Mustafa Abdullayev

Regularization Assignment with R

AY6015 – Intermediate Analytics

Term: Winter 2020 - A

Instructor: Joseph Manseau

01/27/2020

# Introduction

This is a Microsoft Word Report accompanying R Script. It contains my R code, outputs, my comments, and findings. In my analysis, I used the [“House Prices: Advanced Regression Techniques”](https://www.kaggle.com/ngee379k/house-prices-advanced-regression-techniques) dataset. There are 81 variables and 1460 observations in this data. 80 of them are explanatory variables such as Lot area and General Living Area. We must use these features to predict the target column namely Final Sale Price (SalePrice). Data was not cleaned and contained missing values. There are both numeric and factor variables. Further explanations can be found [here.](https://www.kaggle.com/lespin/house-prices-dataset#data_description.txt) Since my main aim is to practice regularization techniques, I considered only numeric values to make things a little simpler. My main goal is to utilize R and its regression techniques to analyze data and find insights. I mainly used the glmnet package to create LASSO Regression. I will define the OLS model also to compare it with LASSO model. I used the Root Mean Square Error (RMSE) to measure my models’ accuracy. Also, Powerful R built-in functions and graphs, such as correlation plots, are utilized to dive deeper to observe hidden patterns and visually communicate my findings to the audience. Since I also provided R script with all the codes and comments, I removed some of the codes and comments from my report (such as package loading). It is due to keep my report brief, succinct and to the point.

# Data Preparation and EDA

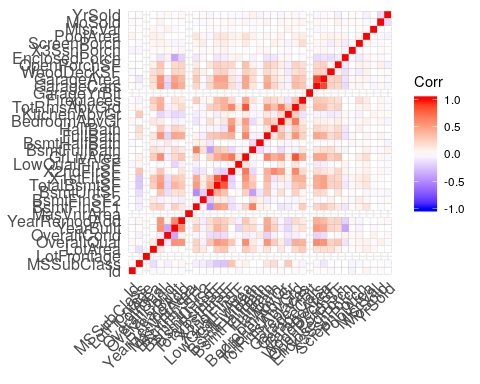
Since I am using only numeric variables for this analysis, I drop all columns with factor type. This will decrease both the accuracy and reliability of my model since there are some important columns such as “Neighborhood”. But, since my goal is to practice the Regularization technique, it is not vital at this point.

The main benefit of the LASSO model over ordinary regression models is that it helps to reduce variance caused by collinearity among variables. But, in the correlation plot, I observed that the correlation between variables is not too high. That hinted to me that LASSO regression is not likely to be extremely better than the ordinary regression model.

## Reading data from computer  
data <- read.csv("housetrain.csv")  
## In order to make things simple, I will only consider numeric values.  
## I will drop columns with factors  
to\_drop <- colnames(data %>% select(which(sapply(.,is.factor))))

my\_data <- data[,!names(data) %in% to\_drop]  
## Lets check the correlation plot   
attach(my\_data)  
ggcorrplot(cor(my\_data[,-c(38)]))

detach(my\_data)



Secondly, I checked if there are columns with missing values. 3 columns, namely LotFrontage, MasVnrArea, and GarageYrBlt had missing values. The percentage of missing values was critical for only the LotFrontage variable (17.74%) but I did not drop it. Instead, I imputed missing values for all three columns with mean values.

## Let’s define a function to check the percentage of missing values for each column  
pMiss <- function(x){ sum(is.na(x))/length(x)\*100}  
apply(my\_data,2,pMiss)

## Id MSSubClass LotFrontage LotArea OverallQual   
## 0.0000000 0.0000000 17.7397260 0.0000000 0.0000000   
## OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1   
## 0.0000000 0.0000000 0.0000000 0.5479452 0.0000000   
## BsmtFinSF2 BsmtUnfSF TotalBsmtSF X1stFlrSF X2ndFlrSF   
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000   
## LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath   
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000   
## HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces   
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000   
## GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF   
## 5.5479452 0.0000000 0.0000000 0.0000000 0.0000000   
## EnclosedPorch X3SsnPorch ScreenPorch PoolArea MiscVal   
## 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000   
## MoSold YrSold SalePrice   
## 0.0000000 0.0000000 0.0000000

## Filling missing values with mean values  
my\_data <- mice(my\_data,m=5,maxit=50,meth='pmm',seed=500)

my\_data <- complete(my\_data,1)

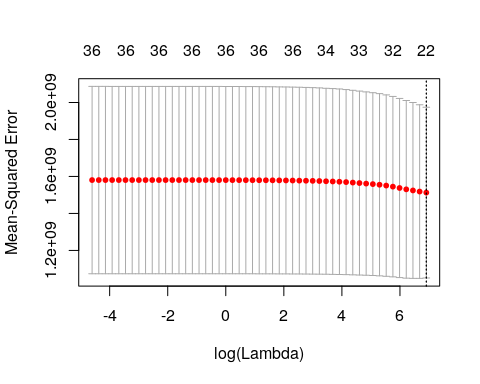
# Regression Models

In the second part, I proceeded to create two separate regression models. Data was partitioned randomly to Train(70 %) and Test(30 %) datasets. I did not need Validation data since I used cross-validation (k-fold) technique to find optimal hyperparameters for my model. For LASSO regression, I only have one hyperparameter – lambda. From the below graph, it is obvious that MSE error decrease as we increase our lambda. According to my code, the best value for lambda was 1000. But due to 2 reasons So, I decided to not to choose lambda to be 1000. Firstly, we can see from the graph that, after some point, increment in lambda affects MSE value very little. Secondly, extremely big lambda values result in model underfitting. Therefore, I choose lambda to be 100.

## Separating data into training and test data  
## Since I will use k-fold validation, I do not need validation subset  
set.seed(1)  
spec <- c(train = .7, test = .3)  
g <- sample(cut(seq(nrow(my\_data)),   
 nrow(my\_data)\*cumsum(c(0,spec)),  
 labels = names(spec)  
))  
res <- split(my\_data, g)  
my\_data\_train <- res$train  
my\_data\_test <- res$test  
  
## Setting my lambda values to check  
lambdas <- 10^seq(3, -2, by = -.1)

## Lets train my lasso model  
## I will use matrix form since glmnet does not accept data frames  
attach(my\_data\_train)  
x <- as.matrix(my\_data\_train[, !(colnames(my\_data\_train) %in% c("SalePrice"))])  
y <- as.matrix(my\_data\_train$SalePrice)  
find\_lambda <- cv.glmnet(x,y,alpha=1,lambda = lambdas)  
plot(find\_lambda)

## I choose my lambda to be 100  
my\_lambda = 100



After creating my model, I observed beta values for all 37 variables. One point to mention, 4 of them reduced to 0, i.e. no effect on our model’s decision. LASSO model decided that these features do not contribute enough to the signal to offset their contribution to noise. Also, our model exaggerated the impact of variables like the General Living Area (GrLivArea). That makes sense since living area is extremely important for everyone who wants to buy a house.

Finally, I put my model to test with Test data. MRSE value for our model is 37509. That means, on average, there is a 37509 USD difference between what the LASSO model shows us and the actual price of a home. Considering the average value of 190000 USD, this is almost a 20% difference.

## Defining my model  
lasso\_reg <- glmnet(x,y,alpha= 1,lambda = my\_lambda,intercept = TRUE)

lasso\_reg$beta

## 37 x 1 sparse Matrix of class "dgCMatrix"  
## Id -1.314719e+00  
## MSSubClass -1.902103e+02  
## LotFrontage -3.929584e+00  
## LotArea 4.805409e-01  
## OverallQual 1.768762e+04  
## OverallCond 4.384516e+03  
## YearBuilt 1.960834e+02  
## YearRemodAdd 1.043633e+02  
## MasVnrArea 2.356203e+01  
## BsmtFinSF1 1.250197e+01  
## BsmtFinSF2 1.972779e+00  
## BsmtUnfSF .   
## TotalBsmtSF 2.125195e+00  
## X1stFlrSF 3.722701e+00  
## X2ndFlrSF .   
## LowQualFinSF -1.321121e+01  
## GrLivArea 4.842001e+01  
## BsmtFullBath 8.877593e+03  
## BsmtHalfBath -2.052030e+03  
## FullBath 4.497180e+03  
## HalfBath .   
## BedroomAbvGr -9.963658e+03  
## KitchenAbvGr -1.441343e+04  
## TotRmsAbvGrd 3.539954e+03  
## Fireplaces 2.869999e+03  
## GarageYrBlt 1.775151e+02  
## GarageCars 1.540266e+04  
## GarageArea -1.175793e+01  
## WoodDeckSF 2.178727e+01  
## OpenPorchSF 3.041770e+00  
## EnclosedPorch 1.018937e+01  
## X3SsnPorch 1.597783e+01  
## ScreenPorch 5.320969e+01  
## PoolArea -1.606587e+01  
## MiscVal .   
## MoSold 1.756601e+02  
## YrSold -9.692876e+02

detach(my\_data\_train)  
## Lets Calculate RMSE - Square root of MSE

attach(my\_data\_test)  
MSE\_lasso <- mean((predict(lasso\_reg,as.matrix(my\_data\_test[,-c(38)])) - my\_data\_test[,38])^2)   
RMSE\_reg <- sqrt(MSE\_lasso)  
print(RMSE\_reg)

## [1] 37509.68

detach(my\_data\_test)

To see if LASSO regression was worth using in that case, I created ordinary regression (OLS). On my train data, that model had an R-squared value of 0.8094. That means our model can explain 80% of all the variability of the response data around its mean. Finally, I used that model to predict Test data and get an RMSE value of 37545 USD. That means, on average, there is a 37545 USD difference between what the OLS model shows us and the actual price of a home.

## Now, Let's define an ordinary least squares regression to see the difference  
attach(my\_data\_train)  
reg\_ols <- lm(SalePrice ~ MSSubClass + LotFrontage + LotArea + OverallQual + OverallCond + YearBuilt + YearRemodAdd + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF +  
 X1stFlrSF + X2ndFlrSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +Fireplaces + GarageYrBlt + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + MiscVal + MoSold + YrSold)

## Multiple R-squared: 0.8158, Adjusted R-squared: 0.8094   
detach(my\_data\_train)  
  
## Calculatin RMSE for OLS Model  
attach(my\_data\_test)   
MSE\_OLS <- mean((predict(reg\_ols,my\_data\_test[,-c(38)]) - my\_data\_test[38])^2)

RMSE\_ols <- sqrt(MSE\_OLS)  
print(RMSE\_ols)

## [1] 37545.43

detach(my\_data\_test)

# Conclusion

To conclude, I analyzed the house price dataset obtained from Kaggle. I performed data preparation and EDA. Also, I utilized both LASSO regression and OLS regression for predictive analysis. In the beginning, since the correlation between variables was small, I hypothesized that the LASSO model is not likely to extremely outperform the OLS model. That is indeed a case. They have almost identical RMSE values (LASSO: 37509 vs OLS: 37545). Moreover, there is an issue in my analysis since I dropped all categorical variables and utilized only numerical ones. There were some important columns such as “ Neighborhood”. The location of the home definitely affects its price. Finally, the last point to mention was about the LotFrontage feature. Almost 17% of its values were missing. Since I already dropped factor columns, I did not want to lose any numerical variables and filled missing values with the mean value.

# References

1. Gaur. (2018, February 12). House Prices: Advanced Regression Techniques. Retrieved from [house-prices-advanced-regression-techniques](https://www.kaggle.com/ngee379k/house-prices-advanced-regression-techniques)