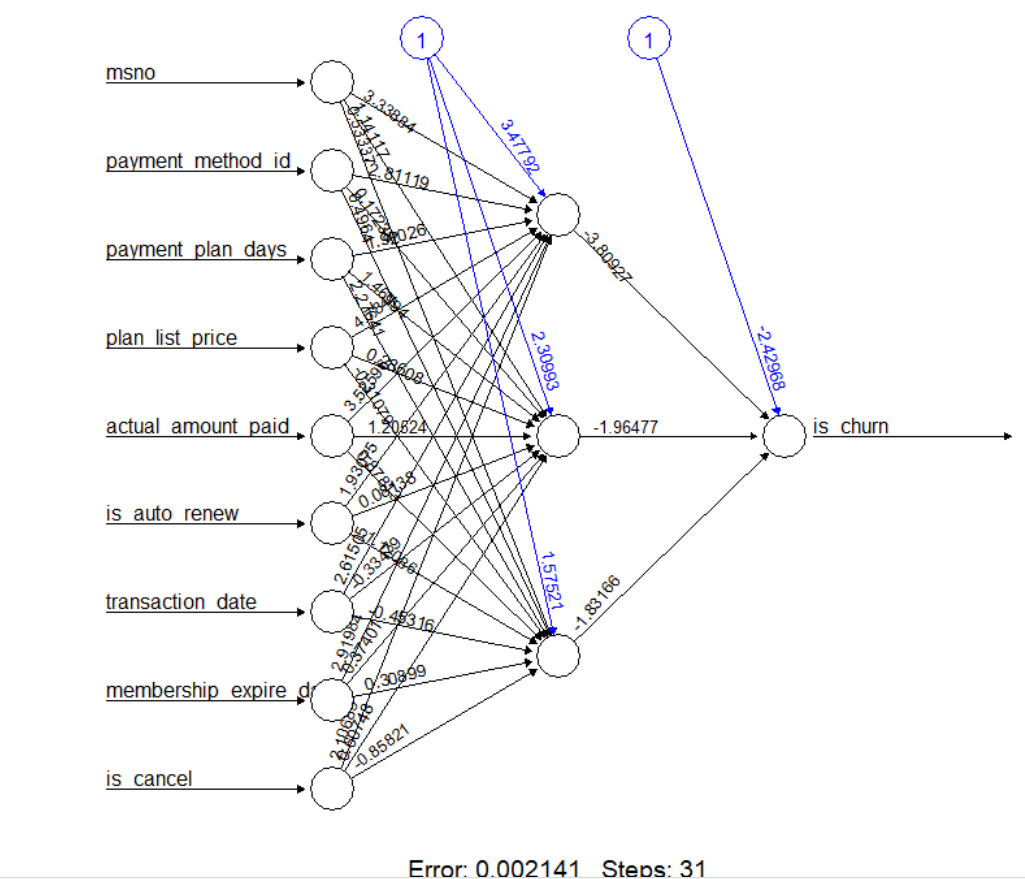
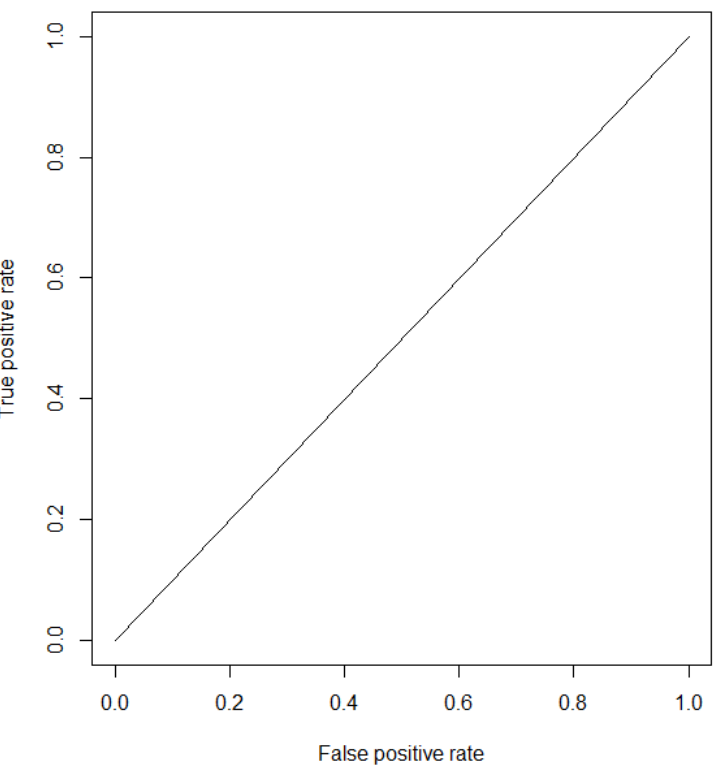
Connor Watson  
Last HW CS 636  
Neural Network Portion





area under curve is 0.5

#Connor Watson

#Lab9 Codes

train\_data = read.csv('/Users/watson/Documents/cs636/kkbox/train\_v2.csv', header = T)

trans\_data = read.csv('/Users/watson/Documents/cs636/kkbox/transactions\_v2.csv', header = T)

merged <- merge(train\_data, trans\_data, by="msno")

sample\_data <- merged

sample\_data$is\_churn <- as.numeric(sample\_data$is\_churn)

sample\_data$msno <- as.numeric(sample\_data$msno)

data <- (sample\_data - min(sample\_data, na.rm=TRUE))/(max(sample\_data,na.rm=TRUE) -

min(sample\_data, na.rm=TRUE))

set.seed(17)

ind <- sample(2, nrow(data), replace =T, prob = c(0.7,0.3))

train\_1 <-data[ind==1,]

test\_1 <- data[ind==2,]

library(neuralnet)

names <- names(train\_1)

formula <- as.formula(paste("is\_churn ~", paste(names[!names %in% "is\_churn"], collapse = " + ")))

nmodel <- neuralnet(formula, data=train\_1, hidden=3,

linear.output = FALSE)

plot(nmodel)

p <- nmodel$net.result[[1]]

pred <- compute(nmodel,train\_1[,-1])

p\_res <- pred$net.result

p\_res <- ifelse(p\_res>0.5,1,0)

tab\_res <- table(predicted=p\_res, Actual = train\_1$is\_churn)

tab\_res

(1- sum(diag(tab\_res)/sum(tab\_res))) \* 100

(sum(diag(tab\_res)/sum(tab\_res))) \* 100

pred <- compute(nmodel,test\_1[,-1])

p\_res\_2 <- pred$net.result

p\_res\_2 <- ifelse(p\_res\_2>0.5,1,0)

tab\_res\_2 <- table(predicted=p\_res\_2, Actual = test\_1$is\_churn)

tab\_res\_2

(1- sum(diag(tab\_res\_2)/sum(tab\_res\_2))) \* 100

(sum(diag(tab\_res\_2)/sum(tab\_res\_2))) \* 100

detach(package:neuralnet, unload = T)

library(ROCR)

nn.pred = prediction(p\_res\_2, test\_1$is\_churn)

pref <- performance(nn.pred, "tpr", "fpr")

plot(pref)

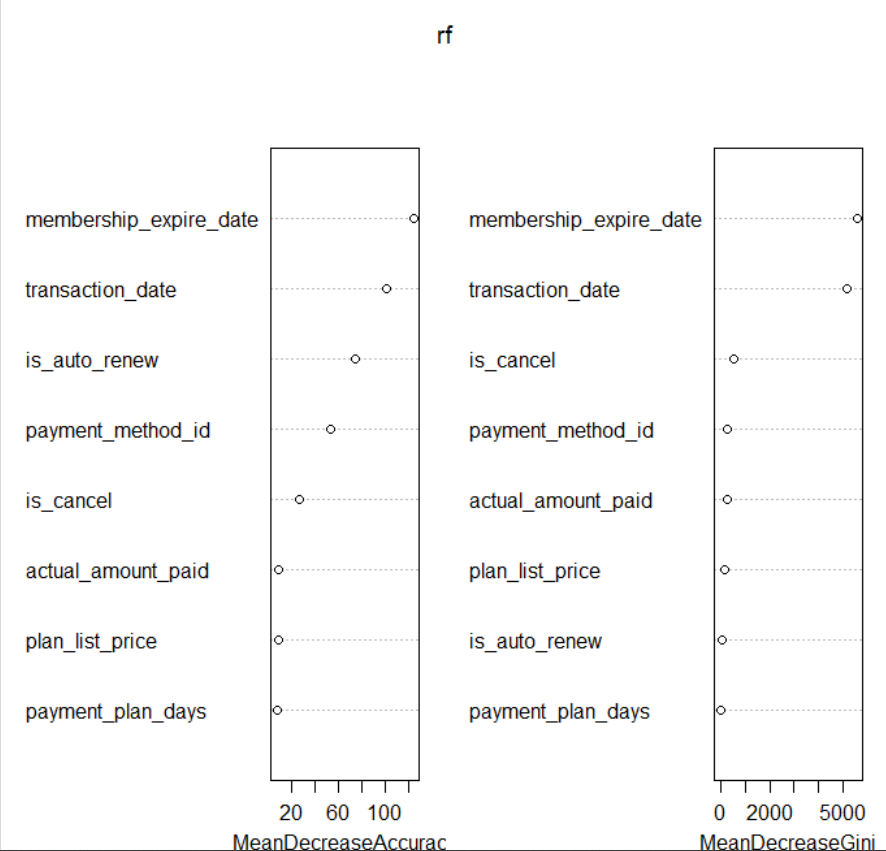
auc.perf = performance(nn.pred, measure = "auc")

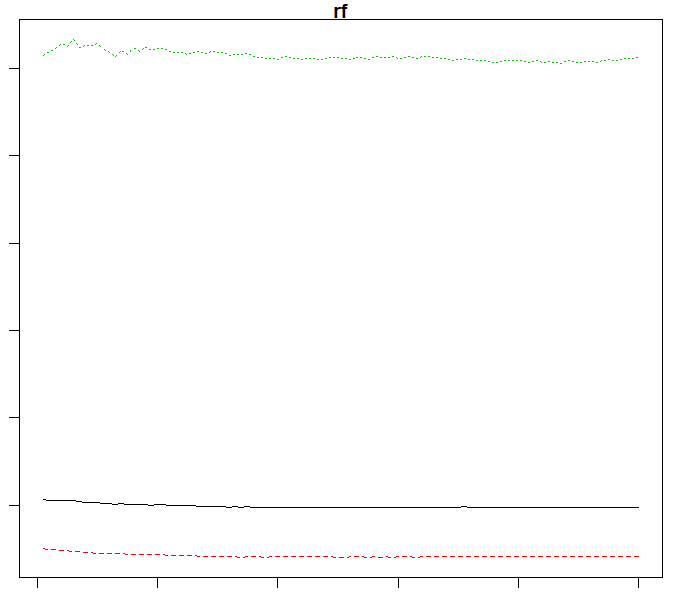
auc.perf@y.values #area is 0.5

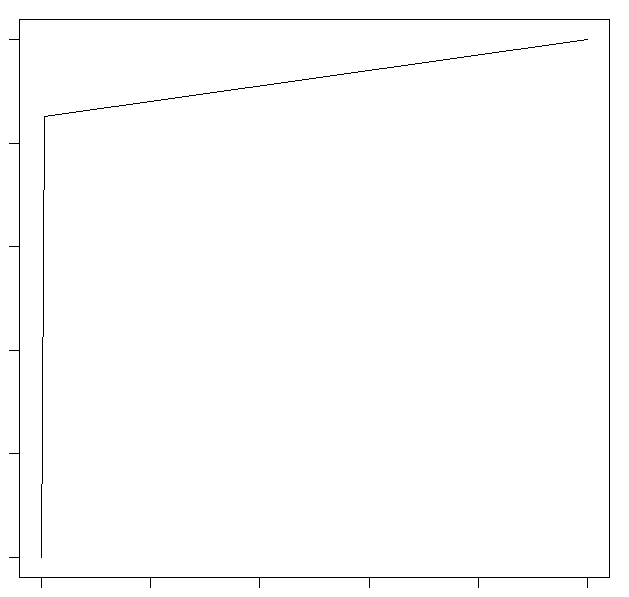
rm(list = ls())

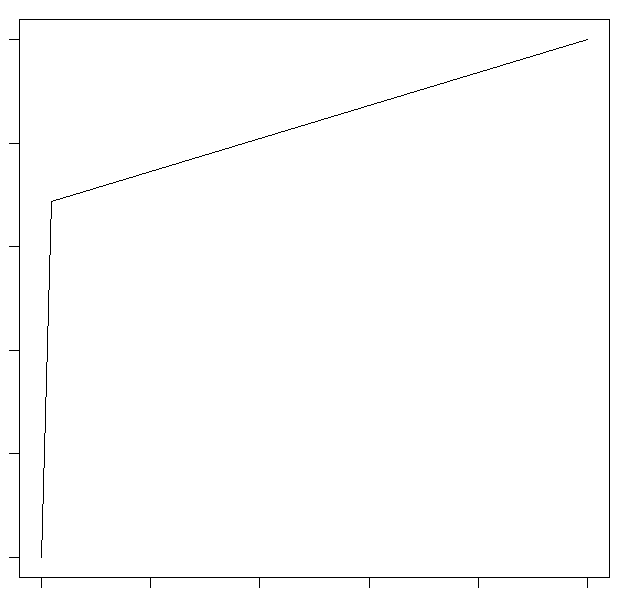
detach(package:ROCR, unload = T)

Random Forest Portion





 Train Plot area = 0.923212683

Test Plot area = 0.8346216455

train\_data = read.csv('/Users/watson/Documents/cs636/kkbox/train\_v2.csv', header = T)

trans\_data = read.csv('/Users/watson/Documents/cs636/kkbox/transactions\_v2.csv', header = T)

merged <- merge(train\_data, trans\_data, by="msno")

set.seed(17)

ind <- sample(2, nrow(merged), replace = T, prob = c(0.1, 0.9))

sample\_data <- merged[ind == 1,]

data <- sample\_data

data$is\_churn <- as.factor(data$is\_churn)

set.seed(17)

ind <- sample(2, nrow(data), replace = T, prob = c(0.7, 0.3))

train\_2 <- data[ind==1,]

test\_2 <- data[ind==2,]

#install.packages("randomForest")

library(randomForest)

rf <- randomForest(is\_churn ~ + payment\_method\_id + payment\_plan\_days + plan\_list\_price + actual\_amount\_paid +

is\_auto\_renew + transaction\_date + membership\_expire\_date + is\_cancel, train\_2,

importance = T, ntree = 100, mtry = 8)

rf

#install.packages('e1071', dependencies=TRUE)

library(caret)

library(ROCR)

library(pROC)

pred\_1 <- predict(rf, train\_2)

confusionMatrix(pred\_1, train\_2$is\_churn)

pred1\_ints <- ifelse(as.numeric(pred\_1)==1,0,1)

pred1\_ints

train\_pred <- prediction(pred1\_ints, train\_2$is\_churn)

train\_perf <- performance(train\_pred, 'tpr', 'fpr')

plot(train\_perf)

auc.perf\_1 = performance(train\_pred, measure = "auc")

auc.perf\_1@y.values

pred\_2 <- predict(rf, test\_2)

confusionMatrix(pred\_2, test\_2$is\_churn)

pred2\_ints <- ifelse(as.numeric(pred\_2)==1,0,1)

pred2\_ints

test\_pred <- prediction(pred2\_ints, test\_2$is\_churn)

test\_perf <- performance(test\_pred, 'tpr', 'fpr')

plot(test\_perf)

auc.perf\_2 = performance(test\_pred, measure = "auc")

auc.perf\_2@y.values

varImpPlot(rf)

plot(rf)