Zillow House Price Prediction

Wei Zou(wz299) Jiahe Xu(jx266)

December 5, 2017

Note: that in project proposal and midterm report, the goal of our team is to predict the log(error) of Zillow Estimation. However, when we process further the project, we notice that it focuses more on the feature engineering. Hence, we decide to change our dataset and choose another similar dataset which is to predict the house price rather than predict the log(error) of Zillow Estimation.

Abstract

Zillow home provides a dataset including 21613 house prices with 19 related features which have potential impact on the sale price. This project aims to build multiple models to predict the house sale prices. The models based on random forest, quadratic and huber loss funtions with l_1,l_2 and without regularization were evaluated with their MSE. Results turn out random forest performs best with MSE lower than 0.03. Further improvement is discussed in the end.

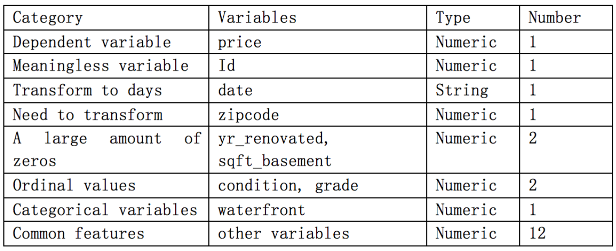
1 Introduction

Zillow's Zestimate home valuation has influenced the U.S. real estate industry since first released 11 years ago. They devote themselves to collecting as much as possible related data and extracting useful information to help the first-time consumers access to home values information at no cost. In this report, the goal is to use a number of house features from Zillow to predict the house price. Also, the team is going to figure out which factor play a significant role that determine the house price. It could help Zillow to figure out the important factors that buyers may concern. Here we use several combinations of loss functions and regularizations, and random forest model to fit the dates sets.

2 Data Analysis

2.1 Data Description

This dataset contains house sale prices for King County, which includes Seattle and it includes homes sold between May 2014 and May 2015. Specifically, our dataset consists of 19 house features plus the price and the id columns, along with 21613 observations. Because there are very limited variables in the dataset, we list all the features and consider them all carefully in section 2.3 DATA CLEAN.



Category	Variables	Type	Number
Dependent variable	price	Numeric	1
Meaningless variable	Id	Numeric	1
Transform to days	date	String	1
Need to transform	zipcode	Numeric	1
A large amount of zeros	yr_renovated, sqft_basement	Numeric	2
Ordinal values	condition, grade	Numeric	2
Categorical variables	waterfront	Numeric	1
Common features	other variables	Numeric	12

Figure 1: Features Table

2.2 Data Visualization

First of all, we perform the correlation matrix for all variables and plot it. As we can see, positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients. In the right side of the correlogram, the legend color shows the correlation coefficients and the corresponding colors.

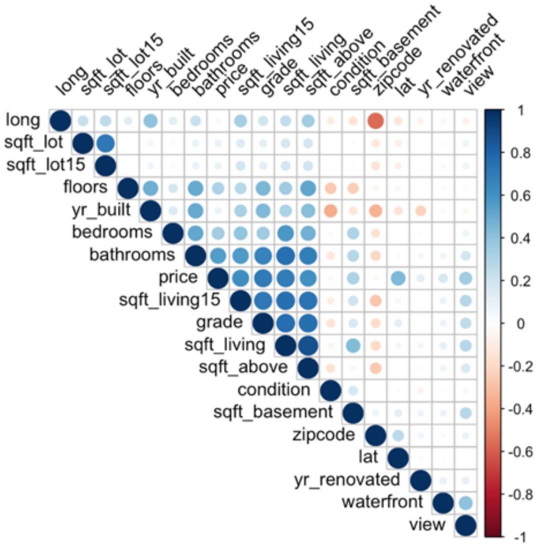


Figure 2: Correlation Matrix Plot

From the figure, it indicates that the house price is highly related to the variables below:
grade: overall grade given to the housing unit.
sqft_living: square footage of the home.
sqft_labove: square footage of house apart from basement.
sqft_lliving15: living room area in 2015.

Hence, for a further investigation we continue to plot correlation matrix with just these 4 variables above plus house price as below. The distribution of each variables is shown on the diagonal. The bivariate scatter plots with a fitted line are displayed on the bottom of the diagonal. The value of the correlation plus the significance level as stars on the top of the diagonal.

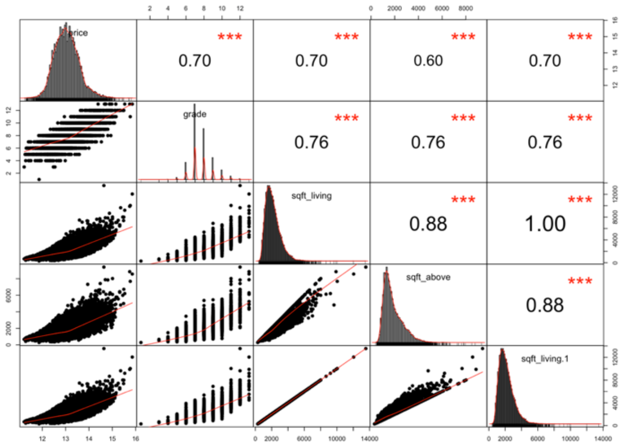


Figure 3: Partial Correlation Matrix Plot

From the first column of the figure above, for each of those four variables, it indicates that the house price will increase as each of them increases. Take the plot of second column and first row for example, it indicates that the correlation between price and grade is 0.70 and their relation is significant with three stars.

2.3 Data Cleaning

2.3.1 Data Transformation

After plotting the price, we find the distribution is skewed to the left, which means the house price tend to be lower level. In order to make the model we fit perform better, it is common to take log to price as the dependent variable.

Note that in this report, we mention MSE, it actually means:

$$MSE = MEAN((log(\text{predicted price}) - log(\text{true price}))^2)$$

Among all the 19 features, it contains one date-formatted variable date(Transaction date) (eg.20140810T000000), however, regression problems could not handle this kind of variable with type string. Also, we cannot consider it as a categorical variable and separate it to many dummy variables, which may be really multifarious. Hence, our team transform this variable into date_days(the number of days from transaction till now).And then it may could be a useful variables and we plot it below. From the plot, it is a stable variable, at every time period, the house with different have stable transactions.

As for one categorical variables water_front(), it is already transformed to dummy variable with value 0 and 1. We can just leave it.

Moreover, it contains two ordinal variablescondition(1-5) and grade(1-10). There are two alternatives to deal with them. One is to create corresponding number of dummy variables (eg.creating 5 dummy variables for feature condition). Second alternative is that we do not need to do any transformation for them. Both

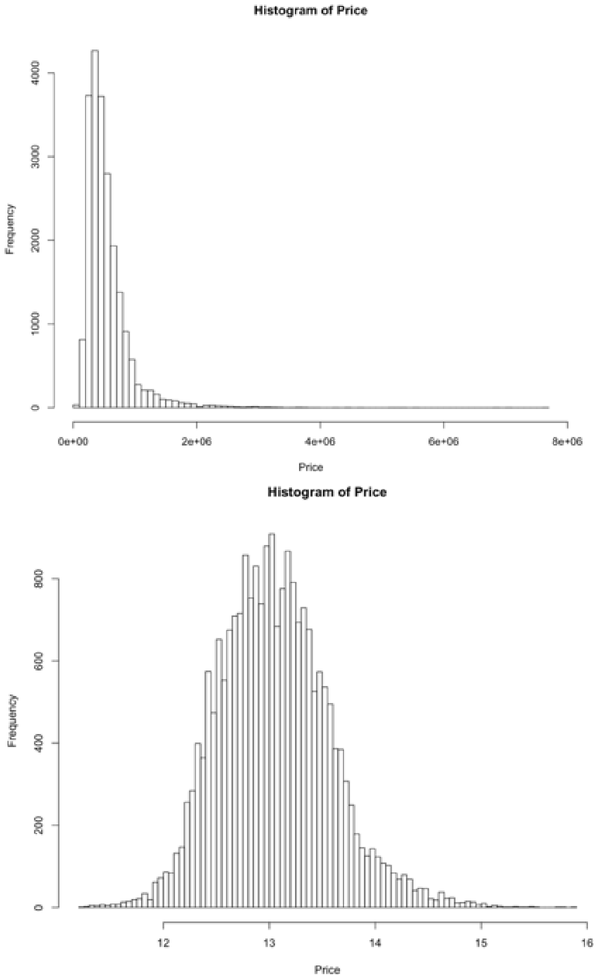


Figure 4: Histogram of Price

of them could be considered as continuous numeric variables and it makes sense that as the grade increases, the house price increases accordingly.

One more variable we should consider is the variable zipcode. Obviously, taking it as numeric variable does not make sense for fitting a model. There also are two alternatives to deal with this feature. One alternative is to translate it to a categorical variable. Each zip code corresponds to a variable and we could separate this variable into many variable, each represented by dummy variable with value 0 and 1. Another alternative is to use the latitude and longitude of the center point of the zipcode as two new variables. Notice that our dataset has already had longitude and latitude. Hence, we just discard this feature.

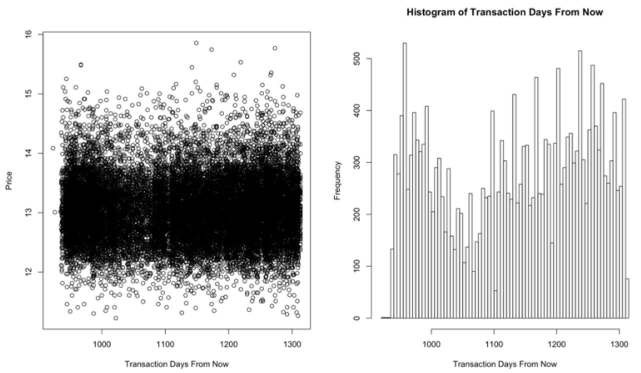


Figure 5: Plot of date_days(new features)

2.3.2 Missing Value

This dataset is pretty clean and hence we do not need to deal with multifarious missing value.

There only two variable that have a huge amount of zero value are yr_renovated(Year when the house was renovated) and sqft_basement(Square Footage of Basement).

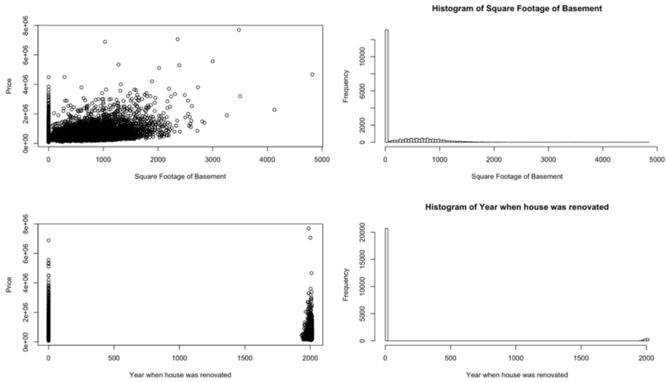


Figure 6: Variables with a large amount of zeros

We have the issue that we do not know that whether the zero value means that these records miss a value or it means these records do not satisfied that feature. Take yr_renovated for example, does zero mean this house have never been renovated or does it mean we lost that value? The percentage of zeros in those two variables plotted above are 61% and 96%, respectively. Considering that there is no requirement that the features should have a specific distribution, so we just leave these two variables.

3 Model

In this section, our team split our dataset randomly into training set(4/5) and test set(1/5). Training data is used to train the model and then use the test set to see the performance of the models based on the MSE.

3.1 Tree Based Method

Tree-based method is a popular method for either classification problem or regression problems. In lecture we have learned bagging, random forest and boosting. Considering that the random forest method is more well-performed than bagging generally, thus random forest and boosting will be discussed in this section.

In general, we choose the number of predictor, which needs to be performed in every split, as one third of for regression problems. The number of trees is set to 500 since it will not be overfitting for random forest and 500 trees is enough for our fitting.

After performing the random forest model, we get MSE 0.02948759 and plot the predicted price and the true price.

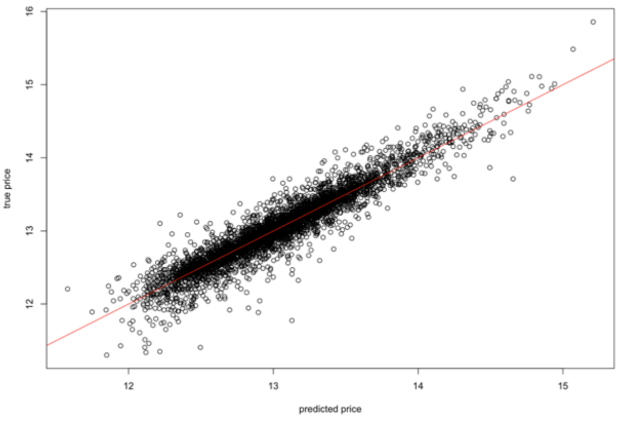


Figure 7: Prediction Result of Random Forest

Just as a further investigation we would like to view the importance of each variable, so our team measure the total decrease in node impurity that results from splits over that variable, average all trees as the right panel of Figure 5. Based on the mean decrease of accuracy in predictions on the out of random forest samples when a given variable is excluded from the model, the result is shown as the left panel of Figure 5.

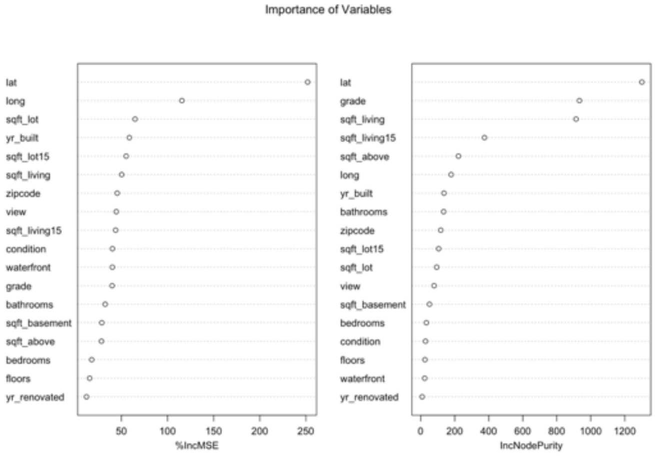


Figure 8: Importance of Variables

From this result, it means that if variable lat is deducted from this model, then there might be much impact on prediction performance of this model. It is may because this variable lat is chosen very early in the split process.

3.2 Boosting

For the boosting, it is to fit a new tree to the RESIDUAL of the previous model. The Algorithm for boosting regression trees is show as below.(ISLR 8.2.3)

Algorithm (for boosting regression tree):
1 Initialization: $\widehat{f} \equiv 0$ and $r_i = y_i, i = 1, ..., n$
2 For $b = 1, ..., B$:

a Fit residual: Fit a new tree $\widehat{f}^b$ with $d+1$ leaves to the data X, r

b Shrink and combine: Update $\widehat{f}$ by adding a shrunken version of $\widehat{f}^b$:

$$\widehat{f}(x) \leftarrow \widehat{f}(x) + \lambda \widehat{f}^b(x)$$

c Update residuals:

$$r_i \leftarrow r_i + \lambda \widehat{f}^b(x_i) \quad i = 1, ..., n$$

We take the depth as 4 and choose the number of trees through cross validation as figure 9, here, we choose the maximum number we set 5000.And then we perform that training model on test dataset. The MSE is 0.0414706.

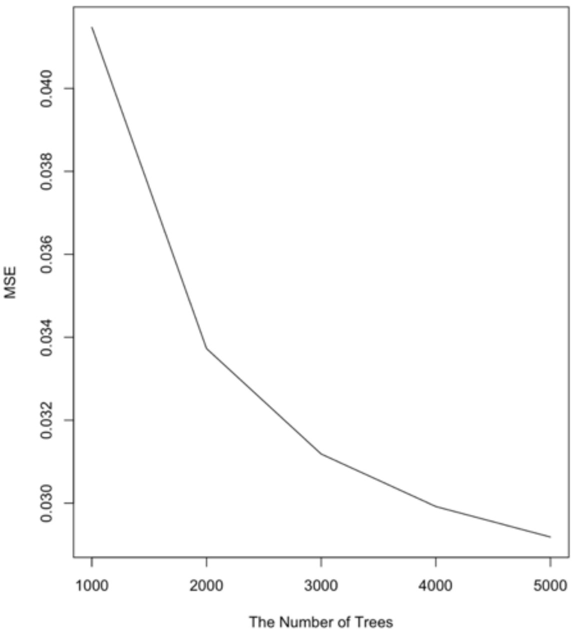


Figure 9: Cross Validation for the Number of Trees

4 Regression

In the lecture, we discuss quadratic, Huber and quantile loss functions, and lasso and ridge regularization for regression problems. Here we will perform several different combinations.

4.1 Quadratic Regression without Regulation

For fitting the quadratic regression without regularization, our goal is to minimize:

$$minimize \quad \Sigma_{i=1}^{n}(y_i - w^T x_i)^2$$

Firstly, we train a model using training data set, and then get MSE 0.06192281 when performing the model on the test data. In order to see the performance of the model, we plot the predicted price and the true price as we as the straight line(y = x) to compare them.

From the plot and the MSE we got, the MSE seems pretty good. However,as the Log(price) tend to be the middle value, the residuals became larger.

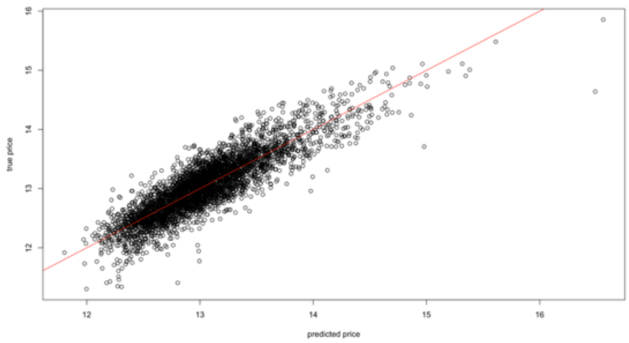


Figure 10: Quadratic Loss Function without Regularization

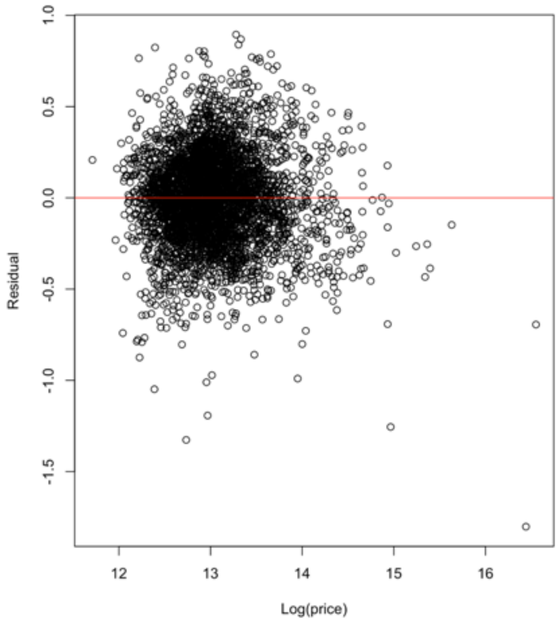


Figure 11: Residuals and Log(price)

4.2 Quadratic Regression with LASSO Regularization

To improve the model, we are going to apply shrinkage methods using ridge regression and lasso regression, both of which need to tune the parameter λ. Every λ corresponds to a regularization function, so we are going to perform cross validation on training data set to determine the λ that gives the least MSE.

For the lasso problem, the goal is to minimize the function below:

$$minimize \quad \Sigma_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda\Sigma_{i=1}^{n}|w|$$

Note that lasso regularization can shrink the coefficient to zeros, which is useful for dealing with those nonsignificant variables in predicting process.

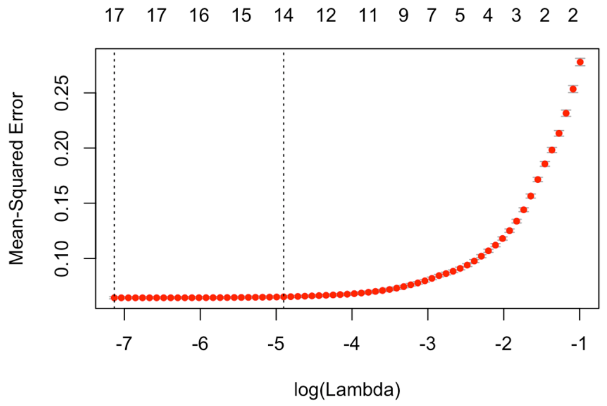


Figure 12: Result of Cross Validation for Quadratic Regression with Lasso Regularization

After performing the cross-validation process using 10-folds, it outputs the λ 0.0007979959 (the dashed line in the figure above) with the smallest MSE.

Next step, we use the 0.0007979959 for λ to training the model and then figure out the MSE = 0.0619331 on test set.

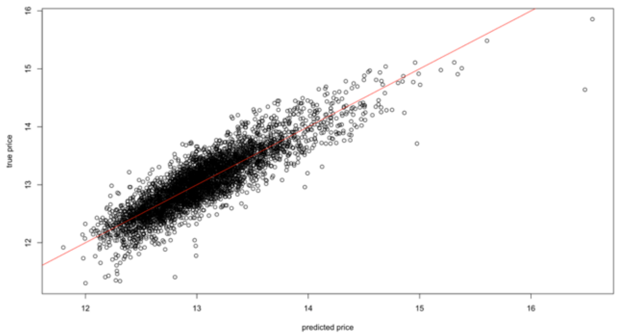


Figure 13: Quadratic Loss Function with Lasso Regularization

The residuals plot is pretty same with that in section 4.1, the Log(price) tend to be the middle value, the residuals became larger.

4.3 Quadratic Regression with Ridge Regularization

For the ridge problem, the goal is to minimize the function below:

$$minimize \quad \Sigma_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda \Sigma_{i=1}^{n} w^2$$

Compared to lass regularization, ridge regression also could shrink the coefficients, the difference is that the ridge regression can NOT shrink the coefficients to zero. The advantage is that it can leave and consider the effect of all variables.

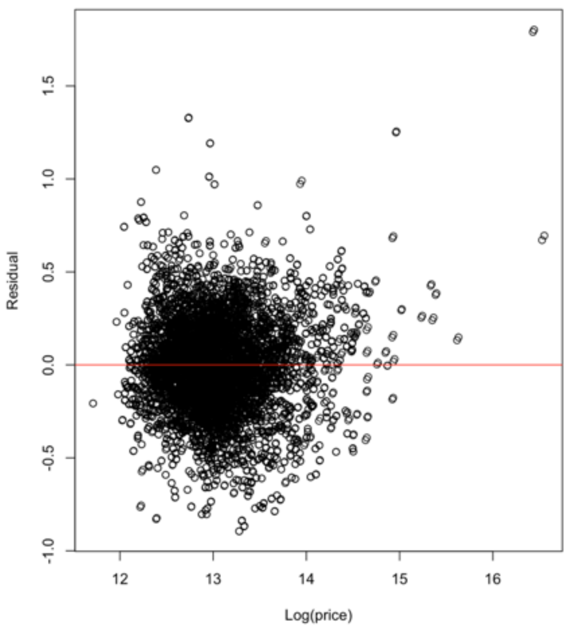


Figure 14: Residuals and Log(price) for Quadratic Regression with Lasso Regularization

After performing the cross-validation process using 10 folds, it outputs the λ 0.04065101 with the smallest MSE.

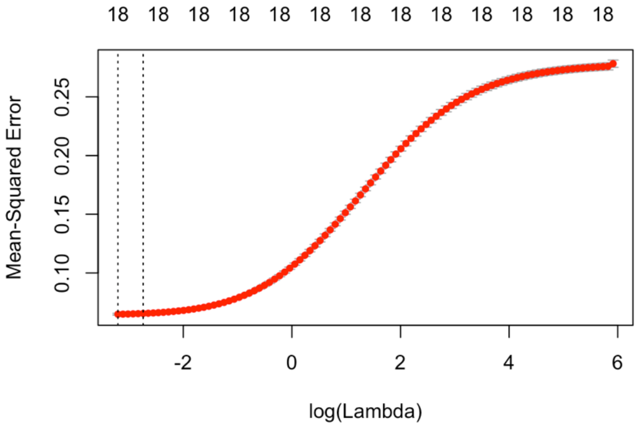


Figure 15: Result of Cross Validation for Quadratic Regression with Ridge Regularization

Next step, we use the 0.04065101 for λ to training the model and then figure out the MSE 0.06217825 on test set.

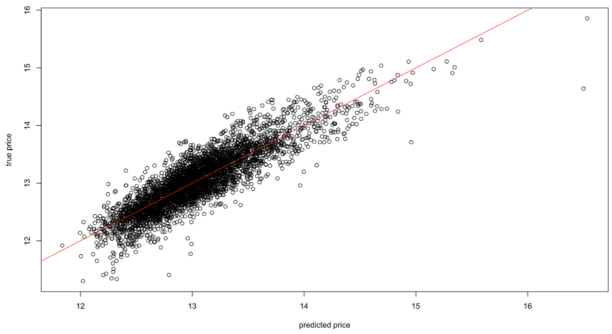


Figure 16: Quadratic Loss Function with Quadratic Regression with Ridge Regularization

4.4 Huber Regression with LASSO Regularization

For this ridge problem, the goal is to minimize the function below:

$$minimize \quad \frac{1}{n}\Sigma_{i=1}^{n}\mathbf{huber}(y_i - w^T x_i)^2 + \lambda|w|^2$$

where,

$$\mathbf{huber(z)} = \begin{cases} \frac{1}{2}z^2 & |z| \leq k \\ k(|z| - \frac{1}{2}k) & |z| \geq k \end{cases}$$

This loss function is quadratic for small values of z and linear for large values, with equal values and slopes of the different sections at the two points where —z—= k. The variable z often refers to the residuals, that is to the difference between the observed and predicted values z = log(predicted price)-log(true price).

Likewise, we choose λ as 0.004526363 from cross validation and then perform the model on test dataset to figure out MSE 0.06243748.

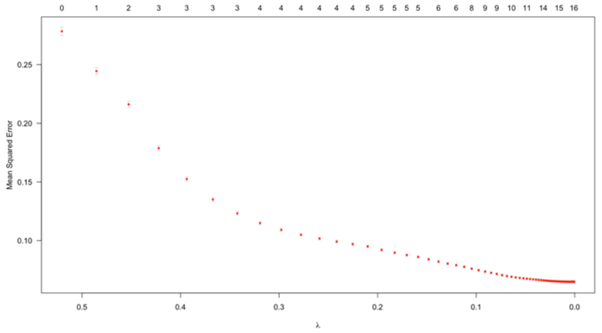


Figure 17: Result of Cross Validation for Huber with Lasso Regularization

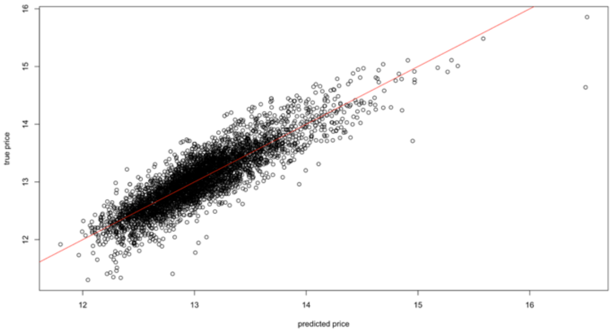


Figure 18: Huber Loss Function with Lasso Regularization

4.5 Huber Regression with Ridge Regularization

For this ridge problem, the goal is to minimize the function below:

$$minimize \quad \frac{1}{n}\Sigma_{i=1}^{n}\mathbf{huber}(y_i - w^T x_i)^2 + \lambda|w|^2$$

where,

$$\mathbf{huber(z)} = \begin{cases} \frac{1}{2}z^2 & |z| \leq k \\ k(|z| - \frac{1}{2}k) & |z|^2 \geq k \end{cases}$$

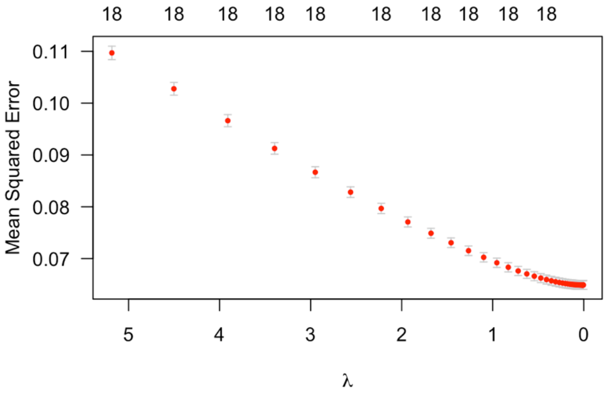


Figure 19: Result of Cross Validation for Huber with Ridge Regularization

Likewise, we choose λ as 0.01390546 from cross validation and then perform the model on test dataset to figure out MSE 0.06246271.

5 Summary

From the figure 21, it indicates that the best method for this dataset I have performed is random forest with the least MSE.

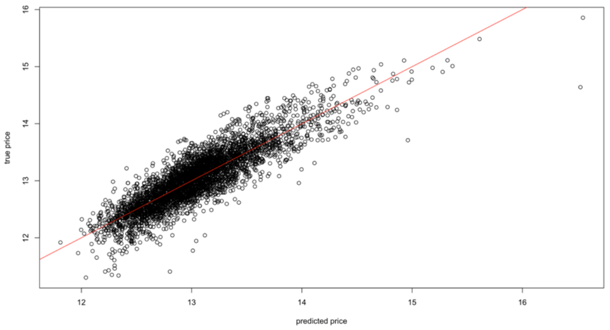
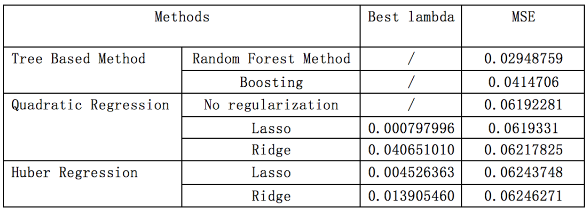


Figure 20: Huber Loss Function with Ridge Regularization

It is really strange that no matter what loss functions (Huber or quadratic regression) that we use with lasso or ridge regularization. We keep getting very small value of λ. And because the smaller it is, the power of its penalty to the coefficients is less. As the λ we get tends to be zero, it is the same with the loss function without regularization. Hence, the plots within the same loss function look pretty same.



Methods		Best lambda	MSE
Tree Based Method	Random Forest Method	/	0.02948759
	Boosting	/	0.0414706
Quadratic Regression	No regularization	/	0.06192281
	Lasso	0.000797996	0.0619331
	Ridge	0.040651010	0.06217825
Huber Regression	Lasso	0.004526363	0.06243748
	Ridge	0.013905460	0.06246271

Figure 21: MSE Table

6 Future Improvement

6.1 Polynomial Regression

It could be considered that to perform polynomial regression to avoid under-fitting and try several different orders of the features and use cross validation to determine the order than gives the least MSE.

6.2 Future Section

We also could do feature selection to choose the proper number of features to fit the model using backward selection, forward selection or best selection. Considering that the dataset just has 18 features, we could use the best selection which is to find the global optimal solution. We plot the relationship between the number of variables used with adjust square of R.

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

Where N is the number of points in data set and K is the number of independent features.

Adjust R-square indicates how well terms fit a curve, but adjusts for the number of terms in a model. If more useless features are added into the model, the adjust R-square will decrease.

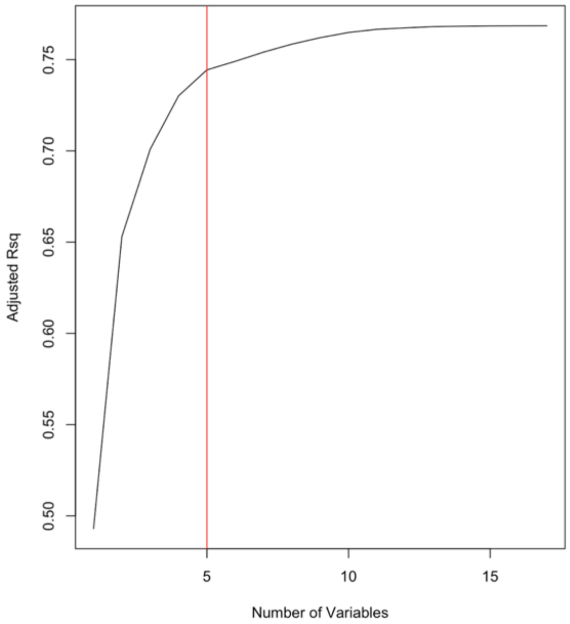
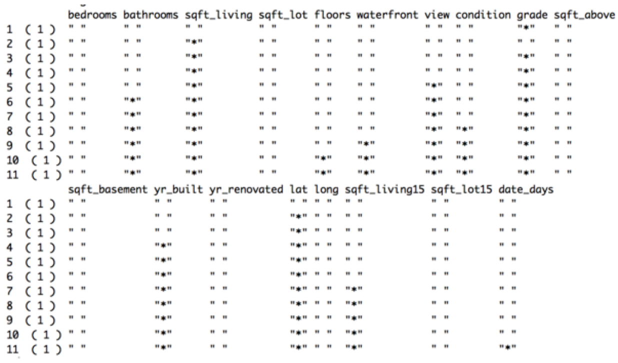


Figure 22: Adjust Rsq plot



```
          bedrooms bathrooms sqft_living sqft_lot floors waterfront view condition grade sqft_above
1  ( 1 )  " "      " "       " "         " "      " "    " "        " "  " "       "*"   " "
2  ( 1 )  " "      " "       "*"         " "      " "    " "        " "  " "       " "   " "
3  ( 1 )  " "      " "       "*"         " "      " "    " "        " "  " "       "*"   " "
4  ( 1 )  " "      " "       "*"         " "      " "    " "        " "  " "       "*"   " "
5  ( 1 )  " "      " "       "*"         " "      " "    " "        "*"  " "       "*"   " "
6  ( 1 )  " "      "*"       "*"         " "      " "    " "        "*"  " "       "*"   " "
7  ( 1 )  " "      "*"       "*"         " "      " "    " "        "*"  " "       "*"   " "
8  ( 1 )  " "      "*"       "*"         " "      " "    " "        "*"  "*"       "*"   " "
9  ( 1 )  " "      "*"       "*"         " "      " "    "*"        "*"  "*"       "*"   " "
10  ( 1 ) " "      "*"       "*"         " "      "*"    "*"        "*"  "*"       "*"   " "
11  ( 1 ) " "      "*"       "*"         " "      "*"    "*"        "*"  "*"       "*"   " "
          sqft_basement yr_built yr_renovated lat long sqft_living15 sqft_lot15 date_days
1  ( 1 )  " "           " "      " "          " " " "  " "           " "        " "
2  ( 1 )  " "           " "      " "          "*" " "  " "           " "        " "
3  ( 1 )  " "           " "      " "          "*" " "  " "           " "        " "
4  ( 1 )  " "           "*"      " "          "*" " "  " "           " "        " "
5  ( 1 )  " "           "*"      " "          "*" " "  " "           " "        " "
6  ( 1 )  " "           "*"      " "          "*" " "  " "           " "        " "
7  ( 1 )  " "           "*"      " "          "*" " "  "*"           " "        " "
8  ( 1 )  " "           "*"      " "          "*" " "  "*"           " "        " "
9  ( 1 )  " "           "*"      " "          "*" " "  "*"           " "        " "
10  ( 1 ) " "           "*"      " "          "*" " "  "*"           " "        " "
11  ( 1 ) " "           "*"      " "          "*" " "  "*"           " "        "*"
```

Figure 23: Choosing the Number of Variables

From the plot above, it is acceptable to only use 5 features in our model which gives a relative high adjust R-square. Also, based on the figure, we choose sqft)_living, view, grade, yr)_built and lat to training our model using the techniques in previous sections.

7 Reference

(1) A PEEK INSIDE THE HOUSE PRICING BLACK BOX:
https://github.com/woshibobo/ORIE-4741Project/blob/master/final)_report.pdf

(2) Adjust R-square:
https://www.quora.com/What-is-the-difference-between-R-squared-and-Adjusted-R-squared

(3) ISLR:
http://www-bcf.usc.edu/ gareth/ISL/

(4) Huber Loss:
https://en.wikipedia.org/wiki/Huber)_loss