suppressPackageStartupMessages({  
 library(tidyverse)  
 library(caret)  
 library(ranger)  
 })

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

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

blood = read.csv("Blood.csv")  
blood = blood %>%  
 mutate(DonatedMarch = as.factor(DonatedMarch)) %>%  
 mutate(DonatedMarch = fct\_recode(DonatedMarch, "No" = "0","Yes" = "1"))

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

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

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.

**Task 3 response: the most important variable is the TotalDonations, and the least important is Mnths\_Since\_Last.**

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

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

confusionMatrix(predRF,train$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   
##

**Task 5 & 6 responses: The accuracy of our model on the training set is 0.9027, with a sensitivity of 0.6560 and specificity of 0.9799. Our model is significantly better than the naive model, which has an accuracy of 0.7615.**

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

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

confusionMatrix(predRF\_test,test$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   
##

**Task 7 response: It appears that the model might be overfitting our training set, as the accuracy on the test data comes in at 0.7723, which is still better than the naive model of 0.7634, but obviously much closer than the accuracy of our training set. Also, unlike the training set confusion matrix, the p-value between models is not significant, indicating that perhaps there is not a material difference between our predictions and the naive model. I think the model would still be valuable though, since the accuracy is still higher, I would have confidence in our model being able to predict blood donations in the month of March.**

**Task 8 response: I think this would be a good real-world application for the organization collecting this data to estimate the amount of blood they are expecting to receive in the month of march. It could also help them identify the people most likely to donate again, and send them targeted marketing to remind them about upcoming drives, and not waste time or resources sending materials to people designated as unlikely to donate again by our model, so we could get some cost savings on more specific marketing as well as build a forecast of future collection. The concern I have is the one I mentioned in task 7 above, that our model might be overfitting the training data set; however, perhaps we could have used a higher number of trees to build a more comprehensive random forest and improve the model for data that it hasn’t seen yet. Despite my fear of overfitting, the accuracy of the model with testing data was still higher than the naive model, so I think it’s appropriate to suggest that it would handle more predictions well and give the organization strong estimates.**