**Assignment 1: Advanced Classification**

**Student Name: W. Joseph Wadford**

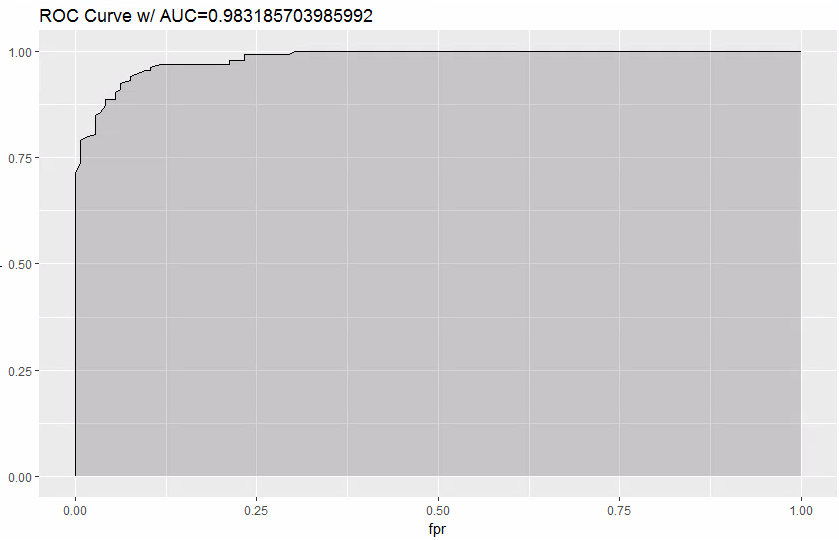
1. Use train\_data to build a random forest classifier with 10 trees. Use library(randomForest).
   1. **(2 points)** What is the OOB estimate of error rate? 9.02%
   2. **(2 points)** How many variables R tried at each split? **5**
   3. **(4 points)** Now use 20 trees.
      1. What is OOB estimate of error rate? 8.46%
      2. How many variables R tried at each split? 5
   4. **(4 points)** Now use 30 trees.
      1. What is OOB estimate of error rate? 5.85%
      2. How many variables R tried at each split? 5
   5. **(2 points)** Increase the number of trees in 10 increments (e.g. 40, 50, …). Using OOB error rate to evaluate your random forest classifier, how many trees would you recommend?
      1. I would recommend 60 trees
   6. **(2 points)** Use tuneRF() function to find the best value for mtry. Here is the code:

*mtry <- tuneRF(train\_data[-26], train\_data$gen\_election, ntreeTry=n, stepFactor=1.5, improve=0.01, trace=TRUE, plot=TRUE, , na.action=na.exclude).* Replace n with the number of trees you recommended in 9.5. What is the recommended value for mtry? 4

* 1. **(2 points)** Use your recommended number of trees and mtry value to build a new random forest classifier using train\_data. What is OOB estimate of error rate? 6.77%
  2. **(8 points)** Use library(caret)[[1]](#footnote-1) and the code in Module 6 to create the confusion matrix for test\_data. Fill out the confusion matrix in below. Use “W” as the value of option positive in confusionMatrix() function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | W | L |
| W | **124** | **11** |
| L | **9** | **135** |

* + 1. What is the value of accuracy? 0.9283
    2. What is the value of TPR? 0.9323
    3. What is the value of FPR? 0.0753
  1. **(4 points)** Use the code in Module 6 to calculate AUC and create the ROC curve.
     1. What is the value of AUC? 0.983185703985992
     2. Paste the ROC curve in the space below:

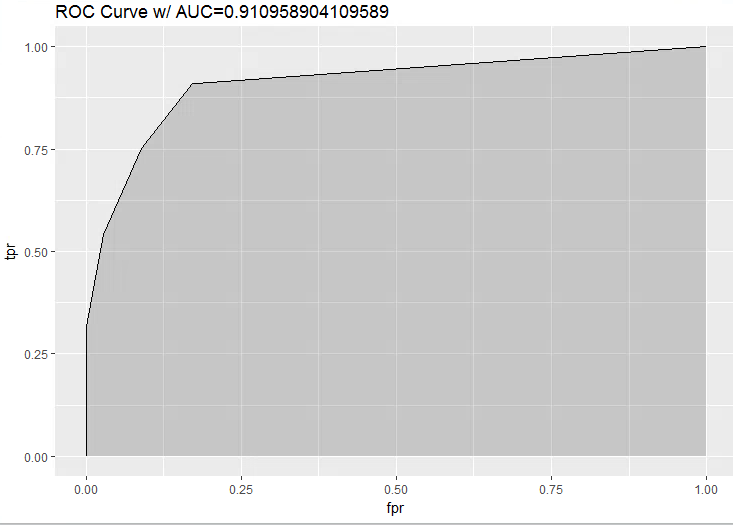


* 1. **(4 points)** Use varImpPlot() to create the plot for variable importance. What are the type five important variables when we use MeanDecreaseAccuracy?
     1. (Opp\_fund) Opposition fund, ending cash, contributions from other political committees, beginning cash, and facebook.

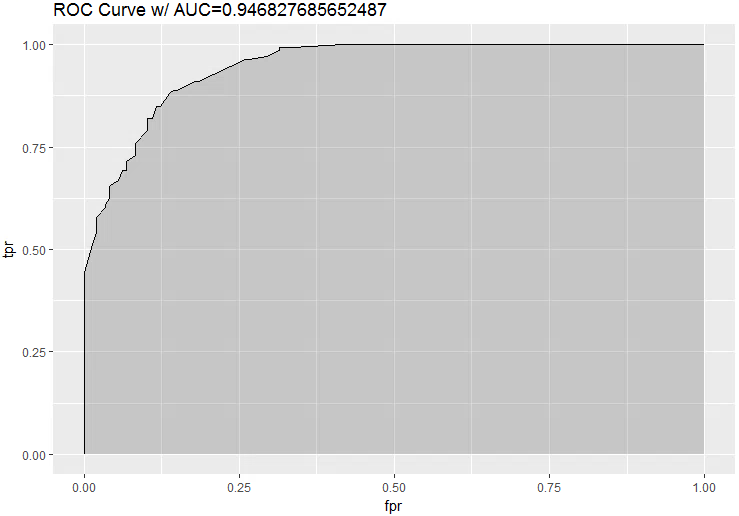
1. Use library(nnet) and the code in Module 6 to build a neural network classifier.
   1. **(20 points)** Use 5 hidden nodes in your ANN.
      1. How many input nodes are in the ANN? 39
      2. How many weights are in the ANN? 206
      3. Use library(caret) and the code in Module” 6 to create the confusion matrix for test\_data. Fill out the confusion matrix in below. Use “W” as the value of option positive in confusionMatrix() function.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual | | |
| Predicted |  | W | L |
| W | **100** | **13** |
| L | **33** | **133** |

* + 1. What is the value of sensitivity? 0.7519
    2. What is the value of specificity? 0.9110
    3. Use the code in Module 6 to calculate AUC and create the ROC curve.
       1. What is the value of AUC? 0.910958904109589
       2. Paste the ROC curve in the space below:

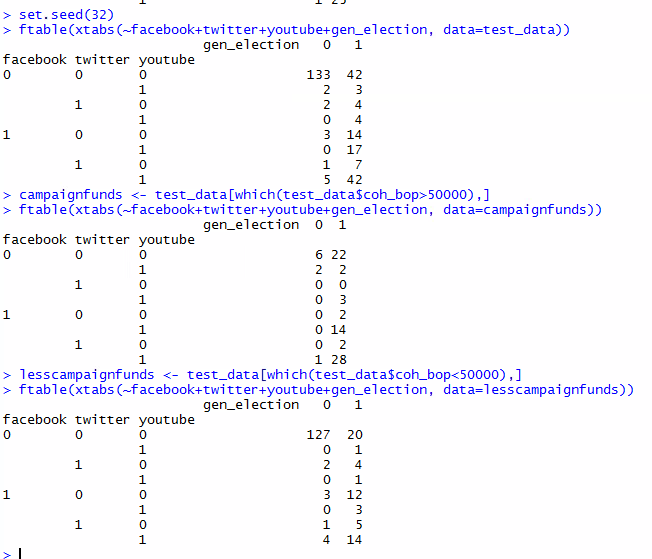


* 1. **(6 points)** Increase the number of hidden nodes until you get the following error: “*Error in nnet.default(x, y, w, entropy = TRUE, ...): too many (1026) weights.”* Use the maximum number of hidden nodes that you can use to build your ANN classifier.
     1. What is the maximum number of hidden nodes that we could use? 24
     2. Use the code in Module 6 to calculate AUC and create the ROC curve.
        1. What is the value of AUC? 0.946827685652487
        2. Paste the ROC curve in the space below:



1. **(5 points)** Among the three classifiers that you built, which classifier would you finally use for predicting the election’s outcome? Please explain.
   1. I would use the random forest classifier for predicting the election’s outcome. The AUC for the ROC curve is the highest for the random forest classifier. This means that the random forest classifier maximizes our ability to find a decision that maximizes our specificity and sensitivity. In simpler, yet vaguer, terms – a AUC closer to 1 increases our ability to discriminate between correctly and incorrectly predicting a general election win or loss.
2. **(10 points)** The buzz from the 2008 election motivated the candidates for political offices to employ social media campaigns to get their message across. Imagine that you are an advisor to a candidate who is running for a Congressional seat. Based on your analysis, would you recommend sparing money and resources to create social media campaigns? If so, among the three social media platforms (Facebook, Twitter, and YouTube), which platform would you recommend to invest in? Please explain. You can use function ftable() in Module 3 to support your recommendation.
   1. Generally speaking, campaigns with social media accounts had much better win-loss ratios than campaigns without social media accounts. After creating a subset of the test data where coh\_bop > $50,000 and comparing that to another subset wherein coh\_cop < $50,000, the results become much more clear. When candidates had more than $50,000 at the beginning of their campaign, the difference in win-loss ratio was significant but not as dramatic as when candidates had less than $50,000. In other words, when candidates had less money – having a social media account made a huge difference. Based on this table alone, I would conclude that having a social media account is very important ESPECIALLY if your candidate has less money.

It seems as though having a facebook is the most important social media account to have but facebook also seems to work well in conjunction with twitter and youtube.

* 1. 

1. **(10 points)** Given your analysis, would you agree with this statement: “Money Buys Political Power”? Please explain.
   1. Yes, money does buy political power. You can see that the more money a campaign has, the more likely it is to win (see table below in section 5.2). However if you review the analysis for question 4, you can see that social media helps to reduce the effect of money in politics.
   2. 
2. **(10 points)** Imagine that you are an advisor to a candidate who is running for a Congressional seat. Based on your analysis, what are your prescriptions for success for your candidate? Please explain.
   1. My advice would be to network in order to maximize donations, especially from other political committees. Additionally, I would highly recommend that my candidate hire social media experts to leverage the power of social media in order to maximize the efficiency of their cash flow. I would also highly recommend utilizing the social media capabilities of other political committees to her/his benefit. For reference see the answers for questions: 1.10.1; 1.9.2; 4.1; 4.2; 5.1; 5.2.
3. **(5 points)** Please paste your R code in the space below:

data <- read.csv("~/MBAD 6211 advanced business analytics/assignments/election\_campaign\_data.csv", sep=",", header=T, strip.white=T, na.strings=c("NA","NaN","","?"))

summary(data)

data$cand\_id <- NULL

data$last\_name <- NULL

data$first\_name <- NULL

data$twitterbirth <- NULL

data$facebookdate <- NULL

data$facebookjan <- NULL

data$youtubebirth <- NULL

data$twitter <- factor(data$twitter)

data$facebook <- factor(data$facebook)

data$youtube <- factor(data$youtube)

data$cand\_ici <- factor(data$cand\_ici)

data$gen\_election <- ifelse(data$gen\_election == "W", 1, 0)

data$gen\_election <- factor(data$gen\_election)

summary(data)

set.seed(32)

data<- data[complete.cases(data), ]

complete.cases(data)

n = nrow(data)

trainIndex = sample(1:n,

size = round(0.7\*n),

replace=FALSE)

train\_data = data[trainIndex,]

test\_data = data[-trainIndex,]

summary(train\_data)

library(randomForest)

rf <-randomForest(gen\_election~., data=train\_data, ntree=10, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=20, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=30, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=40, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=50, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=60, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=70, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=80, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=90, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=100, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=110, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=120, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=130, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=140, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=150, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=160, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=170, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=180, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=190, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=200, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=210, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=220, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=230, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

rf <-randomForest(gen\_election~., data=train\_data, ntree=240, na.action=na.exclude, importance=T,

proximity=T)

print(rf)

set.seed(32)

mtry <- tuneRF(train\_data[-26], train\_data$gen\_election, ntreeTry=60, stepFactor=1.5, improve=0.01, trace=TRUE, plot=TRUE, , na.action=na.exclude)

best.m <- mtry[mtry[, 2] == min(mtry[, 2]), 1]

print(mtry)

print(best.m)

set.seed(32)

rf <-randomForest(gen\_election~., data=train\_data, mtry=best.m, importance=TRUE, ntree=60)

print(rf)

library(caret)

library("e1071")

set.seed(32)

predicted\_values <- predict(rf, test\_data,type= "prob")

head(predicted\_values)

threshold <- 0.5

pred <- factor( ifelse(predicted\_values[,2] > threshold, 1, 0) )

levels(test\_data$gen\_election)[2]

confusionMatrix(pred, test\_data$gen\_election,

positive = levels(test\_data$gen\_election)[2])

set.seed(32)

predicted\_values <- predict(rf, test\_data, type = "prob")[,2]

pred <- prediction(predicted\_values, test\_data$gen\_election)

perf <- performance(pred, measure = "tpr", x.measure = "fpr")

auc <- performance(pred, measure = "auc")

auc <- auc@y.values[[1]]

roc.data <- data.frame(fpr=unlist(perf@x.values),

tpr=unlist(perf@y.values),

model="RF")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +

geom\_ribbon(alpha=0.2) +

geom\_line(aes(y=tpr)) +

ggtitle(paste0("ROC Curve w/ AUC=", auc))

set.seed(32)

varImpPlot(rf)

set.seed(32)

ann <- nnet(gen\_election ~ ., data=train\_data, size=5, maxit=1000)

summary(ann)

set.seed(32)

predicted\_values\_ann <- predict(ann, test\_data, type = "raw")

head(predicted\_values)

threshold <- 0.5

pred\_ann <- factor(ifelse(predicted\_values\_ann[,1] > threshold, 1, 0))

head(pred)

levels(test\_data$gen\_election)[2]

confusionMatrix(pred\_ann, test\_data$gen\_election,

positive = levels(test\_data$gen\_election)[2])

pred\_ann <- prediction(predicted\_values\_ann, test\_data$gen\_election)

perf\_ann <- performance(pred\_ann, measure = "tpr", x.measure = "fpr")

auc <-performance(pred\_ann, measure="auc")

auc <- auc@y.values[[1]]

roc.data\_ann <- data.frame(fpr=unlist(perf\_ann@x.values),

tpr=unlist(perf\_ann@y.values),

model="ANN")

ggplot(roc.data\_ann, aes(x=fpr, ymin=0, ymax=tpr)) +

geom\_ribbon(alpha=0.2) +

geom\_line(aes(y=tpr)) +

ggtitle(paste0("ROC Curve w/ AUC=", auc))

set.seed(32)

ann <- nnet(gen\_election ~ ., data=train\_data, size=24, maxit=1000)

predicted\_values\_ann <- predict(ann, test\_data, type = "raw")

head(predicted\_values)

pred\_ann <- prediction(predicted\_values\_ann, test\_data$gen\_election)

perf\_ann <- performance(pred\_ann, measure = "tpr", x.measure = "fpr")

auc <-performance(pred\_ann, measure="auc")

auc <- auc@y.values[[1]]

roc.data\_ann <- data.frame(fpr=unlist(perf\_ann@x.values),

tpr=unlist(perf\_ann@y.values),

model="ANN")

ggplot(roc.data\_ann, aes(x=fpr, ymin=0, ymax=tpr)) +

geom\_ribbon(alpha=0.2) +

geom\_line(aes(y=tpr)) +

ggtitle(paste0("ROC Curve w/ AUC=", auc))

set.seed(32)

ftable(xtabs(~facebook+twitter+youtube+gen\_election, data=test\_data))

campaignfunds <- test\_data[which(test\_data$coh\_bop>50000),]

ftable(xtabs(~facebook+twitter+youtube+gen\_election, data=campaignfunds))

lesscampaignfunds <- test\_data[which(test\_data$coh\_bop<50000),]

ftable(xtabs(~facebook+twitter+youtube+gen\_election, data=lesscampaignfunds))

set.seed(32)

lesscampaignfunds <- test\_data[which(test\_data$coh\_bop<50000),]

table(lesscampaignfunds$gen\_election)

campaignfunds <- test\_data[which(test\_data$coh\_bop>50000),]

table(campaignfunds$gen\_election)

morecampaignfunds <- test\_data[which(test\_data$coh\_bop>250000),]

table(morecampaignfunds$gen\_election)

mostcampaignfunds <- test\_data[which(test\_data$coh\_bop>500000),]

table(mostcampaignfunds$gen\_election)

1. You may get an error that will require you to use library(e1071). In this case, install this library using install.packages(“e1071”). [↑](#footnote-ref-1)