# Random Forests

## Brian Fortier

### Module 4 Assignment 2

Loading Packages:

#install.packages("tidyverse")  
#install.packages("caret")  
#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)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(ranger)

Loading Blood and Converting Variables:

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

Creating Train/Test Sets:

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

Creating Random Forest Model:

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

Random Forest Details:

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## TotalDonations 100.00  
## Mnths\_Since\_First 78.15  
## Mnths\_Since\_Last 18.51  
## Total\_Donated 0.00

rf\_fit

## Random Forest   
##   
## 748 samples  
## 4 predictor  
## 2 classes: 'Yes', 'No'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 673, 673, 673, 673, 673, 673, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.7674054 0.2638705  
## 2 extratrees 0.7860721 0.2920425  
## 3 gini 0.7527027 0.2242451  
## 3 extratrees 0.7594234 0.2418346  
## 4 gini 0.7460721 0.2171814  
## 4 extratrees 0.7460721 0.2140371  
##   
## 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.

Looking at the variable importance, we see that TotalDonations is viewed by the model as the most important variable with Total\_Donated to be the least improtant variable.

Predictions on Training Set:

predRF = predict(rf\_fit, train)  
head(predRF,6)

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

Looking at these predictions, we have 2 “Yes” predictions followed by 2 “No”, then 2 more “Yes” to finish off the top 6 results.

Confusion Matrix on Train Set:

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 74 3  
## No 51 396  
##   
## Accuracy : 0.8969   
## 95% CI : (0.8677, 0.9216)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 1.614e-15   
##   
## Kappa : 0.6732   
## Mcnemar's Test P-Value : 1.596e-10   
##   
## Sensitivity : 0.5920   
## Specificity : 0.9925   
## Pos Pred Value : 0.9610   
## Neg Pred Value : 0.8859   
## Prevalence : 0.2385   
## Detection Rate : 0.1412   
## Detection Prevalence : 0.1469   
## Balanced Accuracy : 0.7922   
##   
## 'Positive' Class : Yes   
##

After looking over the confusion matrix, we see the accuracy of our model to be at 89.69% with the sensitivity and specificity to be at 0.5840 and 0.9950, respectively.

Comparing this accuracy to the naive approach, we see the naive approach to be at 76.15% compared to our 89.69%.

Predictions on the Testing Set:

predRF2 = predict(rf\_fit, test)  
head(predRF2, 6)

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

A little different on the testing set predictions, we see 3 of each outcome, “Yes” and “No”.

Confusion Matrix on Test Set:

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 26 3  
## No 27 168  
##   
## Accuracy : 0.8661   
## 95% CI : (0.8144, 0.9078)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 9.092e-05   
##   
## Kappa : 0.5606   
## Mcnemar's Test P-Value : 2.679e-05   
##   
## Sensitivity : 0.4906   
## Specificity : 0.9825   
## Pos Pred Value : 0.8966   
## Neg Pred Value : 0.8615   
## Prevalence : 0.2366   
## Detection Rate : 0.1161   
## Detection Prevalence : 0.1295   
## Balanced Accuracy : 0.7365   
##   
## 'Positive' Class : Yes   
##

As we would expect, we have a slight decrease in model accuracy at 86.61% with sensitivity and specificity values at 0.4906 and 0.9825. We also have a little closer difference between our accuracy and the naive model accuracy of 76.34%.