## Random Forest - MOd 4 Assignment 3

Load Libraries

options(tidyverse.quiet = TRUE)  
library(tidyverse)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(ranger)

Reading Data

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

Blood = Blood %>% mutate(DonatedMarch = as\_factor(DonatedMarch)) %>%  
mutate(DonatedMarch = fct\_recode(DonatedMarch,  
"No" = "0",  
"Yes" = "1"))

Structure and summary

str(Blood)

## '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 ...

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   
##   
##   
##   
##

Split the data (training and testing)

set.seed(1234)  
train.rows = createDataPartition(y = Blood$DonatedMarch , p=0.7, list = FALSE) #70% in training  
train = slice(Blood,train.rows)   
test = slice(Blood,-train.rows)

Now that we have the split data, let’s build a classification tree. Here we use caret to manage the model building.

fit\_control = trainControl(method = "cv",   
 number = 10) #set up 10 fold cross-validation  
  
set.seed(123)   
rf\_fit = train(x=Blood[,-5], y=Blood$DonatedMarch,  
 method = "ranger",   
 num.trees = 100,  
 importance = "permutation",  
 trControl = fit\_control)

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## Total\_Donated 100.000  
## TotalDonations 60.911  
## Mnths\_Since\_Last 2.779  
## Mnths\_Since\_First 0.000

rf\_fit

## Random Forest   
##   
## 748 samples  
## 4 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 674, 674, 673, 673, 673, 673, ...   
## Resampling results across tuning parameters:  
##   
## mtry splitrule Accuracy Kappa   
## 2 gini 0.7620901 0.2512608  
## 2 extratrees 0.7794414 0.2659173  
## 3 gini 0.7406847 0.2094663  
## 3 extratrees 0.7567387 0.2353788  
## 4 gini 0.7460901 0.2246573  
## 4 extratrees 0.7474234 0.2139096  
##   
## 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 Total\_Donated and the least is Mnths\_Since\_first.

Predictions

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

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

Confusion matrix

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 391 54  
## Yes 8 71  
##   
## Accuracy : 0.8817   
## 95% CI : (0.8509, 0.9081)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : 2.719e-12   
##   
## Kappa : 0.6272   
##   
## Mcnemar's Test P-Value : 1.097e-08   
##   
## Sensitivity : 0.5680   
## Specificity : 0.9799   
## Pos Pred Value : 0.8987   
## Neg Pred Value : 0.8787   
## Prevalence : 0.2385   
## Detection Rate : 0.1355   
## Detection Prevalence : 0.1508   
## Balanced Accuracy : 0.7740   
##   
## 'Positive' Class : Yes   
##

The accuracy is 88% , sensitivity is .56, and Specificity is .9799.

The model accuracy of 88% is better than the naive model, which has an accuracy of 76.15%.

Predictions on test data

predRF\_Test = predict(rf\_fit, test)  
head(predRF\_Test)

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

Confusion matrix on test data

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 169 20  
## Yes 2 33  
##   
## Accuracy : 0.9018   
## 95% CI : (0.8551, 0.9374)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 8.252e-08   
##   
## Kappa : 0.692   
##   
## Mcnemar's Test P-Value : 0.0002896   
##   
## Sensitivity : 0.6226   
## Specificity : 0.9883   
## Pos Pred Value : 0.9429   
## Neg Pred Value : 0.8942   
## Prevalence : 0.2366   
## Detection Rate : 0.1473   
## Detection Prevalence : 0.1562   
## Balanced Accuracy : 0.8055   
##   
## 'Positive' Class : Yes   
##

The model accuracy on the test data is even better at 90% compared to the model on training data. The naive model accuracy is 76.34%.

This model can be used to predict how many might donate in the next month or two, based on the Mnths\_since\_Last and totalDonations. While the accuracy and specificity are high, sensitivity which captures "positive obsevations is low. This model might predict lower donations.