Author: Ajit Bhanot (abhanot2@illinois.edu)

CS 598: Practical Statistical Learning: Project 1

Project I: Predicting the Housing Prices in Ames

1. Part I: Predicting with 2 Models

Method: Data Preparation

The data given to us is different from the usual Kaggle data in the sense that it has been cleaned to a large extent. There are some data transformations that can be done (some of them were given to us as hints in Piazza). Along with the same, I took out some of the data points that had a high correlation with each other, for instance, the basement area is quite highly correlated with the first floor area, therefore I got rid of one of the variables.

I combined some of the variables to make a full variable (Example bathrooms can be thought of as the total number of bathrooms on the property as opposed to the individual floors). The cumulative number of bathrooms were used. Some of the variables, when removed, did not add much value such as combining the total area gave us results on the contrary and therefore were added back.

I did apply winsorization on the data for the numeric fields.

Data Inspection through Visualization and Plotting

Upon plotting some of the data, some of the outliers were apparent. Some of them were also mentioned in the piazza post here. In addition to those, I did remove some of the outliers that had an extraordinarily high price.

Inspection of the qq plot for the Sale Price (response) variable also reveals that the data is not normally distributed. I transformed this variable using log scale for our modeling to normalize and then convert it back when calculating the RMSE value for the results.

Model Selection:

1. Lasso Regression

   I first used a the lasso regression from the glmnet function. I used the automatic lambda sequence from the glmnet function to train. The lambda min and lambda.1se values are obtained and used to get the predicted values for the response. Lambda.min produced a better result and therefore was used in the final submission to predict the Sale_Price variable.

2. Extreme Gradient Boosted
   I used the model as my second model for the Part I. R provides a wide variety of parameters to tune. I specified a gaussian distribution (see comment above in the plotting section where the response function was transformed).

2. Part 2: Coordinate Descent for Lasso

This part focusses on writing my own function *using the pseudocode provided on piazza*. I pretty much used the same variables that were used for Part I and nearly the same transformations as in Part I.

3. Results

The results for all three models are below. The execution was done on a Windows 10, Intel i7-7700 CPU @3.60GHz with 16GB RAM.

	Performance			Time		
	Model1: Lasso	**Model2:XBG**	**Model3: CD**	**Model 1: Lasso**	**Model 2: XBG**	**Model3: CD**
Split1	0.1200834	0.1378582	0.1215115	1.25	27.58	**7.4**
Split2	0.1189062	0.1307218	0.1218517	1.53	29.28	**7.31**
Split3	0.1153163	0.1272669	0.1193228	1.49	32.366	**7.26**
Split4	0.1133222	0.1345314	0.1139569	1.52	30.83	**7.42**
Split5	0.1084679	0.1251015	0.1095974	1.49	32.47	**7.42**
Split6	0.1133301	0.1350162	0.1163424	1.47	30.7	**7.25**
Split7	0.1031186	0.1313583	0.1083795	1.5	30.34	**7.35**
Split8	0.1145624	0.1300911	0.115756	1.46	28.16	**7.36**
Split9	0.1217016	0.1379329	0.1248545	1.45	31.2	**7.29**
Split10	0.1138357	0.1450384	0.1147128	1.5	30.73	**7.24**