## Daniel Sanchez

## BAN 502

## Dr. Stephen Hill

## Module 4 Assignment 3

## June 8th, 2020

library(tidyverse)  
library(caret)  
library(ranger)

Blood = read.csv("Blood.csv")  
  
Blood = Blood %>% mutate(DonatedMarch = as\_factor(as.character(DonatedMarch))) %>%  
mutate(DonatedMarch = fct\_recode(DonatedMarch,  
"No" = "0",  
"Yes" = "1"))

**Task 1: Split the dataset into training (70%) and testing (30%) sets. Use set.seed of 1234.**

set.seed(1234)  
train.rows = createDataPartition(y=Blood$DonatedMarch, p=0.7, list=FALSE)  
train = slice(Blood,train.rows)  
test = slice(Blood,-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 function to set up 10 fold cross-validation. Use a random number seed of 123. Use 100 trees.**

fit\_control = trainControl(method = "cv",   
 number = 10)  
set.seed(123)   
rf\_fit = train(x=train[,-5], y=train$DonatedMarch,  
 method = "ranger",   
 importance = "permutation",  
 trControl = fit\_control,  
 num.trees = 100)

**Task 3: Using varImp, what is the most important variable in the model, what is the least important?**

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## TotalDonations 100.00  
## Mnths\_Since\_First 66.28  
## Total\_Donated 14.10  
## Mnths\_Since\_Last 0.00

rf\_fit

## Random Forest   
##   
## 524 samples  
## 4 predictor  
## 2 classes: 'Yes', 'No'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 472, 472, 471, 471, 471, 472, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.8035922 0.3828006  
## 2 extratrees 0.8074020 0.3812627  
## 3 gini 0.7787010 0.3328348  
## 3 extratrees 0.7997097 0.3719429  
## 4 gini 0.7902032 0.3585442  
## 4 extratrees 0.7785922 0.3266171  
##   
## 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 in the model is TotalDonations. The least important variable is Mnths\_Since\_last.

**Task 4: Use the model to develop predictions on the training set. Use the “head” function to display the first six predictions.**

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

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

**Task 5: Use the model to create a confusion matrix using caret’s confusionMatrix function for the training set. What is the accuracy, sensitivity, and specificity of the model?**

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 77 3  
## No 48 396  
##   
## Accuracy : 0.9027   
## 95% CI : (0.874, 0.9267)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6943   
##   
## Mcnemar's Test P-Value : 7.218e-10   
##   
## Sensitivity : 0.6160   
## Specificity : 0.9925   
## Pos Pred Value : 0.9625   
## Neg Pred Value : 0.8919   
## Prevalence : 0.2385   
## Detection Rate : 0.1469   
## Detection Prevalence : 0.1527   
## Balanced Accuracy : 0.8042   
##   
## 'Positive' Class : Yes   
##

In this model, the Accuracy is 90.27%, the Sensitivity is 61.6%, and the Specificity is 99.25%.

**Task 6: How does the accuracy of the model compare to a naive model that assumes that all observations are in the majority class?**

The 90.27% accuracy is this model is substantially better than the naive model, which would have a 76.15% accuracy.

**Task 7: Use the model to develop predictions on the test set. Develop a confusion matrix. How does the model perform on the testing set?**

predRFtest = predict.train(rf\_fit, test)  
confusionMatrix(predRFtest, test$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 13 17  
## No 40 154  
##   
## Accuracy : 0.7455   
## 95% CI : (0.6832, 0.8012)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.762475   
##   
## Kappa : 0.1716   
##   
## Mcnemar's Test P-Value : 0.003569   
##   
## Sensitivity : 0.24528   
## Specificity : 0.90058   
## Pos Pred Value : 0.43333   
## Neg Pred Value : 0.79381   
## Prevalence : 0.23661   
## Detection Rate : 0.05804   
## Detection Prevalence : 0.13393   
## Balanced Accuracy : 0.57293   
##   
## 'Positive' Class : Yes   
##

The model does not perform as well on the testing set, with an Accuracy of 74.55%. This is lower than the No Information Rate of 76.34% with a naive model.

**Task 8: Comment on how this model might be used in the “real-world.” Would you recommend this model for real-world use? What if any concerns would you have about using the model?**

In practice, this model may be used to determine inventory and staffing for a blood drive in March. A planning team may be able to target the important variables, such as TotalDonations, Total\_Donated, and Mnths\_since\_First. The team may consider implementing incentives to motivate donors to act, based on their number of donations or newness to the system. My concern with using this model is the disparity in the model’s performance between the training and testing sets.