

Random Forests

```
knitr::opts_chunk$set(eval = FALSE)

library(caret)

## Loading required package: lattice
## Loading required package: ggplot2
## Registered S3 methods overwritten by 'ggplot2':
##   method      from
##   [.quosures   rlang
##   c.quosures   rlang
##   print.quosures rlang

library(randomForest)

## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##   margin
library(ranger)

##
## Attaching package: 'ranger'
## The following object is masked from 'package:randomForest':
##
##   importance
library(rpart)

cog_data <- readRDS("./data/cog_data_preproc.RDS")
cog_train <- readRDS("./data/cog_train_preproc.RDS")
cog_test <- readRDS("./data/cog_test_preproc.RDS")

set.seed(1)
train_index <- createDataPartition(cog_data$cdr, p = 2/3, list = FALSE, times = 1)

ctrl1 <- trainControl(method = "repeatedcv",
                      repeats = 5,
                      summaryFunction = twoClassSummary,
                      classProbs = TRUE)

#caret
#not exactly sure what mtry and min.node.size should be
rf.grid <- expand.grid(mtry = 1:6,
                      splitrule = "gini",
                      min.node.size = 1:6)
```

```

#Error: Setting row names on a tibble is deprecated
set.seed(1)
rf.fit <- train(x = cog_train[3:10],
               y = cog_train$cdr,
               method = "ranger",
               tuneGrid = rf.grid,
               metric = "ROC",
               trControl = ctrl1,
               importance = 'permutation')
summary(rf.fit$finalModel)

saveRDS(rf.fit, file = "./data/rf.fit.RDS")
rf.fit = readRDS("./data/rf.fit.RDS")

ggplot(rf.fit, highlight = TRUE)

#importance
barplot(sort(ranger::importance(rf.fit$finalModel),
          decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("darkred", "white", "darkblue"))(8))

rf.pred <- predict(rf.fit, newdata = cog_test, type = "prob")[,1]

error_rate_rf <- mean(rf.pred != cog_data$cdr[-train_index])

```