# Fields, Alex

## BAN 502

### Module 4 Assignment 3 - RF

#### Task 3 - Q&A

The most important variable is *TotalDonations* with a score of 100.00 and *Mnths\_Since\_Last* at 98.25. The least important variable being *Total\_Donated at 0*.

#### Task 5 - Q&A

When running the confustion matrix on the training set I was able to see the model had an Accuracy of 0.8931, Sensitivity of 0.6080 and Specificity of 0.9825.

#### Task 6 - Q&A

When looking at the Naive model, it seems that the accuracy is a little better. This could be it is assuming all observations are in the majority.

#### Task 7 - Q&A

The testing data seems to be less accurate than the training set. It drops almost 10% while the p-value (significance) becomes larger which is not what you want to see. The Model for the Testing data is not what we would like to have.

#### Task 8 - Q&A

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? **This model could be used for blood donations like we have tested but for accuracy we would need a lot more data. We coud also use this model in predicting the weather/temperature in a certain region. I would recommend this model, its not the best but it is still fairly accurate. My only concern is the need for more data when using this model.**

### Library

options(tidyverse.quiet = TRUE)  
library(tidyverse)  
library(VIM) #visualizing missingness  
library(ranger) #for random forests  
library(caret)

### Cleaning Data

library(readr)  
Blood <- read\_csv("Blood.csv")  
  
Blood = Blood %>% mutate(DonatedMarch = as.factor(DonatedMarch)) %>%   
 mutate(DonatedMarch = fct\_recode(DonatedMarch, "No" = "0", "Yes" = "1" ))

### Training/Testing Split

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

### Random Forest Generation

fit\_control = trainControl(method = "cv",   
 number = 10) #set up 10 fold cross-validation  
  
  
set.seed(1233)   
rf\_fit = train(DonatedMarch ~.,  
 data = train,   
 method = "ranger",   
 importance = "permutation",  
 trControl = fit\_control,  
 num.trees = 100)

### Validating variable importance

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## Total\_Donated 100.000  
## TotalDonations 74.243  
## Mnths\_Since\_First 6.022  
## Mnths\_Since\_Last 0.000

### Predictions on training

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

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

### Confusion Matrix on training

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 393 44  
## Yes 6 81  
##   
## Accuracy : 0.9046   
## 95% CI : (0.8761, 0.9283)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7067   
##   
## Mcnemar's Test P-Value : 1.672e-07   
##   
## Sensitivity : 0.6480   
## Specificity : 0.9850   
## Pos Pred Value : 0.9310   
## Neg Pred Value : 0.8993   
## Prevalence : 0.2385   
## Detection Rate : 0.1546   
## Detection Prevalence : 0.1660   
## Balanced Accuracy : 0.8165   
##   
## 'Positive' Class : Yes   
##

### Predictions on test

predRF\_test = predict(rf\_fit, newdata = test)

### Confusion matrix on test

confusionMatrix(predRF\_test, test$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 152 38  
## Yes 19 15  
##   
## Accuracy : 0.7455   
## 95% CI : (0.6832, 0.8012)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.76248   
##   
## Kappa : 0.1962   
##   
## Mcnemar's Test P-Value : 0.01712   
##   
## Sensitivity : 0.28302   
## Specificity : 0.88889   
## Pos Pred Value : 0.44118   
## Neg Pred Value : 0.80000   
## Prevalence : 0.23661   
## Detection Rate : 0.06696   
## Detection Prevalence : 0.15179   
## Balanced Accuracy : 0.58595   
##   
## 'Positive' Class : Yes   
##