##### Kaggle #5 Code SL.nnet

Data = read.csv("Data.csv", stringsAsFactors = FALSE)

n = dim(Data)[1] # The number of rows of the data.frame

### ID is the 43th column in the dataset

# The code below generates a spread sheet with 101 predictors (X1 - X101)

# and the word corresponding to the names of predictors

iters = c(5:7, 13:42, 44:dim(Data)[2])

k = integer(1)

predictor.table = data.frame(predictor = character(101), idno = 1:101, stringsAsFactors = FALSE)

for (i in iters) {

k = k + 1

predictor.table$predictor[k] = names(Data)[i]

names(Data)[i] = paste("X", k, sep = "")

}

write.csv(predictor.table, "predictor\_table.csv", row.names = FALSE)

predictors = Data[,-c(1:4,8:13,43)]

#superLearner

if(!require(SuperLearner)) {

install.packages("SuperLearner"); require(SuperLearner)}

if(!require(gam)) {

install.packages("gam"); require(gam)}

if(!require(gbm)) {

install.packages("gbm"); require(gbm)}

if(!require(randomForest)) {

install.packages("randomForest"); require(randomForest)}

if(!require(nnet)) {

install.packages("nnet"); require(nnet)}

if(!require(glmnet)) {

install.packages("glmnet"); require(glmnet)}

if(!require(polspline)) {

install.packages("polspline"); require(polspline)}

SL.library <- c("SL.glm", "SL.mean",

"SL.glmnet","SL.gam","SL.polymars", "SL.nnet")

#"step", "SL.gbm", "SL.randomForest"

set.seed(100)

imputed\_missForest = read.csv("imputed\_missFOrest.csv", stringsAsFactors = FALSE)

predictors$X5=imputed\_missForest$X5

#here Gaussian() fits a continuous Y and binomial() fits a binary Y

Yelpstars.SL <-SuperLearner(Y=Data$stars,X=predictors, SL.library=SL.library,

family=gaussian(), method="method.NNLS", verbose=TRUE)

save.image()

Yelpstars.SL\_simple = Yelpstars.SL

Test1 = read.csv("imputed\_Test1.csv", stringsAsFactors = FALSE)

result1 = predict(Yelpstars.SL\_simple, newdata = Test1)

result.frame1 = data.frame(Id = 1:dim(Test1)[1], Prediction = result1$pred)

Test2 = read.csv("imputed\_Test2.csv", stringsAsFactors = FALSE)

result2 = predict(Yelpstars.SL\_simple,newdata = Test2)

result.frame2 = data.frame(Id = 1:dim(Test2)[1], Prediction = result2$pred)

n1 = dim(Test1)[1]

n2 = dim(Test2)[1]

id = 1:(n1 + n2)

k = integer(1)

m = integer(1)

result.frame = data.frame(Id = id, Prediction = numeric(length(id)))

for (i in id) {

k = k + 1

if (k <= n1) {

result.frame$Prediction[i] = result.frame1$Prediction[k]

} else {

m = m + 1

result.frame$Prediction[i] = result.frame2$Prediction[m]

}

}

for (i in seq\_len(length(result.frame$Id))) {

if (result.frame$Prediction[i] < 1) {

result.frame$Prediction[i] = 1

} else if (result.frame$Prediction[i] > 5) {

result.frame$Prediction[i] = 5

}

}

write.csv(result.frame, "group1.csv", row.names = FALSE)