# Module 4 Assignment 2

## Bass, Robert

### Random Forests

Libraries

library(tidyverse)

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

## v ggplot2 3.1.0 v purrr 0.3.2   
## v tibble 2.1.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 v stringr 1.4.0   
## v readr 1.3.1 v forcats 0.4.0

## -- 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)

Reading in the data

blood\_original <- 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()  
## )

Converting variables

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

## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 748 obs. of 5 variables:  
## $ Mnths\_Since\_Last : num 2 0 1 2 1 4 2 1 2 5 ...  
## $ TotalDonations : num 50 13 16 20 24 4 7 12 9 46 ...  
## $ Total\_Donated : num 12500 3250 4000 5000 6000 1000 1750 3000 2250 11500 ...  
## $ Mnths\_Since\_First: num 98 28 35 45 77 4 14 35 22 98 ...  
## $ DonatedMarch : Factor w/ 2 levels "Yes","No": 1 1 1 1 2 2 1 2 1 1 ...

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  
## Yes:178   
## No :570   
##   
##   
##   
##

Task 1

Split the data set into training and testing sets

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

Task 2

Create a random forest model

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

Task 3:

Random Forest details using varImp

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: 'No', 'Yes'   
##   
## 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 in this model is “Total\_Donated” and the least important variable being “Mths\_Since\_Last”.

Task 4

Predictions

predRF <- predict(rf\_fit, train)  
head(predRF)

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

Task 5

Confusion Matrix

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

## Warning in confusionMatrix.default(predRF, train$DonatedMarch, positive  
## = "Yes"): Levels are not in the same order for reference and data.  
## Refactoring data to match.

## 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 0.91, naive is 0.76 with a p-value < .05. Sensitivity is .65 and specificity is .99.

Task 6

This model is a large improvement with an accuracy of 0.91 and a naive score of 0.76 with the p-value < .05. This model is more accurate because we used a k-fold of 10 then developed 100 decision trees with an output of the best model.

Task 7

Predictions on the test set

predRF\_test <- predict(rf\_fit, newdata = test)  
head(predRF\_test)

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

Confusion Matrix on the test set

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

## Warning in confusionMatrix.default(predRF\_test, test$DonatedMarch,  
## positive = "Yes"): Levels are not in the same order for reference and data.  
## Refactoring data to match.

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 15 12  
## No 38 159  
##   
## Accuracy : 0.7768   
## 95% CI : (0.7165, 0.8296)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.351547   
##   
## Kappa : 0.2562   
##   
## Mcnemar's Test P-Value : 0.000407   
##   
## Sensitivity : 0.28302   
## Specificity : 0.92982   
## Pos Pred Value : 0.55556   
## Neg Pred Value : 0.80711   
## Prevalence : 0.23661   
## Detection Rate : 0.06696   
## Detection Prevalence : 0.12054   
## Balanced Accuracy : 0.60642   
##   
## 'Positive' Class : Yes   
##

Accuracy is 0.7768, naive is 0.7634 with a p-value > .05. Sensitivity is .28 and specificity is .93. The accuracy is improved, which is a good thing. The sensitivity is very low which could cause some concern. This is an OK model.

Task 8

This model could be used in the real world for blood drives. Marketing strategies could be used with this data, targeting people who have already donated the most blood first, since they are more likely to come back. The concerns I have with this model is the very low sensitivity score (this could lead to incorrectly detecting positives) and the large p-value. This model still has an improved accuracy than a naive model which is good. This model would be OK to use depending on its applications. In regards to getting the word out about blood donations and to attempt to get more people to donate, this model is fine. If this model was being used for funding purposes, like obtaining grant money for example, I would be hesitant.