# Random Forests

## June 5th, 2020

### Stephen Kiser

Libraries

library(tidyverse)

## -- Attaching packages ------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.5  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- 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)

Read-in Dataset

blood <- read\_csv("Blood.csv")

## Parsed with column specification:  
## cols(  
## Mnths\_Since\_Last = col\_double(),  
## TotalDonations = col\_double(),  
## Total\_Donated = col\_double(),  
## Mnths\_Since\_First = col\_double(),  
## DonatedMarch = col\_double()  
## )

Changing variables

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

## Observations: 748  
## Variables: 5  
## $ Mnths\_Since\_Last <dbl> 2, 0, 1, 2, 1, 4, 2, 1, 2, 5, 4, 0, 2, 1, 2, 2, 2...  
## $ TotalDonations <dbl> 50, 13, 16, 20, 24, 4, 7, 12, 9, 46, 23, 3, 10, 1...  
## $ Total\_Donated <dbl> 12500, 3250, 4000, 5000, 6000, 1000, 1750, 3000, ...  
## $ Mnths\_Since\_First <dbl> 98, 28, 35, 45, 77, 4, 14, 35, 22, 98, 58, 4, 28,...  
## $ DonatedMarch <fct> Yes, Yes, Yes, Yes, No, No, Yes, No, Yes, Yes, No...

summary(blood)

## Mnths\_Since\_Last TotalDonations Total\_Donated Mnths\_Since\_First  
## Min. : 0.000 Min. : 1.000 Min. : 250 Min. : 2.00   
## 1st Qu.: 2.750 1st Qu.: 2.000 1st Qu.: 500 1st Qu.:16.00   
## Median : 7.000 Median : 4.000 Median : 1000 Median :28.00   
## Mean : 9.507 Mean : 5.515 Mean : 1379 Mean :34.28   
## 3rd Qu.:14.000 3rd Qu.: 7.000 3rd Qu.: 1750 3rd Qu.:50.00   
## Max. :74.000 Max. :50.000 Max. :12500 Max. :98.00   
## DonatedMarch  
## No :570   
## Yes:178   
##   
##   
##   
##

Training/Testing Sets

set.seed(1234)  
train.rows.blood <- createDataPartition(y=blood$DonatedMarch, p=0.7, list = FALSE)  
  
train\_blood <- slice(blood, train.rows.blood)  
test\_blood <- slice(blood, - train.rows.blood)  
  
summary(train\_blood)

## Mnths\_Since\_Last TotalDonations Total\_Donated Mnths\_Since\_First  
## Min. : 0.000 Min. : 1.000 Min. : 250 Min. : 2.0   
## 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.: 500 1st Qu.:16.0   
## Median : 8.000 Median : 4.000 Median : 1000 Median :28.0   
## Mean : 9.819 Mean : 5.557 Mean : 1389 Mean :34.9   
## 3rd Qu.:14.000 3rd Qu.: 7.000 3rd Qu.: 1750 3rd Qu.:51.0   
## Max. :74.000 Max. :50.000 Max. :12500 Max. :98.0   
## DonatedMarch  
## No :399   
## Yes:125   
##   
##   
##   
##

str(train\_blood)

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 524 obs. of 5 variables:  
## $ Mnths\_Since\_Last : num 2 0 1 2 1 5 4 2 1 2 ...  
## $ TotalDonations : num 50 13 16 20 12 46 23 10 13 15 ...  
## $ Total\_Donated : num 12500 3250 4000 5000 3000 11500 5750 2500 3250 3750 ...  
## $ Mnths\_Since\_First: num 98 28 35 45 35 98 58 28 47 49 ...  
## $ DonatedMarch : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...

### Task 2

fit\_control = trainControl(method = "cv", number=10)  
  
set.seed(123)  
rf\_fit = train(x=as.matrix(train\_blood[,-5]), y=as.matrix(train\_blood$DonatedMarch),  
 method = "ranger",  
 importance = "permutation",  
 trControl = fit\_control,  
 num.trees = 100)

### Task 3

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## TotalDonations 100.00  
## Total\_Donated 57.72  
## Mnths\_Since\_First 35.32  
## 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: 472, 472, 471, 471, 471, 472, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.7519956 0.2410225  
## 2 extratrees 0.7767779 0.2904529  
## 3 gini 0.7406386 0.2326286  
## 3 extratrees 0.7577649 0.2447921  
## 4 gini 0.7482946 0.2498497  
## 4 extratrees 0.7424528 0.2099090  
##   
## 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 model is TotalDonations, and the least important variable is Mnths\_Since\_Last.

### Task 4

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

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

### Task 5

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 391 43  
## Yes 8 82  
##   
## Accuracy : 0.9027   
## 95% CI : (0.874, 0.9267)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7036   
##   
## Mcnemar's Test P-Value : 1.927e-06   
##   
## Sensitivity : 0.6560   
## Specificity : 0.9799   
## Pos Pred Value : 0.9111   
## Neg Pred Value : 0.9009   
## Prevalence : 0.2385   
## Detection Rate : 0.1565   
## Detection Prevalence : 0.1718   
## Balanced Accuracy : 0.8180   
##   
## 'Positive' Class : Yes   
##

The accuracy of the model is 90.27%. The sensitivity of the model is 0.6560 and the specificity is 0.9799.

### Task 6

The naive accuracy is 76.15% while the data set we created was 90.27%. This is a 14.12 percent difference in accuracy. The model’s accuracy is much better than the naive model’s accuracy.

### Task 7

predRF\_test = predict(rf\_fit, newdata = test\_blood)  
  
confusionMatrix(predRF\_test, test\_blood$DonatedMarch, positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 154 34  
## Yes 17 19  
##   
## Accuracy : 0.7723   
## 95% CI : (0.7118, 0.8255)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.41192   
##   
## Kappa : 0.2913   
##   
## Mcnemar's Test P-Value : 0.02506   
##   
## Sensitivity : 0.35849   
## Specificity : 0.90058   
## Pos Pred Value : 0.52778   
## Neg Pred Value : 0.81915   
## Prevalence : 0.23661   
## Detection Rate : 0.08482   
## Detection Prevalence : 0.16071   
## Balanced Accuracy : 0.62954   
##   
## 'Positive' Class : Yes   
##

This new Confusion Matrix on the test dataset is worse than our training confusion matrix. The accuracy of the test data is 76.79% while our accuracy for the training data was 90.27%. This means that the test dataset has more wrong predictions then the training dataset.

### Task 8

This model might be used in the real world to find out how many people would donate blood in the next month. We could use the data that was collected for March, and see if we could enter in new donations to our model to predict how many people will donate in the following month. We would have to update this model every month to keep it up to date. I would recommend it because there is a 90.27% accuracy in predicting those who would donate. My concern is that we would have to update this model every month because the new data, and those who donated in March might not donate for several months. They might wait a couple of months in between donations because they have just recently donated blood.