## Module 4 Random Forest Assignment

### Madison West

load libraries

library(tidyverse)

##

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

##   
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## 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 3.6.2

read in data

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()  
## )

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

Task 1: Split the dataset into training (70%) and testing (30%) sets. Use set.seed of 1234.

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

Task 2: Create a random forest model on the training set to predict DonatedMarch using all of the variables in the dataset. Use caret’s trainControl function to set up 10 fold cross-validation. Use a random number seed of 123. Use 100 trees (Note you can specify the number of trees by adding a line num.trees = 100 to the rf\_fit block of code).

fit\_control = trainControl(method = "cv",   
 number = 10) #set up 10 fold cross-validation  
  
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: Using varImp, what is the most important variable in the model, what is the least important?

varImp(rf\_fit)

## ranger variable importance  
##   
## Overall  
## TotalDonations 100.00  
## Mnths\_Since\_First 40.88  
## Total\_Donated 23.41  
## Mnths\_Since\_Last 0.00

**According to the RF model, the most important variable is TotalDonations, while Mnths\_Since\_Last is the least important.**

Task 4: Use the model to develop predictions on the training set. Use the “head” function to display the first six predictions.

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

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

Task 5: Use the model to create a confusion matrix using caret’s confusionMatrix function for the training set. What is the accuracy, sensitivity, and specificity of the model?

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 77 3  
## No 48 396  
##   
## Accuracy : 0.9027   
## 95% CI : (0.874, 0.9267)  
## No Information Rate : 0.7615   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6943   
##   
## Mcnemar's Test P-Value : 7.218e-10   
##   
## Sensitivity : 0.6160   
## Specificity : 0.9925   
## Pos Pred Value : 0.9625   
## Neg Pred Value : 0.8919   
## Prevalence : 0.2385   
## Detection Rate : 0.1469   
## Detection Prevalence : 0.1527   
## Balanced Accuracy : 0.8042   
##   
## 'Positive' Class : Yes   
##

**This models accuracy is 0.9027, with a sensitivity of 0.6160 and a specificity of 0.9925.**

Task 6: How does the accuracy of the model compare to a naive model that assumes that all observations are in the majority class?

summary(train)

## Mnths\_Since\_Last TotalDonations Total\_Donated Mnths\_Since\_First  
## Min. : 0.000 Min. : 1.000 Min. : 250 Min. : 2.00   
## 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.: 500 1st Qu.:16.00   
## Median : 7.500 Median : 4.000 Median : 1000 Median :28.00   
## Mean : 9.614 Mean : 5.651 Mean : 1413 Mean :35.16   
## 3rd Qu.:14.000 3rd Qu.: 7.000 3rd Qu.: 1750 3rd Qu.:52.00   
## Max. :74.000 Max. :44.000 Max. :11000 Max. :98.00   
## DonatedMarch  
## Yes:125   
## No :399   
##   
##   
##   
##

naiveacc = 399/nrow(train)  
naiveacc

## [1] 0.7614504

**Based on the naive model which assumes all records are ‘No’ (did not donate), the accuracy is 0.7615, which is significantly lower than the accuracy of the random forest accuracy.**

Task 7: Use the model to develop predictions on the test set. Develop a confusion matrix. How does the model perform on the testing set?

predRF\_test = predict(rf\_fit, newdata = test)  
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 13 17  
## No 40 154  
##   
## Accuracy : 0.7455   
## 95% CI : (0.6832, 0.8012)  
## No Information Rate : 0.7634   
## P-Value [Acc > NIR] : 0.762475   
##   
## Kappa : 0.1716   
##   
## Mcnemar's Test P-Value : 0.003569   
##   
## Sensitivity : 0.24528   
## Specificity : 0.90058   
## Pos Pred Value : 0.43333   
## Neg Pred Value : 0.79381   
## Prevalence : 0.23661   
## Detection Rate : 0.05804   
## Detection Prevalence : 0.13393   
## Balanced Accuracy : 0.57293   
##   
## 'Positive' Class : Yes   
##

**Based on the predicitions from the RF model on the test data set we see that the model performance is not as great as the model on the training dataset. The accuracy of this mondel is 0.7455, with a sensitivity of 0.24528, and a specificity of 0.90058. However, this accuracy is higher than the naive model.**

Task 8 Comment on how this model might be used in the “real-world.” Would you recommend this model for real-world use? What if any concerns would you have about using the model?

**This model may be used in the real-world for the blood bank to reach out to those subjects that are predicting to not donate in March to ask for donations/send additional offers to peruade those subjects to donate. Concerns may include not reaching out to subjects who were predicted to donated in March that actually did not, which would cause a lack of donations that potentially could have been prevented.**