**Checkpoint 3 Report**

**Dingrong Wang, Shixuan Guo, Zulun Zhu**

1. Data Preprocess

In the data preprocess section, we remove the outlier points, fix the problem of missing value by removing the entry or inserting a suitable value. And then, we calculate the correlation matrix to give polynomial feature expansion to the top ten important features with respect to the response, housing price. Next, we manually address each feature by discretizing numerical variables to categorical variables, transforming categorical variables to dummy variables or ordered numbers and discarding some constant or unimportant features to make the data set easier to be modeling. In the last step, using some clever statistical methods, we make the features and the response in the training data set more normalized to satisfy some machine learning algorithms’ assumptions.

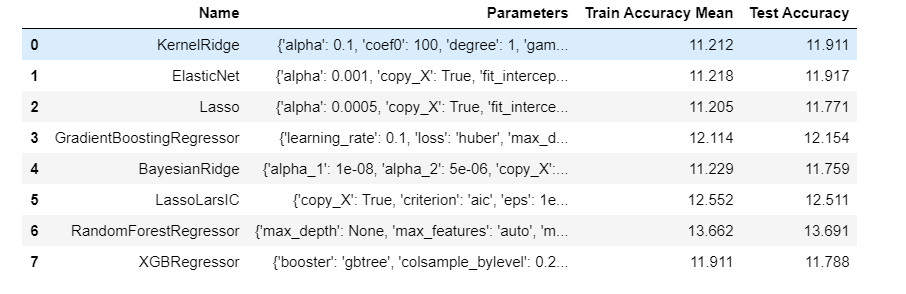
2. Decision of the core algorithm

Basically, we test 8 candidate algorithms to find the best performance algorithm as the core algorithm. Specifically, they are:

1. Kernel Ridge Regression
2. Elastic Net
3. Lasso
4. Gradient Boosting
5. Bayesian Ridge
6. Lasso Lars IC
7. Random Forest Regressor
8. XGBoost

We use Root Mean Square Error (RMSE) of the validation data set to measure the performance of these algorithms. Using a separate validation data set can avoid the problem of overfitting.

AIC，BIC or adjusted R2 are also OK, but since the data is enough here, we just directly measure the test error. Using the default parameter setting configuration will make these models perform poorly. So, the first thing we do is to use GridSearchCV method to choose the best parameter setting for these algorithms. And then, we train these models respectively and see their performance in the validation set. Experiments show that the RMSE of Lasso method is lower than the other methods in the validation set. So, we choose Lasso as our meta-model for stacking (core algorithm), just as the following figure shows.



3. Improve the model performance by fine tuning

The stacking process is like this: First using the other 7 estimators trained from the training set, we predict the response for each term in the validation set, then we form a new training set, which has 7 features corresponding to the prediction result of each estimator for every data entry in the validation set. Then using this new training data set as the predictors and the true label of validation set as the response to train the meta-model. Finally, we use this trained meta-model to make predictions for the future data. Experiments show that the stacking-model here can improve the performance of the last optimized lasso model, just as the following figures show.

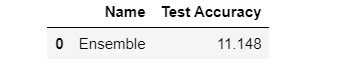
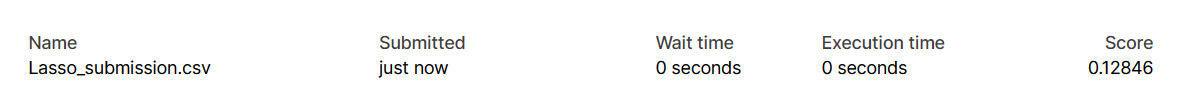


Figure1: The RMSE w.r.t the log of the validation result, which has been multiplied by 100 to compare with the previous models’ results.

Figure2: The RMSE w.r.t the log of the prediction results