setwd(choose.dir())

###uploading the data

data1= read.csv("LSM\_data\_latest.csv")

head(data1)

rm(data1)

## split

library(caTools)

set.seed(3210)

s=sample(1:nrow(data1),0.7\*nrow(data1))

train=data1[s,]

split <- sample.split(data1$mvo,0.7)

str(data)

train <- subset(data1,split ==T)

test <- subset(data1,split ==F)

head(data1)

rm(split)

rm(data1)

summary(test)

rm(y)

nrow(train)

ifelse (train$state=="Jharkhand", nrow(train$state),0)

y<- subset(test,test$state=="Jharkhand")

nrow(y)

###############################GLM###################################################

### use as.factor where binary flag

#####Build Model

model <- glm(mvo ~ brand+

               TYPE+

               exwar+

               custtype+

               zone+

               AO+

               state+

               accbkt+

               servbkt+

               spndbkt+

               odmbkt+

               compbkt+

               as.factor(month)

             ,data = train,family = 'binomial')

summary(model)

rm(split)

predictions <- predict(model, data =test,type = 'response')

#CONFUSION MATRIX

table(actual = test$mvo,predicted = predictions>0.4)

#####Predict

predictions <- predict(model,data= test,type='response')

head(predictions,10)

summary(predictions)

write.csv(predictions,"./pred.csv")

getwd()

summary(predictions)

##AUC to check threshold

install.packages("aod")

library(aod)

library(ROCR)

library(caret)

confusionMatrix(test)

predict.rocr <- prediction(predictions,test$mvo)

plot(performance(predict.rocr,'tpr','fpr'),colorize=T)

predictions\_10 <- as.numeric(predictions>0.4)

#### this gives the confusion matrix####

table(actual=test$mvo,predicted = predictions\_10)

### auc value

auc <- as.numeric(performance(predict.rocr,'auc')@y.values)

auc

results <- as.data.frame(cbind(actual=test$mvo,predicted = predictions))

library(dplyr)

results$decile <-  ntile(-results$predicted,10)

library(data.table)

results.dt <- data.table(results)

results.dt [order(decile),.(count = .N,Actual\_ones = sum(actual),perc\_conv =sum(actual)/.N) , decile]

summary(model)

####################out of time validtaion

#####Predict remaining data

test1=read.csv('DATA\_SET\_OTV.csv')

predictions <- predict(model,newdata= test1,type='response')

summary(predictions)

summary(predictions)

##AUC to check threshold

library(ROCR)

predict.rocr <- prediction( predictions,test1$TARGET)

plot(performance(predict.rocr,'tpr','fpr'),colorize=T)

predictions\_10 <- as.numeric(predictions>0.19)

table(actual=test1$TARGET,predicted = predictions\_10)

auc <- as.numeric(performance(predict.rocr,'auc')@y.values)

auc

results <- as.data.frame(cbind(actual=test1$TARGET,predicted = predictions))

library(dplyr)

results$decile <-  ntile(-results$predicted,10)

library(data.table)

results.dt <- data.table(results)

results.dt [order(decile),.(count = .N,Actual\_ones = sum(actual),perc\_conv =sum(actual)/.N) , decile]

summary(model)