MATH 4432 Mini Project 2: House Price Prediction

Zizheng Lin[1] and Yanbang Wang[2]

[1]Department of Mathematics, HKUST
[2]Department of Computer Science and Engineering, HKUST

1. Introduction

This project aims at predicting housing price for Ames Housing dataset through various regression techniques, which includes generalized linear models as well as boosting methods. Furthermore, intensive data preprocessing and feature engineering practices are also deployed so as to facilitate subsequent regression process. We recorded our procedures for each step of engineering the data, which includes preprocessing, feature engineering and selection, and also make detailed analysis as well as comparison between different algorithm.

2. House Value Dataset

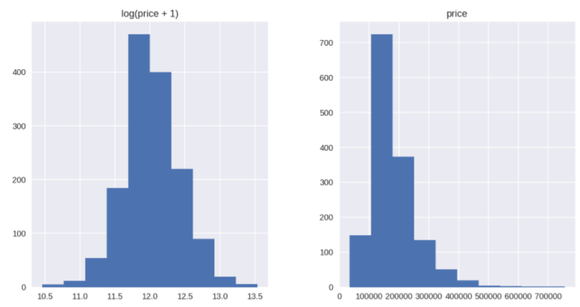
This data set incorporates a moderate number of houses sale information. The 79 variables functioning as predictors provide a quite comprehensive description of almost all aspects of a house's condition. Sale Price is given in the training set as the response variable and is to be predicted for the test set. Note that a considerable portion of the predictors are not meant to be utilized directly, due to various reasons such as containing missing values, qualitative values, being excessively skewed, etc. Therefore, a good amount of work has to be conducted to meticulously clean the data for the best of their usage, which would be elaborated in the following section.

3. Data preprocessing and Feature Engineering

Convert missing and NA data The Ames Housing dataset contains some fields where certain values are not available. For example, in GarageQual (Garage Quality) field, the 'NA' represents that the house has no garage, whereas in LotFrantage (Linear feet of street connected to property) field, the 'NA' denotes that the value is missing. Since such values were all regarded as 'NAN' in python pandas package, which hinders the consequent data analysis, appropriate measures were taken to convert them into suitable numbers. Essentially, we adopted 4 different filling strategy for corresponding fields. The first is converting the 'NA' into 'None' string, such as GarageQual and PoolQC. Second one is replacing 'NA' with number 0, such as GarageArea. Third one is utilizing mean value of houses in the neighborhood to estimate the missing value, such as LotFrontage. Final one is using the mode value of other records as substitution, such as Functional. The detailed implementation can be viewed in the source code.

Handle qualitative variables In terms of the qualitative attributes in the data set, we create dummy variables for each of them. Simply quantizing those attributes tend to integrate into those data the non-existed ordering inherited by those quantities, which is abandoned by us.

Log Transformation We examined the skewness of the predictors and response, as a necessary preprocessing step for the later adoption of generalized linear models such as LASSO and Ridge Regression. For those that exhibit skewness to a certain extent, logarithm transformation is applied, which very perceivably renders them a more center-oriented distribution. [2]



Feature Selection Backward recursive feature selection is adopted to select the best set of features for regression. Start from the data with full features, we fit them into the Lasso regression model, and remove the feature that has the least statistical significance, and the new data set with one feature removed is again fit into the model and the process repeats till only 1 feature left. During each iteration, the cross-validation score of current model is recorded, and the feature set with the highest cross-validation score is selected as the optimum.

4. Models

Ridge regression This model addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares, Here, α is a complexity parameter that controls the amount of shrinkage: the larger the value of , the greater the amount of shrinkage, and thus the coefficients become more robust to collinearity. [1]

$$\min_w ||Xw - y||_2{}^2 + \alpha||w||_2{}^2$$

LASSO Regression The Lasso is a linear model that estimates sparse coefficients. It is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent. For this reason, the Lasso and its variants are fundamental to the field of compressed sensing.

$$\min_w \frac{1}{2n_{samples}}||Xw - y||_2^2 + \alpha||w||_1$$

Bagging A Bagging regressor is an ensemble meta-estimator that fits base regressors each on bootstrapped instance of the original dataset and then average their individual predictions to form a final prediction. This average effects over bootstrapped dataset reduce the variance of each single base regressor since sampling from given dataset leads to a good approximation of the ground truth distribution[1]. The predictions from bagging can be depicted as following formula:

$$\frac{1}{B}\sum_{b=1}^{B}\hat{f}_b$$

Random Forest Random Forest algorithm further enhance the heterogeneity of the base classifier by integrating randomness into the searching pool of the features used to split. Despite the slight increase in bias introduced by the extra randomness, the algorithm very perceivably reduce the variance, with its gain outweighing the bias increase.

Gradient Boosting Tree regression A Gradient Boosting Tree (GBT) is an additive model over numbers of weak learners. Each weak learner is a decision tree set to learn the negative gradient steps on previous predictive function, and then use the prediction of optimal terminal node as the update step. The base regression tree is fed with randomly selected subset of training data in order to reduce the correlation among base regressors[2]. The update formula of prediction in each step is as follows:

$$\hat{F}_m(x) = \hat{F}_{m-1}(x) + \lambda\rho_k(x)$$

Where ρ_e denotes the prediction of optimal node of base regressor k, is a hyperparameter representing the learning rate. Since the learning rate reflects the magnitude of change for the model, a larger number of base regression trees usually requires smaller learning rate to avoid sub-optimums.

5. Result and Analysis

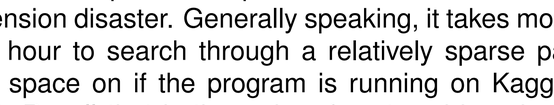
5.1 Result

the feature selection procedure shrinks the number of features from 79 to 59, leaving behind 'MSSubClass', 'MSZoning' and so on.

We perform grid search over the hyperparameters of each model. For bagging model, we search over the range of number base regressors in range 500, 600, 700. For GBT, we search over the range of number base regressors in range 6000, 7000, 8000, maximum depth of each regression tree in range 3,4,5, learning rate in range 1e-2, 1e-1, minimum number of samples in each leaf node of base regressor in range 60, 80. More details could be find with reference to our Jupyter Notebook submitted alongside.

Model	Test MSE	Best Param's
Ridge Regression	0.153391	'alpha': 1
LASSO Regression	0.227646	'alpha': 0.5
Bagging	0.152950	'n_estimators': 700
Random Forest	0.150630	'max_features': 15, 'n_estimators': 700
Gradient Boosting	0.149058	'learning_rate': 0.01, 'max_depth': 3, 'min_samples_leaf': 60, 'n_estimators': 5000

As shown in the table above, the gradient boosting methods gives birth to the best performance among all models. Therefore, gradient boosting is applied to the test set, the result of which is submitted to the leather board. The ranking is returned as follows:



5.2 Analysis

With reference to the ranking result, there are multiple aspects to be looked at when we approach this prediction problem, including reexamining the preprocessing steps, questioning the validity of parameter tunning, and analyzing the model evaluation and their effectiveness in terms of accuracy and time cost.

Preprocessing With respect to this specific problem, it is worth noting that the data preprocessing could latently make large difference to the final accuracy. As has been introduced in House Value Dataset Section, how to handle the missing values pose a large challenge. While we address this problem by a brute rule of replacing the original values with 0's, means or modes, it remains questionable especially when we are dealing with missing values that account for the majority. Meanwhile, we have applied log transformation to all of the skewed numerical values, but we do not really examine distributions for each numerical predictor. Therefore, it is possible that the transformed values could be even more skewed.

Parameter Tuning The empirical study indicate that parameter tunning in this case could be relatively time consuming, especially for boosting and random forest, where there exist quite a few parameters that could lead to dimension disaster. Generally speaking, it takes more than an hour to search through a relatively sparse parameter space on if the program is running on Kaggle Kernels.Recall that both random forest and bagging would variate dramatically when n_estimators is relatively small. Therefore, the time cost constrains seriously limit the parameter tunning and can easily lead to us ending up with local minimum.

Model selection It is worth noting that LASSO results in a noticeably high RMSE. A possible explanation for this looks back at LASSO's assumption that coefficients follow Laplace distribution. This might be extensively violated in this problem when we take natural log to all the numerical values. Indeed, this could be evident when we look at ridge regression outperforming in this setting, which bases its assumption that the coefficients follow a normal distribution. The assembled method does perform with robustness in this setting. With gradient boosting tops among them. The cross validation proves to be reliable when we submit the predictions (0.14970 is very close to our estimated test set RMSE 0.149058). This makes sense because our training set is moderately large, and the competition's test set and training set are generate by random split (so they are i.i.d's), all of which make it possible to infinitely approach or simulate the real distribution of data set with bootstrapped samples.

6. Conclusion

In this mini-project we approach the house values prediction problem using the various models covered in classroom. Large amount of work has been done with respect to data preprocessing, parameter tunning and feature engineering. We end up ranking in the middle on Kaggle's leather-board, which indicates that we still have much space for further improvements. Our analysis reveal that our future work could concentrate even more on various other various fields, especially appropriately preprocessing the data.

References

[1] Generalized linear models¶.

[2] Regularized linear models.