Analysis of the Regression Model based on Boston Housing Dataset

Group CSTWY
Chloe Morrison
Shuman Tang
Ting Li
William Nolan
Yuwei Tie

May 11, 2019

1 Introduction

This problem requires us to build a data set based on the price data given by the Kaggle website, establish three regression prediction models and analyze the three models. This is a supervised learning problem. The data set for this problem comes from house prices in the surrounding area of Boston in the 1970s. The data set is very large. Except the ID, the training set has a total of 80 columns, of which 79 columns are independent variables and the last column is the training target (SalePrice). Therefore, the test set should have 79 columns (independent variables). There are some data loss parts and outliers in the data set. These values affect the results in the data set and need to be adequately processed. First, we performed an overall correlation analysis of the data.

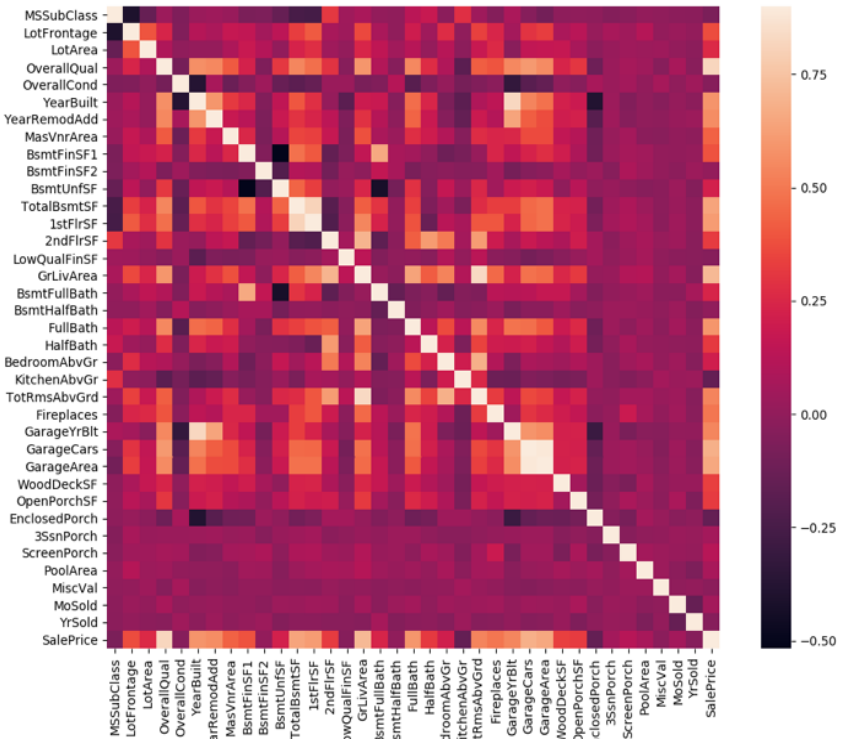

Figure 1: Data overall correlation analysis

It can be seen that some of the data have a greater impact on house prices (The lighter part). Then, we used the program to filter the ten variables that affect the final price.

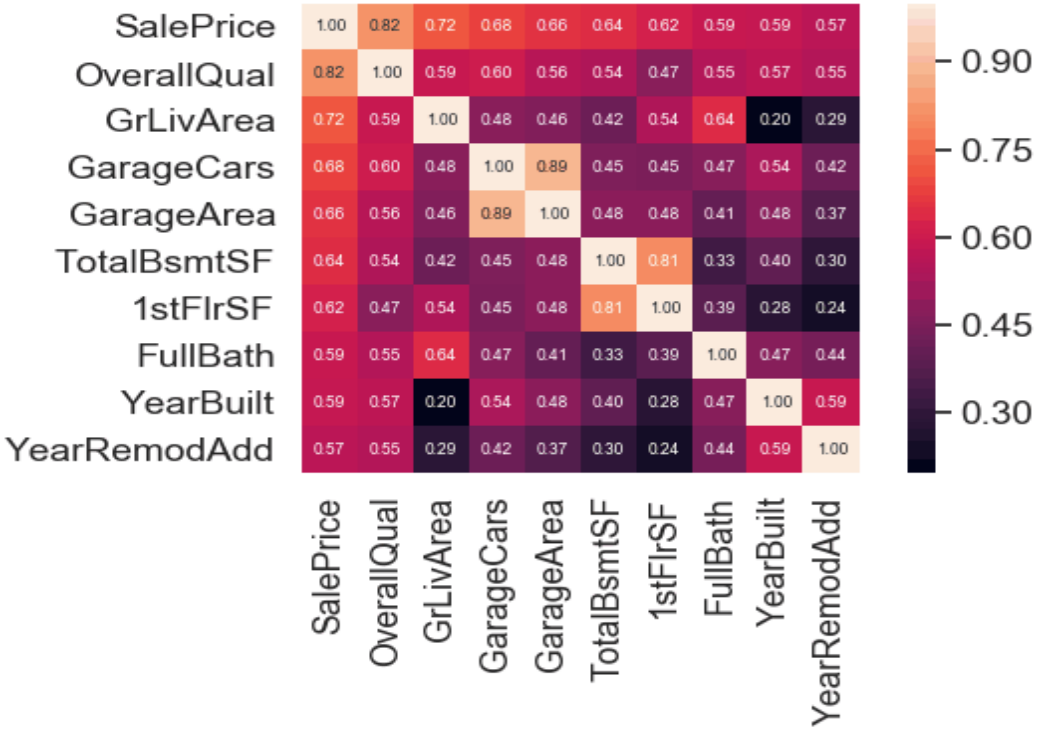

Figure 2: Ten variables that affect the final price

Based on these variables, we have a preliminary understanding of the data set. It can be seen from the figure that OverallQual, GrLivArea and TotalBsmtSF are strongly related to SalePrice. GarageCars and GarageArea only need one of the variables because they are similar. The main task of the training model is to fully fit the data. If there is an under-fitting condition, the training error will be larger; if there is over-fitting, the model may capture the special relationship between the variables, resulting in high variance and large test error. In training, we need to make a trade-off between bias and variance. In this experiment, we used three methods to predict the data. The mean squared error (MSE) and R-square predicted by the three methods are calculated separately. The smaller the value of MSE, the better the accuracy of the prediction model describing the experimental data. R-squared = SSR/TSS=1-RSS/TSS,it can measure the accuracy of the model. After sufficient data processing, all three models have a good performance. Among them, the accuracy of linear regression is about 78%, the accuracy of kernel regression is about 88%, and the accuracy of lasso regression is the highest, about 89%.

2 Data preparation

The House Prices Dataset is divided into a training set and a test set. The training set has 1460 sets of training data, and the test set has 1459 sets of test data, a total of 80 features, and 1 target type.

2.1 Data preprocessing

Since the Id feature has no effect on the prediction results, the Id column is removed from the training set and the test set. Therefore, there are 79 characteristics of the data, of which 36 are numeric types and 43 are category types.

2.2 Data cleaning

2.2.1 Outliers processing

The 1460 training data of 36 numerical type features are shown in graphical. By observing the scatter plot of each feature, the individual outliers are found and deleted in the data set. If the data is not particularly bad, the data is retained to ensure the generalization ability of the model. For example, the TotalBsmtSF feature. As shown in the figure below, the point in the lower right corner is obviously not in same rhythm with other points. This outlier means the house with the largest basement area but the house price is very low. There may be special circumstances, but this point belongs to the minority as an outlier which should be deleted first. After careful screening, a total of five features which have outliers were deleted, namely LotFrontage, BsmtFinSF1, TotalBsmtSF, 1stFlrSF and GrLivArea.

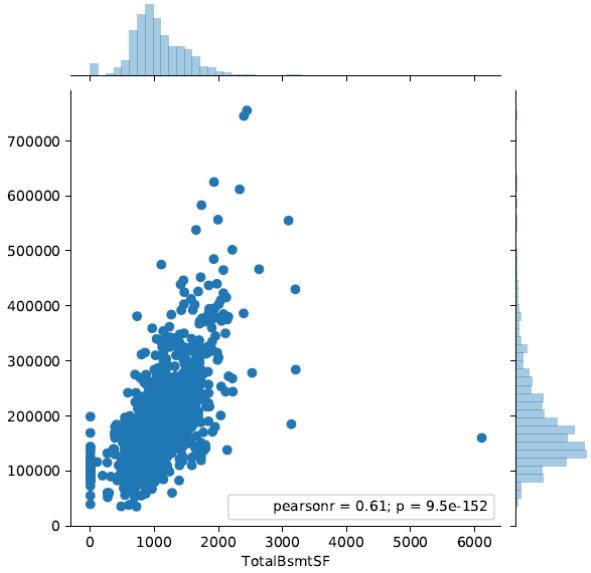

Figure 3: scatter plot of TotalBsmtSF

2.2.2 Normalization of target type

Because the linear model requires a normal distribution of the target value to maximize its effect, the normal distribution of the house price is made to be shown in graphical firstly. It can be clearly seen from Figure 4 that the dataset is biased to the left as a whole. And then it is known that it belongs to the right-biased distribution by the normal probability plot. Due to the large skewness of the data, the log conversion of the target value of the house price is required to restore the normality of the target value in order to make the error of the model as small as possible. The final result is shown in Figure 5. It can be seen that the house price is approximately normal distribution at this time.

Table 1: Data set feature missing rate

Feature	Total	Percent	Feature	Total	Percent
PoolQC	2909	99.69157	MasVnrArea	23	0.788211
MiscFeature	2813	96.40164	MSZoning	4	0.13708
Alley	2720	93.21453	BsmtHalfBath	2	0.06854
Fence	2347	80.4318	Utilities	2	0.06854
FireplaceQu	1420	48.66347	Functional	2	0.06854
LotFrontage	486	16.65524	BsmtFullBath	2	0.06854
GarageCond	159	5.448938	BsmtFinSF2	1	0.03427
GarageQual	159	5.448938	BsmtFinSF1	1	0.03427
GarageYrBlt	159	5.448938	Exterior2nd	1	0.03427
GarageFinish	159	5.448938	BsmtUnfSF	1	0.03427
GarageType	157	5.380398	TotalBsmtSF	1	0.03427
BsmtCond	82	2.810144	Exterior1st	1	0.03427
BsmtExposure	82	2.810144	SaleType	1	0.03427
BsmtQual	81	2.775874	Electrical	1	0.03427
BsmtFinType2	80	2.741604	KitchenQual	1	0.03427
BsmtFinType1	79	2.707334	GarageArea	1	0.03427
MasVnrType	24	0.822481	GarageCars	1	0.03427

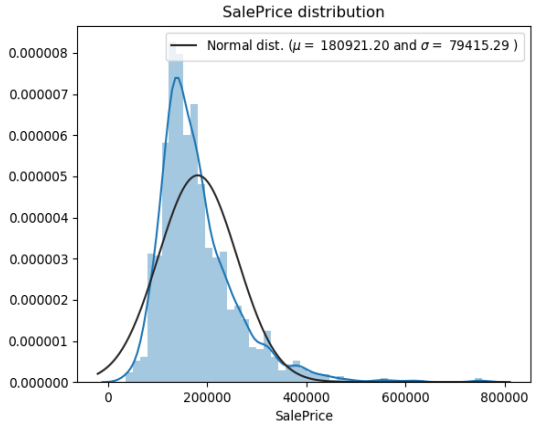

Figure 4: before log conversion

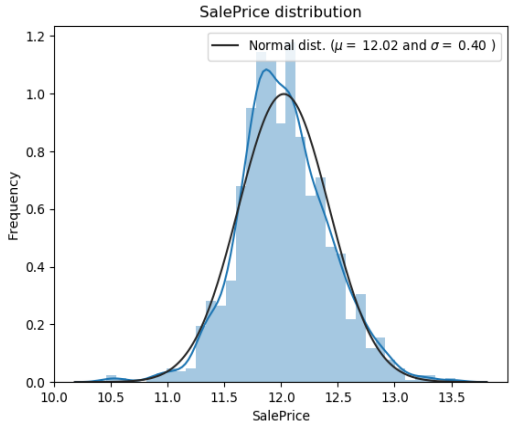

Figure 5: after log conversion

2.3 Missing value processing

In order to perform feature missing statistics better, the data of the training set and the test are combined together. There were originally 2919 sets of data, but seven sets of data have been deleted after the processing of outliers. So only the remaining 2912 sets of data will do missing values processing. First, the missing data of the various features in the overall data set are counted, and a total of 34 features have missing values.

2.3.1 Category feature

It is firstly to consider the feature with a large miss rate and fill the vacancy position of the category feature with NA. In this dataset, features with the number of missing greater than 1000 can be filled with null values. These features shown in Table 1 are PoolQC, MiscFeature, Alley, Fence and FireplaceQu.Then, it is to process the features with a missing rate greater than 1% and these features are similarly filled with NA. The remaining features have missing rate less than 1%. Since there are only a few missing or even one or two, the missing values location are filled by the most attributes of this feature. For example, the

GarageCars feature only has one value missing and 2 of this feature appears the most, so this blank value is filled with 2. In addition, for the Utilities feature, it has only two categories in which the category NoSeWa only appears once and the rest are all AllPub categories. Therefore, the research of this feature is of little significance and delete the column directly.

2.3.2 Numeric type feature

The LotFrontage feature is processed first. Because the Linear feet of street connected to property may be similar to other houses in the neighborhood, the median value of the Neighborhood feature is used to fill the missing data. This is followed by features with a missing rate greater than 1%, which are directly filled with zeros. The remaining feature missing values of less than 1% are also filled by the most attribute.

2.4 Feature coding

2.4.1 Numeric type feature

Some data type features are actually category-type features. For example, OverallCond is a feature that evaluates the overall condition of the house. It gives ten digits of one to ten, but it is actually a category. So it is converted into a string category and there have other 13 features belonging to this type. For the remaining 22 numerical features, it is firstly to calculate their data distribution skewness, as shown in Figure 6. Log conversion is performed on features that do not conform to the normal distribution so that they conform to the normal distribution.

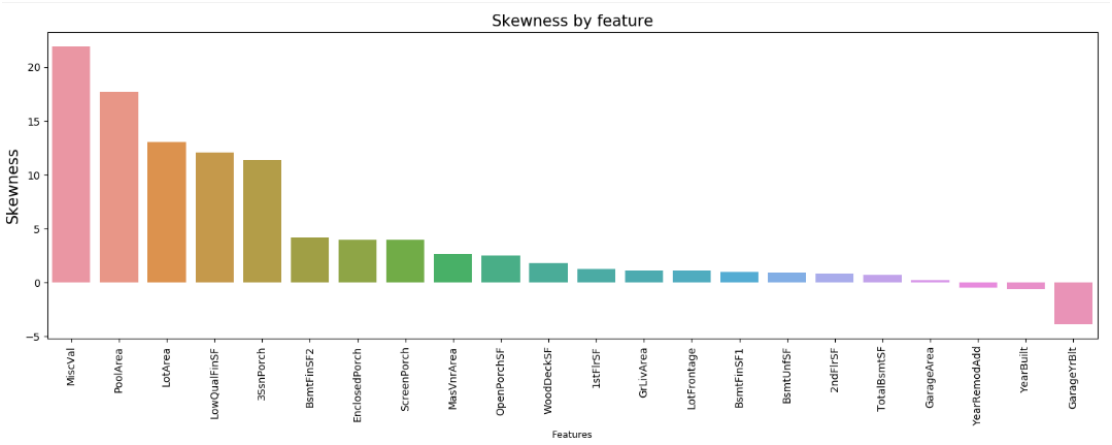

Figure 6: Numerical feature

2.4.2 Category feature

For category features, one-hot encoding is used. Because if the classification is directly represented by numbers, there may be a problem that an invisible size sequence is added to the category. For example, if 1 is used to behalf men and 2 behalf women. 2 is large than 1,but actually it should be the same status in gender description. Furthermore, different values in the process of model training may make the weight of the same feature in the sample change. If the feature is directly encoded into 1000, it is more affected than encoded 1 to model. In order to prevent the negative impact on the model caused by the problem represented by the classification value during the training process, the one-hot encoding is introduced to uniquely encode the feature of the category type. In this data set, in addition to the original 43 category-type features, 14 data-type features of stringification should be added.

2.5 Data set division

After all the data has been processed, the training set and the test set are re-separated. The original test set is then divided into 60% training samples and 40% test samples using the train_test_split function.

3 Predictor

3.1 Linear regression

Every sample of the House Prices datasets has several different features. Therefore the data is predicted by multiple linear regression. Multiple linear regression means that for a sample i with n features, its regression results can be written as follows:

$$\hat{y_i} = w_0 + w_1 x_{i1} + w_2 x_{i2} + ... + w_n x_{in}$$

w is collectively referred to as model parameters, where w_0 is called intercept and $w_1 ... w_n$ is called regression coefficient. y is the target variable, which is the label. $x_{i1} ... x_{in}$ is a different feature on sample i. For data with m samples, it can be expressed as:

$$\begin{bmatrix} \hat{y_0} \\ \hat{y_1} \\ \hat{y_2} \\ ... \\ \hat{y_m} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} \end{bmatrix} * \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ ... \\ w_m \end{bmatrix}$$

$$\hat{y} = X * w$$

Define the cost function. The cost function is used to describe the difference between the linear regression model and the actual data. If there is no difference at all, then this linear regression model fully describes the relationship before the data. To find the best fit linear regression model, the cost function is to be minimized. The relevant formula is described as follows:

$$\sum_{i=1}^{m} (y_i - \hat{y_i})^2$$

$$\sum_{i=1}^{m} (y_i - X_i * w)^2$$

y_i is the real label of the sample i, and $\hat{y_i}$ is the prediction label of the sample i under the next set of parameters w. The cost function calculates the distance between the true label and the predicted value. It measures the difference between model predictions and real labels. Therefore the solution goal is transformed into a minimized the cost function:

$$\min_{w} {||y - X * w||_2}^2$$

Next, the minimized residual sum of squares is introduced in order to solve the minimum of the cost function. The minimized residual sum of squares (RSS) between the true and predicted values will be find.

The RSS is then derived from the parameter vector w. The solution process is as follows:

$$\begin{aligned}
\frac{\partial RSS}{\partial w} &= \frac{\partial ||y - Xw||_2{}^2}{\partial w} \\
&= \frac{\partial (y - Xw)^T (y - Xw)}{\partial w} \\
&= \frac{\partial (y^T - X^T w^T)(y - Xw)}{\partial w} \\
&= \frac{\partial (y^T y - X^T w^T y - y^T Xw + w^T X^T Xw)}{\partial w} \\
&= \frac{\partial y^T y - \partial X^T w^T y - \partial y^T Xw + \partial w^T X^T Xw}{\partial w} \\
&= 0 - X^T y - X^T y + 2X^T Xw \\
&= X^T Xw - X^T y
\end{aligned}$$

The optimal solution of the parameter w is such that the first derivative after derivation is zero. Because the feature matrix X is not a matrix with all elements being zero. Therefore $X^T X$ will not be 0. Continue to derive:

$$\begin{aligned}
X^T Xw - X^T y &= 0 \\
X^T Xw &= X^T y \\
w &= (X^T X)^{-1} X^T y
\end{aligned}$$

In this case the minimum value of w can be obtained. There must be a linear relationship between the feature and the dependent variable. Therefore linear regression is very sensitive to outliers. It can seriously affect the regression line and the final predicted value.In multiple linear regression, higher order polynomials are used to fit lower degree polynomials. This error can lead to overfitting.

3.2 Kernel ridge regression

Ridge regression is similar to linear regression. To some extent, the over-fitting problem of linear regression can be solved. Ridge regression adds a regularization term to the cost function. The cost function of the ridge regression is as follows:

$$\sum_{i=1}^{m} (y_i - X_i * w)^2 + \lambda \sum_{j=1}^{n} {w_i}^2$$

Where λ is called the regularization parameter. If the selection of λ is too large, all parameters ω are minimized, resulting in under-fitting. If the selection of λ is too small, it will lead to improper resolution of the over-fitting problem.

Kernel ridge regression is to add the Kernel trick Φ on the ridge regression. The feature X will be calculated:

$$X = \begin{bmatrix} \Phi(x_0) \\ \Phi(x_1) \\ \Phi(x_2) \\ \cdots \\ \Phi(x_m) \end{bmatrix}$$

Therefore the solution goal is transformed into a minimized the new cost function:

$$\min_{w} ||y - \phi(X) * w||_2^2$$

Because Kernel Ridge Regression uses the Kernel trick. It is flexible and suitable for complex fittings. In this way, the complexity of the calculation is related to the amount of data. This method is not suitable if the amount of data is large.

3.3 Lasso regression

Lasso regression adds a regularization term to the cost function. The cost function of Lasso regression is as follows:

$$\sum_{i=1}^{m}(y_i - X_i * w)^2 + \lambda \sum_{j=1}^{n} |w_i|$$

Therefore the solution goal is transformed into a minimized the cost function:

$$\min_{w} ||y - X * w||_2^2$$

Lasso regression is able to train some parameters with smaller features to zero. Then get a sparse solution. In other words, in this way, the purpose of dimension reduction (feature screening) is achieved in the process of training the model. When lasso regression is called in sklearn, the parameter alpha is λ.

4 Evaluation

For regression problem, we have 2 model evaluation approaches here such as train and test on the same dataset and train/test split. We used the second method to build prediction model and test it because the first method may caused high training accuracy and out-of-sample accuracy, which was caused by over-fitting. In addition, we have 3 main methods as evaluate and compared metric:

(1) Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n} ln \sum_{i=1}^{n} |y_i - \hat{y_i}|$$

(2) Mean Squared Error (MSE) is the mean of the squared errors:

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

(3) Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

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

We observe the three formulas above, because the error shown by using MSE is larger, which can better reflect the accuracy of the model, we choose MSE as the metric of model prediction accuracy.

After several tests, the prediction results of the three regression models are very stable. The MSE and accuracy values of the test results are shown as table 2. It can be clearly seen from the table that the MSE of Kernel ridge Regression and Lasso Regression are smaller and their prediction accuracy is more higher than Linear Regression.

Table 2: Regression MSE and Accuracy

	MSE	Accuracy(%)
Linear Regression	0.0335	78%
Kernel ridge Regression	0.0178	88%
Lasso Regression	0.0156	89%

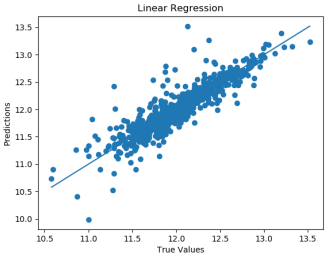

Figure 7: linear

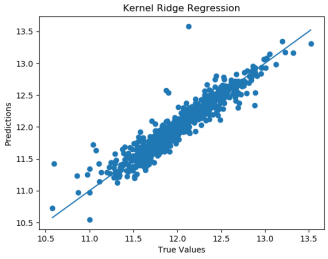

Figure 8: Kernel ridge

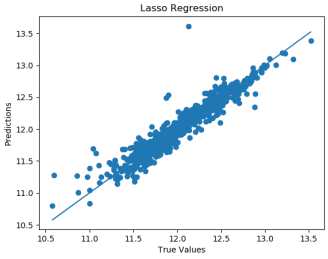

Figure 9: Lasso

The figure above shows the distribution diagram of the results of the linear Regression model and Kernel ridge Regression. In the figure, the ordinate is the predicted result and the abscissa is the actual value, which means that the more coordinate points are concentrated on the oblique line y=x, the more accurate the prediction will be. The figure below shows the comparison of partial predicted values and actual values of the three regression models.

In a word, the method for evaluation is to calculate the MSE value and accuracy. By comparing these values, it can be seen that the prediction accuracy of final Kernel ridge Regression and Lasso Regression is better than Linear Regression. At the same time, the distribution of correlation between the predicted value and the actual value observed in the picture also shows that Linear Regression is looser than Kernel ridge Regression.

Figure 10: Regression result comparison

5 Conclusion

Here we summarize two main results: the modeling effect and the main factors affecting housing pricing.

Firstly, in this report, we established three predictor models: Linear Regression, Kernel ridge Regression and Lasso Regression. And there are several differences among them during the process of modeling which is mainly shown in parameters' setting.

For Linear Regression, there is no special parameter which should be defined in training.

Then, for Kernel ridge Regression, alpha is set as 2, and coef0 is 1 which means the coefficient of the constant term is 1 in sigmoidkernel. Degree is 3 which is a parameter for polynomial kernel. And kernel function is linear.

While for Lasso Regression, there are many parameters, and two of the main parameters are alpha and the maximum number of iterations which are set as 0.0007 and 1000 respectively in this model.

The above descriptions are the differences of the three models in setting parameters. What's more, the size of training set of the three models is all 1460. There are 80 columns in the training set, 79 of which are independent variables and the last one is training target (sales price). Its format is csv and it is saved as tarin.csv.

And finally, we test these three models in the test set, they obtain different accuracy which can be shown as follows:

Table 3: Accuracy of different models

Models	Accuracy(%)
Linear Regression	78.43%
Kernel ridge Regression	88.53%
Lasso Regression	89.95%

As shown in the accuracy table above, we can conclude that the Lasso Regression has the best performance in the prediction of house sale prices, as it has the highest accuracy among three models which is 89.95%. While Linear Regression has the worst result, as it has the lowest accuracy among three models that is 78.43%.

Secondly, we also conclude that there are some factors that have a great impact on pricing house to a certain extent. These are analyzed from dataset: OverallQual, GrLivArea and TotalBsmtSF can be dominant in determining a house's price.

These three factors mean the overall material and finish of the house, Above grade (ground) living area square feet, and Total square feet of basement area respectively.

Therefore, if one of the three factor is worse, it will lead to lower house prices, for example, like bad overall materials, small living area above the ground or small basement area.

In contrast, if the overall material and decoration of a house is better, the larger the living area above the ground and the larger the total area of the basement, the house may also have higher pricing.