# Random Forests

## Kinser, Shannon

### BAN-502 Module 4, Assignment 2

#install.packages("ranger")

library(tidyverse)

## -- Attaching packages --------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

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

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" ))

#### Task 1

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

#### Task 2

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

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## Total\_Donated 100.00  
## TotalDonations 56.38  
## Mnths\_Since\_First 49.77  
## Mnths\_Since\_Last 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 0.7499637 0.2186920  
## 2 extratrees 0.7518868 0.1872728  
## 3 gini 0.7441945 0.1985484  
## 3 extratrees 0.7481132 0.2116928  
## 4 gini 0.7404572 0.2055315  
## 4 extratrees 0.7365747 0.1913026  
##   
## 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 Mnths\_Since\_First and the least important is Mnths\_Since\_Last.

#### Task 4

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

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

#### Task 5

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 393 47  
## Yes 6 78  
##   
## Accuracy : 0.8989   
## 95% CI : (0.8698, 0.9233)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 5.838e-16   
##   
## Kappa : 0.6862   
## Mcnemar's Test P-Value : 3.920e-08   
##   
## Sensitivity : 0.6240   
## Specificity : 0.9850   
## Pos Pred Value : 0.9286   
## Neg Pred Value : 0.8932   
## Prevalence : 0.2385   
## Detection Rate : 0.1489   
## Detection Prevalence : 0.1603   
## Balanced Accuracy : 0.8045   
##   
## 'Positive' Class : Yes   
##

The accuracy of this model is 90.27%, the sensitivity is .60 and the specificity is .9975.

#### Task 6

The accuracy of the model is better than the naive model, with an accuaracy of 90.27% vs the naive model at 76.15%.

#### Task 7

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 160 39  
## Yes 11 14  
##   
## Accuracy : 0.7768   
## 95% CI : (0.7165, 0.8296)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.3515469   
##   
## Kappa : 0.2444   
## Mcnemar's Test P-Value : 0.0001343   
##   
## Sensitivity : 0.2642   
## Specificity : 0.9357   
## Pos Pred Value : 0.5600   
## Neg Pred Value : 0.8040   
## Prevalence : 0.2366   
## Detection Rate : 0.0625   
## Detection Prevalence : 0.1116   
## Balanced Accuracy : 0.5999   
##   
## 'Positive' Class : Yes   
##

The testing set is less accurate than the naive model, with an accuracy of just 75.45% vs 76.34%.