## Module 4 Assignment 2 - Random Forests

### Nicholas Barrett, 02/16/19

options(tidyverse.quiet = TRUE)  
library(tidyverse)  
library(caret)

## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(ranger)

## Warning: package 'ranger' was built under R version 3.5.2

blood2 = read\_csv("Blood.csv")

## Parsed with column specification:  
## cols(  
## Mnths\_Since\_Last = col\_integer(),  
## TotalDonations = col\_integer(),  
## Total\_Donated = col\_integer(),  
## Mnths\_Since\_First = col\_integer(),  
## DonatedMarch = col\_integer()  
## )

blood2 = blood2 %>% mutate(DonatedMarch = as.factor(DonatedMarch))%>%  
 mutate(DonatedMarch = fct\_recode(DonatedMarch, "Yes" = "1", "No" = "0"))

### Task 1 - Split the dataset into training and testing sets. Use set.seed of 1234.

set.seed(1234)  
  
train.rows = createDataPartition(y = blood2$DonatedMarch, p=0.7, list = FALSE)   
train = blood2[train.rows,]   
test = blood2[-train.rows,]

### Task 2 - Create a random forest model on the training set to predict DonatedMarch using all of the variables in the dataset. Use caret’s trainControl functon to set up 10 fold cross-validation. use random number seed of 123. use 100 trees.

fit\_control = trainControl(method = "cv", number = 10)  
  
set.seed(123)  
rf\_fit = train(DonatedMarch ~.,  
 data = train,  
 method = "ranger",  
 importance = "permutation",  
 num.trees = 100,  
 trControl = fit\_control)

### Task 3 - Use varImp, determine the most and least important variable in the model.

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## TotalDonations 100.00  
## Mnths\_Since\_Last 84.41  
## Total\_Donated 64.23  
## Mnths\_Since\_First 0.00

rf\_fit

## Random Forest   
##   
## 524 samples  
## 4 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 471, 471, 472, 472, 471, 472, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.7499274 0.2370165  
## 2 extratrees 0.7631713 0.2420563  
## 3 gini 0.7271045 0.1985620  
## 3 extratrees 0.7422351 0.2063049  
## 4 gini 0.7195210 0.1812501  
## 4 extratrees 0.7212627 0.1875149  
##   
## Tuning parameter 'min.node.size' was held constant at a value of 1  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were mtry = 2, splitrule =  
## extratrees and min.node.size = 1.

The most important variable is TotalDonations and the least important variable is Mnths\_Since\_First.

### Task 4 - Use the modoel to develop predictions on the training set. Use the “head” funchion to display the first six predictions.

predRF = predict(rf\_fit, train)  
head(predRF)

## [1] Yes Yes No No Yes Yes  
## Levels: No Yes

### Task 5 - Use the the model to create a confusion using caret’s confusionMatrix function for the training set. Determine the accuracy, sensitivity, and specificity of the model.

confusionMatrix(predRF, train$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 394 50  
## Yes 5 75  
##   
## Accuracy : 0.895   
## 95% CI : (0.8656, 0.9199)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 4.371e-15   
##   
## Kappa : 0.6703   
## Mcnemar's Test P-Value : 2.975e-09   
##   
## Sensitivity : 0.6000   
## Specificity : 0.9875   
## Pos Pred Value : 0.9375   
## Neg Pred Value : 0.8874   
## Prevalence : 0.2385   
## Detection Rate : 0.1431   
## Detection Prevalence : 0.1527   
## Balanced Accuracy : 0.7937   
##   
## 'Positive' Class : Yes   
##

The accuracy of the random forest model for the training set is 89.5%. The sensitivity of the training set is 60%. The specificity of the training set of the random forest model is 98.7%.

### Task 6 - Determine the accuracy of the model compared ot a naive model that assumes that all observations are in the majority class.

The 89.5% accuracy of the model is better than the naive model’s 76.1% accuracy.

## Task 7 - Use the model to develop predictions on the test set. Develop a confusion matrix. Determine how the model performs on the testing set.

predRF2 = predict(rf\_fit, test)  
head(predRF2)

## [1] No No No Yes Yes Yes  
## Levels: No Yes

confusionMatrix(predRF2, test$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 163 41  
## Yes 8 12  
##   
## Accuracy : 0.7812   
## 95% CI : (0.7213, 0.8336)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.2945   
##   
## Kappa : 0.2288   
## Mcnemar's Test P-Value : 4.844e-06   
##   
## Sensitivity : 0.22642   
## Specificity : 0.95322   
## Pos Pred Value : 0.60000   
## Neg Pred Value : 0.79902   
## Prevalence : 0.23661   
## Detection Rate : 0.05357   
## Detection Prevalence : 0.08929   
## Balanced Accuracy : 0.58982   
##   
## 'Positive' Class : Yes   
##

The random forest model for the testing dataset has an accuracy of 78.1% whereas the naive model’s accuracy is 76.3%. It is a better predictor than the naive model.