rm(list = ls())

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(1:4,8:12,43)]

#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.randomForest")

#"step", "SL.gbm",

set.seed(100)

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

predictors=imputed\_missForest

#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()