CS583A: Course Project

Zihan Chen

December 1, 2019

1 Summary

We pariticipe an active competition of Predict sales prices and practice feature engineering, RFs, and gradient boosting. The final model we choose is Model Stacking, a stacking model based on base models: ElasticNet, Gradient Boosting, Kernel Ridge and Lasso. It takes 79 variables as input and outputs the prediction price of each house sample.We implement the stacking model and one full-connected neural network using Keras, and run the code on Google Colaboratory. Performance is evaluated on the mean absolute error. In the public leaderboard, our score is 0.11555; we rank 506 among the 5384 teams. The result on the private leaderboard is not available until now.

2 Problem Description

Problem. This dataset contains 79 explanatory variables with every aspect of residential homes, and the aim is to predict the final price of each home. This is a regression problem with machine learning skills. The competition is at `https://www.kaggle.com/c/house-prices-advanced-regression-techni` `overview`.

Data. The data contains two part: training samples and test samples, each with 70 variables. The number of training samples is $n = 1460$. The number of training samples is $n = 1459$. The output is the price of each house.

Challenges. There are two main challenges in our dataset. The first one is the high proportion of missing values. What's more, missing value has totally different meaning in each variables. Therefore, we need to solve it separately. Secondly, the output of our data is skewed, which will leads to problem when we used methods like OLS. Hence, we also need to handle this problem.

3 Solution

Model. The model we finally choose is the stacking model based on four base line model: ElasticNet, Gradient Boosting, Kernel Ridge and Lasso. We firstly build a simple stacking model by averaging base models, then add a meta-model on averaged base models and use the out-of-folds predictions of these base models to train our meta-model. A description of ResNet is online: `https://en.wikipedia.org/wiki/Ensemble_learning`

(a) The mean absolute error on the training set and validation set.

(b) The loss on the training set and validation set.

Figure 1: The convergence curves.

Implementation. We implement the stacking models using Google Colaboratory. Our code is available at `https://github.com/ZihanChen1995/CS583-Final`. We run the code on a MacBook Pro with one Intel i7 CPU and 16 GB memory, as well as Google Colaboratory.

Settings. In the full connected neural network. The loss function is categorical mean absolute error. The opitmizer is Adam. And the batch size is 32, epoch is 500. In the stacking model, the percentage we assigned to three model is: Stacked model: 0.80, Xgboost:0.10, LGBM: 0.10

Cross-validation. We tune the parameters using a 5-fold cross-validation. Figure 1 plots the the convergence curves on 80% training data and 20% validation data.

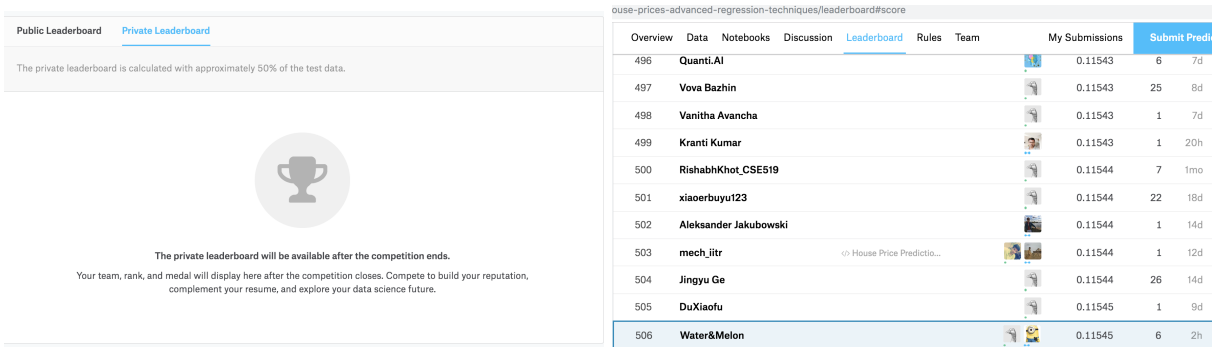
4 Compared Methods

Fully-connected neural network. We implemented a 3-layer fully-connected neural network. The width of the layers (from bottom to top) are all 256, as output is linear. The training and validation mean absolute error are respective 0.0907 and 0.09686

Random Forest Regressor. We use the random forest regressor model provided by SKlearn, as well as cross validation method. We set folds as 5, and report the mean score of mean absolute error. The mean score of this model is 0.1041. (Result is in Model 3)

Bayesian Regression. We use the linear model, BayesianRidge model provided by SKlearn. as well as cross validation method. We set folds as 5, and report the mean score of mean absolute error. The mean score of this model is 0.08393. (Result is in Model 3)

Baseline Model We also fit our model with some baseline model. Following are the results.



(a) Private leaderboard.

(b) Public leaderboard.

Figure 2: Our rankings in the leaderboard.

- LASSO Regression. Alpha is 0.0005, Random state is 1, the score is 0.1115 (standard deviation is 0.0074)
- Elastic Net Regression. Alpha is 0.0005, l1 ratio is 0.9, random state is 3, the score is 0.1116 (standard deviation is 0.0074)
- Kernel Ridge Regression Alpha is 0.6, kernel is 'polynomial', degree is 2, coefficient is 2.5. The score is 0.1153 (standard deviation is 0.0075)
- Gradient Boosting Regression n estimators is 3000, learning rate is 0.05, max depth is 4. The score is 0.1177 (standard deviation is 0.0080)
- XGBoost Gamma is 0.0468, learning rate is 0.05, max depth is 3,The score is 0.1161 (standard deviation is 0.0079)
- LightGBM Objective is 'regression',num leaves is 5, learning rate is 0.05, n estimators is 720, max bin is 55. The score is 0.1157 (standard deviation is 0.0067)

5 Outcome

We participated in an active competition. Our score is 0.11555 in the public leaderboard (private leaderboard not available). We rank 506/5384 in the public leaderboard. (We just have one member in our team) The screenshots are in Figure 2.