Ames Housing Data

Ahmad Al-Dhalaan

10/9/2020

Pre-processing

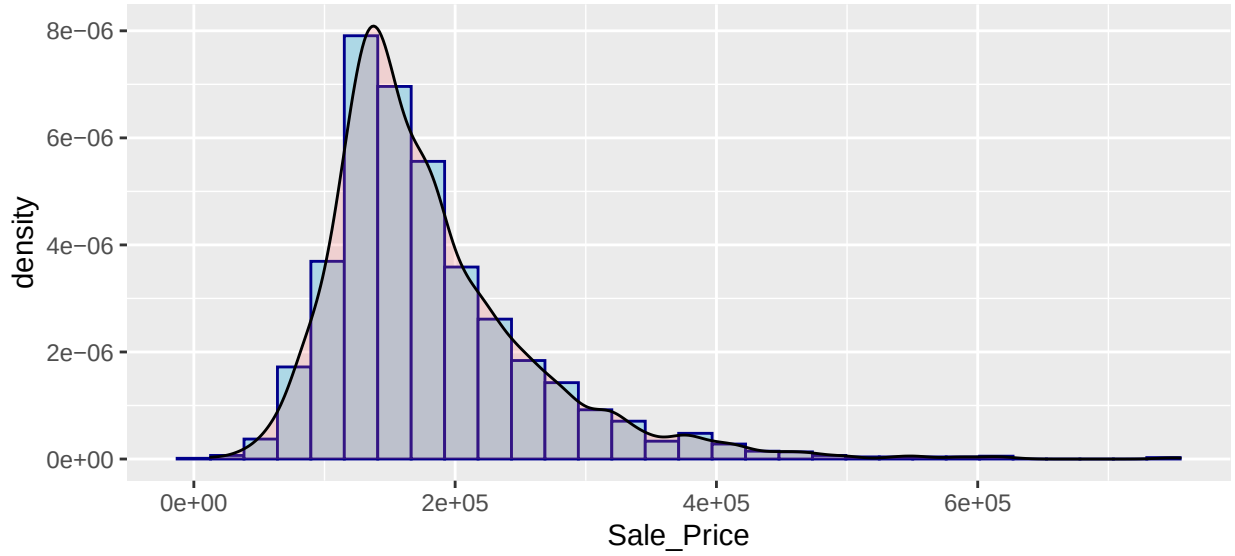
In this report, we build two predictive models to predict the sale price of houses in Ames, Iowa. The dataset consists of 2930 rows and 83 columns. Each row represents a sale (from 2006 to 2010). One column represents the identification number, one column represents the sale price and 81 columns represent possible explanatory variables. To simulate a realistic predictive model, we pre-process the training and test data seperatley.

We note missing values for the variable Garage_Yr_Built. We set those missing values to 0. The variable Year_Remod_Add has many values set at 1950, which are houses built before 1950 with no remodelling information. Therefore, we set these values to 0. We are also more interested in whether remodelling happended or not, so we convert this variable to categorical (1:remodelled, 0:not remodelled). Finally, we introduce a new variable, Total_Area, which adds eight area variables in the dataset into one value. Total area of a house is likely a strong predictor of price and having a variable for it sounds like a reasonable choice.

We remove 11 varaibles from the dataset: 'Street', 'Utilities', 'Condition_2', 'Roof_Matl', 'Heating', 'Pool_QC', 'Misc_Feature', 'Low_Qual_Fin_SF', 'Pool_Area', 'Longitude','Latitude'. These variables contain many missing values and are redundant.

We perform winsorization on 17 continous variables to limit extreme values and reduce the effects of outliers. Most are house area variables with some very big houses that could influence the predictions of our models.

Finally, our the prediction variable Sale_Price is right skewed with a few expensive houses affecting the distribution. We log transform the variable to reduce the influence of these expensive houses.



Implementation

We build two predictive models: one a linear regression model with Lasso penalty and the other a boosting tree model.

The linear regression model is implemented with the glmnet package using cross-validation and an alpha of 1 (representing a Lasso penalty). We use the lambda.min value from cross-validation as it gives better prediction results.

The boosting tree model is implemented with the gbm package. We use a grid search to determine the optimal hyperparameters: shrinkage, interaction depth and bag fraction. We loop through the grid using 1000 trees and determine the hyperparameters that produce the smallest RMSE. These are interaction.depth = 3, shrinkage = 0.1 and bag.fraction = 0.5. We then use 5-fold cross-validation to fit the model using 1000 trees.

We use the dummyVars function from the caret package to convert our categorical variables to dummy variables. Some levels of some categorical variables are present in the train data but not in the test data, and vice versa. Therefore, we perform two actions on the test data. Firstly, we add training data levels (not present in test data) to test data and set as 0. Secondly, we remove levels in test data not in train data. These steps are required to ensure the test data has the same dimensions as the train data, which the model expects in order to predict the response variable. This is specific for the glmnet package and is not done for the gbm package boosting tree model.

Results

We use a 64-bit Windows Operating System - Intel Core i5-3317U CPU with 4GB of RAM to run the ten splits. The running time for the linear model for all ten train/test splits was 36.98 seconds, wheras the running time for the tree model for all ten train/test splits was 469.01 seconds. At a seed of "1234" we produce the following RMSE results. All the RMSE results are less than the benchmark except for the fourth split of the linear model, which is at 0.130.

Table 1: RMSE for train/test splits

Split	Linear with Lasso	Gradient Boosting Machine
1	0.1227187	0.1155072
2	0.1207645	0.1132854
3	0.1233881	0.1075251
4	0.1301672	0.1119097
5	0.1131309	0.1091390
6	0.1325327	0.1266231
7	0.1273417	0.1278648
8	0.1215726	0.1213108
9	0.1309426	0.1281325
10	0.1247869	0.1252639

We note a mean value of 0.0022 for our ten cross-validated λ_{min}. While using λ_{1se} produces a more parsimonious model with less predictors, we were unable to meet the benchmark RMSE results with it. Therefore, λ_{min} produced better accuracy for our purposes and was finally selected for all splits. Since the intent of this project is prediction accuracy this selection should be appropriate.

Our tree model does, however, shed some light on the most influential predictors. The "Overall_Quality" variable, the rate of the overall material and finish of the house (from Very Poor to Very Excellent), was by far the most influential predictor in all splits. The variable "Neighborhood", physical locations within Ames, was the second most influential predictor for seven times, while the variable "Gr_Liv_Area", above grade (ground) living area square feet, was the second most influential predictor for three times.

Table 2: Most influential prediction variables

Split	First	Second	Third
1	Overall_Quality	Neighborhood	Gr_Liv_Area
2	Overall_Quality	Gr_Liv_Area	Neighborhood
3	Overall_Quality	Neighborhood	Gr_Liv_Area
4	Overall_Quality	Gr_Liv_Area	Neighborhood
5	Overall_Quality	Neighborhood	Gr_Liv_Area
6	Overall_Quality	Neighborhood	Gr_Liv_Area
7	Overall_Quality	Gr_Liv_Area	Neighborhood
8	Overall_Quality	Neighborhood	Gr_Liv_Area
9	Overall_Quality	Neighborhood	Gr_Liv_Area
10	Overall_Quality	Neighborhood	Gr_Liv_Area