5\_Report\_ChoosingBestMethod.R

setwd("")  
library(Metrics)  
########################################################################  
#Compare all models based on AUC  
########################################################################  
#we will perform a model comparison across all types of models that we've learned about so far  
#Decision Trees  
#Bagged Trees  
#Random Forest  
#Gradient Boosting Machine  
#The models were all trained on the same training set  
credit <- read.csv("credit.csv", stringsAsFactors = TRUE)  
########################################################################  
#Split data in 80% 20%  
########################################################################  
# Total number of rows in the credit data frame  
n <- nrow(credit)  
# Number of rows for the training set (80% of the dataset)  
n\_train <- round(0.8 \* n)   
# Create a vector of indices which is an 80% random sample  
set.seed(123)  
train\_indices <- sample(1:n, n\_train)  
# Subset the credit data frame to training indices only  
credit\_train <- credit[train\_indices, ]   
# Exclude the training indices to create the test set  
credit\_test <- credit[-train\_indices, ]  
########################################################################  
#Retrieve the prediction from the four models we studied  
########################################################################  
dt\_preds <- readRDS("dt\_preds")  
bag\_preds <- readRDS("bag\_preds")  
rf\_preds <- readRDS("rf\_preds")  
gbm\_preds <- readRDS("gbm\_preds")  
  
########################################################################  
#Compare all models based on AUC  
########################################################################  
  
# Generate the test set AUCs using the two sets of predictions & compare  
actual <- ifelse(credit\_test$default == "yes", 1, 0)  
dt\_auc <- auc(actual = actual, predicted = dt\_preds)  
bag\_auc <- auc(actual = actual, predicted = bag\_preds)  
rf\_auc <- auc(actual = actual, predicted = rf\_preds)  
gbm\_auc <- auc(actual = actual, predicted = gbm\_preds)  
  
# Print results  
sprintf("Decision Tree Test AUC: %.3f", dt\_auc)

## [1] "Decision Tree Test AUC: 0.686"

sprintf("Bagged Trees Test AUC: %.3f", bag\_auc)

## [1] "Bagged Trees Test AUC: 0.776"

sprintf("Random Forest Test AUC: %.3f", rf\_auc)

## [1] "Random Forest Test AUC: 0.812"

sprintf("GBM Test AUC: %.3f", gbm\_auc)

## [1] "GBM Test AUC: 0.786"

#Random Forest performed the best on the test set  
#a bit more tuning of the GBM, the performance might be closer to that of the Random Forest  
  
########################################################################  
#Plot & compare ROC curves  
########################################################################  
#We conclude this course by plotting the ROC curves for all the models   
# The ROCR package provides the prediction() and performance() functions which generate the data required for plotting the ROC curve  
  
#The more "up and to the left" the ROC curve of a model is, the better the model.  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

# List of predictions  
preds\_list <- list(dt\_preds, bag\_preds, rf\_preds, gbm\_preds)  
  
# List of actual values (same for all)  
m <- length(preds\_list)  
actuals\_list <- rep(list(credit\_test$default), m)  
  
# Plot the ROC curves  
pred <- prediction(preds\_list, actuals\_list)  
rocs <- performance(pred, "tpr", "fpr")  
plot(rocs, col = as.list(1:m), main = "Test Set ROC Curves")  
legend(x = "bottomright",   
 legend = c("Decision Tree", "Bagged Trees", "Random Forest", "GBM"),  
 fill = 1:m)

