Zillow House Price Prediction

Wei Zou(wz299) Jiahe Xu(jx266)

December 4, 2017

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 the house features provided by Zillow to estimate the house price. Also, the team is going to figure out which factors play significant roles that determine the house price. It could help Zillow to determine important factors that buyers will 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. It includes homes sold between May 2014 and May 2015. This dataset consists of 19 house features plus the price and the id columns, along with 21613 observations. Among all the 19 features, one variable is the date of the transaction, two variables are ordinal and the remaining all are numeric variables.

2.2 Data Visualization

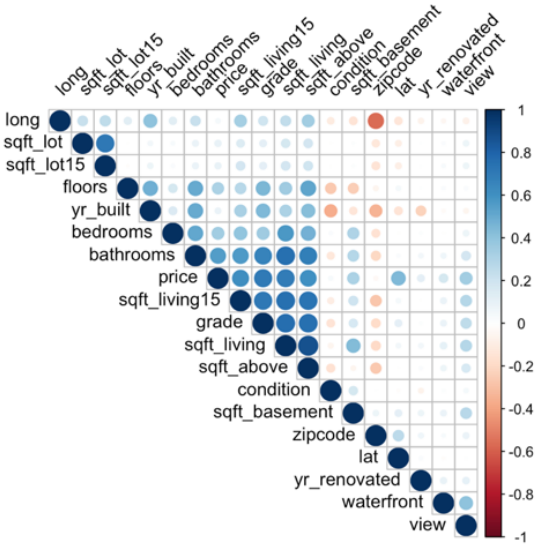

Figure 1: Correlation Matrix Plot

First of all, we perform the correlation matrix for all variables and plot it as above. 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.

From the figure, it indicates that the house price is more 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.

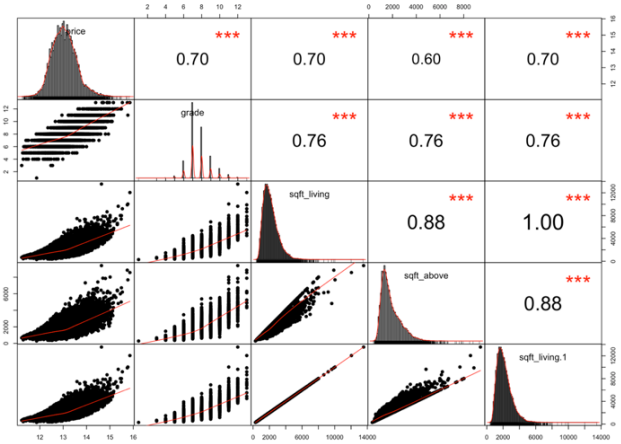

Figure 2: Partial Correlation Matrix Plot

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.

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

As for two ordinal variable "condition" (1-5) and "grade" (1-10), we do not need to do

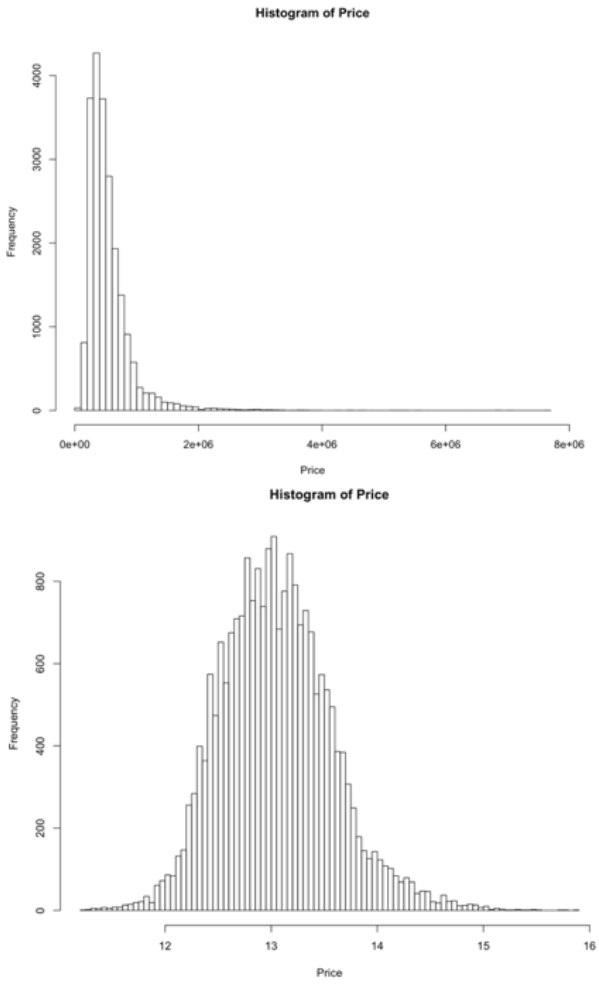

Figure 3: Histogram of Price

any transformation for them. Both 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. We could 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. The other option is to delete this variable.

2.3.2 Missing Value

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

There only one variable we should consider is yr_renovated(Year when the house was ren-

ovated). This dataset contains houses that have been renovated with value 0 or NOT have been renovated with value renovated year. It makes more sense that whether this house have been renovated than when the house was renovated. Hence, we translate this variable into dummy variable with value 0 (representing NOT been renovated) and 1(representing have been renovated)

3 Model

In this section, our team split our dataset 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 and boosting, thus random forest 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.

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.

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.

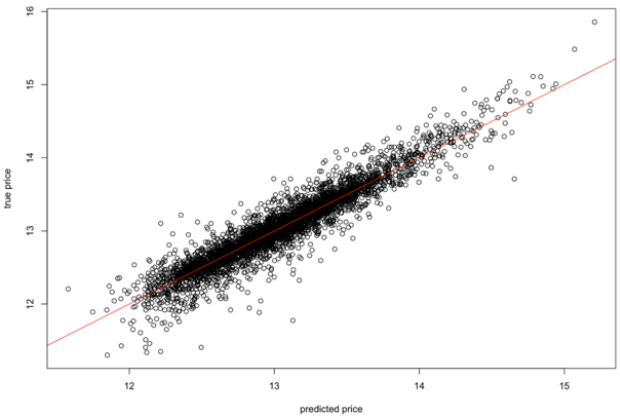

Figure 4: Prediction Result of Random Forest

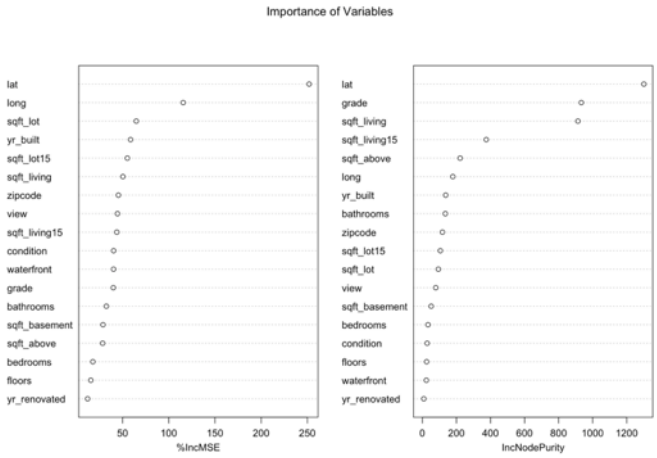

Figure 5: Importance of Variables

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 a well-performed model, our goal is to minimize:

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

After performing this model, we get MSE 0.06192281 and plot the predicted price and the true price.

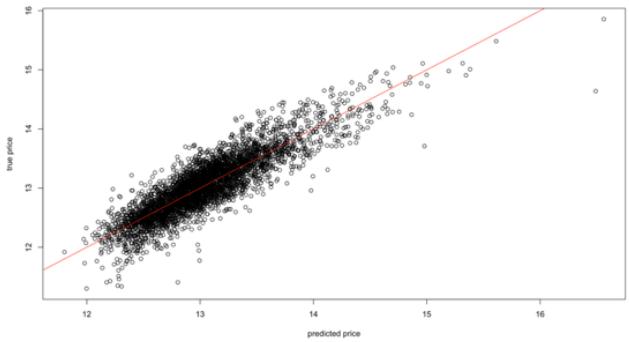

Figure 6: Quadratic Loss Function without Regularization

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.

4.2 Quadratic Regression with LASSO Regularization

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

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

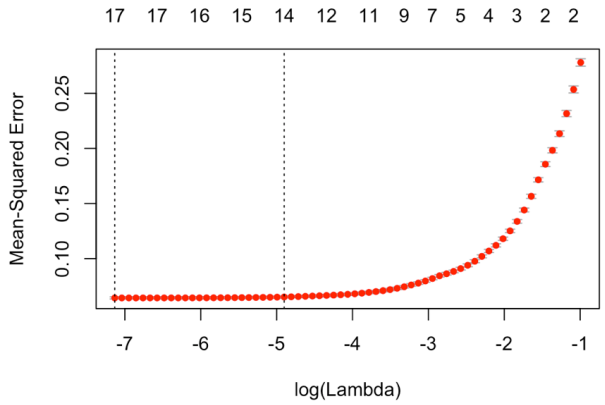

Figure 7: Result of Cross Validation

After performing the cross-validation process using 10 folds, it outputs the λ 0.0007979959 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.

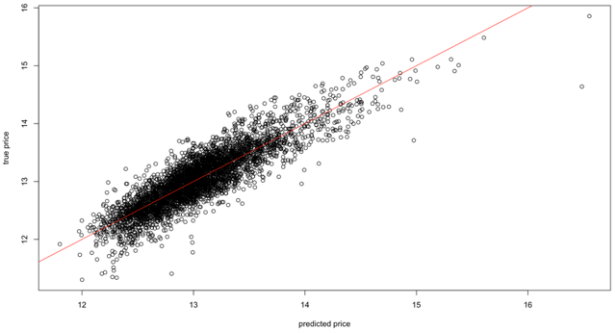

Figure 8: Quadratic Loss Function with Lasso Regularization

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

4.3 Quadratic Regression with Ridge Regularization

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

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

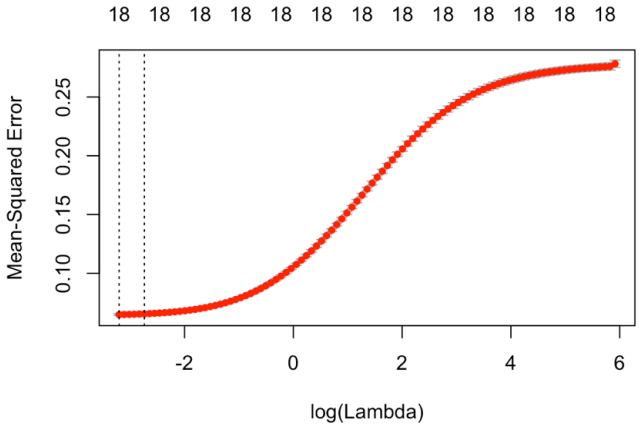

Figure 9: Result of Cross Validation

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

Compared to lass regularization, ridge regression also could shrink the coefficients, the difference is that the ridge regression cannot

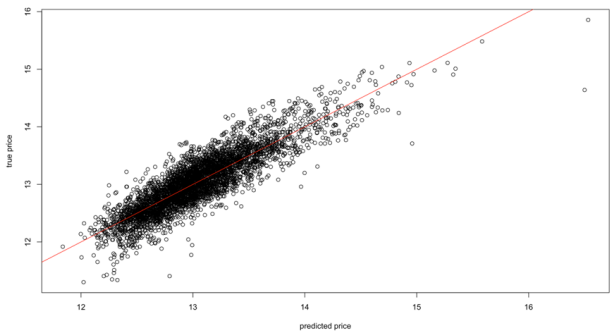

Figure 10: Quadratic Loss Function with Ridge Regularization

shrink the coefficients to zeros. The advantage is that it can leave and consider the effect of all variables.

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,

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

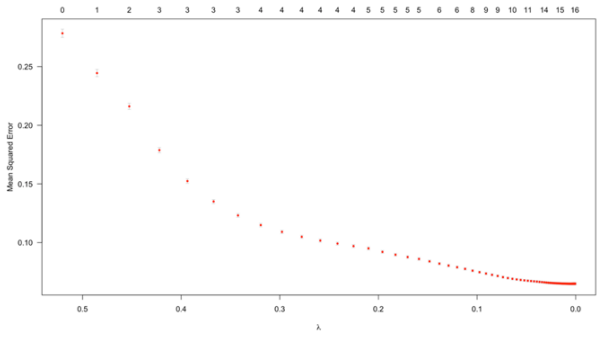

Figure 11: Result of Cross Validation

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

4.5 Huber Regression with Ridge Regularization

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

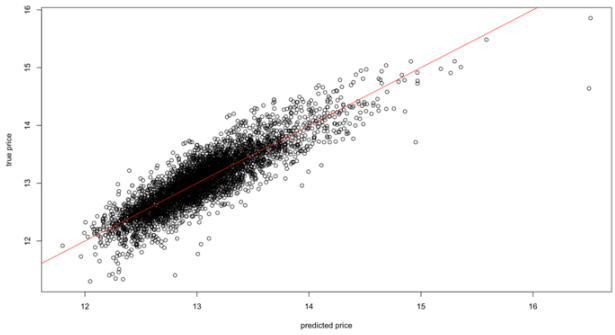

Figure 12: Huber Loss Function with Lasso Regularization

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

where,

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

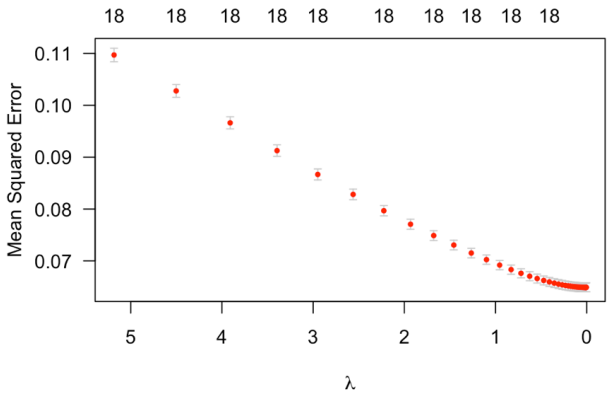

Figure 13: Result of Cross Validation

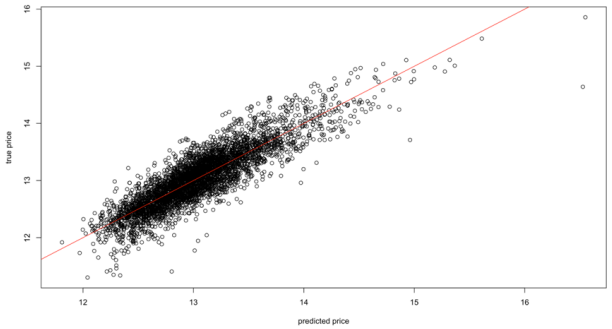

Figure 14: Huber Loss Function with Ridge Regularization

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

Methods		Best lambda	MSE
Random Forest Method		NONE	0.01390546
Quadratic Regression	No regularization	NONE	0.06192281
	Lasso	0.000797996	0.0619331
	Ridge	0.04065101	0.06217825
Huber Regression	Lasso	0.004526363	0.06243748
	Ridge	0.01390546	0.06246271

Figure 15: MSE Table

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

It is really strange that whether we use Huber or quadratic regression 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.

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 variables 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 and forward selection.