Stat 542 Project I Jinran Yang

Computer System

MacBook Pro (Retina, 13-inch, Mid 2014)
Processor:2.6 GHz Intel Core i5
Memory: 8 GB 1600 MHz DDR3

2.Part I

2.1Data pre-processing

(1) Remove missing value
Replace the missing value in *Garage_Yr_Blt* with the value of Y*ear_Built*. Since there exist some missing value of *Garage_Yr_Blt* and the value of *Year_Built* is likely to be very similar to the corresponding value of *Garage_Yr_Blt*.

(2) Combine/Replace/Remove highly correlated data
I combined some highly correlated data according to the correlations matrix, and created some new variables.

New variable added:
total bath by using *Full_Bath, Half_Bath,Bsmt_Full_Bath* and *Bsmt_Half_Bath;*
Total Square Feet by using G*r_Liv_Area,Total_Bsmt_SF;*
Total Porch in square feet, by using *Open_Porch_SF*, *Enclosed_Porch*
Year by using *Year_Remod_Add*
BsmtFinTotal by using B*smtFin_SF_1* and *BsmtFinSF2*

(3) Remove categorical variables that contain one "dominating" category and some unimportant/irrelevant variables

Drop Variables:
'Longitude', 'Latitude','Full_Bath', 'Half_Bath', 'Bsmt_Full_Bath', 'Bsmt_Half_Bath', 'Gr_Liv_Area','Total_Bsmt_SF','First_Flr_SF', 'Second_Flr_SF', 'Open_Porch_SF', 'Enclosed_Porch', 'Three_season_porch', 'Screen_Porch', 'Bedroom_AbvGr', 'Year_Built','Year_Remod_Add', 'BsmtFin_SF_1','BsmtFin_SF_2','BsmtFin_SF_1', 'BsmtFinSF2', 'Street','Low_Qual_Fin_SF','Condition_2','Pool_QC','Heating','Roof_Matl', 'Utilities','Land_Slope','Misc_Feature','Garage_Cond','Garage_Yr_Blt','Land_Contour','Alley','Central_Air','Misc_Val'

(4) Winsorization

(5) Pre-preprocessing categorical variables—dealing with the levels just exist in the test dataset
Combined some levels of train dataset with few observations, name them *"others"*.
As for the categorical levels which are just in test dataset but not train dataset, if the level is ordered, replace them with the closest ordered categorical level, otherwise, replace them with the most frequent categorical level.

(6) Pre-preprocessing numeric variables—transforming some number variables into factor (E.g Month, Year)

(7) Log_Sale_Price

2.2 Performance

I used Random Forest and Boosting models to predict Sale_Price. The parameter of Xgboost I used are eta = 0.1, max_depth = 10, nrounds = 1000.
The RMSEs for ten spilt datasets are as follow:

```
> result
   RandomForset     Boost
1     0.1257762 0.1253957
2     0.1361554 0.1295987
3     0.1529050 0.1467233
4     0.1435789 0.1349956
5     0.1145866 0.1107200
6     0.1466977 0.1334542
7     0.1299516 0.1162680
8     0.1242585 0.1300983
9     0.1483541 0.1299735
10    0.1283559 0.1253112
> sapply(result,mean)
RandomForset        Boost
   0.1350620    0.1282539
```

Generally, Boost is more accurate. Besides, the computation time for each spilt is around 40 using random forest and 35 using Boost (Xgboost).

3.Part II

Using the same data Pre-preprocessing as the part I and using the Lasso I implement.
And according to the result of "glmnet", I set lambda =20.
The performances are as follow:

```
> RMSE
[1] 0.1389857
> Running_time3 = proc.time()[3]-start_time
> cat("Running_time of mylasso is ",Running_time3,"s")
Running_time of mylasso is  34.267 s
```