House Price Prediction

Team

1. Aman Arora (amana4@illinois.edu): Worked on ridge regression modelling and performance evaluation.
2. Gaurav Dubey (kgdubey2@illinois.edu): Worked on data pre-processing and XGboost modeling

Introduction

This is a popular data set and the details were available from web (see acknowledgments). It contains 2039 rows. The data set contains 82 explanatory variables describing every aspect of the home. The dataset is heterogeneous containing both ordinal, nominal, continuous and discrete attributes

Goal

The Goal is to fit any two models as per the specification and predict the Sales price for the test data. The train and test data will split in 70-30 ratio 10 times based on the test ID file which is provided. The end goal is to train the model to achieve test RMSE value for first five split is less than 0.125 and next five splits less than 0.135.

Overall Approach

Read Test data > Data exploration > Pre-Process > Build model > Read Train Data > Predict > Evaluate performance on test data.

Details

1. Read the data from master csv file, which is csv and based on given conditions,
split them into training and testing dataset.

2. It was important to **preprocess test and training dataset separately**, although it would make our life easier but in real world, we don’t have test data while we train model.

3. The basic approach was to **check nulls** and missing data in the data set for the initial
analysis and we could only find Garage_Yr_Blt.

4. First, we used Cook's distance method and find outliers which later we transform with
winsorization.

5. Many of the model like Xboost only work on numerical fields, thus **characterization** was used to convert character features into levels.

6. **Xboost and Lasso/ridge** were attempted and results are described later in detail.

7. The Predicted Sales price was written in two files for each model and later accuracy was calculated using the difference of log of actual price and predicted price.

8. Submission file 1 contains result from prediction from Lasso/Ridge and Submission 2
contains prediction from Xboost.

Feature Engineering

1. It is mandatory to check the data consistency as a first step of exploratory data analysis

2. Checked for the data consistency, whether there are some NULL values in the dataset. Garage_Yr_Blt was found and replaced with all NA values with mean from Ames data. 0 was not assigned to avoid introducing outliers.

3. There were few columns which were removed mainly because 90% of the values were missing like pool area, misc. features.

4. Conversion of character features – they are encoded to numeric types, using a function. This was done since Xboost only take numerical matrix. K dummy variables were created which had binary values.

5. Some numerical features had outliers, e.g. Garage_Yr_Blt there was a value 2207, it was transformed to use quantile value of .95 i.e decreased the magnitude of extreme value. Excluded sales price since it was the response and PID which do not have bearing on the prediction.

6. While pre-processing the test data, same approach was followed. **Train and test data were preprocessed separately.**

7. Same Winsorization function was used on test data but passed the test data quantile values for the most extreme values.

8. Same categorical Imputation function was used on test data. While predicting from model an error “dimensionality mismatch” was seen.

9. To overcome “dimensionality mismatch, we used train data set and all the levels which were not present in test data set were passed to test dataset with default value 0. For the levels which were not in train dataset but in test dataset were removed from test since they will not be part of training the data and default value.

Model Fitting

Linear model with Ridge:

We created a linear model with alpha 0.7 which is with range α∈[0,1] α=1 is the lasso (default) and α=0 is for ridge. This value was adjusted based on the performance on train data.

We then used cv.glmnet ridge regression using 10 folds to calculate the optimized value of **lamda(lambda.min)** which was then passed to the prediction model.

Xboost :

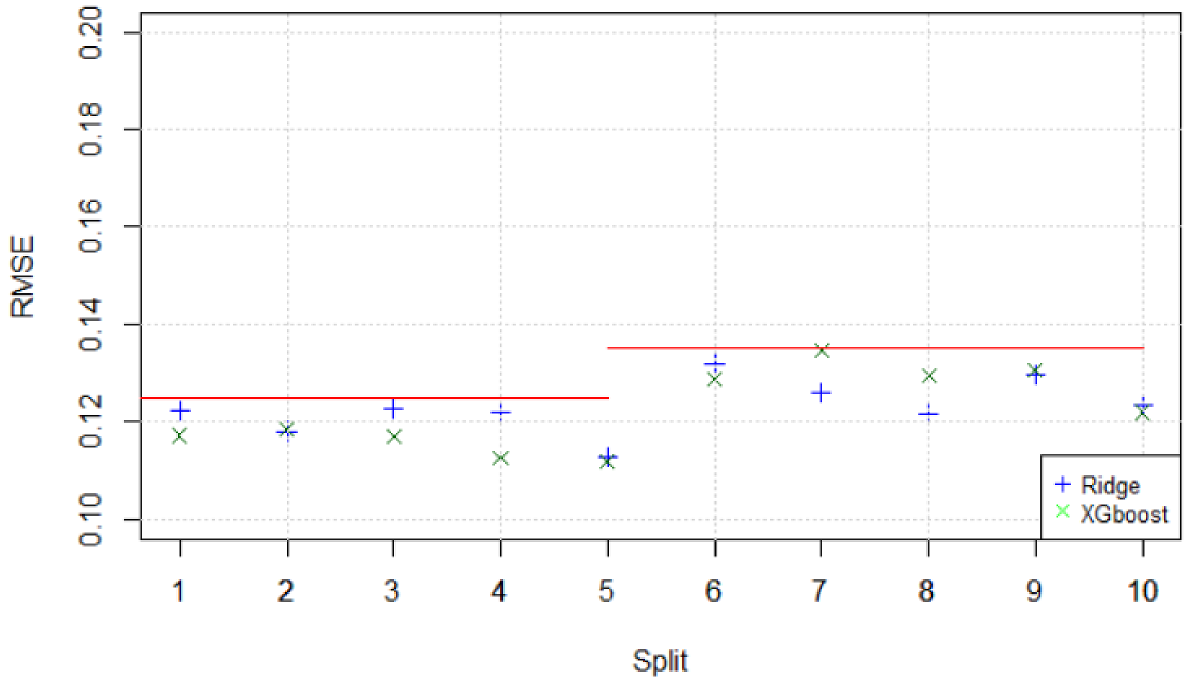
The data was converted into matrix and we took log(response) to improve run time. We kept depth of the trees as 6 since our model was moderate in size and we don’t want the trees to become deep and increase run time and overfit it. (this is also default).
We tried different values with eta and nrounds and came to conclusion that lower eta must be supported by increased nrounds . If we want to decrease the training rate of model (eta-which shrinks the feature weights to make the boosting process more conservative), it is good to increase the number of iterations which is nrounds. Modifying nround=10000 increase the run time by 25%. Setting subsample to 0.5 means that XGBoost would randomly sample all of the training data prior to growing trees and will prevent overfitting.

Evaluation

The models are used to predict on 10 spltis and the results are plotted. We can clearly see that Ridge and Xboost are meeting the evaluation criteria for all 10 splits in below table.

Split	Ridge RMSE	Ridge Runtime (min)	Xgboost RMSE	Xgboost Runtime (min)
1	0.122075503	7.432651043	0.118238907	43.45833087
2	0.118062167	10.92075205	0.118441232	53.79969096
3	0.12284229	7.174040079	0.11708455	58.17491508
4	0.121967522	11.91139889	0.112488774	47.00849199
5	0.112855846	8.637755871	0.111954753	43.22899103
6	0.132054265	10.23921084	0.128714815	53.51037097
7	0.126133742	8.197932005	0.134514693	59.11187911
8	0.12147159	7.992703915	0.129367566	43.9148519
9	0.129547222	4.975697041	0.13065158	39.23618317

10	0.123588276	8.409538031	0.121917233	36.87486506



System specs Windows intel core i7, dual core 16 GB ram.

Learnings –

• Data preparation (cleaning, outlier detection, feature engineering) is the most time-consuming task and if done correctly can boost the performance significantly

• XGboost has slightly higher performance but much longer run times than ridge regression. There is a trade off between run time and performance that is a consideration while choosing the type of model to fit based on application.

• Train and test data should not be preprocessed together while doing outlier detection and winsorization.

References

1. https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html
2. https://www.pluralsight.com/guides/linear-lasso-and-ridge-regression-with-r
3. https://www.kaggle.com/c/house-prices-advanced-regression-techniques