House Prices: Advanced Regression Techniques

The Trung Le

A1784927

School of Computer Science, The University of Adelaide

Abstract

"House Prices: Advanced Regression Techniques" is a famous problem in Kaggle. In order to solve this problem, I analyzed the dataset, handled the missing value, transforming the categorical variables into numerical variables, log-transforming the target variable to reduce the skewness problem. After that, I splitted the train dataset into train set and validation set, then I built Linear Regression model and Random Forest Regression model and applied into these two sets. Lastly, I evaluated the results of these two models on both sets and chose a final model to apply into test dataset, then submitted the result into Kaggle. For this dataset, Linear Regression model worked better on train set but worse on validation set compare to Random Forest Regression model. Hence, the final model I chose to apply on test dataset is Random Forest Regression model and it achieved the RMSE of 0.1479 on Kaggle.

Introduction

These days, the demand of buying a house is increasing worldwide since the population in big cities is rising dramatically. However, many people do not understand about what factor define the price of a house and they want to buy a good house with reasonable prices. Furthermore, several real estate agents want to buy a house with lower price than its' actual price. Hence, a model which can predict the price of a house based on that house's features is essential for those customers.

In this project, I analyzed the housing data and build a model that can predict the sale price of a house. I used the dataset of "House Prices: Advanced Regression Techniques" problem in Kaggle[1].

House Price Database

This data set is made by Dean De Cock of Truman State University in 2011 and it is available in Kaggle. This problem has 1460 data in train dataset and 1459 data in test dataset. There are 80 features variables including Id of a house and 79 features of that house. There is 1 target variable – the Sale Price, that is available in train dataset.

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

Implementation Step

In order to solve this problem, I followed 4 steps:

Step 1: Data analysis – analyzing the dataset, understanding the construct of dataset, finding the missing values of the dataset and the reason of missing values, exploring about the target variable.

Step 2: Data preprocessing – handling the missing value, transforming the categorical variables into numerical variables and transforming the target variable to reduce the “skewness” problem.

Step 3: Build models – building models using the train dataset to predict the target variable.

Step 4: Evaluation – Comparing and evaluation the result of each model, then choosing the best model.

Data Analysis

The aim of this problem is using the features of a house to predict the Sale Price of that house. Since Sale Price is a continuous variable, this problem is a regression problem.

There are 81 variables in train dataset (including the “target” variable – “Sale Prices” and the “Id variable) and 80 variables in test dataset (excluding “Sale Price” and including the “Id” variable).

In the train dataset, there are 37 numerical variables and 43 categorical variables (excluding the “Id” variable). Over 37 numerical variables, the most five variables which have the highest correlation with the “Sale Price” are: OverallQual (0.791), GrLivArea (0.709), GarageCars (0.640), GarageArea (0.623) and TotalBsmtSF (0.613). Exploring deeper about the correlation between these five variables and Sale Price, it showed that when OverallQual, GrLivArea, GarageArea and TotalBsmtSF increased, the Sale Price of that house increased. Figure 1 below contains the scatter plots of these five variables and Sale Price.

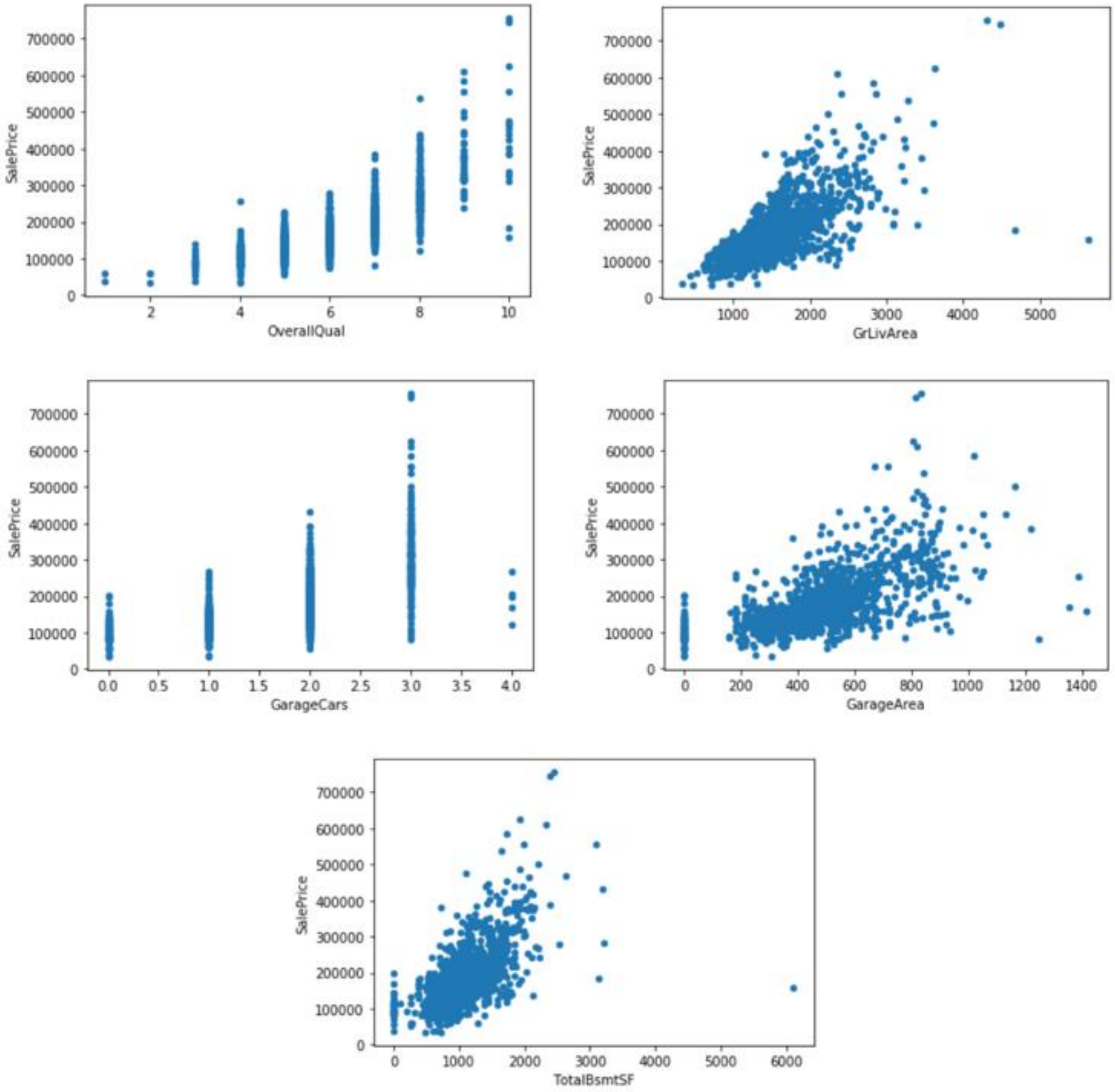


Figure 1 Scatter Plot between five variables and Sale Price

In the dataset, there are 19 features which have missing values, including 3 numerical features (LotFrontage, MasVnrArea and GarageYrBlt) and 16 category features. Both these features with missing values have relationships with other features. For example: if a house has no pool, the PoolQC (Pool Quality) of this house is NA (a missing value) and the PoolArea is 0. If a house has no garage, both the GarageType, GarageYrBlt, GarageFinish, GarageQual and GarageCond of this house are NA. This relationship also explains the reason why these five features have the same number of missing value (81 in train dataset). The feature with the most missing value is PoolQC (1453 in train dataset) – it means that almost houses do not have a pool.

About the target feature of the dataset – Sale Price. From the analysis, the minimum value of Sale Price is 34900 and the maximum value is 755000. However, the majority of Sale Price is from 130000 to 210000, with the median value of Sale Price is 163000 and the mean value of Sale Price is 180921. Hence, the distribution of Sale Price is right-skewed. The histogram plot of Sale Price represented this problem (figure 2 below)

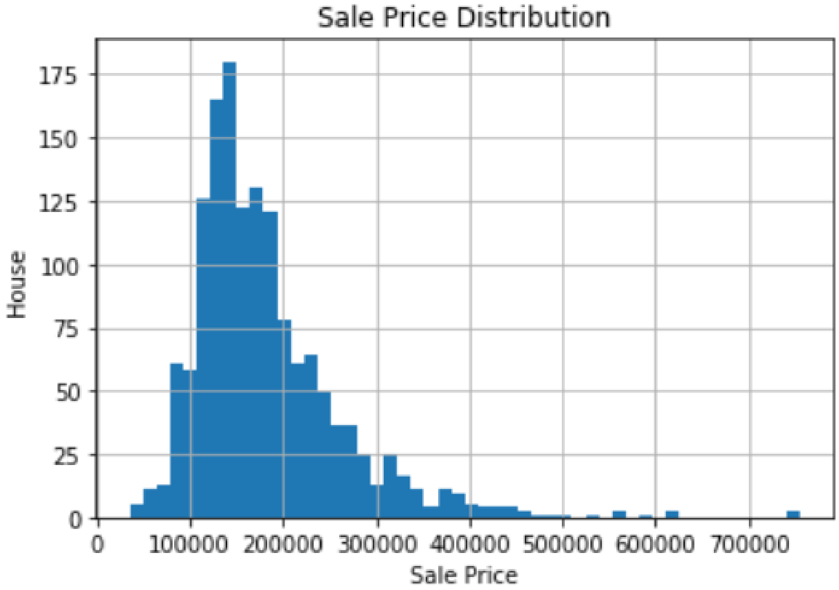


Figure 2 Histogram plot of Sale Price

Data Preprocessing

Firstly, I handled the missing values. All the missing values in this dataset existed due to the reason that the house does not have the object containing these variables. Hence, for category features, both missing values are replaced by “None”. For numerical features, both missing values are replaced by 0. For example, I replaced both the missing values of GarageType, GarageFinish, GarageQual and GarageCond by “None” because these houses do not have a garage.

Secondly, I handle the categorical features. In order to use regression models, both features are required to be numerical features. Furthermore, the unique values in each categorical feature are not in order – they do not order how the unique values affect the Sale Price. Hence, for every category feature, first I ordered the unique values in each categorical feature by compare the median value of Sale Price of houses that have these unique values. After that, I transformed the unique values after ordering into number, start from 0.

Lastly, to solve the “skewness” problem of Sale Price variable, I used log transformation on Sale Price. This method can reduce the right – skewed distribution of Sale Price. Figure 3 show the histogram plot of Sale Price after using log transformation.

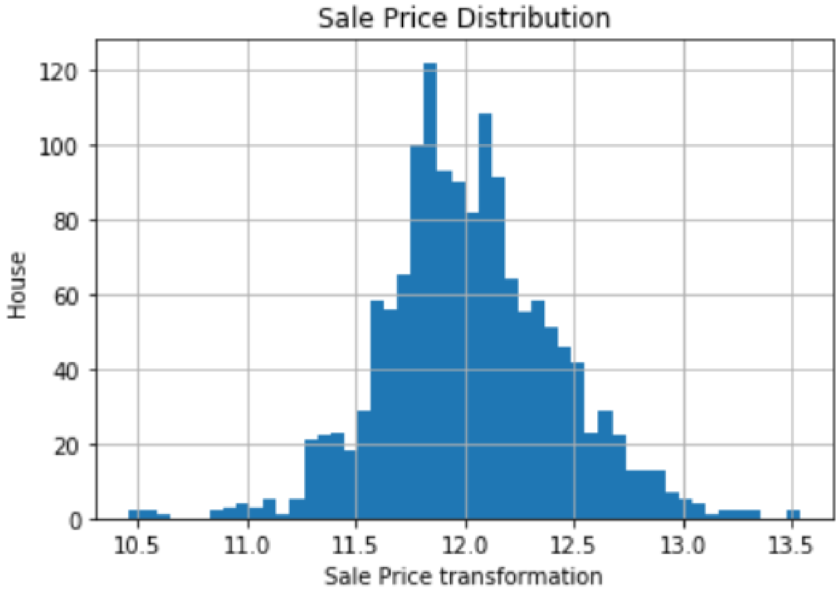


Figure 3 Histogram plot of Sale Price after log transformation

Building Models

Before building models, I split the train dataset into 2 set – train set and validation set. Train set contains the first 70% data of train dataset and validation set contains the last 30% data of train the dataset. The reason of this splitting is I wanted to use the train set to train the models, then used validation set to evaluate the models.

For this dataset, I chose two models. The first model is Linear Regression. This is a simple model that works well when dealing with regression problem. Compare to Decision Tree Regression model, Linear Regression model does not have “overfitting” problem (high variance but high bias). The main drawback of this model is “underfitting” problem (low variance). The second model is Random Forest Regression. The reason I choose Random Forest Regression model is I want to keep the strength of the Decision Tree Regression model (high variance) and improve the weakness of that model (low bias). Random Forest Regression model is the model that have several Decision Trees. Each tree used a part of the training data. The predicted value is the average of the values provided by these Decision Tree. As the result, that model prevents the "overfitting" problem of the Decision Tree Regression model.

Evaluating

For this dataset, I evaluated the results of these two models by using root-mean-squared-error (RMSE). The 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 [2]. I also performed 10-fold cross validation on each model to reduce the risk of over-fitting.

The result is represented on table 1 below:

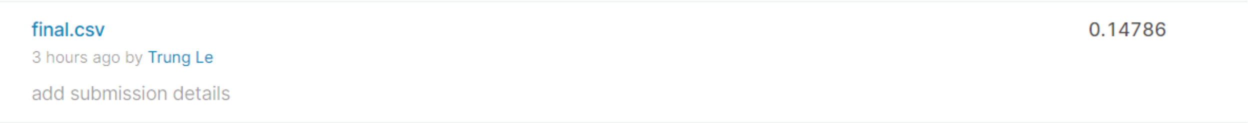
Table 1 RMSE results of 2 models

	Linear Regression	Random Forest Regression
Train set (10-fold cross validation)	0.1291	0.1434
Validation set	0.1551	0.1457

From this result, it can be saw that the Linear Regression model performed better on train set, but worse on validation set compare to Random Forest Regression model. Hence, I chose Random Forest Regression model as my final model.

After that, I applied my Random Forest Regression model on the test dataset. Firstly, I handled the missing values and transformed the categorical variables into numerical variables in the test dataset like I did in the train dataset. After that, I fitted my Random Forest Regression model on the test dataset and got the prediction. Lastly, I transformed the prediction back to normal (because the prediction is already log-transformed) by using exponential transformation.

After applying Random Forest Regression model on the test dataset and submitted the result into Kaggle, the RMSE of my model is 0.1479, as show in figure 4 below.



final.csv

3 hours ago by Trung Le

add submission details

0.14786

Figure 4 Kaggle Result

References

Kaggle. House Prices: Advanced Regression Techniques. URL: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

[2] https://en.wikipedia.org/wiki/Root-mean-square_deviation