The prediction of house price

Abstract

1. Introduction of project

1.1. Background

House price is a continuously hot topic. In fact, a lot of factors should be taken into consideration if given this topic. Even though we seldom think about the type and material of roof or the height of basement ceiling, they indeed have impacts on the determination of the house price.

That is why we find the prediction of house price a really complicated problem and worth furthering. In this project, we would like treat it with methods from machine learning. Using a combination of regularization and xgboost along with neural network, we hope to perfect our models and predict the house price with minimal errors.

1.2. Description of data

The data was compiled by Dean De Cock for use in data science education(**?**), and published on the Kaggle competition. Here we have 79 explanatory variables describing (almost) every aspect of residential home in which 36 numerical variables and 43 categorical variables. There are 1460 examples in training set and 1459 in test set.

For the 79 variables, we can see the correlation between each two variables in the figure1. In this figure, the most interesting thing to get our attention is the four colored squares, the 'TotalBsmtSF' and '1stFlrSF' variables, the 'GarageCars' and 'GarageArea' variables, the 'GrLivArea' and 'TotRmsbvGrd' variables and the 'GarageYrBlt' and 'YearBuilt' variables. They are so highly correlated that maybe they give almost the same information. So when we do the data preprocessing, we keep just one of these variables. The other things we have interest to look is the variables highly correlated with the house price which is displayed in figure2.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

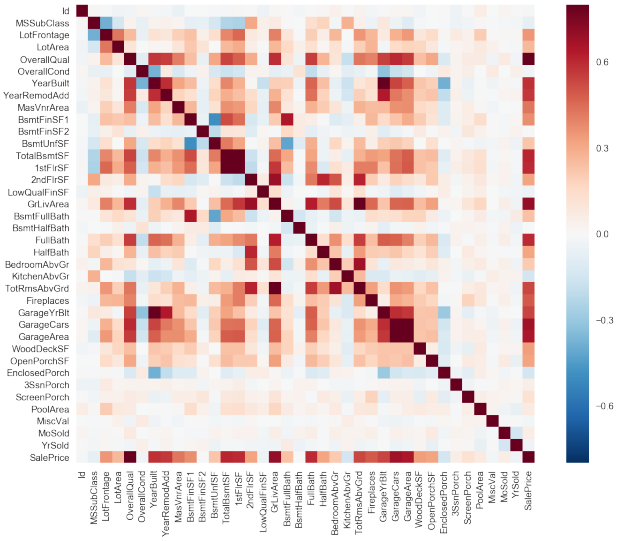

Figure 1. Heatmap

2. Data preprocessing

Data preprocessing aims to transform the raw data to produce outputs that can be smoothly used as inputs in data modeling.

2.1. Convert categorical variable to numerical variable

Many machine learning algorithms require numbers as inputs, it needs to be coded as numbers in some way. With the help of the function "get_dummies" in pandas packages, data frame is roughly converted from 79 columns to 288 columns.

2.2. Filling up missing values

Most algorithms require all variables to have values in order to use it for training the model. For the missing values in categorical variables, when we use the function "get_dummies", automatically it will add 0 to each column, meaning that no string used to describe this kind of feature. So there is no need to make some change for categorical variables. But when it comes to the numerical variables, we replace the missing values with the mean of their re-

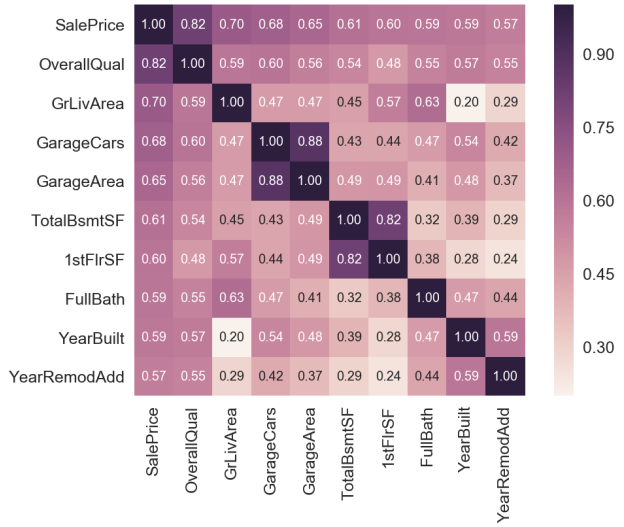

Figure 2. Top 10 variables correlated with sale price

spective columns.

2.3. Log transform of sale price

Log transform, it can transform data to ones that are single-peaked and symmetric. Its most important feature is that, relatively, it moves big values closer together while it moves small values farther apart. And it is easier to describe the relationship between variables when it's approximately linear. In order to treat the data of sale price up to hundred thousand dollars, log transformation works well in modeling.

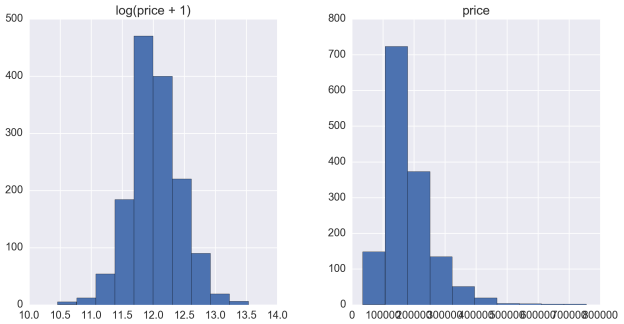

Figure 3. Histogram of sale price with/without log transformation

3. Data modeling

3.1. Regression with regularization

Regression analysis is a statistical process for estimating the relationships among variables and it is widely used in prediction and forecasting. In the simple linear regression model, the ordinary least squares(OLS) estimates are obtained by minimizing the residual squared error. There are two limitations of this method. The first is prediction accuracy: the OLS estimates often have low bias but large variance; prediction accuracy can sometimes be improved by shrinking or setting to 0 some coefficients. By doing so we sacrifice a little bias to reduce the variance of the predicted values and hence may improve the overall prediction accuracy. The second reason is interpretation. With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effects.(**?**)

In our problem, we have 79 variables. In order to avoid over-fitting in the analysis of high-dimensional data, we choose to use regularization methods: linear regression model with regularization (L-1 : ridge et L-2 : lasso). And a function has been defined to calculate Root-mean-square error (RMSE) for measuring the performance of each machine learning algorithms. The RMSE represents the sample standard deviation of the differences between predicted values and observed values:

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

3.1.1. Ridge regression

Ridge regression regularizes the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares,

$$\hat{\beta}^{ridge} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{P} \beta_j^2 \right\} \tag{2}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the degree of regularization, or we call it the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage.

The criterion in matrix form with parameter λ,

$$RSS(\lambda) = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \tag{3}$$

and the ridge regression solutions are easily seen to be

$$\hat{\beta}^{ridge} = \left(X^T X + \lambda I \right)^{-1} X^T y, \tag{4}$$

where I is the p × p identity matrix. The solution adds a positive constant to the diagonal of $X^T X$ before inversion. This makes the problem nonsingular, even if $X^T X$ is not of full rank, and was the main motivation for ridge regression when it is used**?**.

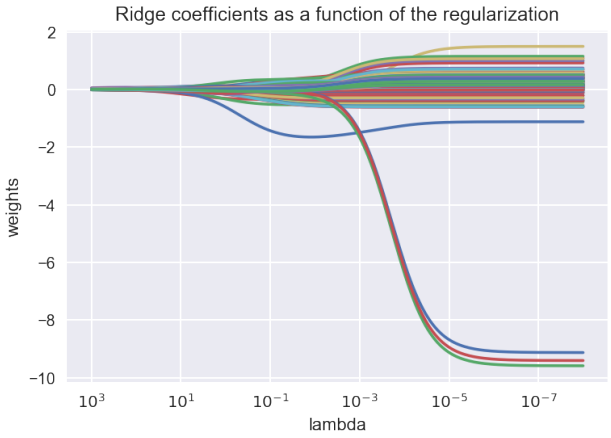

Figure 4. The values of β with the change of λ

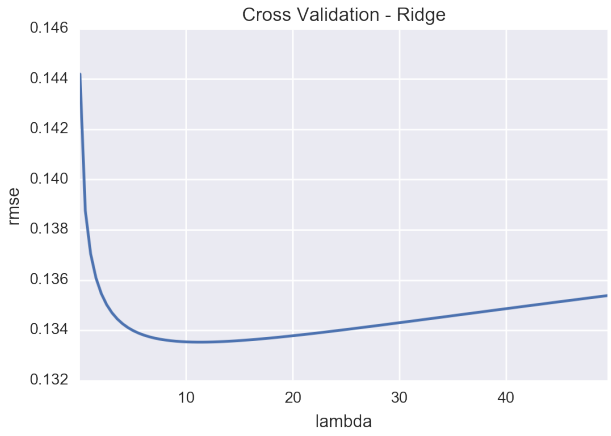

Figure 5. Cross validation of regression ridge

Figure4 and Figure5 show the changes of β and the changes of RSME with function of the effective degrees of freedom implied by the penalty λ. Their phenomena are closely related. When λ is too large, in Figure4 we can see that with the increase of λ, β values change little and they are all round 0. In fact, the change of RSME is also relatively small. That is to say, the shrinkage is so strong that the model trends to be under-fitting. However, when λ is too small, in Figure4 the weights of β vary a lot, at the same time from Figure5 the values of RSME also change a lot as λ increases, so the model is not sufficiently stable and it has a risk of over-fitting.

According to formula (1), we choose λ value = 10 is about right based on Figure5. Clearly, from Figure4, λ = 10 corresponds to the weights of β which are close to 0 while not equal to 0. The minimization value of $\hat{\beta}^{ridge}$ = 0.1358.

To sum up, compared to OLS model, ridge regression is a continuous process that shrinks coefficients and hence is more stable. However, it does not set any coefficients to 0 (based on Figure4) and hence does not give an easily interpretable model(**?**). So in next part, we explore the method of lasso regression.

3.1.2. Lasso regression

Lasso is also a shrinkage method like ridge but with a little difference, it uses the L1 penalty like the following:

$$\hat{\beta}^{ridge} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{P} \|\beta\| \right\} \tag{5}$$

As the lasso regression uses the absolute sum of the coefficients as the penalty item, we can shrink the coefficients to 0, which is illustrated in the figure 6. Suppose that we have 2 variables in our model, the solid green area are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$ respectively and the red ellipses are the contours of the least squares error functions. It is evident that for the lasso regression, we can easily shrink the coefficient to 0 while not that easy for ridge regression. As a result, we can select the variables in our model which is a really great advantage for lasso regression.

In our case, we have 79 variables in total, then after some preprocessing, especially the transformation of the categorical variables, we get 284 variables. Then we use the lasso regression. It selects 113 variables among the 284 variables. By using the cross validation with the fold 3, the β it has chosen is 0.0005 and the minimization of the error $\hat{\beta}^{ridge}$ = 0.1169 which is much better than the ridge regression. So we decide to use lasso regression rather than ridge regression. Afterwards, we display the coefficients of the 10 most positively correlated variables and 10 most negatively correlated variables in figure7. The variable 'RoofMatl_ClyTile' has the biggest negative coefficient meaning that the roof material on clay or tile has a great negative influence on house price. The next variable having the great impact is the variable 'MSZoning_C' which means that when the house locates in the commercial zone, it will decrease the house price. For the positive aspect, the most influential variable is 'Neighborhood-

StoneBr' indicating that these houses who are situated on the street 'Stone Brook' may be sold with a good price.

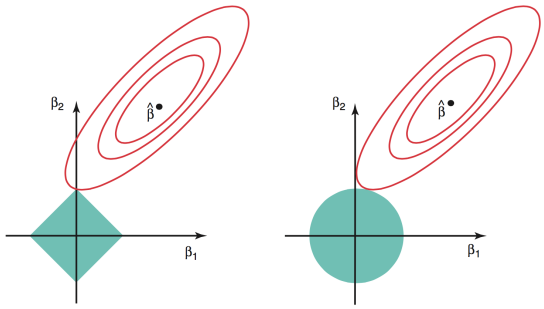

Figure 6. Estimation picture of the lasso(left) and ridge regression(right)

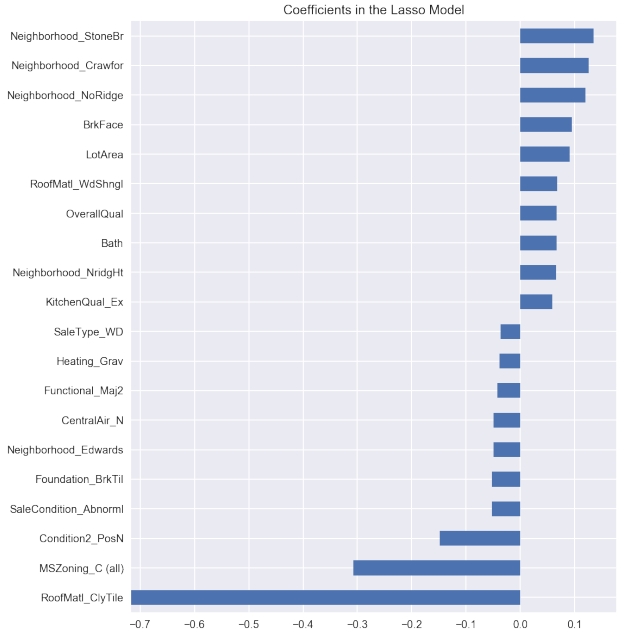

Figure 7. Coefficients in lasso model

3.2. Regression with principal component

3.2.1. Principal Component Regression(PCR)

Principal component regression is a technique for analyzing multiple regression data that suffer from multicollinearity, which arises when two or more of the explanatory variables are close to being collinear. when multicollinearity occurs, the variance of OLS are large so that they may be far away from the true value.

PCR is to perform a Principal Components Analysis (PCA) of the X matrix and then use the principal components of X as regressors on Y as a dimension reduction. The orthogonality of the principal components eliminates the multicollinearity problem.The ideal of PCR is to find k<p new variables to replace the initial p variables:

$$Y = X_1^* \beta_1^* + \cdots + X_k^* \beta_k^* + \varepsilon \tag{6}$$

Where X^* correspond to the principal components.

We start by performing PCA and perform 10-fold cross-validation to see how it influences the MSE in order to select the appropriate number of principal components(Figure 8). According to the figure, We see that the cross-validation error increase when M $>$ 60 components are used, and the smallest cross-validation error occurs when M = 52. As a result, we choose 52 principal components to do the regression and the MSE on the test data is 0.02648.

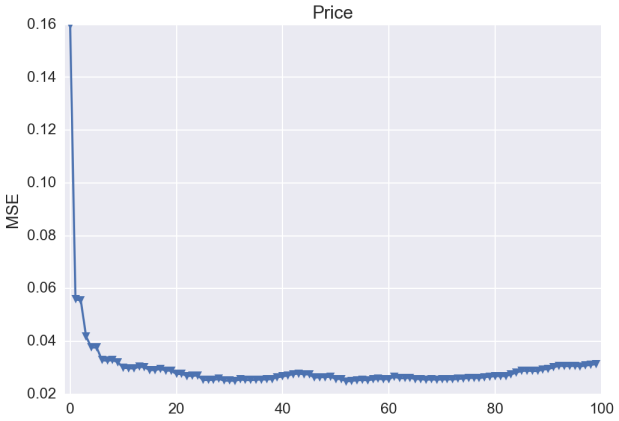

Figure 8. Number of principal components in regression PCR

3.2.2. Partial least squares(PLS)

Whereas in PCR the response variable, Y, plays no role in identifying the principle component directions, nothing guarantees that the principal components X^* are relevant for Y. By contrast, Partial least squares (PLS), finds components from X that are also relevant for Y. Specially, PLS regression searches for a set of components (called *latent vectors*) that performs a simultaneous decomposition of X and Y with the constraint that these components explain as much as possible of the *covariance* between X and Y.

We can find that the lowest cross-validation error occurs when only M = 5 partial least squares directions are used

from Figure 9. The corresponding MSE is 0.01927, much better than the model PCR, value of 0.02299. As the number of components augment from 5, the MSE increase slowly. When we perform the model on the testing data, the test MSE is 0.02938.

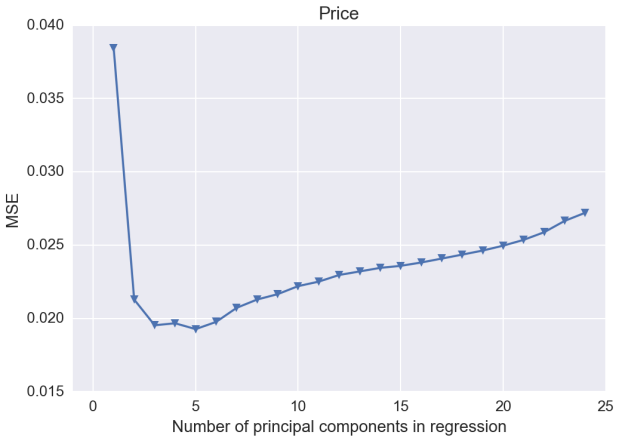

Figure 9. Number of principal components in regression PLS

3.3. Random forests model

Random forests(**?**)(**?**) operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of over-fitting to their training set(**?**).

As we have said in 3.1 and 3.2, lasso regression model has more advantages compared to PCR and PLS methods. So our random forests model is based on the selection of variables in lasso regression result. To apply random forest model, to begin with, we explore the theoretically accurate number of trees in the forest. Here the score of model is defined as the coefficient of determination (R square),

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \tag{7}$$

where SS_{res} is the residual sum of squares, SS_{tot} is the total sum of squares. So the more the value of R square is, the more precisely the model fit the training set. To avoid the risk of over-fitting when the number of tree is too large, we apply the method of cross validation. The result is shown in Figure10. And we choose the number of trees in forest equals 40 in order to maximize the value of R square. At that time, R^2 = 0.87 supports the accuracy of model for training set.

Afterwards, the model is fitted with training set and realize the predictions with it.What's more, we compare the prediction results between lasso modeling and random forests regression modeling and the result is shown in Figure11.

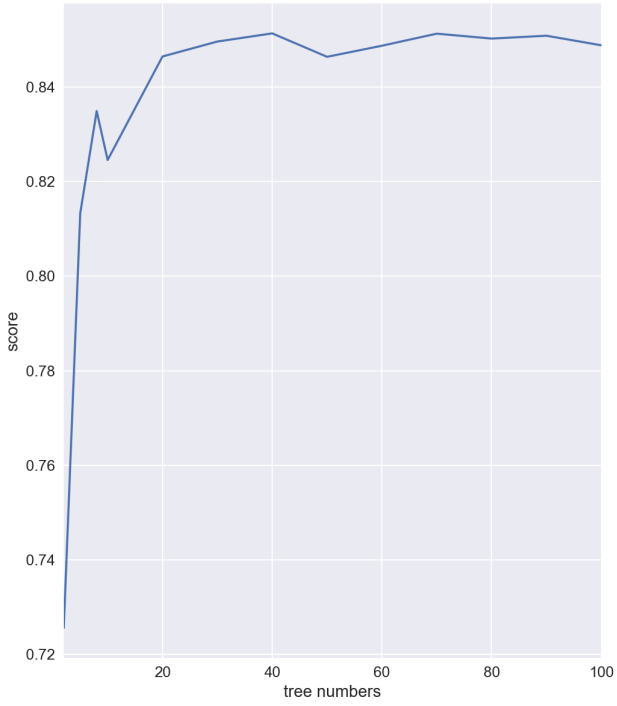

Figure 10. The scores of random forests regression modeling with changes in number of trees

Figure11. demonstrates the similarity degree between the two given models. Based on it, we can see that, when the prediction value is relatively small, the difference of the 2 models are quite small. However, when the value of prediction is much larger, especially from 300,000, the 2 prediction results trends to be dispersive. Given this phenomenon, for final prediction we will combine at least the two models together.

To go further in evaluation of the random forest model, we divide the original training set to 2 parts. The first part includes $\frac{2}{3}$ of the original training and it is set to be the new training set. The rest part is converted to be new test set with accurate value of house sale price. With this measure, we can evaluate more precisely and accurately the model. However, from Figure.12., we are faced with the problem of high bias in real value and prediction result. This figure shows our model is not so precise for testing. And we will continue to work on this point, like looking for the optimal parameter for model, etc.

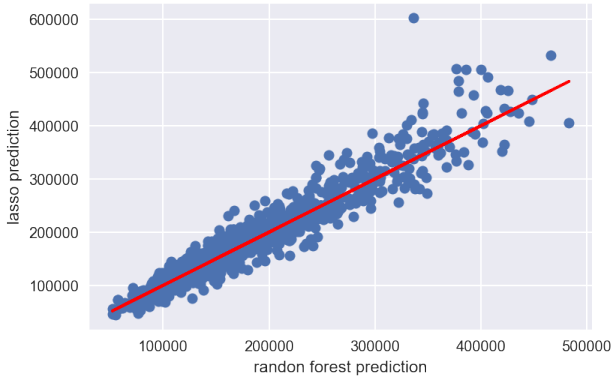

Figure 11. The comparison of prediction results between lasso modeling and random forests regression modeling

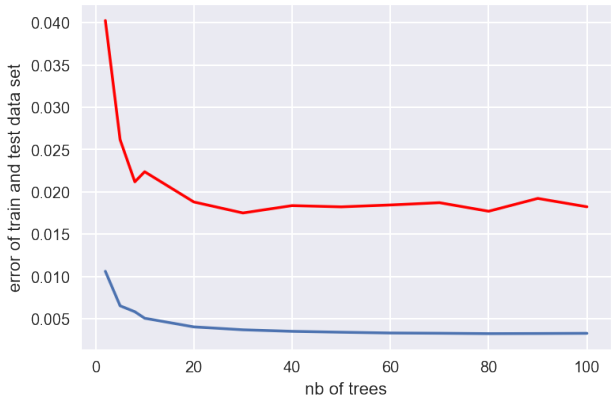

Figure 12. The comparison between the prediction results and the real values in test