## Module 4 Assignment 2

#install.packages("ranger")

library(tidyverse)

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

## v ggplot2 3.1.0 v purrr 0.3.0  
## v tibble 2.0.1 v dplyr 0.7.8  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.3.1 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)

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()  
## )

View(blood)

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

set.seed(1234)  
train.rows2 = createDataPartition(y = blood$DonatedMarch, p=0.7, list = FALSE) #70% in training  
train2 = blood[train.rows2,]   
test2 = blood[-train.rows2,]

fit\_control = trainControl(method = "cv",   
 number = 10) #set up 10 fold cross-validation  
  
set.seed(123)   
rf\_fit = train(DonatedMarch ~.,   
 data = train2,   
 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.

The most important variable is Total\_Donated. The least important variable is Mnths\_Since\_Last.

predRF = predict(rf\_fit, train2)  
head(predRF)

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

confusionMatrix(predRF, train2$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   
##

The accuracy on this model is very high at .9065. We also see that is it quite a bit higher than if we just went with the majority class of .7615 and a naive model. Also, we can tell that there is significance to this model because of the P-value that is so much lower than .05. The sensitivity of the model is .6480 and the specificity of the model is .9875.

fit\_control1 = trainControl(method = "cv",   
 number = 10) #set up 10 fold cross-validation  
  
set.seed(123)   
rf\_fit1 = train(DonatedMarch ~.,   
 data = test2,   
 method = "ranger",   
 importance = "permutation",   
 num.trees = 100,  
 trControl = fit\_control1)

varImp(rf\_fit1)

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

rf\_fit1

## 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.

predRF1 = predict(rf\_fit1, test2)  
head(predRF1)

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

confusionMatrix(predRF1, test2$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 actually performed even better on the test model. We see that the accuracy is higher than the training mdoel at .9196 and the P-value is extremly small. I would say this is an ideal model. At least from the various data sets that we have looked as thus far.