House Prices: Advanced Regression Techniques

ABSTRACT

In this Kaggle competition, we are going to predict the prices of houses by the data given. The project starts with data preprocessing, follows with data analysis and ends with modeling and prediction.

KEYWORDS

House Prices, Kaggle Competition, Random Forest, Xgboost, Ridge.

1 Methodology

The goal of this project is to use datasets from "House Prices: Advanced Regression Techniques" from Kaggle competition and then to conduct a comparative study on different advanced regression algorithms, which are used to predict sale price for each house given data of numerous descriptive features of the house. The accuracy of prediction will be evaluated via RMSE (Root Mean Square Error) taken between logarithm of the predicted price computed by a certain algorithm and the logarithm of the actual price from the training data.

To do preprocessing of the data, the data will be cleaned and features with a high percentage of missing values will be removed. More than 80% of missing data will be considered a high percentage. The percentage of missing data in the dataset is as follows: PoolQC(99%), MiscFeature(96%), Alley(93%), Fence(81%)...

Feature selection will be performed on the data to select the top features for fitting he classification models. Four classifier models will be fitted to the data. These include Random Forest, Xgboost, and Ridge. The performance of the models will be compared for the best accuracy. In addition, prediction of sale price for the test data will also be outputted using the best algorithm from the comparative study for reference.

2 Experiment

2.1 Data Description

The dataset contains three files: data_description.txt, train.cvs, and test.cvs. The description file provides basic information of data, such as the meaning of attribute variables. The other two file are the training data and test data, respectively. There are 1460 observations with 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa from 2006 to 2010. Among explanatory variables, there are 37 integer variables, such as Id, MSSubClass, LotFrontage, and 43 factor variables, such as MSZoning, Street, LotShape. We see that test has only 80 columns, while train has 81. This is due to the fact that the test data do not include the final sale price information.

2.2 Data Analysis and Visualization

In this section, we provide more insight into the independent variable "SalePrice", including basic descriptive statistics, relationship with some significant explanatory variables. Moreover, correlation between all variables will be explored.

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

Figure 1: Figure of summary of descriptive statistics of the variable SalePrice

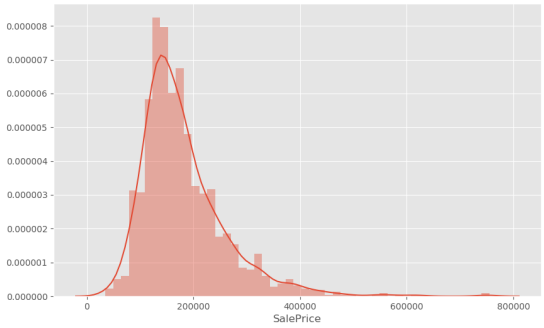

Figure 2: Figure of histogram of SalePrice

From the graph we can see a clear positively skewed distribution, implying that a log transformation on SalePrice will be needed in data processing. The skewness is calculated as 1.882876.

We select a few variables that are likely to have strong association with sale price of a house for presentation and visualization. Scatter plots are used for variables with quantitative values, whereas box plots are used for those with qualitative values.

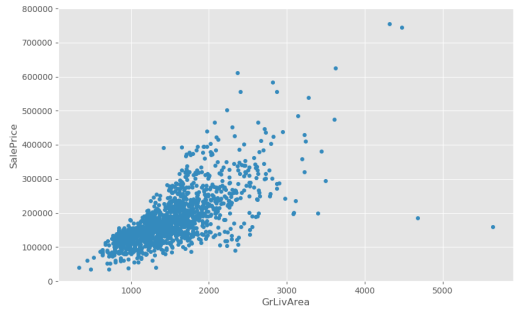

Figure 3: Figure of Scatter plot of SalePrice against GriLivArea

From the graph we can see that these two variables are positively correlated. However, the two data points (observations) near the bottom right corner of the plot should be identified as outliers.

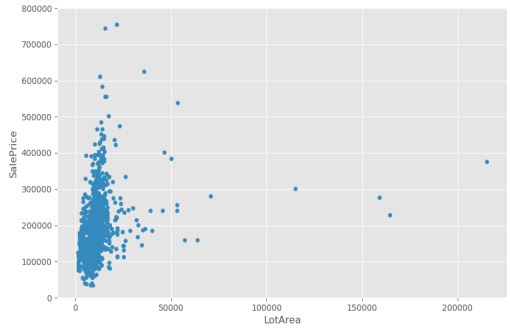

Figure 4: Figure of Scatter plot of SalePrice against LotArea

Likewise, they are positively correlated. The data points lying on the right may represent large houses located in inexpensive region. They can be treated as outliers.

Figure 5: Figure of Scatter plot of SalePrice against TotalBsmtSF

Two variables are positively correlated. One outlier lies on the right edge of the plot.

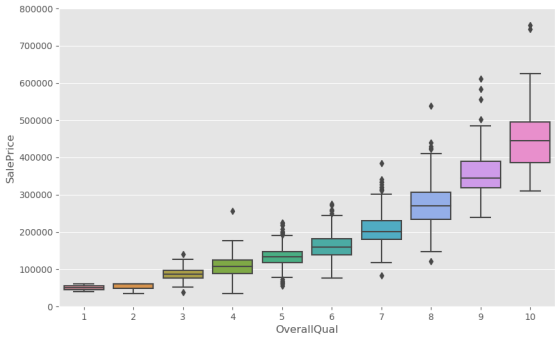

Figure 6: Figure of Box plot of SalePrice against OverallQual

Two variables are positively correlated.

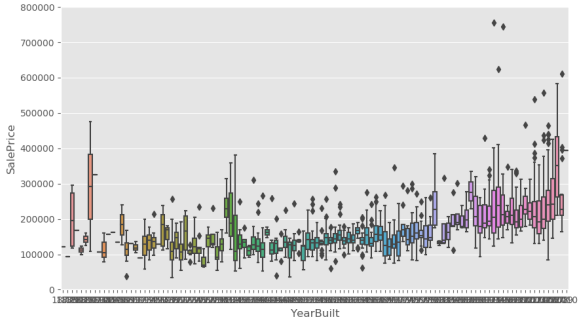

Figure 7: Figure of Box plot of SalePrice against YesrBuilt

It shows an increasing trend of sale price over years. New house may worth more in the market.

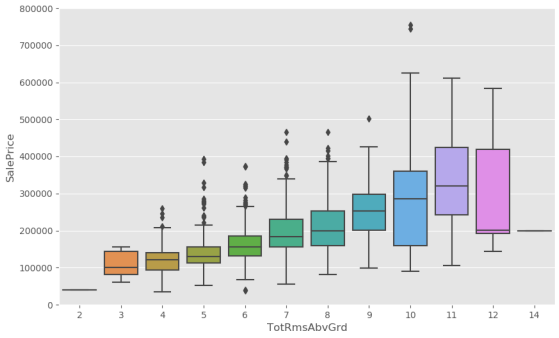

Figure 8: Figure of Box plot of SalePrice against TotRmsAbvGrd

Sale price increase as the number of rooms increases, and it reaches a maximum when TotRmsAbvGrd=11.

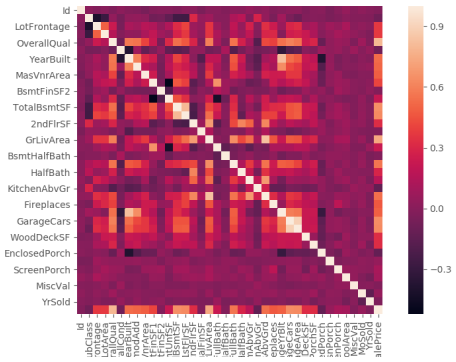

Figure 10: Figure of correlation matrix

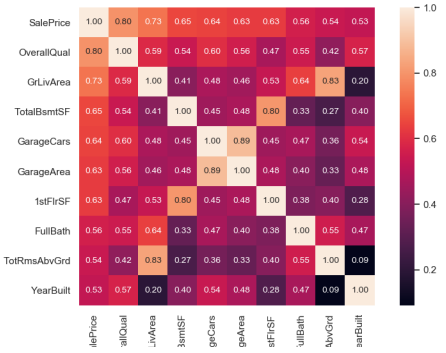

Figure 11: Figure of correlation matrix with number

The scale of correlation is shown on the right side of the plot. Some of the most correlated variables to SalePrice are: OverallQual, GriLivArea, TotalBsmtSF, GarageCars, etc. On the other hand, we also discover some highly correlated variables that may cause multicollinearity, implying that some need to be handled in data processing.

2.3 Data Preprocessing

This is an overview of the variables with missing values and the corresponding number and percentage of missing values:

	Count	Percent
PoolQC	1448	0.995873
MiscFeature	1402	0.964237
Alley	1363	0.937414
Fence	1173	0.806740
FireplaceQu	690	0.474553
LotFrontage	256	0.176066
GarageCond	81	0.055708
GarageType	81	0.055708
GarageYrBlt	81	0.055708
GarageFinish	81	0.055708
GarageQual	81	0.055708
BsmtExposure	38	0.026135
BsmtFinType2	38	0.026135
BsmtFinType1	37	0.025447
BsmtCond	37	0.025447
BsmtQual	37	0.025447
MasVnrArea	8	0.005502
MasVnrType	8	0.005502
Electrical	1	0.000688

Figure 12: Figure of missing values and percentage accordingly

For features that are missing too many values (over 80% observations), we drop them because of lack of significance of these variables. Specifically, variables to be dropped are: PoolQC, MiscFeature, Alley and Fence.

For features that a house not necessarily has, we fill in the value “0” for continuous variables and “None” for categorical variables. On the other hand, for features every house must have, we fill in the mode value of the feature. Specifically, missing values of "MasVnrArea", "BsmtUnfSF", "TotalBsmtSF", "GarageCars", "BsmtFinSF2", "BsmtFinSF1", "GarageArea" are filled with 0, while those of "FireplaceQu", "GarageQual", "GarageCond", "GarageFinish", "GarageYrBlt", "GarageType", "BsmtExposure", "BsmtCond", "BsmtQual", "BsmtFinType2", "BsmtFinType1", "MasVnrType" are filled with “None”. In addition, missing values of "MSZoning", "BsmtFullBath", "BsmtHalfBath", "Utilities", "Functional", "Electrical", "KitchenQual", "SaleType","Exterior1st", "Exterior2nd" are filled with the mode value.

A special case is handling the variable LotFrontage. For a house with missing value of LotFrontage, we decide to fill it using the median value of the neighborhood where the house is located using the variable Neighborhood. By now there should be no missing value from all the explanatory variables.

2.4 Features Processing

Summary of Most Important Features:

Top feature: OverallQual
Top 2 features: GrLivArea
Top 3 features: GarageCars
Top 4 features: GarageArea

Features are processed before applying regression algorithms to satisfy the underlying assumptions. For categorical variables with ordinal values, we use label encoding. For other categorical variables, we use one-hot encoding, meaning converting them to dummy variables. On the other hand, we log transform numerical variables to achieve normality.

3 Model Prediction and Conclusion

Finding the optimal parameters for regression algorithms using k-fold cross validation. The parameter used for Xgboost is max depth = 5, and the parameter used for Ridge is alpha = 1.7, and the parameter used for Random Forest is m_estimators = 200 and max features = 0.5.

Using the above parameters, we predict the log of sale price using different regressors. RMSE are computed for each between the predicted log transformed price and the original log transformed price. The result of the three models applied are as followed:

RMSE of XGBoost: 0.06270315929133988
RMSE of Random Forest: 0.049894464492623754
RMSE of Ridge: 0.12382062660675872

By comparison, random forest produces the least and thus is the best regressor among 3.

REFERENCES

[1] Travis E, Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).
[2] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)
[3] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007)
[4] https://seaborn.pydata.org/generated/seaborn.distplot.html
[5] https://www.kaggle.com/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda