## Stuart Broach

## Random Forests

#install.packages("ranger")  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- 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(as.character(DonatedMarch))) %>%  
mutate(DonatedMarch = fct\_recode(DonatedMarch,  
"Yes" = "1",  
"No" = "0"))  
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 100.000  
## TotalDonations 38.494  
## Mnths\_Since\_First 7.657  
## Mnths\_Since\_Last 0.000

rf\_fit

## Random Forest   
##   
## 524 samples  
## 4 predictor  
## 2 classes: 'Yes', 'No'   
##   
## 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 0.7804790 0.3105144  
## 2 extratrees 0.7880987 0.3133046  
## 3 gini 0.7804790 0.3284588  
## 3 extratrees 0.7747097 0.2923162  
## 4 gini 0.7689768 0.2939497  
## 4 extratrees 0.7727504 0.2903873  
##   
## 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.

Total\_Donated is the most importqnt variable in the train model and Mnths\_Since\_Last is the least important.

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 81 5  
## No 44 394  
##   
## Accuracy : 0.9065   
## 95% CI : (0.8783, 0.93)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7117   
## Mcnemar's Test P-Value : 5.681e-08   
##   
## Sensitivity : 0.6480   
## Specificity : 0.9875   
## Pos Pred Value : 0.9419   
## Neg Pred Value : 0.8995   
## Prevalence : 0.2385   
## Detection Rate : 0.1546   
## Detection Prevalence : 0.1641   
## Balanced Accuracy : 0.8177   
##   
## 'Positive' Class : Yes   
##

Accuracy is good at .906. The Sensitivity is .648 and the specificty is .987.

The accuracy of the model is MUCH better than the naive model. The accuracy is up 14% from the naive model.

fit\_control2 = trainControl(method = "cv",  
 number = 10)  
set.seed(123)  
rf\_fit2 = train(DonatedMarch ~.,  
 data = test,  
 method = "ranger",  
 importance = "permutation",  
 num.trees = 100,  
 trControl = fit\_control)

varImp(rf\_fit2)

## ranger variable importance  
##   
## Overall  
## Total\_Donated 100.000  
## TotalDonations 32.502  
## Mnths\_Since\_First 4.427  
## Mnths\_Since\_Last 0.000

rf\_fit2

## Random Forest   
##   
## 224 samples  
## 4 predictor  
## 2 classes: 'Yes', 'No'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 201, 202, 202, 201, 202, 202, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.6791173 -0.05268216  
## 2 extratrees 0.7057971 -0.02192663  
## 3 gini 0.6656621 -0.06790315  
## 3 extratrees 0.6927372 0.03616767  
## 4 gini 0.6694664 -0.05644777  
## 4 extratrees 0.6741601 -0.01483741  
##   
## 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.

predRF2 = predict(rf\_fit2, test)  
head(predRF2)

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 35 0  
## No 18 171  
##   
## Accuracy : 0.9196   
## 95% CI : (0.876, 0.9517)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 8.154e-10   
##   
## Kappa : 0.748   
## Mcnemar's Test P-Value : 6.151e-05   
##   
## Sensitivity : 0.6604   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9048   
## Prevalence : 0.2366   
## Detection Rate : 0.1562   
## Detection Prevalence : 0.1562   
## Balanced Accuracy : 0.8302   
##   
## 'Positive' Class : Yes   
##

The model performed even better on the testing set. It predicted one more Yes than the train set and the accuracy and specificty and sensitivity all went up.