Kaggle Competition: House Price - Project Report

Team Golden Squirrels: Lingjie Qiao, Minsu Kim, Kevin Liao, Cheng Peng

December 4, 2016

Abstract

This paper summarizes the background, problem, methodology and results of our team's final project in the course Stats 159 Reproducible and Collaborative Statistical Data Science. To make full use of statistical models and predictive tools we have learned from the class and challenge ourselves to the next level, our team choose to complete **"House Price: Advanced Regression Techniques"** from Kaggle Competition and enter the competition with our work.

Competition Link: https://www.kaggle.com/c/housepricesadvancedregression techniques

The goal of this project is to present the use of predictive modeling process and utilize software tools that effectively communicate the results. While the competition only emphasizes the accuracy of predicted values, our team at the same time are dedicated to maintain project reproducibility and provide both objective and personal reflections upon regression analysis.

1 Introduction

The House Price project thoroughly explores the predictive modeling process and advanced regression techniques. From previous study, in order to understand the relationship of one dependent variable with several independent variables, we fit a multiple linear regession with Ordinary Least Squares. However, since OLS may have high variance and include irrelevant variables, Predictive Modeling Process can improve the results in terms of ***Prediction Accuracy*** and ***Model Interpretability***.

The competition sets the background of the project: Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition requires participants to predict the final price of each home. Our team therefore follows the idea of model prediction and tries to use different techniques in order to most accurately predict the final sales price of each house.

2 Data

The datasets are obtained from the Kaggle Competition website. We have access to four files:
1. *data description*, which gives the official definition for fields
2. *train.csv*, which provides 1459 real observations that can be used for model construction
3. *test.csv*, which is used to fit the predictive model and create submission entry for the final sales price of 1460 observations
4. *sample submission*, which gives an example of how the fitted values should be submitted.

The train dataset has in total 80 variables, 79 potential predictors and 1 dependent variable called *SalePrice*. We observe both categorial predictors, such as *FireplaceQu*, *GarageCond* and *MasVnrType* as well as numerical predictors, such as *PoolArea*, *EnclosedPorch* and *YrSold*. Since we can potentially create a lot of different new variables, our goal is to understand the relationship between *SalePrice* and these predictors with statistical fitting procedures that minimizes Mean Square Error.

3 Exploratory Data Analysis

should be done

4 Methodology

The goal of this analysis is to accurately predict the final price of each home. Therefore, we frame this problem as a regression problem, and decide to use the L2 loss function [10] which is often used in regression problem. Taking this objective into account, we preprocess original dataset so that regression models can work well. Furthermore, we extract more features by involving feature engineering. Finally, we fit two shrinkage models and two ensemble models. The details are explained in the following subsections.

4.1 Evaluation and objective Loss Function

We specifically use root mean squared logarithmic error [5], which makes more sense in our problem setting because errors in predicting expensive houses and cheap houses should affect the result equally. The following is the formula of RMSLE.

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(p_i+1) - log(a_i+1))^2}$$

Since L2 loss function minimizes the squared differences between the estimated and existing target values [10], L2 error will be much larger in the case of outlier compared to L1 and therefore L2 loss function is highly sensitive to outliers in the dataset. So, in the preprocessing step, we eliminate outliers to remedy this issue.

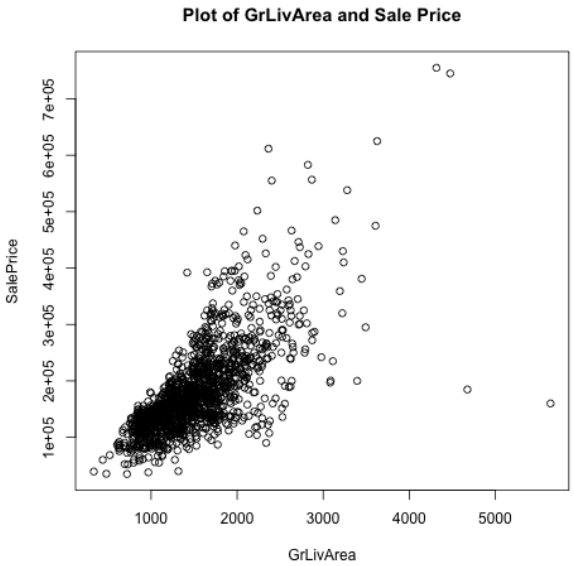
4.2 Preprocess

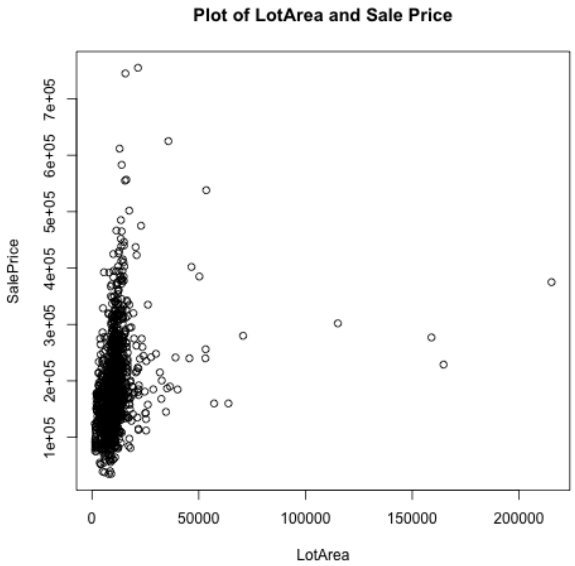
According to the exploratory data analysis, we find a lot of **NA values** in most of predictors. Since regression models cannot handle missing data, we need to either remove or impute data using appropriate methods [7]. The data description provided by client [6] indicates that some of the missing values are actually none value. In that case, we replace **NA** value with factor variable named **None**. However, there are some numerical predictors with missing values in an unsystematical manner. In that case, we impute them with **mean values** of predictors.

As a next step, we apply **log transformations** to **area related predictors** such as GrLivArea and LotArea as well as target variable. The log transformation has an effect to remedy skewness of data [8] by making original distributions of predictors to more normally distributed. Consequently, it helps regression model to work better.

After the log transformation, we still notice few outliers and eliminate lowest and highest 0.1% data points. Furthermore, majority of predictors are categorical under which regression models cannot be used directly. Therefore, we apply **one-hot encoding** [9] to convert categorical values to numerical ones. It consequently expands features from 79 to approximately 500. The Table 1 below summarizes this procedure.

Table 1: Data Preprocess and Variable Transformation





Factorization	Log Transformation	Removing Outliers
MSSubClass	SalePrice	17 points -> below 10.91511
YearBuilt	LotArea	and above SalePrice
YrSold	GrLivArea	
MoSold		
GarageYrBlt		

4.3 Data Preparation

Before fitting the model, we first split dataset into train and test. We could have a separate validation set. However, R library caret has a built-in cross validation as a generic interface. So it automatically takes care of cross validation. We use train data to train and tune our models using 5-kold cross validation, and later compare RMSLE using hold-out test data [13].

4.4 Featuring Engineering

Considering the complexity of the problem as well as the number of observations and predictors, we assume that the success of this analysis is largely dependent on **informative, feature engineered predictors** that can reveal the subtle relationship to our target variable. Given the small size of dataset with 1460 observations, we conclude that feature learning, which is a set of techniques that learn a feature: a transformation of raw data input to a representation that can be effectively exploited in machine learning tasks [1], is not a feasible option because feature learning often involves very complicated models with multiple layers, which tends to cause an overfitting issue when dataset is small [3].

With this observation, we therefore focus more on **manual feature engineering** [4]. This process is a important stepping-stone in that it helps reveal significant predictors that are previously not represented well in original dataset. By explicitly designing what the input x's should be, our predictive models can solve a problem easily.

4.5 Model description and hyper-parameter tuning

While training each model, we need to find optimal paramters for each model. In order to effectively select hyper-parameters, we use 10-fold cross-validation. For lasso and ridge, lambda is the tuning parameter. It determines how much we will penalize models for high weights on predictors. If lambda is high, it penalizes models more and ends up generating sparse models.

This kind of models is called shrinkage method because they shrinken weights or even remove predictors by penalizing models. These models are especially good options when there are many predictors. By penalziing or removing unnecessary predictors, they provide more interpretable results. Thus, they are often utilized in genomic and pharametical analysis. For PCR and PLSR, the number of principal components is the tuning parameter. They both internally use the precipal component analysis to obtain pricipal components, which are linear combination of original predictors in dataset. Based on spectral theorem, the eigenvector corresponding to highest eigenvalue is the direction that explains the most about the variability in data, which is the first principal component. We need to find the optimal number of subsets of PCs that summarize the entire data set without harming accuracy. Again, we use 10-kold cross validation to obtain the optimal number of principal components.

4.6 Modeling

We utilize both **shrinkage regression models** and **ensemble models**. Practitioners often favor ensemble models [11] due to their conveniences. Ensemble models such as Random Forest [12], which uses the averaged result from the randomly grown decision trees such as CART [13], often work well with unscaled, missing data and are used for both classification and regression problems. Also, since our dataset contains a hug number of predictors, shrinkage methods such Lasso and Ridge regressions [13], which penalize predictors by shrinking their weights, can be highly effective. Furthermore, we utilize a dimension reduction technique called PCA[13] in order to compress information into lower dimension. The results of modeling will be further explained in the following section in detail.

4.7 Model comparison

As mentioned on in Evaluation section, we use RMSE to compare models and select the best model. Although RMSE does not provide an absolute means of model accuracy, it provides a relative measure to compare models. Thus, we finalize our model with the lowest RMSE. In summary, the following is the the procedure for each model. 1. Split the data into train and test, 802. Train a model using train data with 5-fold cross-valdation. 3. Pick the optimal hyper-parameters. 4. Predict balance using the model with the optimal parameters. 5. Calculate Mean Square Error. 6. Record both Mean Square Error and coefficients.

5 Analysis

Due to the high dimensionality of dataset, we first experiment to decide whether to either project predictors into lower dimensional bases or select predictors using shrinkage methods. First, we try Principal component analysis (PCA) which is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [15]. Since our one-hot encoded dataset has a lot of predictors that have near zero variances, we first need to remove near zero variances. This is because PCA considers variability of data and compress the predictors with high variances into first principal components. After removing near zero variance 519 -> 128 -> This in-
dicates that a great number of predictors have apporiximately identical across observations. Figure $(\text{model}_{p}ca_{s}cree_{p}lot.png)showsaPCAscreeplotforeachprincipalcomponent.Itindicatesthatfirstfewprincipalcompon$
Furthermore, we try to fit both lasso and re-
gression and tune lambda values using 5-fold cross validation and then examine coefficients. Figure $\text{model}_{l}asso_{l}ambda.png, model_{r}idge_{l}ambda.pngshowsMSEforcorrespondinglambdavaluesforbothmodels.Figure(m$
$5and -$
$1.15respectively.Basedontheseresults, wedecideoptimallambdaforbothmodel.Wealsoexaminecoefficientsoflassore$

6 Results

We fit shrinkage regression models and ensemble model, experiment the several methods including dimension reduction techniques, feature importance measures.

Figure named (model_ lasso_ lambda.png) shows that MSE exponentially increases as log(lambda) values increases. The optimal lambda that minimizes MSE turns out to be 0.02020202.

7 Conclusion

In conclusions, while participating in the Kaggle competition - "House Price: Advanced Regression Techniques", we get a chance to dig down into real-life dataset that have complicated variables, missing values and large number of data entries. Based our analysis on the train dataset, we try to find the regression model and advanced techniques that most accurately predict the housing price, and use the exact model to predict new entries from test file. Exploring and creating visualization of data, we compare the usage of different regression models to understand the relationship between dependent variable *SalePrice* and 79 potential predictors. Fortunately, we ranked No.10 among all 2000 participating teams with our prediction, which not only shows the precision of our own predictive model, but also suggests room of improvement and growth.

This course gives us great opportunity to practice **project reproducibility**, **advanced statistical model learning** and **teamwork delegation**. By deligating work to each team mate and implementing the model to generate results, we aim to maintain project reproducibility and publicity so that more people can learn from our model and benefit for future researches. *If you have any questions or concerns regarding any part of our analysis, please feel free to contact the authors for further clarification.*