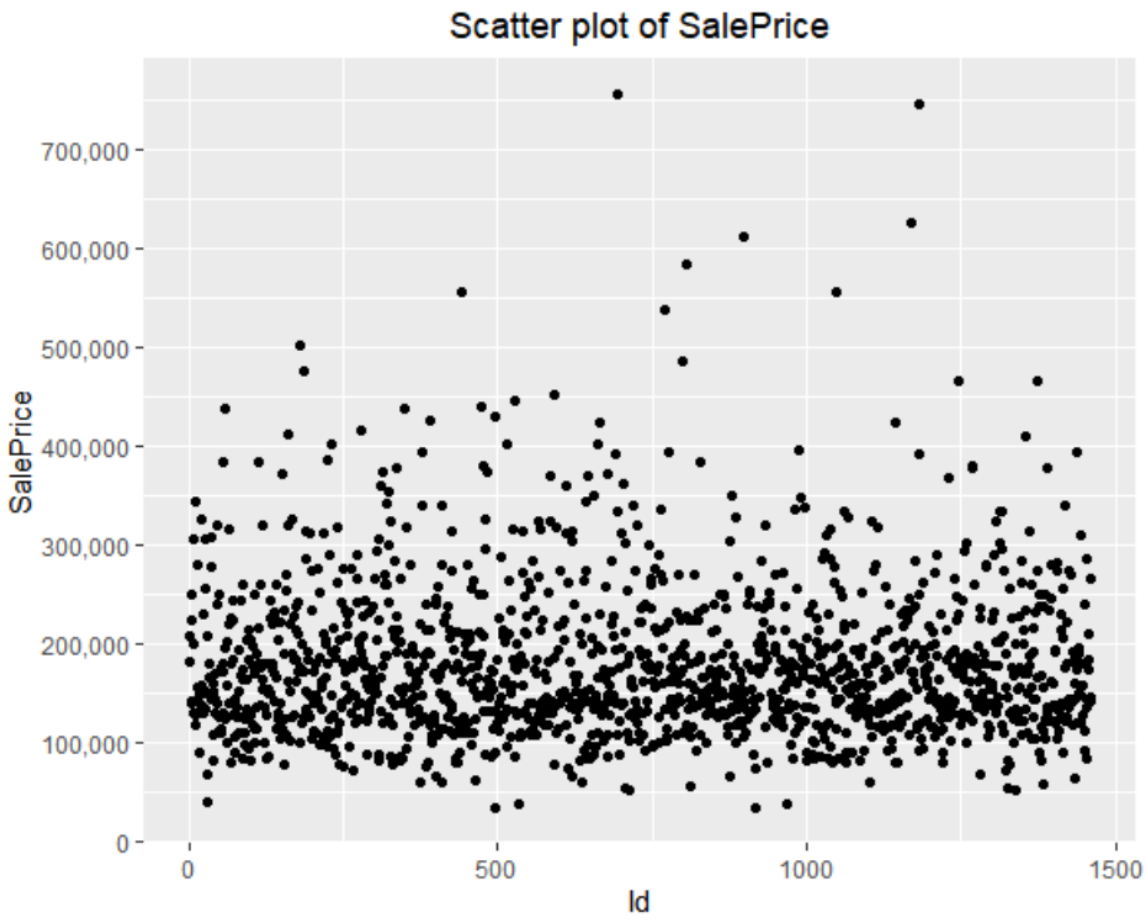


Figure 1

Figure 1 gives us the scatter plot of the sale price. Most of the points are assembled on the bottom. And there seems to be no large outliers in the sale price variable

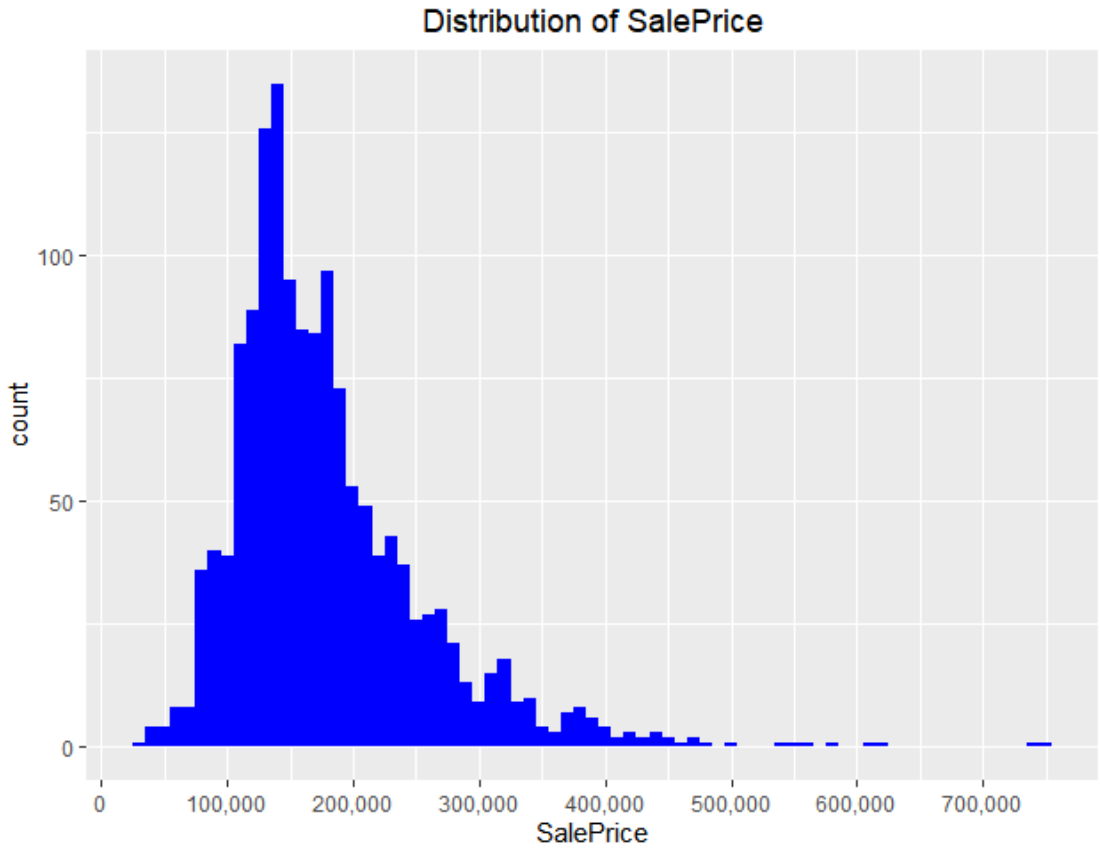


Figure 2

Figure 2 shows that the distribution of sale prices are right skewed, which shows the distribution of the sale prices isn't normal. It is reasonable because few people can afford very expensive houses. I need to take transformation to the sale prices variable before model fitting.

2.2 The most related numeric predictors

I decided to see which numeric variables have a high correlation with the Sale Price by making a correlation plot.

2.2.1 Correlations

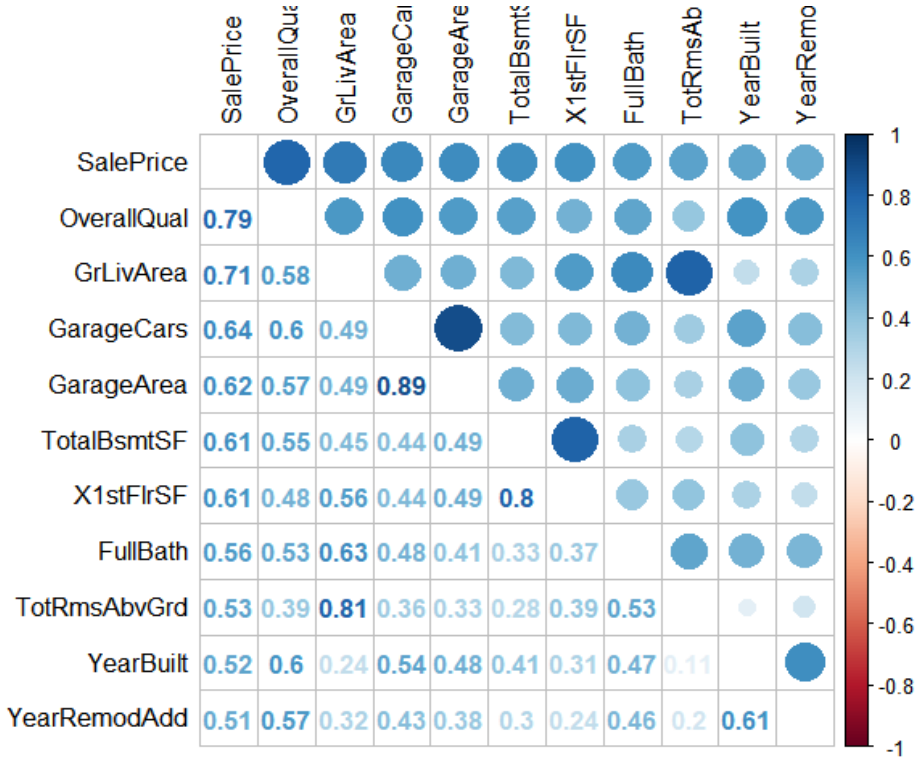


Figure 3

There are 10 numeric variables with correlations of at least 0.5 with Sale Price. All those correlations are positive.

I will visualize the relation between Sale Price and the two predictors with the highest correlation with Sale Price; Overall Quality and the Above Grade Living Area.

It also becomes clear the multicollinearity is an issue. For example: the correlation between GarageCars and GarageArea is very high (0.89), and both have similar (high) correlations with Sale Price. The other 6 six variables with a correlation higher than 0.5 with SalePrice are: -TotalBsmtSF: Total square feet of basement area -1stFlrSF: First Floor square feet -FullBath: Full bathrooms above grade -TotRmsAbvGrd: Total rooms above grade (does not include

bathrooms) -YearBuilt: Original construction date -YearRemodAdd: Remodel date (same as construction date if no remodeling or additions).

2.2.2 Overall Quality

We find that the highest correlation 0.79 which is between the overall quality and sale price. This overall quality variable rates the overall material and finish of the house as follows:

OverallQual: Rates the overall material and finish of the house

10	Very Excellent
9	Excellent
8	Very Good
7	Good
6	Above Average
5	Average
4	Below Average
3	Fair
2	Poor
1	Very Poor

Figure 4 shows the counts of the Overall Quality variable, it is very hard to judge whether the distribution of this variable is normal, so we need to check its skew.

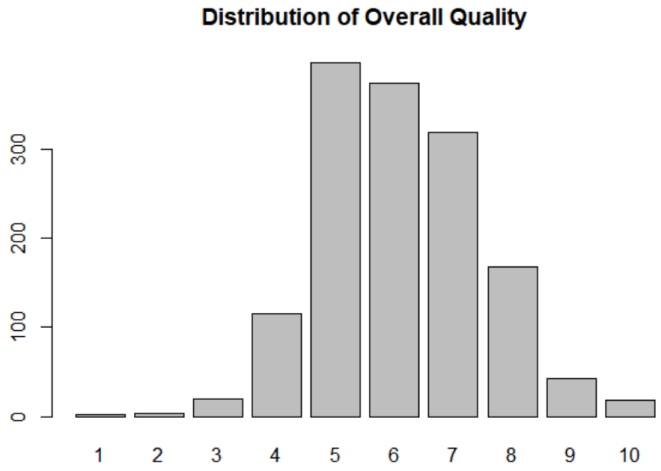


Figure 4

The skew of overall quality is 0.21 which is very low. Therefore, the distribution of the overall quality could be regarded as normal.

Figure 5 shows the relationship between the overall quality and the sale price.

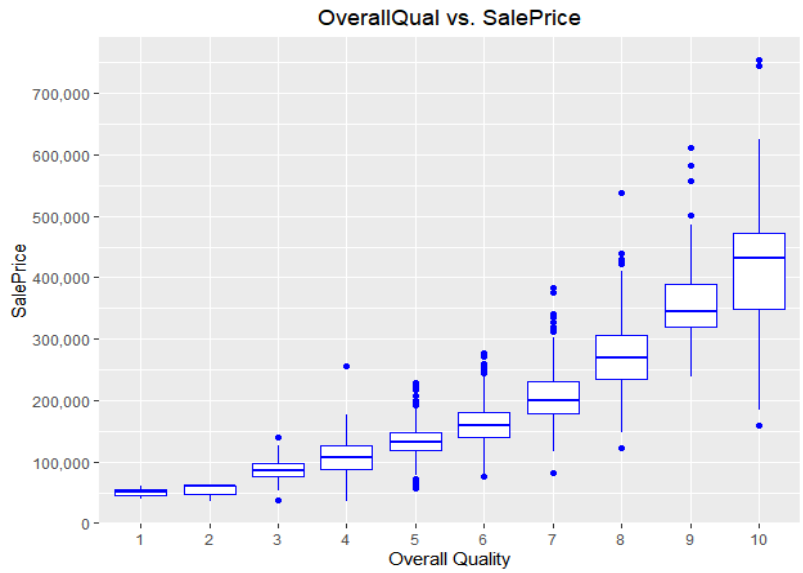


Figure 5

We find that there is a positive relationship between the Overall Quality and Sale Price. And it seems like a quadratic relationship or something else like that rather than the linear relationship. This relationship seems easy to be understood. If a house keeper want to improve the overall quality of his house from very poor to poor, he will only need to spend a little money and buy a few items. However, if the house keeper want to improve the overall quality of his house from excellent to very excellent, it will be very difficult and costs he much money.

2.2.3 Above Grade (Ground) Living Area (square feet)

The correlation between this numeric variable and sale price is 0.71 which the second highest. We can give interpretations to the high correlations. Large above grade (ground) living area means large house, and large house means expensive sale price. This makes sense a lot.

The counts of Above Grade (Ground) Living Area which is shown as follows:

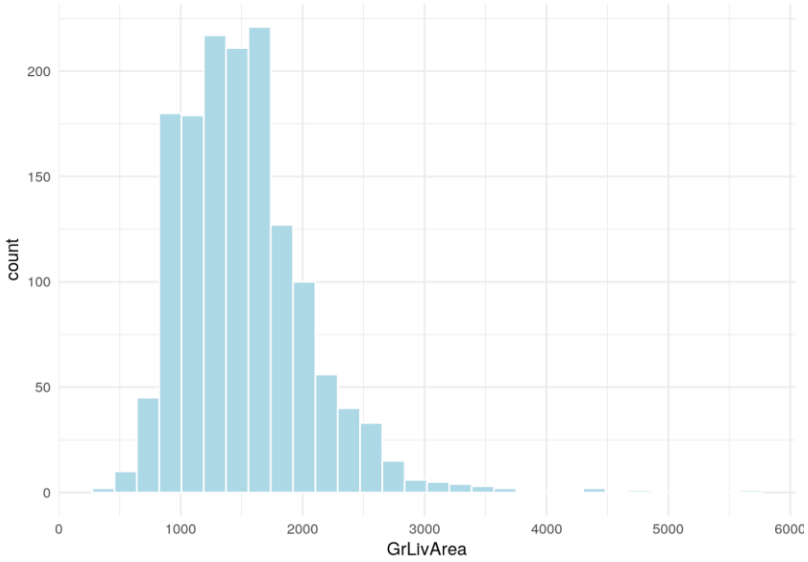


Figure 6

Most houses have low Above Grade (Ground) Living Area, and only a few houses have very high Above Grade (Ground) Living Area. The distribution of this variable isn't normal. I need to take transformation to the sale prices variable before model fitting.

I also need to see the relationship between this variable and Sale price.

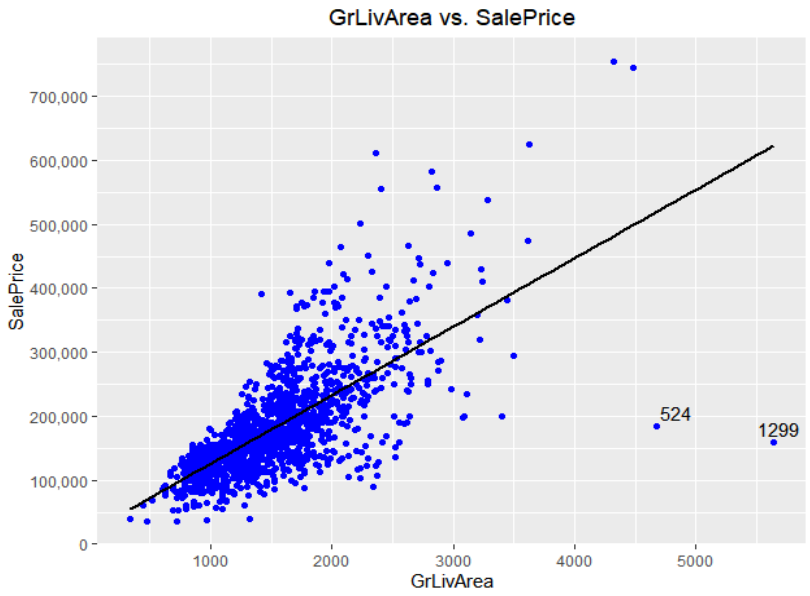


Figure 7

There are two obvious outliers with high above grade living area but low sale price. Actually, I will not easily delete these two outliers. Because there may be some reasons accounting for the low sale price. I just analysis the overall quality variable, therefore I think they may have low overall quality.

Table 1

Id	*SalePrice*	*OverallQual*	*GrlivArea*
524	184750	10	4676
1299	160000	10	5642

However, from table 1, we can see the two houses also have high overall quality. There may be some other reasons accounting for their low prices, but I will keep houses 1299 and 524 in mind as prime candidates to take out as outliers.

2.3 Missing Values

Table 2 shows which variables contain missing values.

Table 2

PoolQc	**MiscFeature**	**Alley**	**Fence**	**SalePrice**
2909	2814	2721	2348	1459
FireplaceQu	**LotFrontage**	**GarageYrBlt**	**GarageFinish**	**GarageQual**
1420	486	159	159	159
GarageCond	**GarageType**	**BsmtCond**	**BsmtExposure**	**BsmtQual**
159	157	82	82	81
BsmtFinType2	**BsmtFinType1**	**MasVnrType**	**MasVnrArea**	**MSZoning**
80	79	24	23	4
Utilities	**BsmtFullBath**	**BsmtHalfBath**	**Functional**	**Exterior1st**
2	2	2	2	1
Exterior2nd	**BsmtFinSF1**	**BsmtFinSF2**	**BsmtUnfSF**	**TotalBsmtSF**
1	1	1	1	1
Electrical	**KitchenQual**	**GarageCars**	**GarageArea**	**SaleType**
1	1	1	1	1

We can see that there are 34 predictor variables containing missing values. And I need to fix these NAs in these variables.

2.4 Inputing missing data

2.4.1 Pool Quality

The pool quality variable has the most missing values. So, we need to analyze this variable and fix these missing values.

The description of the Pool Quality will help us fix the NAs.

PoolQC: Pool quality

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
NA	No Pool

NA means no pool. Therefore, we can replace the NA by 0 which means no pool. Also, we will replace other levels by ordinal numbers as follows:

Table 3

ORIGINAL LABEL	NEW LABEL
NONE	0
FA	1
TA	2
GD	3
EX	4

2.4.2 Miscellaneous feature

This variable has the second largest number of NA in this dataset. The description of the Miscellaneous feature variable is shown as follows:

MiscFeature: Miscellaneous feature not covered in other categories

Elev Elevator
Gar2 2nd Garage (if not described in garage section)
Othr Other
Shed Shed (over 100 SF)
TenC Tennis Court
NA None

The levels of this variable are not ordinal. Therefore, I choose to convert this variable to factor.

Figure 8 shows the relationship between this variable and sale price

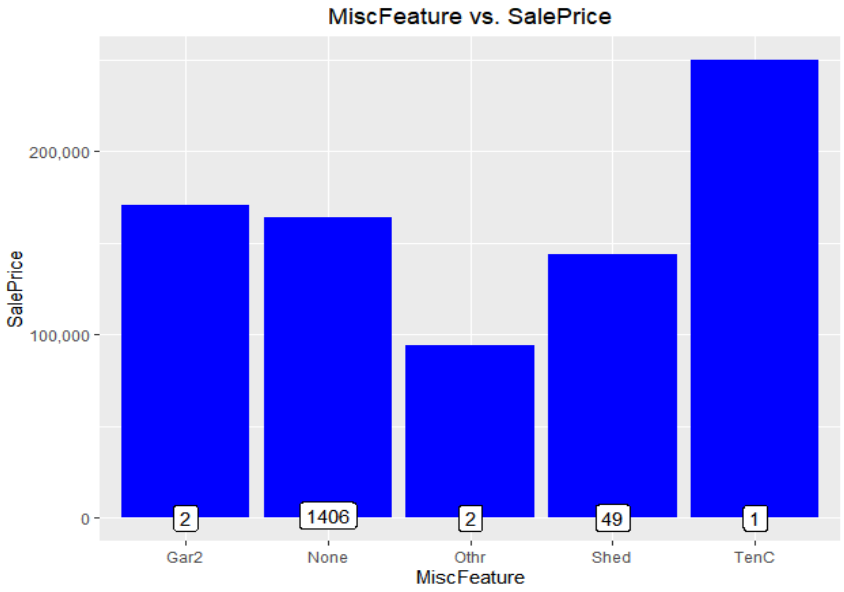


Figure 8

I think this variable may not have much influence on the sale price. It sounds reasonable that the sale price of a house with a tennis court is high. And there is only one house with a tennis court in this dataset.

2.5 Dealing with Character Varibales

After I fix the missing values in some variables, I still need to be care of the rest character variables without NAs.

Table 4

Street	LandContour	LandSlope
Condition1	Condition2	BldgType
HouseStyle	RoofStyle	RoofMatl
Foundation	Heating	HeatingQC
CentralAir	PavedDrive	Neighborhood

All in all, there are 15 remaining character variables. I want to convert them into factors or ordinal numbers. I will give table 5 to show the conclusions rather than too many details.

Table 5

Variables	***Original Type***	***New Type***
Foundation	Character	Factor
Heating	Character	Factor
HeatingQc	Character	Number
RoofStyle	Character	Factor
RoofMatl	Character	Factor
LandContour	Character	Factor
LandSlope	Character	Number
BldgType	Character	Factor
HouseStyle	Character	Factor
Neighborhood	Character	Factor
Condition1	Character	Factor
Condtion2	Character	Factor
Street	Character	Number
PavedDrive	Character	Number

2.6 Changing some numeric variables into factors

After I fix the missing values and do label encoding, all of the character variables are converted into factors or numeric labels.

Nevertheless, I find that there are some numeric variables that should be categorical variables.

2.6.1 Year and Month sold

The year sold variable only contains 5 years. Therefore, I think it may be reasonable for me to convert it to factors.

The month sold variable is originally a numeric variable. However, the month is actually categorical. Therefore, I need to convert this month sold variable into factors.

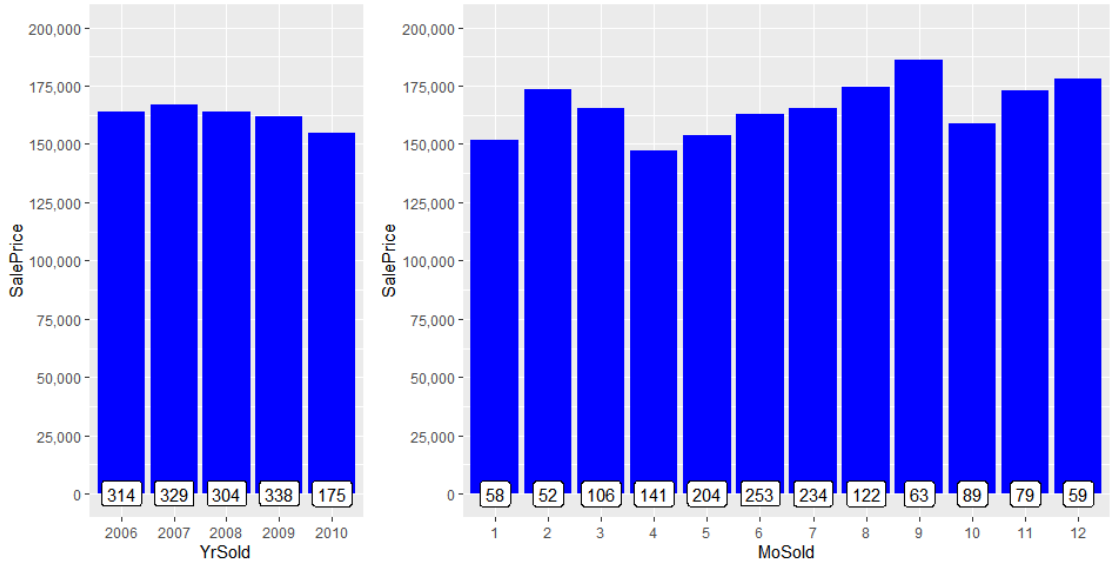


Figure 9

We can find some facts from the figure 9.

The right plot is about the sale price and the month sold. We can see there is seasonality related to the month. This seasonality seems to have obvious effects on the housing price.

The left plot is about the sale price and the year sold. From this plot, we can see the effects of the Financial Crisis happening 2008. The median housing sale price came to the highest in 2007 and then gradually decreased from 2008.

The financial crisis of 2008, also known as the global financial crisis and the 2008 financial crisis, was a severe worldwide economic crisis considered by many economists to have been the

most serious financial crisis since the Great Depression of the 1930s, to which it is often compared. [2]

2.6.2 MSSubClass

The description of this MSSubClass variable is shown as follows:

MSSubClass: Identifies the type of dwelling involved in the sale.

20	1-STORY 1946 & NEWER ALL STYLES
30	1-STORY 1945 & OLDER
40	1-STORY W/FINISHED ATTIC ALL AGES
45	1-1/2 STORY - UNFINISHED ALL AGES
50	1-1/2 STORY FINISHED ALL AGES
60	2-STORY 1946 & NEWER
70	2-STORY 1945 & OLDER
75	2-1/2 STORY ALL AGES
80	SPLIT OR MULTI-LEVEL
85	SPLIT FOYER
90	DUPLEX - ALL STYLES AND AGES
120	1-STORY PUD (Planned Unit Development) - 1946 & NEWER
150	1-1/2 STORY PUD - ALL AGES
160	2-STORY PUD - 1946 & NEWER
180	PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
190	2 FAMILY CONVERSION - ALL STYLES AND AGES

From the description, we can find that the MSSubClass variable actually identifies the type of dwelling of the houses. However, it is coded as numeric. Therefore, I need to convert it to factors.

2.7 Importance of variables

After I explore some variables, convert some character variables into factors or numbers and convert 3 numeric variables into factors, now I find that there are 56 numeric variables and 23 factor variables in this new dataset.

2.7.1 Correlations between variables (once again)

I plan to check the correlations again and to whether they have changed after I deal with so many variables.

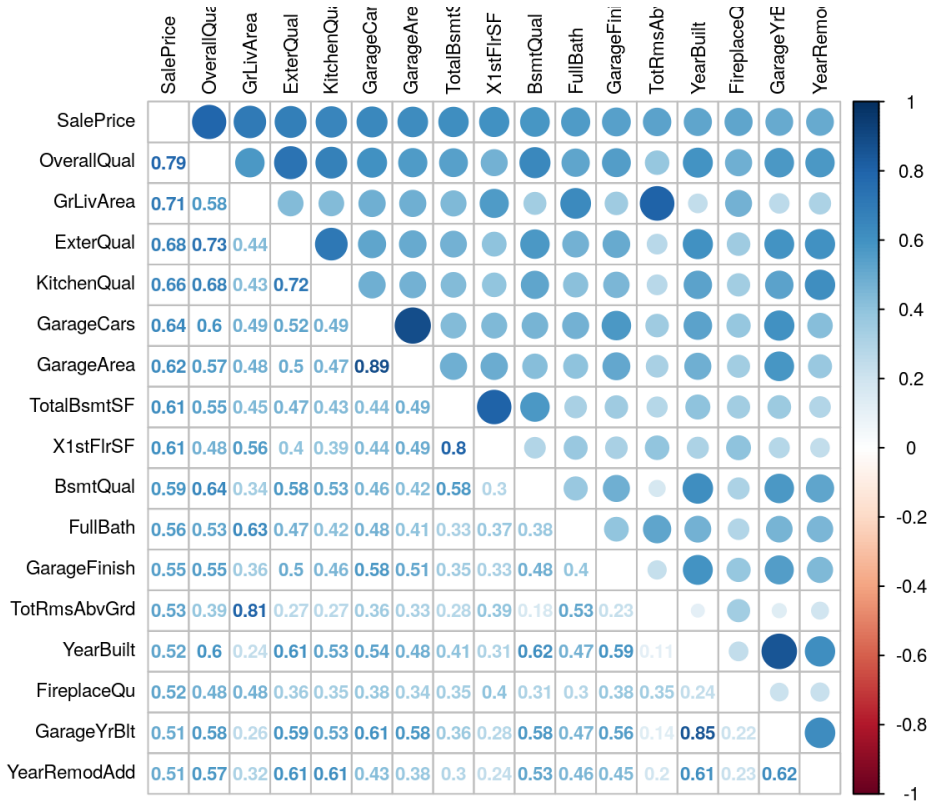


Figure 10

Comparing the correlations in this section to the correlations in section 2.2.1, We can see that the number of numeric variables having at least 0.5 correlations with the SalePrice has increased from 10 to 16, which may shows that my data analysis is reasonable.

2.7.2 Get Importance of variables

From the correlations, we can get an overview of some important numeric variables such as the Overall quality and Above Grade (Ground) Living Area. Also, we can find some variables such as the Garage Car and the Garage Area have multicollinearity.

However, I want to get more details about the importance of variables which including the factor variables. Therefore, I conduct a simple and quick random forest with only 100 trees and see the most 20 important variables.

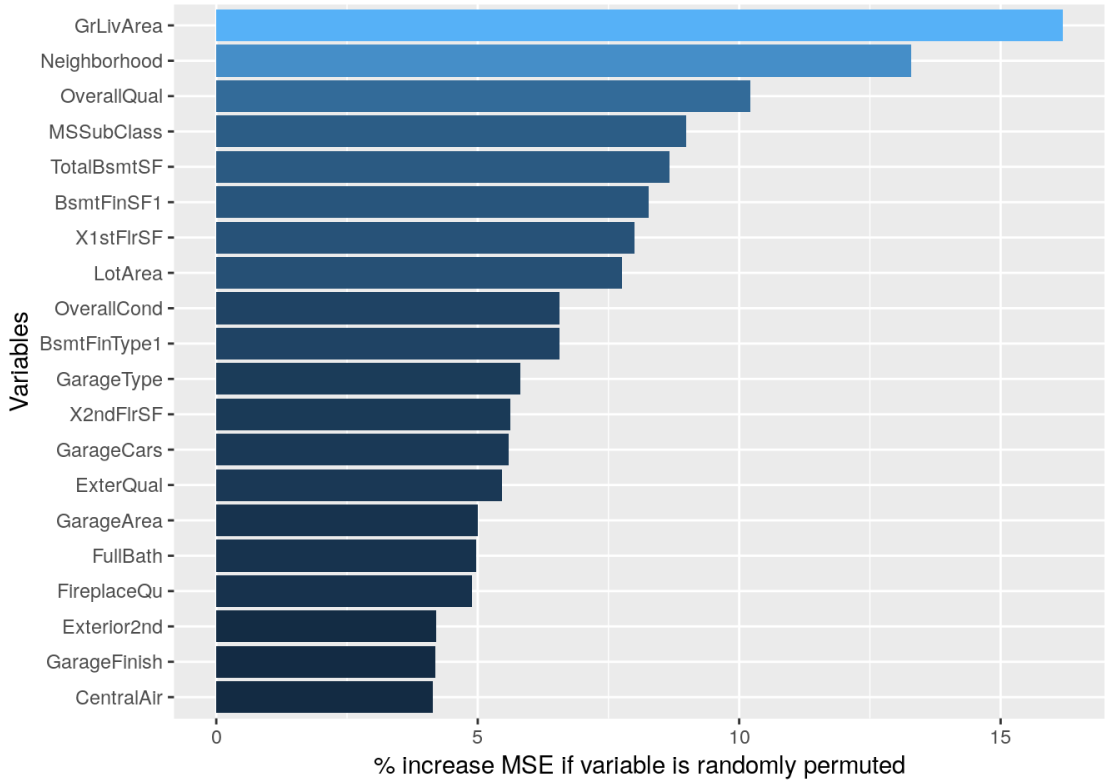


Figure 11

In the most 20 important variables, only 3 are factors. Neighborhood, MSSubClass, and GarageType.

2.8 Feature Engineering

2.8.1 Bathrooms Variables

There all totally 4 bathrooms variables - FullBath, HalfBath, BsmtFullBath and BsmtHalfBath. As far as I am concerned, there is no need to have so many variables about bathrooms. Also, the importance of them are not very high. Therefore, I want to combine them into one variable by adding them all so that this predictor is likely to become more correlated with sale price.

The descriptions of these four variables are shown as follows:

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

However, I need to know what is half bathroom.

“A half-bath, also known as a powder room or guest bath, has only two of the four main bathroom components-typically a toilet and sink.” [3]

Therefore, I plan to count the half bathroom as half. I will create a new variable about bathrooms as follows:

$$TotalBathrooms = FullBath + 0.5HalfBath$$

$$+BsmtFullBath + 0.5BsmtHalfBath$$

Figure 12 help me visualize the new variable - totalbathrooms.

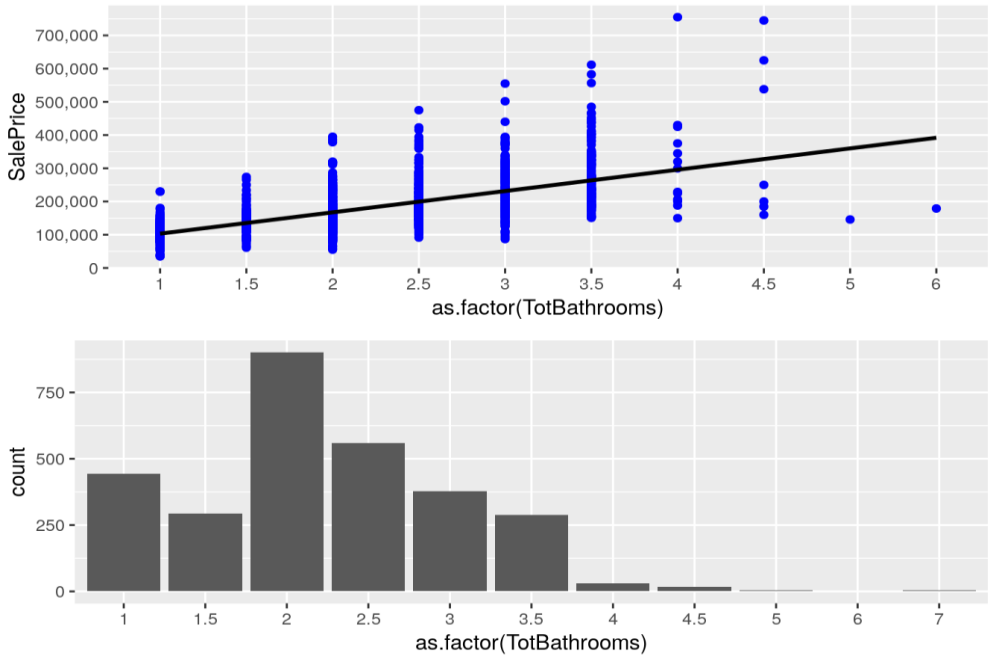


Figure 12

The first plot shows the relationship between the median housing sale price and the TotalBathrooms. From this plot we can find that with the number of total bathrooms increasing, the sale price tends to increase, too.

The second plot gives us the counts houses in each TotalBathrooms in this dataset. From this plot, we can find that most houses have 2 TotalBathrooms.

Also, the correlation between totalbathrooms and sale price is 0.63, which seems very suitable for me to analysis this variable.

2.8.2 Total Square Feet

When people want to buy a house, the total square feet of this house is an very important factor. This dataset contains two variables about the total square feet - GrLivArea and TotalBsmtSF.

GrLivArea means above grade (ground) living area. And TotalBsmtSF means total square feet of basement area.

Therefore, I want to create a new variable - Total Square Feet by just adding them.

$$TotalSqFeet = GrLivArea + TotalBsmtSF$$

Figure 13 shows the relationship between this total square feet and sale price

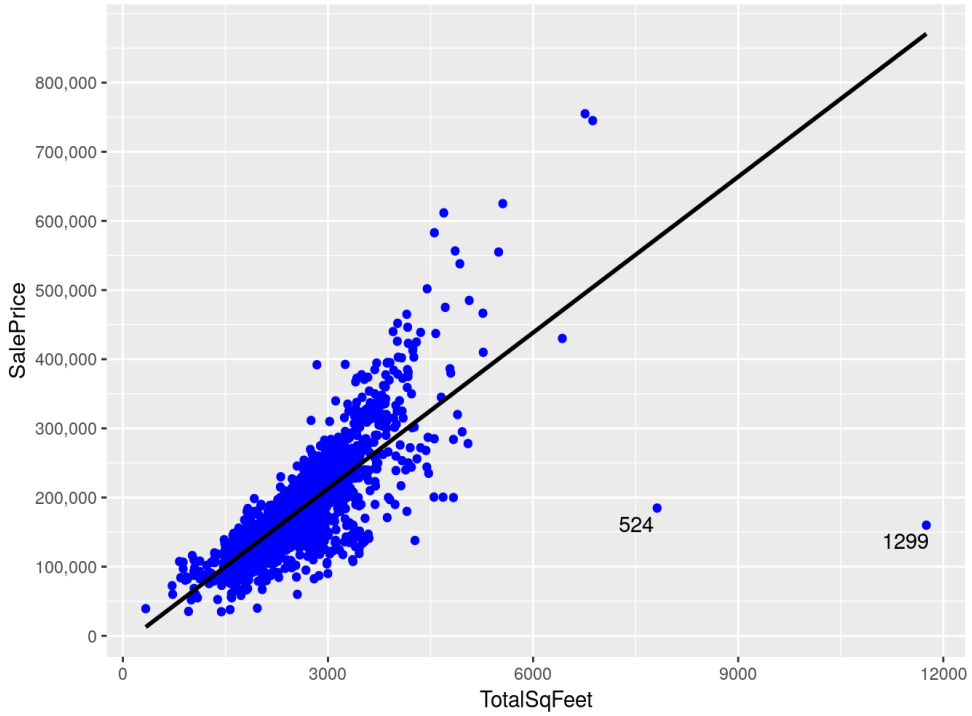


Figure 13

It is very similar to the relationship between GrLivArea and sale price (Section 2.2.3). There are also two potential outliers. (House 524 and 1299).

The correlation between the TotalSqFeet and SalePrice is around 0.78 which is higher than the correlation between GrLivArea and SalePrice. If we delete the two potential outliers, the correlation will increase by 5%. (0.82)

2.8.3 Neighborhood

We have visualized the neighborhood variable in section 5.2.1. There are so many labels in this variable, and it will be very hard to analysis. Therefore, I want to combine them into only a few labels (3 or 4).

In order to achieve my goal, I want to see the relationship between this neighborhood and the median sale price as well as the mean sale price.

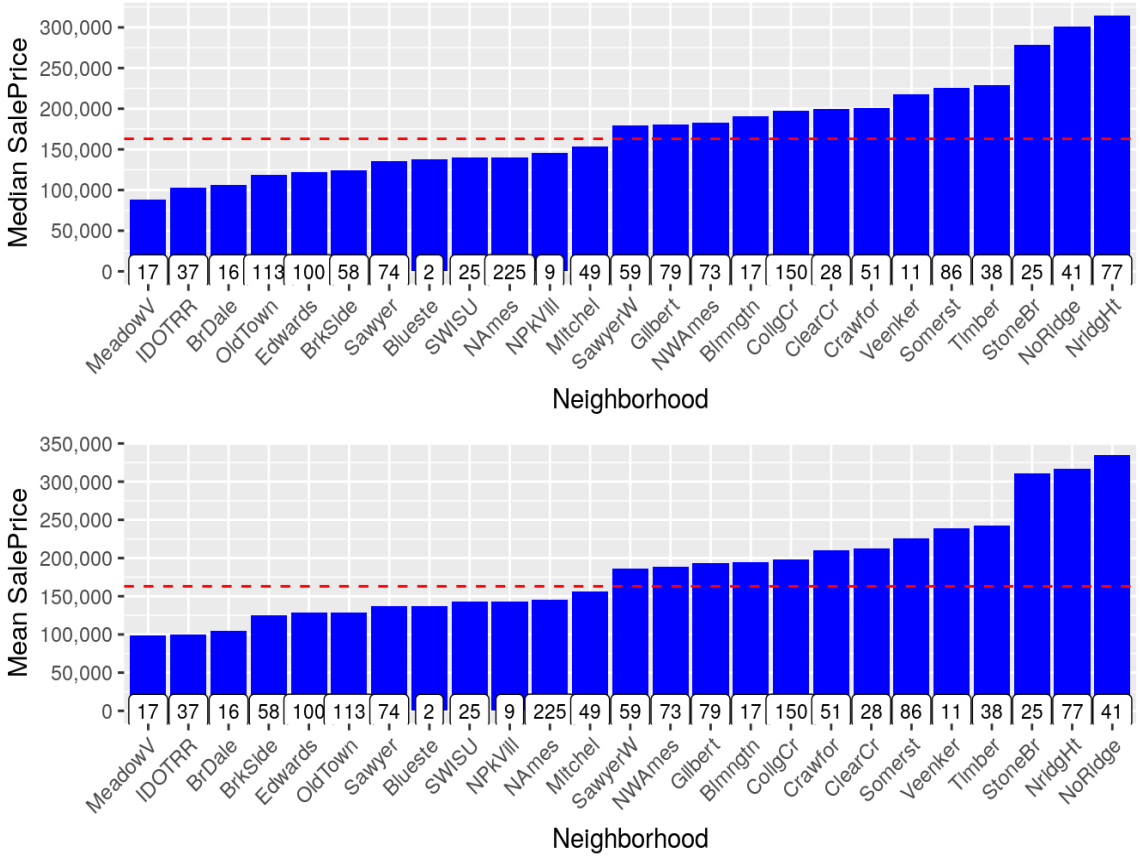


Figure 14

From the figure 14, we can find some results.

Both the median and mean sale price in ‘StoneBr’, ‘NridgHt’, ‘NoRidge’ are very high. Also,

both the median and mean sale price in ‘MeadowV’, ‘IDOTRR’, ‘BrDale’ are the three lowest. Therefore, I want to combine these labels into only 3 labels as follows:

Table 6

ORIGINAL LABEL	NEW LABEL
'STONEBR', 'NRIDGHT', 'NORIDGE'	2
'MEADOWV', 'IDOTRR', 'BRDALE', 'STONEBR', 'NRIDGHT', 'NORIDGE'	1
'MEADOWV', 'IDOTRR', 'BRDALE'	0

2.9 Preparing data for modeling

2.9.1 Dropping highly correlated variables

From the correlation plot, we can find that some variables are highly correlated. Therefore, I need to drop a variable if the correlations of two variables are very high. For example: The GarageCars variable and GarageArea variable have a correlation of 0.89. Of those two, I will drop the variable with the lowest correlation with SalePrice (which is GarageArea with a SalePrice correlation of 0.62. GarageCars has a SalePrice correlation of 0.64).

All in all, I will drop 7 variables which are ‘YearRemodAdd’, ‘GarageYrBlt’, ‘GarageArea’, ‘GarageCond’, ‘TotalBsmtSF’, ‘TotalRmsAbvGrd’, ‘BsmtFinSF1’.

2.9.2 Removing outliers

In section 2.2.3, I said that the house 524 and 1299 are prime candidates to be taken out as outliers. Now I plan to just remove the two really big houses with low SalePrice. Maybe I will investigate these two houses more in the future.

2.9.3 Dealing with skewness of response variable

In 2.1 Section, we find that the response variable - SalePrice doesn’t have a normal distribution. Therefore, we need to take transformation to the variable.

The skew of the variable before transformation is 1.87 which indicates the right skew is too high and the QQ plot shows that sale prices are also not normally distributed. To fix this problem, I choose to take the log transformation of SalePrice.

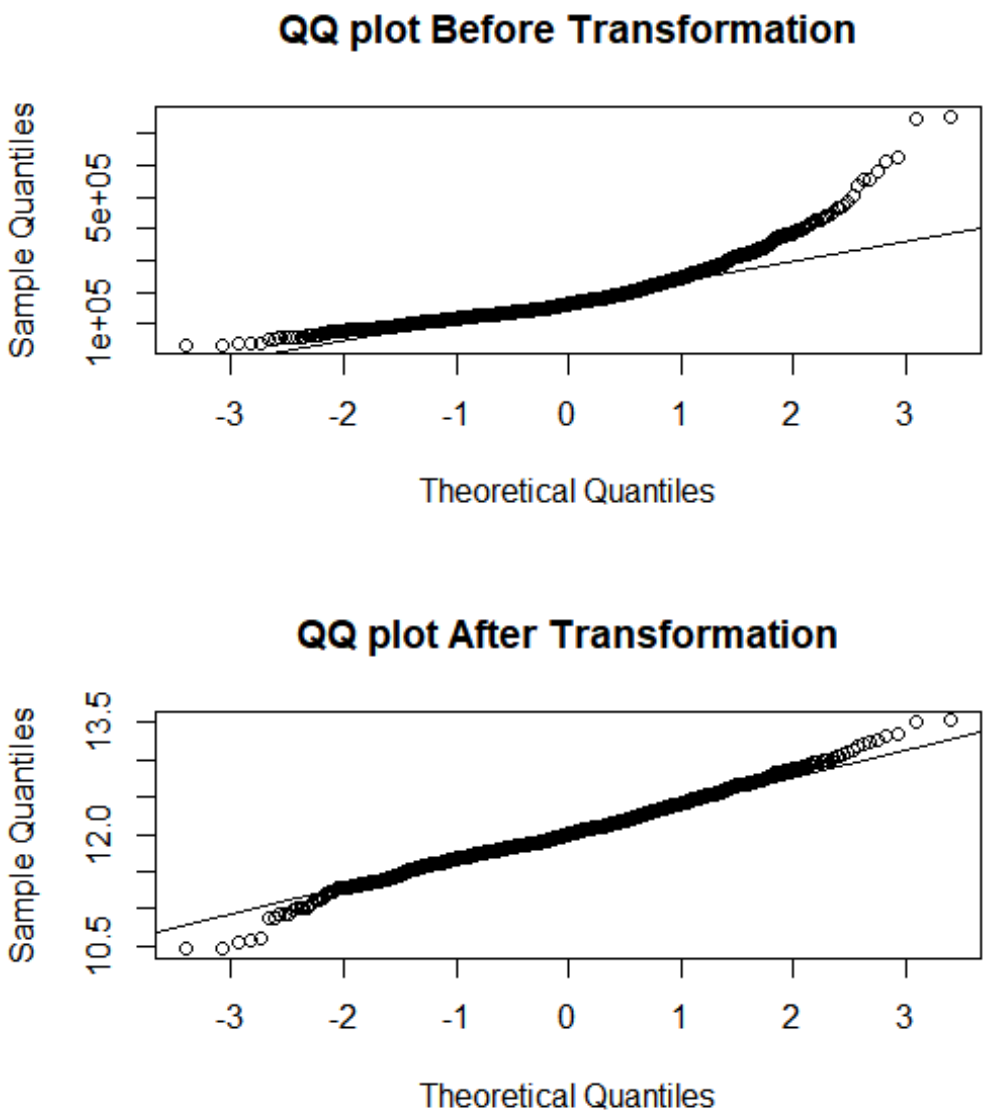


Figure 15

After I take log transformation, the skew is 0.12 which is very low. And the QQ plot looks very good.

2.10 Splitting the data

Before fit model to the data set, I need to split the data into training set and test set.

Each model will be trained on the same training dataset and evaluated on the same test dataset.

The following models will be evaluated by the MSE and R-squared and compared to determine which model is the most effective model.

The data will be splitted into two parts for the modeling process. First, the training dataset contain 70% of the total dataset. And the test dataset will contain 30% of the total data.

CHAPTER 3

3. Criteria to measure performance

3.1 RMSE - root mean square error

The root-mean-square error (RMSE) represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences.

RMSE is the square root of the average of squared errors. The effect of each error on RMSE is proportional to the size of the squared error; thus, larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers.

The RMSE of an estimator $\hat{\theta}$ with respect to an estimated parameter θ is defined as the square root of the mean square error:

$$RMSE(\hat{\theta}) = \sqrt{MSE(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)}$$

For an unbiased estimator, the RMSD is the square root of the variance, known as the standard deviation. [4]

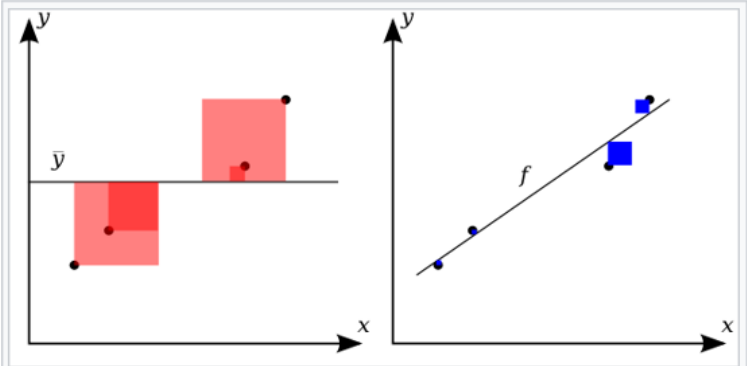
The RMSE of predicted values $\hat{y_i}$ of a regression's dependent variable y_i, with variables observed over n times, is computed for n different predictions as the square root of the mean of the squares of the deviation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{n}}$$

3.2 R Squared - Coefficient of determination

In statistics, the coefficient of determination, denoted R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variables.

When evaluating the goodness-of-fit of simulated (Y_{pred}) vs. measured (Y_{obs}) values, it is not appropriate to base this on the R^2 of the linear regression (i.e., $Y_{obs} = m \times Y_{pred} + b$). The R^2 quantifies the degree of any linear correlation between Y_{obs} and Y_{pred}, while for the goodness-of-fit evaluation only one specific linear correlation should be taken into consideration: $Y_{obs} = 1 \times Y_{pred} + 0$(i.e., the 1:1 line).



$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$

The better the linear regression (on the right) fits the data in comparison to the simple average (on the left graph), the closer the value of R^2 is to 1. The areas of the blue squares represent the squared residuals with respect to the linear regression. The areas of the red squares represent the squared residuals with respect to the average value.

Figure 16

A data set has n values marked $y_1, \ldots, y_n$ (collectively known as y_i or as a vector $y = [y_1, \ldots, y_n]^T$), each associated with a fitted (or modeled, or predicted) value $\hat{y}_1, \ldots, \hat{y}_n$.

Define the residuals as $e_i = y_i - \hat{y}_i$.

If $\overline{y}$ is the mean of the observed data, we have

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

So, the variability of the data set can be measured using three sums of squares formulas:

a. The total sum of squares (proportional to the variance of the data):

$$SS_{tot} = \sum_i (y_i - \overline{y})^2$$

b. The regression sum of squares, also called the explained sum of squares:

$$SS_{reg} = \sum_i (\hat{y}_i - \overline{y})^2$$

c. The sum of squares of residuals, also called the residual sum of squares:

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2 = \sum_i e_i^2$$

The most general definition of the coefficient (R squared) of determination is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R^2 is a statistic that will give some information about the goodness of fit of a model. Suppose $R^2 = 0.49$. This implies that 49% of the variability of the dependent variable has been accounted for, and the remaining 51% of the variability is still unaccounted for. In regression, the R^2 coefficient of determination is a statistical measure of how well the regression predictions approximate the real data points. An R^2 of 1 indicates that the regression predictions perfectly fit the data [5].

CHAPTER 4

4. Model Fitting

4.1 Linear Regression

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their

parameters and because the statistical properties of the resulting estimators are easier to determine. [6]

Given a dataset $\{y_i, x_{i1}, \ldots x_{ip}\}_{i=1}^{n}$ of n statistical units, a linear regression model assumes that the relationship between the dependent variable y and the p-vector of regressors x is linear. This relationship is modeled through a disturbance term or error variable ϵ, which is an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus the model takes the form:

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} = X_i^T \beta + \epsilon$$

Often these n equations are stacked together and written in matrix notation as

$$Y = X\beta + \epsilon$$

All in all, I started with linear regression model which is the simplest in my cases. I will give the RMSE and R Squared of both the training set and the test set in the following table.

Table 7

OLS	TRAINING SET	TEST SET
RMSE	0.0883	0.1262
R^2	0.9512	0.9312

Remember I have taken log transformation to the outcome variable – Sale Price. Therefore, the RMSE value is very low. But it doesn't matter, we can still compare these values.

The result of the linear regression model seems good. However, I need to check the assumption of the linear regression. Figure 17 will help me do the check.

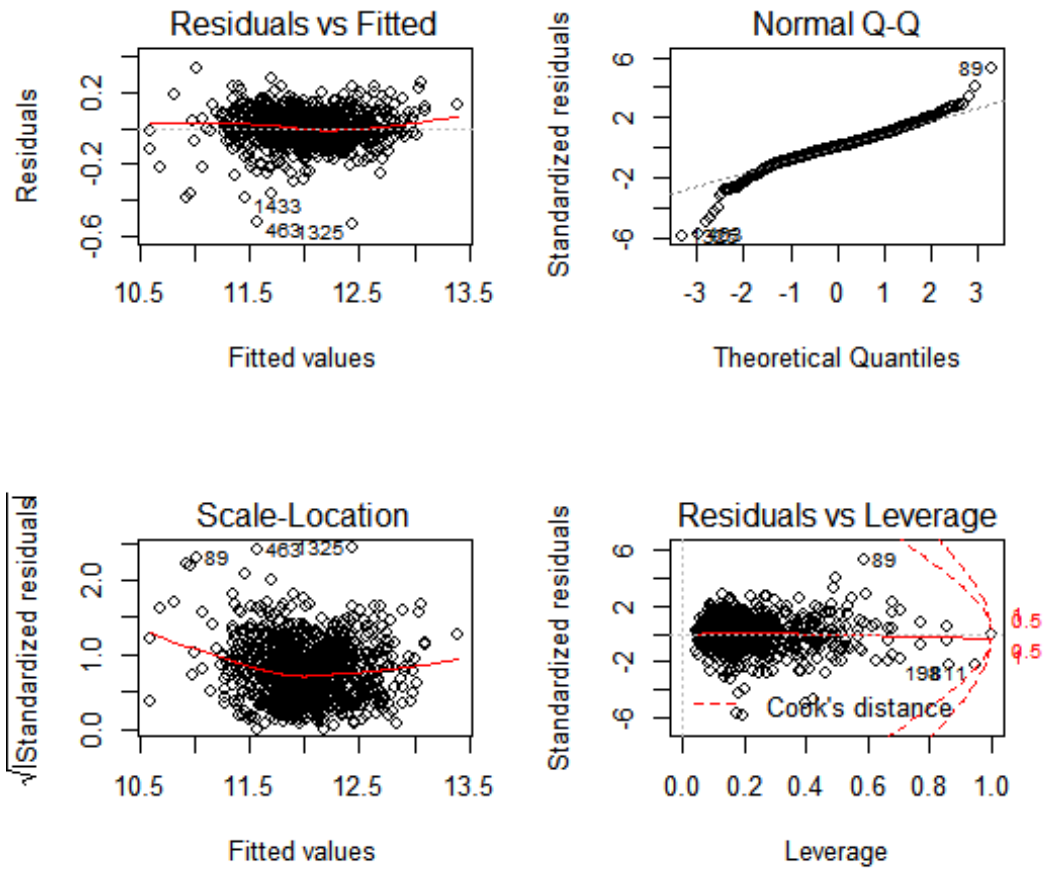


Figure 17

Plot 1 Residuals vs Fitted: Relatively straight line but with patterns. Linearity assumption can be accepted.

Plot 2 QQ plot for normality: Relatively normal. Normality assumption can be accepted.

Plot 3 Scale-Location for assumption of equal variance: Relatively flat line, however it has patterns. We will need to check further using ncv test.

Plot 4 Residuals vs Leverage: No influential case as we barely can see Cook's distance line.

The NCV test shows the p-value << 0.05, which tells us our final linear model doesn't meet the assumption of equality of the variances. Therefore, we can't use the linear model as the candidate of our final model.

4.2 Lasso Regression

In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

Lasso was originally formulated for least squares models and this simple case reveals a substantial amount about the behavior of the estimator, including its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding. It also reveals that (like standard linear regression) the coefficient estimates do not need to be unique if covariates are collinear.

Though originally defined for least squares, lasso regularization is easily extended to a wide variety of statistical models including generalized linear models, generalized estimating equations, proportional hazards models, and M-estimators, in a straightforward fashion. Lasso's ability to perform subset selection relies on the form of the constraint and has a variety of interpretations including in terms of geometry, Bayesian statistics, and convex analysis.

Lasso was introduced in order to improve the prediction accuracy and interpretability of regression models by altering the model fitting process to select only a subset of the provided covariates for use in the final model rather than using all of them. It was developed independently in geophysics, based on prior work that used the l_1 penalty for both fitting and

penalization of the coefficients, and by the statistician, Robert Tibshirani based on Breiman's nonnegative garrote.

Lasso was originally introduced in the context of least squares, and it can be instructive to consider this case first, since it illustrates many of lasso's properties in a straightforward setting. Consider a sample consisting of N cases, each of which consists of p covariates and a single outcome. Let y_i be the outcome and $x_i = (x_1, x_2, \ldots, x_p)^T$ be the covariate vector for the ith case. Then the objective of lasso is to solve.

$$\min_{\beta_0,\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)\}$$

where this formula is subject to $\sum_{j=1}^{p} | \beta_j | \leq t$.

Here t is a prespecified free parameter that determines the amount of regularisation. Let X be the covariance matrix, so that $X_{ij} = (x_{ij})_j$ and x_i^T is the i_{th} row of X, the expression can be written more compactly as

$$\min_{\beta_0,\beta}\{\frac{1}{N} || y - \beta_0 1_N - X\beta ||_2^2\}$$

where this formula is subject to $|| \beta ||_1 \leq t$

Note that $|| \beta ||_p = (\sum_{i=1}^{N} | \beta_i |^p)^{\frac{1}{p}}$ is the standard l^p norm, and 1_N is an $N \times 1$ vectors of ones. Denoting the scalar mean of the data points x_i by $\overline{x}$ and the mean of the response variables y_i by $\overline{y}$, the resulting estimate for β_0 will end up being $\hat{\beta}_0 = \overline{y} - \overline{x}^T\beta$, so that

$$y_i - \hat{\beta}_0 - x_i^T\beta = y_i - (\overline{y} - \overline{x}^T\beta) - x_i^T\beta = (y_i - \overline{y}) - (x_i - \overline{x})^T\beta$$

and therefore, it is standard to work with variables that have been centered (made zero-mean). Additionally, the covariates are typically standardized ($\sum_{i=1}^{N} x_i^2 = 1$) so that the solution does not depend on the measurement scale.

It can be helpful to rewrite

$$\min_{\beta \in R^p} \{ \frac{1}{N} || \, y - X\beta \, ||_2^2 \}$$

where this formula is subject to $|| \, \beta \, ||_1 \leq t$ in the so called Lagrangian form

$$\min_{\beta \in R^p} \{ \frac{1}{N} || \, y - X\beta \, ||_2^2 + \lambda \, || \, \beta \, ||_1 \}$$

where the exact relationship between t and λ is data independent [6].

In section 4.1, our linear regression model doesn't meet the assumption of equality of the variances, so we used the lasso regression to fix this issue. And let's see the results of the lasso regression:

Table 8

LASSO	TRAINING SET	TEST SET
RMSE	0.0992	0.1176
R^2	0.9084	0.8878

Comparing with the result of linear regression model, the result of lasso doesn't seem so good. But this model is robust, and I plan to consider the lasso model as the candidate of my final model.

4.3 Random Forest

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. [7]

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \ldots, x_n$ with response $Y = y_1, \ldots, y_n$ bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1,…,B:

1. Sample, with replacement, n training examples from X, Y; called these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b\,(x')$$

or by taking the majority vote in the case of classification trees.

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de correlating the trees by showing them different training sets.

Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on x':

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}$$

The number of samples/trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees B can be found using cross validation, or by observing the out of bag error: the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample. The training and test error tend to level off after some number of trees have been fit.

The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated. An analysis of how bagging and random subspace projection contribute to accuracy gains under different conditions is given by Ho.

Typically, for a classification problem with p features, $\sqrt{p}$ (rounded down) features are used in each split. For regression problems the inventors recommend $p/3$ with a minimum node size of 5 as the default. In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters [8].

The figure 18 shows the relationship between the error and the number of trees, we find that we need to set the number of trees equal to 100.

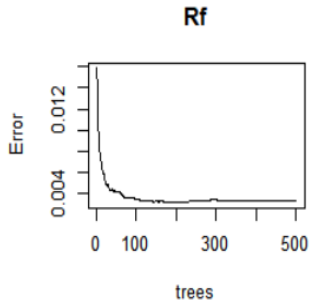


Figure 18

I also want to visualize the important variables, so that I can know which variables in this random forest model tend to have more influence on the sale price.

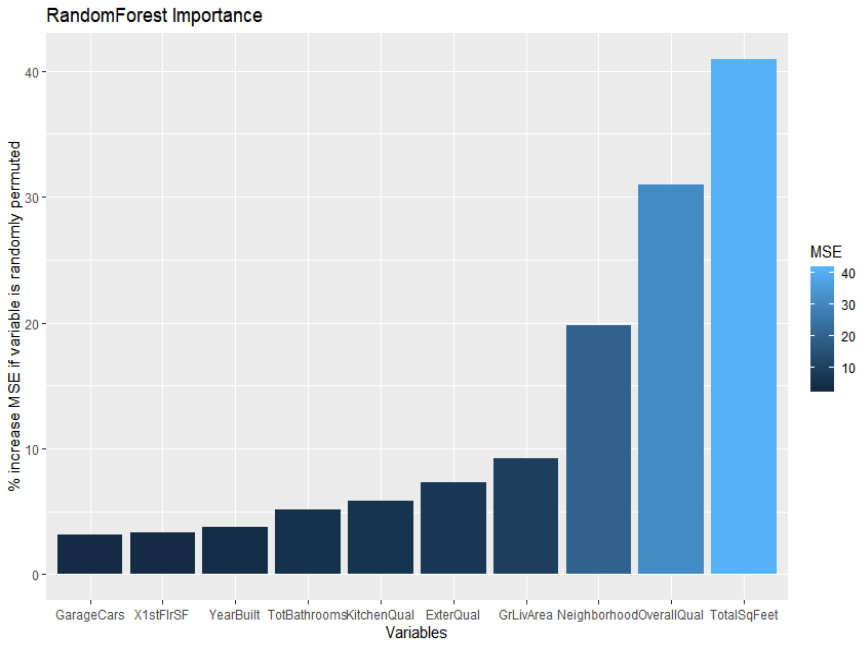


Figure 19

From figure 19, we can see that the TotalSqFeet and OverallQual are the most important two variables.

The RMSE and R squared of my RF model shows as follows:

Table 9

RF	TRAINING SET	TEST SET
RMSE	0.0242	0.0523
R^2	0.9411	0.8673

The RMSE of the random forest model are very low, which tells us the predictions in the random forest model tend to be more central than the regression model.

However, the R squared of the training set is very good. But the R squared of the test set is relatively low, which may show that the random forest model is a little bit overfitting.

4.4 XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

The algorithm differentiates itself in the following ways:

1.A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems.

2.Portability: Runs smoothly on Windows, Linux, and OS X.

3.Languages: Supports all major programming languages including C++, Python, R, Java, Scala, and Julia.

4.Cloud Integration: Supports AWS, Azure, and Yarn clusters and works well with Flink, Spark, and other ecosystems. [9]

XGBoost is the leading model for working with standard tabular data (the type of data you store in Pandas DataFrames, as opposed to more exotic types of data like images and videos).

XGBoost models dominate many Kaggle competitions.

To reach peak accuracy, XGBoost models require more knowledge and model tuning than techniques like Random Forest.

XGBoost is an implementation of the Gradient Boosted Decision Trees algorithm What is Gradient Boosted Decision Trees? We'll walk through a diagram in the figure 20.

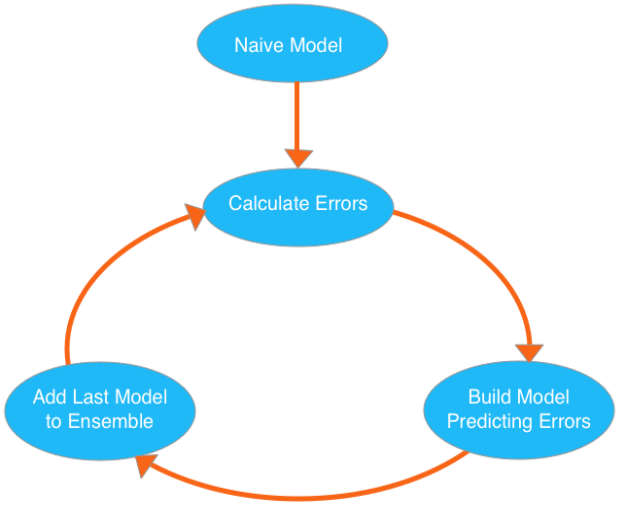


Figure 20

We go through cycles that repeatedly builds new models and combines them into an ensemble model. We start the cycle by calculating the errors for each observation in the dataset. We then build a new model to predict those. We add predictions from this error predicting model to the "ensemble of models."

To make a prediction, we add the predictions from all previous models. We can use these predictions to calculate new errors, build the next model, and add it to the ensemble.

There's one piece outside that cycle. We need some base prediction to start the cycle. In practice, the initial predictions can be pretty naive. Even if its predictions are wildly inaccurate, subsequent additions to the ensemble will address those errors. [10]

Actually, I just worked with the XGBoost package in R directly. The main reason for this was that the package uses its own efficient data structure which help me a lot to get the result.

The importance of the XGboost model is shown as follows:

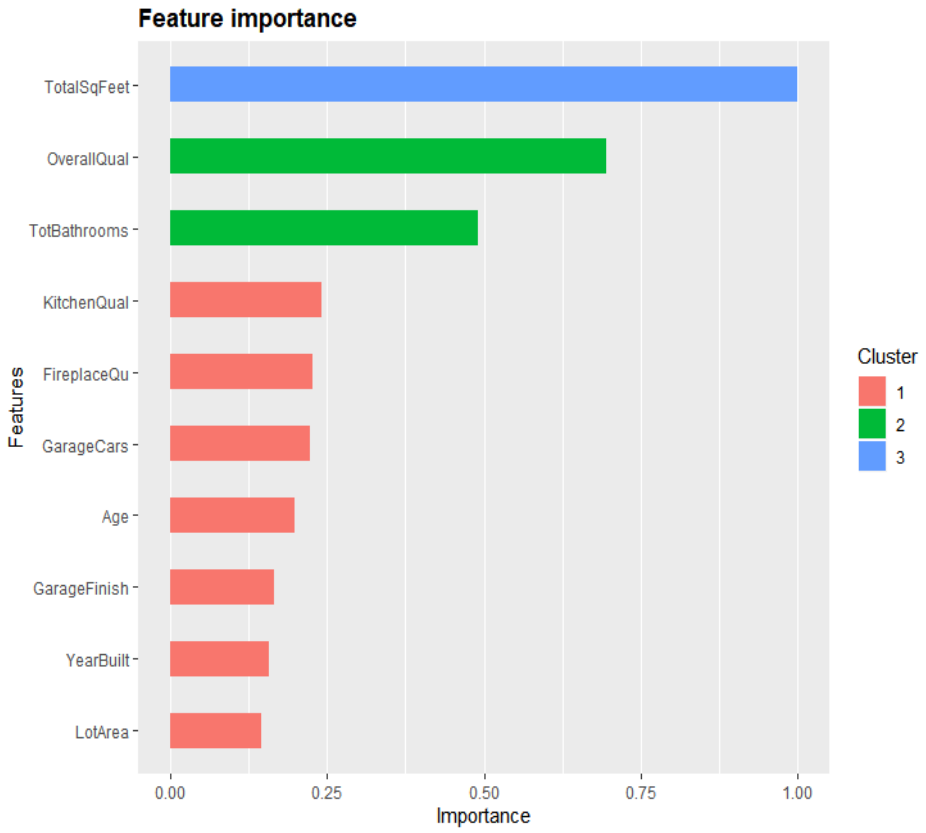


Figure 21

We can see that the TotalSqFeet and OverallQual are the most important two variables which are the same as the result of RandomForest model.

Table 10 gives the RMSE and R squared of my Xgboost model.

Table 10

XGBOOST	TRAINING SET	TEST SET
RMSE	0.0588	0.1212
R^2	0.9495	0.9348

The RMSE of my Xgboost model aren't bad, which tells us the prediction of my xgboost model aren't very far from the true sale price. And the R-squared in the training set and test set are both high, which shows the xgboost model are reasonable and not overfitting. All of the RMSE and R squared seems good, I may use the Xgboost model as my final model.

CHAPTER 5

5. Conclusion

The objective of this paper is to fit models to predict the housing sale price and find some important aspects of the house.

In order to achieve my goal, I fit four models to the dataset: linear regression, lasso regression, random forest and Xgboost. As for the first model - linear regression, it doesn't meet the assumption of equality of the variances. Therefore we can't use the linear model as the candidate of our final model. In order to deal with this problem, I try the second model - lasso regression, but the Rmse and R-squared looks not so good. The third model is Random forest. The Rmse of this model are relative low in both the training set and test set, which shows the predictions seems not bad. What's more, The R squared in this model of training set is very good, but in the test set the R squared is relatively low, which may show the random forest model is a little bit overfitting. Finally, I try the fourth model - Xgboost. All of the results of this xgboost model seem good. Therefore, I will use this xgboost model as my final model to predict the housing price.

What's more, from the feature importance plot of the Xgboost, we can know that the total square feet, the overall quality, and the total number of bathrooms are the three main aspects which influence the housing sale price.

6.Reference

[1] " House Prices: Advanced Regression Techniques ". Kaggle, 2020, https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

[2] "Financial crisis 07-08". Wikipedia, 2020, https://en.wikipedia.org/wiki/Financial_crisis_of_2007–08.

[3] "Half-Bath". Relator, 2020, https://www.realtor.com/advice/buy/what-is-a-half-bath/.

[3] "RMSE". Wikipedia, 2020, https://en.wikipedia.org/wiki/Root-mean-square_deviation/.

[4] "R-squared". Wikipedia, 2020, https://en.wikipedia.org/wiki/Coefficient_of_determination/.

[5] "Linear Regression". Wikipedia, 2020, https://en.wikipedia.org/wiki/Linear_regression/.

[6] "Lasso Regression". Wikipedia, 2020, https://en.wikipedia.org/wiki/Lasso_(statistics)

[7] "Understanding Random Forest". Medium, 2020, https://towardsdatascience.com/understandingrandom-forest-58381e0602d2/.

[8] "Random Forest". Wikipedia, 2020, https://en.wikipedia.org/wiki/Random_forest.

[9] "Xgboost Algorithm: Long May She Reign!". Medium, 2020, https://towardsdatascience.com/httpsmedium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

[10] "Xgboost Model". Kaggle, 2020, https://www.kaggle.com/dansbecker/xgboost