IBM House Price Prediction

Chunso(Eddie) Park

March 2020

1 Introduction

As house prices have increased dramatically in most regions in the world for past few years, people's attention has been gathered to the house price. Some people consider the skyrocketing price of real estate as a profitable investment, others who need their house desperately suffer from the current status. This project intends to analyze the various factors that affect the price using data pre-processing and Multiple Linear Regression Analysis. Final goal is to find a relationship between features and the price and predict price using given value of features and Regression model.

2 Exploratory Data Analysis

I have used data from Kaggle's House Prediction competition. The train dataset contains 79 explanatory features and SalePrice. However, the test dataset only contains 79 features which makes us to predict the Saleprice using values in features. Let's use visualization and pre-processing technique we learned from IBM Data Science Professional course to explore the training set.

2.1 Correlation

To predict the price using regression analysis, we have to find a correlation between each feature and the target value, Saleprice. the top 10 most correlated features are OverallQual, GrLivArea, GarageCars,GarageArea, TotalBsmtSF, 1stFlrSF, FullBath,TotRmsAbvGrd, YearBuilt, YearRemodAdd. It makes sense since the overall quality, the living area and other features related to the size of the house are the most common criteria in differentiating price between houses. Then are those 10 features enough to predict the house price? Let's train a regression model using those features and predict the SalePrice.

2.2 First Submission

Since values in the feature have different range, first standardize the value to produce a better prediction using PolynomialFeatures function and fit tranform function from sklearn. Secondly, we split the train data to train our regression model. Then, we train the regression model using splited data and finally insert the test data given by Kaggle to finish our prediction. Kaggle uses Root Mean Squared Logarithmic Error to evaluate the performance of our predicted values. After submitting the predicted prices, We get a result of 0.17982 for RMSLE and placed around 3300th place which seems to be a decent result to start with.

3 Methodology

In addition to using 10 most correlated features, we can apply more advanced techniques to get a better result. Since the Kaggle uses RMSLE instead of RMSE as a performance indicator, it would be a logical approach that the target value, Saleprice, has an exponential characteristic. To remove its skewness, we use np.log function. Furthermore, machine learning algorithm cannot interpret string value, it is important to deal with categorical variables. To do so, applying pandas' get dummies function would solve the problem. Although MSSubClass has a integer value, it is correct to see the values as a categorical. Therefore, other string columns plus MSSubClass column should be converted with getdummies function.

4 Results

We have used 4 ways to analyze house price data : 10 Most correlated factors, log transform SalePrice column, converting categorical variables with pd.getdummies, and 15 Most correlated factors. RMSLE of the aforementioned methods are 0.17982, 9.45446, 0.22156, and 0.17284, respectively. So far multiple regression that uses 15 features out of 80 results the best performance.

5 Conclusion and Discussion

Even though there are many features in the dataset, including more features does not guarantee a better performance. Using 15 features gives a better performance than 10 features. However, including all categorical variables does not result in a better regression model. As a result, simple multiple regression gives the best performance. There are many potential methods to improve the performance. For example, we can sort out outliers that produce a bias in our result. Or we can look into every variables and adjust accordingly to the characteristic of a variable. For example, instead of dropping a column with more than one Null value, we can convert them into mean value and include it into the regression model. There are many ways to improve our model, but our multiple regression model has a decent performance to start with.