~~~
---
title: "Houses Prices Predictions with Regression-Regularization"
output:
  html_notebook: default
  html_document: default
  word_document: default
---

This is a R Notebook that utilizes Multiple Regression Models to predict House prices.

```{r MultiRegressionModel}
library(doSNOW) # Parallel

RFRun<-TRUE
# Delete all graphs
#
#if(!is.null(dev.list())) dev.off()
node_name<-Sys.info()["nodename"]



# Get Current time
sTime<-Sys.time()
print(sTime)

par(mfrow=c(2,2))

cl <- makeCluster(10, type = "SOCK") #10 parallel processes-RStudio running at the same time


# Register cluster so that caret will know to train in parallel.
registerDoSNOW(cl)

#
# Lasso/Ridge/Elnet with caret Package
#

#MODELS

# set up caret model training parameters
# model specific training parameter
CARET.TRAIN.CTRL <- trainControl(method="repeatedcv",
                                 number=10,
                                 repeats=5,
                                 savePredictions='final',
                                 classProbs = FALSE,
                                 verboseIter=FALSE)

# test out Ridge regression model

lambdas <- seq(1,0,-0.001)

# train model
set.seed(12345)  # for reproducibility
model_ridge <- train(x=X_train,y=y,
                     method="glmnet",
                     metric="RMSE",
                     maximize=FALSE,
                     trControl=CARET.TRAIN.CTRL,
                     tuneGrid=expand.grid(alpha=0, # Ridge regression
                                          lambda=lambdas))


rid.rmse<-mean(model_ridge$resample$RMSE)
rid.mse<-mean(model_ridge$resample$MAE)

model_elnet <- train(x=X_train,y=y,
                     method="glmnet",
                     metric="RMSE",
                     maximize=FALSE,
                     trControl=CARET.TRAIN.CTRL,
                     tuneGrid=expand.grid(alpha=0.5, # Elastic Net regression
                                          lambda=lambdas))



eln.rmse<-mean(model_elnet$resample$RMSE)
eln.mse<-mean(model_elnet$resample$MAE)


# test out Lasso regression model

# train model
model_lasso <- train(x=X_train,y=y,
                     method="glmnet",
                     metric="RMSE",
                     maximize=FALSE,
                     trControl=CARET.TRAIN.CTRL,
                     tuneGrid=expand.grid(alpha=1,  # Lasso regression
                                          lambda=c(1,0.1,0.05,0.01,seq(0.009,0.001,-0.001),
                                                   0.00075,0.0005,0.0001)))
model_lasso

las.rmse<-mean(model_lasso$resample$RMSE, rm.na=TRUE)
~~~

~~~
las.mse<-mean(model_lasso$resample$MSE, rm.na=TRUE)


if (RFRun==TRUE) {

# train model
  model_rf <- train(x=X_train,y=y,
               #      method="rf",
                    method="rf",
                     metric="RMSE",
                     maximize=FALSE,
                     trControl=CARET.TRAIN.CTRL)
  model_rf

  rf.rmse<-mean(model_rf$resample$RMSE, rm.na=TRUE)
  rf.mse<-mean(model_rf$resample$MSE, rm.na=TRUE)

}


stopCluster(cl)

print (paste("Lasso RMSE : ",las.rmse))
print (paste("Ridge RMSE : ",rid.rmse))
print (paste("Elnet RMSE : ",eln.rmse))

if (RFRun==TRUE) {
  print (paste("RF RMSE : ",rf.rmse))
}
# transform train to log+1
# LOG, SQRT, NONE
if (xrf=="LOG") {
    las.preds<-exp(predict(model_lasso,newdata=X_test)) - logC
    rid.preds<-exp(predict(model_ridge,newdata=X_test)) - logC
    eln.preds<-exp(predict(model_elnet,newdata=X_test)) - logC
    if (RFRun==TRUE) {
      rf.preds<-exp(predict(model_rf,newdata=X_test)) - logC
    }
} else if (xrf=="SQRT") {
    las.pred<-predict(model_lasso,newdata=X_test)
    las.preds<-las.pred^2
    rid.pred<-predict(model_ridge,newdata=X_test)
    rid.preds<-rid.pred^2
    eln.pred<-predict(model_elnet,newdata=X_test)
    eln.preds<-eln.pred^2
    if (RFRun==TRUE) {
      rf.pred<-predict(model_rf,newdata=X_test)
      rf.preds<-rf.pred^2
    }
} else {
    las.preds<-predict(model_lasso,newdata=X_test)
    rid.preds<-predict(model_ridge,newdata=X_test)
    eln.preds<-predict(model_elnet,newdata=X_test)
    if (RFRun==TRUE) {
      rf.preds<-predict(model_rf,newdata=X_test)
    }

}


hist(model_ridge$resample$Rsquared)
hist(model_lasso$resample$Rsquared)
hist(model_elnet$resample$Rsquared)
hist(model_rf$resample$Rsquared)


eTime<-Sys.time()
print(paste0("Start Time: ",sTime))

print(paste0("End  Time: ",eTime))

print(paste0("Elapsed Time: ",eTime-sTime))




```
~~~