House Prices Prediction with Machine Learning Algorithms

Chenchen Fan
Electronics Engineering Department, Tsinghua University
ShuangQing Road no.30, Beijing, China
+86-010-62781389
fcc17@mails.tsinghua.edu.cn

Zechen Cui
Beijing Dongzhimen Middle School
ShuangQing Road no.30, Beijing, China
+86-010-62781389
Cuizc0616@163.com

Xiaofeng Zhong
Electronics Engineering Department, Tsinghua University
ShuangQing Road no.30, Beijing, China
+86-010-62781389
zhongxf@tsinghua.edu.cn

ABSTRACT
Based on the data set compiled by D. D. Cock and the competition run by kaggle.com, we propose a house prices prediction algorithm in Ames, lowa by deliberating on data processing, feature engineering and combination forecasting. Our prediction ranks the 35th of the total 2221 results on the public leaderboard of Kaggle.com and the RMSE of predicted results after taking logarithm from all the test data is 0.12019, which shows good performance and small of over-fitting.

CCS Concepts
•Computing methodologies→Supervised learning by regression.

Keywords
House Prices Prediction; Machine Learning Algorithms; Combination Forecasting; Kaggle Competition.

1. INTRODUCTION
Now in the era of Big Data, how to mine valuable information from enormous data set has been a big challenge. And its realization especially for the predictive modeling problem, will result in huge revolution in people's life and work. Meanwhile, machine learning algorithms [1] perform excellently and have been widely used in such problems [2-4]. In that case, this paper utilizes machine learning algorithms to discover the hidden pattern in data and achieves house prices prediction in Ames, lowa based on the contest hosted by Kaggle.com [5].

The Ames Housing data set [6] was compiled by D. D. Cock and it covers all the features necessary for describing a residential house. But among these features provided by this data set, nearly half have missing values, which indeed lays a challenge for the subsequent process. In addition, based on the given features, how to expand them to maximize the useful information and how to select the better ones to reduce the processing time and avoid the effect of irrelevant information also play a key role on the final result. With respect to building models, this paper adopts combination forecasting to optimize the predicted values.

In this work, we present a procedure to model the house prices in Ames, lowa and solve the challenges about missing values, feature engineering and combination forecasting. Our prediction ranks the 35th of the total 2221 results on the public leaderboard of Kaggle.com and the RMSE of predicted results after taking logarithm from all the test data is 0.12019, which shows good performance. The paper structure is as follows: Section 2 introduces the detailed problem description. Section 3 is focused on exploring the data to extract information. Section 4 presents the process of feature engineering. Section 5 shows the models proposed and their combination. Finally, Section 6 presents the final result. Section 7 concludes the paper with remarks and hints about future work.

ICMLC 2018, February 26–28, 2018, Macau, China

ACM ISBN 978-1-4503-6353-2/18/02…$15.00
DOI: http://dx.doi.org/10.1145/3195106.3195133

2. PROBLEM DESCRIPTION
2.1 Data Description
The Ames Housing data set consists of records ranging from 2006 to 2010. Kaggle.com processes this data set and divides it into training set and test set. In training set, each record has 80 features. While in test set, the objective feature—SalePrice which needs to be predicted, is taken out. Among them, 37 features are numeric, and the other 43 ones are categorical. For using more conveniently in subsequent steps, we denote numeric features as $N_{\text{feature\_name}}$, categorical features as $C_{\text{feature\_name}}$ and the objective feature as P.

2.2 Evaluation Function
Training set is publicly available for competitors to train their models, while the test set is used to evaluate the performance of models. Kaggle.com provides two leaderboards to compare the results from all the competitors. The public leaderboard uses approximately 50% of the test data to compute:

$$L_{\text{RMSE}} = \sqrt{\frac{1}{n_{\text{S}}}\sum_{i=1}^{n_{\text{S}}}[\log(O_i) - \log(P_i)]^2} \quad (1)$$

where O_i represents the observed value, P_i represents the predicted value, n_{S} is the amount of records used to compute L_{RMSE}. Taking logarithm means that errors in predicting expensive houses and cheap houses will affect the result equally. Because the specific test records used in public leaderboard are

unknown, this leaderboard can indeed reflect models' performance. Another private leaderboard is not visible until the end of this competition, which uses the rest 50% of the test data. Besides the ranking provided by public leaderboard, we compute the L_{RMSE} with all the test data according to original Ames Housing data set as a more credible measurement index.

3. EXPLORATORY DATA ANALYSIS

3.1 Data Visualization

Data visualization refers to the way of re-displaying data through charts, animations or other graphs. In this way, researchers can quickly understand, absorb the valid information, and discover the implicit patterns in the data. As shown in Figure 1, values of the feature $N_{\text{GrLivArea}}$ are centrally located within 4000 square feet, and its distribution shows positively skewed. Figure 2 is a scatter diagram of feature $N_{\text{GrLivArea}}$ and P in training set. From it we can easily identify the abnormal records, which are not consistent with the overall distribution trend of other records and then delete them.

In this part, we discover the heteroscedasticity of all the numeric features and take logarithm to deal with it. With regard to the abnormal records or values, appropriate measures such as modifying or deleting them, are taken.

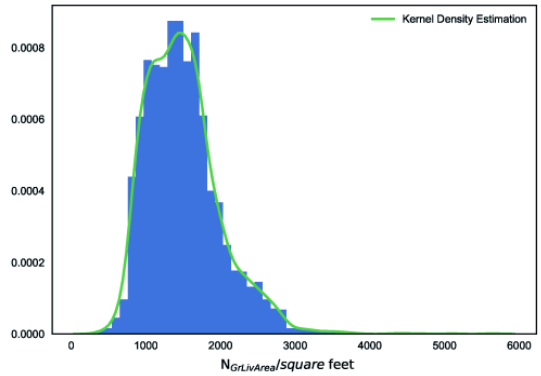

Figure 1. Frequency distribution histogram of $N_{\text{GrLivArea}}$

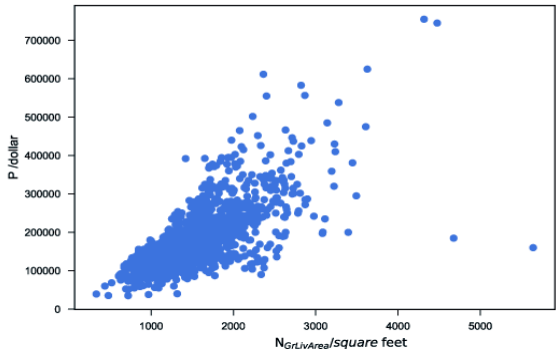

Figure 2. Scatter diagram of $N_{\text{GrLivArea}}$ and P

3.2 Missing Values

Features with missing values and their corresponding loss rate are shown in Table 1. 34 features of the total 80 features contain more than 7000 missing values and some features have fairly high loss rate. Deleting the records or features containing missing values is an efficient coping strategy, but it can risk losing useful information. In contrast, interpolation method needs to have a better understanding of the whole data set and is time-consuming, but the risk is small. Meanwhile reasonable interpolation can even provide convenience for follow-up process.

This paper utilizes three kinds of interpolation methods according to different characteristics of features. The first one relies on the mapping relationship between features. That is to say, filling the missing values by virtue of other closely-related features. This method is used mostly for features with a high loss rate and eventually processes 23 features. The second one is used for categorical features with extremely uneven distribution, and it fills missing values with the category which has the largest frequency. What's more, for some categorical features, the most possible reason for losing values is that there is no suitable category in current value set. In that case, the missing values can be filled with a new defined category based on the analysis of practical situation. This disposal is exactly the third method to interpolate and it is applied to 6 features.

Table 1. Features with missing values

Feature	Number of missing values	Loss rate
C_{PoolQC}	2909	99.657e-2
$C_{\text{MiscFeature}}$	2814	96.403e-2
C_{Alley}	2721	93.217e-2
C_{Fence}	2348	80.439e-2
$C_{\text{FireplaceQu}}$	1420	48.647e-2
$N_{\text{LotFrontage}}$	486	16.650e-2
…	…	…

4. FEATURE ENGINEERING

Jeff Heaton [7] argues that feature engineering is a manually completed and time-consuming task in machine learning applications, and the same model will show very different performance after different feature engineering processes. And the essence of feature engineering is to reasonably expand or shrink the existing feature set.

4.1 Feature Expansion

This paper divides all the features into three kinds and disposes of each kind with different method. For ordinal features and nominal features with balanced value distribution, we create the corresponding numeric feature by finding the mapping relationship from the original categorical one. While for other nominal features with extremely uneven distribution of values, we convert them to binary variables in order to highlight the most influential factor with the largest frequency. And for numeric features, the expansion is more flexible. We utilize the comparison between numeric features, mathematical operation or their unique value characteristics to create new numeric features.

Another problem we are faced with is that only numeric features can be used in subsequent models. So we adopt the common

practice, namely one-hot encoding, to convert all the categorical features to numeric ones.

4.2 Feature Selection

After expansion, the feature set contains 359 features, so feature selection [8] is of great necessity.

There are three pervasive methods for feature selection, including Filter, Wrapper and Embedded. Contrast to conventionally choosing one method at will, this paper uses variance to filter all the features first and then compares 5 different disposals belonging to the three methods. We set the threshold of variance as 0.02, thus filtering out 120 features. Without loss of generality, we use 10-fold cross validation to compute the average of MSE (2) for each disposal based on the training set.

$$\mathrm{MSE} = \frac{1}{n_S}\sum_{i=1}^{n_S}[\log(O_i) - \log(P_i)]^2 \tag{2}$$

It can be seen from Table 2 that when the quantity of deleted features is less than 50, RFE has the lowest mean. In that case, we draw the curve of MSE in Figure 3 to identify the optimum point using RFE. It's obvious the average of MSE is lowest when deleting 35 features.

Table 2. The comparison of different methods

Method / Average Of MSE / Quantity of deleted features	P-value/ Pearson correlation coefficient	Distance correlation coefficient	RF	RFE
10	1.39e-2	1.30e-2	1.30e-2	1.28e-2
50	1.39e-2	1.31e-2	1.30e-2	1.29e-2
100	1.55e-2	1.42e-2	1.35e-2	1.64e-2
150	1.62e-2	1.63e-2	1.40e-2	2.14e-2
200	1.80e-2	1.82e-2	1.54e-2	3.18e-2
220	2.04e-2	2.08e-2	2.03e-2	5.53e-2

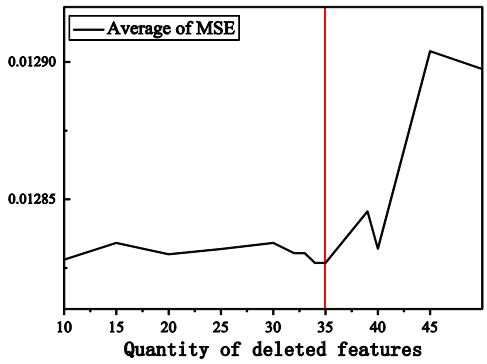

Figure 3. Average of MSE with changing quantity of deleted features

5. MODEL

This paper compares the performance of several models, chooses the better ones, and then combines their predicted values as the final result.

5.1 Introduction to Models and Important Parameters

Lasso and Ridge linear regression models are improvements to the classical linear regression model by adding different penalty terms L_1 and L_2 to objective function (3) (4) respectively. For these two models, only the parameter α in penalty terms needs to be tuned.

$$\min_w \left\{ \frac{1}{2n_S} \left\| XW - O \right\|_2^2 + \alpha \left\| W \right\|_1 \right\} \tag{3}$$

$$\min_w \left\{ \frac{1}{2n_S} \left\| XW - O \right\|_2^2 + \alpha \left\| W \right\|_2^2 \right\} \tag{4}$$

where X is the input vector and W is the coefficient vector, which their product forms the predicted vector. And O is the vector of observed values.

The principle of Support Vector Regression (SVR) [9] is finding a function $f(x)$ (5) to obtain the outputs, which makes the deviation between predicted values and observed values less. The parameters needed to be tuned are C which is the coefficient of slack variable, and kernel which represents the chosen kernel function.

$$f(x) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b,\ \boldsymbol{w} \in \chi,\ b \in R \tag{5}$$

Random Forest (RF) uses Bagging method for tree ensemble and adopts bootstrap sampling to generate different trees. For regression problem, the final result is the average of the outputs from all the trees. Extreme Gradient Boosting (XGB) [10] is an improvement to Gradient Boosting Machine [11] using Boosting method for tree ensemble. And the result of this model is the sum of the outputs from all the trees.

5.2 Parameter Optimization

To improve the performance of models, this paper uses hyper-parameter automatic search module—GridSearchCV to search the optimum values of parameters for each model. Without loss of generality, we use 10-fold cross validation in this module and output the average of MSE with the optimal parameters to measure the performance of each model. The comparison of different models is shown in Table 3.

5.3 Model Selection and Combination Forecasting

As can be seen from Table 3, Lasso linear regression model, Ridge linear regression model and XGB have lower MSE. Based on that, we adopt combination forecasting method [12], which can synthetically utilize the advantages of different models and improve the accuracy of result. The computation formula of linear combination we used is:

$$p = a_1 \times p_{XGB} + a_2 \times p_{Lasso} + a_3 \times p_{Ridge} \quad (6)$$

where p_{XGB}, p_{Lasso} and p_{Ridge} are the predicted values of XGB, Lasso linear regression and Ridge linear regression respectively. And a_1, a_2, a_3 are corresponding weight values for different predictions and their sum equals one.

Therefore, the key issue is determining the proper weight values to maximize the performance of combined models. Based on the measurement index L_{RMSE}, we find that the bigger a_3 is, the larger error is. So a_3 is set as 0. And from Figure 4, we identify the weight values of XGB and Lasso linear regression as 0.6 and 0.4 respectively.

Table 3. The comparison of models with optimal parameters

Machine Learning Algorithm	Average of MSE	Optimal parameters
Ridge Linear Regression	1.263e-2	α : 10.0
Lasso Linear Regression	1.283e-2	α : 3.810e-4
RF	1.782e-2	max_depth: 8, max_features: 100, min_samples_split: 5, n_estimators: 100
SVR (Linear Kernel)	1.757e-2	C : 0.4, kernel: linear
SVR (Gaussian Kernel)	1.506e-2	C : 0.6, kernel: rbf
XGB	1.493e-2	colsample_bytree: 0.9, learning_rate: 0.1, max_depth: 3, subsample: 0.7

6. RESULTS

Our software programming & development environment is JetBrains Pycharm Professional. We utilize the ranking in public leaderboard and L_{RMSE} for all the test data as performance measurements.

Until now, there are 2221 teams and 2390 competitors taking part in this competition. Among all the 2221 results shown in public leaderboard, the L_{RMSE} of our predicted result is 0.11390 and ranks the 35th. Though the private leaderboard is not visible to competitors, we compute the L_{RMSE} for all the records in test set according to the Ames Housing data set and the result is 0.12019, which shows good performance and small of over-fitting.

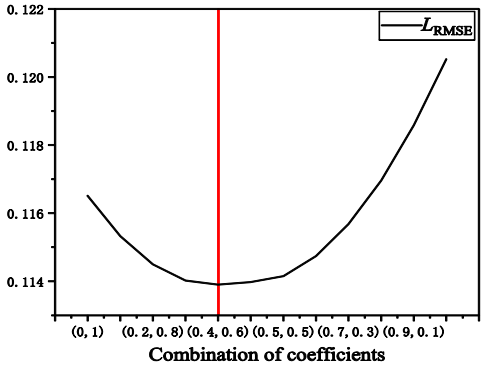

Figure 4. L_{RMSE} with different combination of coefficients

7. CONCLUSION

In this work, we present a procedure to achieve house prices prediction with limited data set. This provided data set has a high loss rate and ill-suited feature set for subsequent process. To solve these problems and improve the predictive accuracy, we manage to interpolate the missing values properly and make efforts in feature engineering, including feature expansion and feature selection. What's more, to build appropriate models, we compare the performance of different models and adopt combination forecasting, which indeed improve the final result.

As future work on the presented problem, we could try other machine learning algorithms depending on the sensitivity to this data set or non-linear combination forecasting to improve the predictive accuracy.

8. ACKNOWLEDGMENTS

This work is supported by Key Laboratory of Universal Wireless Communications (Beijing University of Posts and Telecommunications), and National Natural Science Foundation of China (No. 61631013).

9. REFERENCES

[1] A. L'Heureux, K. Grolinger, H. F. ElYamany, et al. 2017. Machine learning with big data: Challenges and approaches. *IEEE Access* 5(Apr. 2017), 7776-7797. DOI= https://doi.org/10.1109/ACCESS.2017.2696365.

[2] Ola Al Sonosy, S. Rady, N. L. Badr, et al. 2016. A study of spatial machine learning for business behavior prediction in location based social networks. In *Proceedings of the 11th International Conference on Computer Engineering & Systems*, IEEE, Cairo, CAI, 266-272. DOI= https://doi.org/10.1109/ICCES.2016.7822012.

[3] S. Hunta, N. Aunsri, T. Yooyativong. 2015. Drug-Drug Interactions prediction from enzyme action crossing through machine learning approaches. In *Proceedings of the 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, IEEE, Hua Hin, HH, 1-4. DOI= https://doi.org/10.1109/ECTICon.2015.7207126.

[4] Y. Q. Liu, H. J. Zhang. 2016. An empirical study on machine learning models for wind power predictions. In *Proceedings of the 15th IEEE International Conference on Machine*

Learning and Applications, California, CA, 758-763. DOI= https://doi.org/10.1109/ICMLA.2016.0135.

[5] Kaggle. 2016. House Prices: Advanced Regression Techniques. Retrieved from https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

[6] D. D. Cock. 2011. Ames, Lowa: Alternative to the Boston housing data as an end of semester regression project. *J. Journal of Statistics Education*, 19, 3 (Nov. 2011), 15 pages.

[7] J. Heaton. 2016. An empirical analysis of feature engineering for predictive modeling. In *Proceedings of the IEEE Region 3 Technical, Professional, and Student Conference*, IEEE, Norfolk, NF, 1-6. DOI= https://doi.org/10.1109/SECON.2016.7506650.

[8] Aparna. U. R, Shaiju P. 2016. Feature selection and extraction in data mining. In *Proceedings of the International Conference on Green Engineering and Technologies*, IEEE, Kuala Lumpur, KUL, 1-3. DOI= https://doi.org/10.1109/GET.2016.7916845.

[9] A. J. Smola, B. Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing archive*, 14, 3 (Aug. 2004), 24 pages.

[10] T. Chen and C. Guestrin.2016. Xgboost: A scalable tree boosting system. Retrieved from http://arxiv.org/abs/1603.02754

[11] J. H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *J. Annals of Statistics*, 29, 5(Nov. 2011), 1189-1231.

[12] Z. L. Sun, C. H. Zhu, B. Xu, et al. 2011. Research on machine learning method- based combination forecasting model and its application. In *Proceedings of the English International Conference on Fuzzy Systems and Knowledge Discovery*, IEEE, Shanghai, SH, 1226-1231. DOI= https://doi.org/10.1109/FSKD.2011.6019650.