# Module 4 Assignment 2

## BAN 502

### Jennifer Brill

library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ─────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## 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

library(readr)  
blood <- 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()  
## )

blood <- blood %>%  
 mutate(DonatedMarch = as.factor(DonatedMarch)) %>%  
 mutate(DonatedMarch = fct\_recode(DonatedMarch, "No" = "0", "Yes" = "1"))  
str(blood)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 748 obs. of 5 variables:  
## $ Mnths\_Since\_Last : int 2 0 1 2 1 4 2 1 2 5 ...  
## $ TotalDonations : int 50 13 16 20 24 4 7 12 9 46 ...  
## $ Total\_Donated : int 12500 3250 4000 5000 6000 1000 1750 3000 2250 11500 ...  
## $ Mnths\_Since\_First: int 98 28 35 45 77 4 14 35 22 98 ...  
## $ DonatedMarch : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 1 2 1 2 2 ...

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

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)

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## Total\_Donated 100.00  
## TotalDonations 40.25  
## Mnths\_Since\_Last 37.06  
## 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 0.7385341 0.2005381  
## 2 extratrees 0.7689405 0.2634991  
## 3 gini 0.7214441 0.1782345  
## 3 extratrees 0.7346517 0.2119550  
## 4 gini 0.7175617 0.1775164  
## 4 extratrees 0.7231495 0.1853589  
##   
## 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 random tree model in predicitng those who will donate in March is total number of donations (Total\_Donated). The least important variable is months since first donation (Mnths\_Since\_First).

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 393 50  
## Yes 6 75  
##   
## Accuracy : 0.8931   
## 95% CI : (0.8635, 0.9183)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 1.161e-14   
##   
## Kappa : 0.6654   
## Mcnemar's Test P-Value : 9.132e-09   
##   
## Sensitivity : 0.6000   
## Specificity : 0.9850   
## Pos Pred Value : 0.9259   
## Neg Pred Value : 0.8871   
## Prevalence : 0.2385   
## Detection Rate : 0.1431   
## Detection Prevalence : 0.1546   
## Balanced Accuracy : 0.7925   
##   
## 'Positive' Class : Yes   
##

The model is 89% accurate when predicting if someone donated blood in March. The model has a sensitivity of 60% and a specificity of 98%. The model is more accurate compared to a naive model that assumes that all observations are in the majority class. The majority class is No, a person did not donate in March. If we were to say that no one donated blood in March, we would be 76% accurate but our model is 89% accurate.

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 163 42  
## Yes 8 11  
##   
## Accuracy : 0.7768   
## 95% CI : (0.7165, 0.8296)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.3515   
##   
## Kappa : 0.2065   
## Mcnemar's Test P-Value : 3.058e-06   
##   
## Sensitivity : 0.20755   
## Specificity : 0.95322   
## Pos Pred Value : 0.57895   
## Neg Pred Value : 0.79512   
## Prevalence : 0.23661   
## Detection Rate : 0.04911   
## Detection Prevalence : 0.08482   
## Balanced Accuracy : 0.58038   
##   
## 'Positive' Class : Yes   
##

The model performs fairly well on predicting values in the testing set. The no information rate is still 76% but the accuracy rate has gone down from 89% to 77%. I would still interpret the model as being fairly accurate in its predictions on if a person donated blood in March.