RandomForest

Scott McCoy - U80152879

2/24/2021

Random Forest with K-Fold CV

Libraries:

```
set.seed(430)
library(data.table)
library(ggplot2)
library(ggthemes)
library(scales)
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
```

```
theme_set(theme_bw())
```

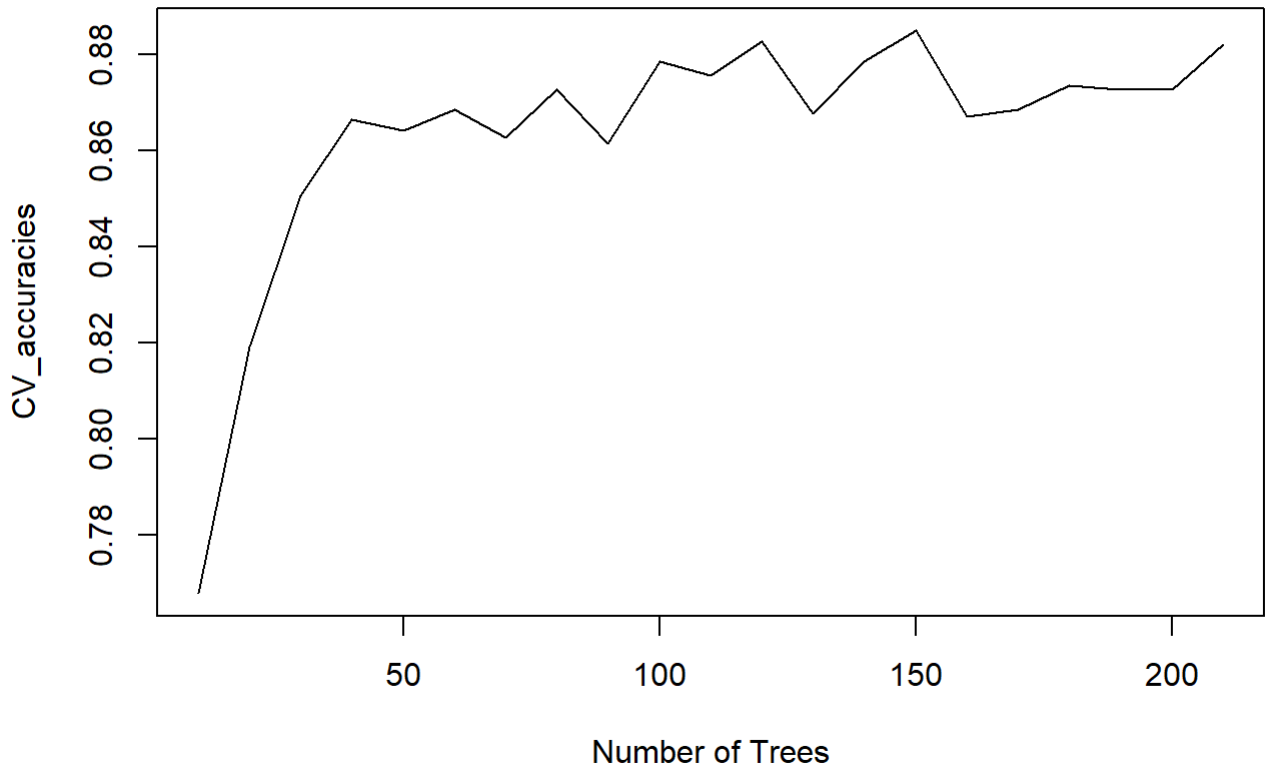
Train-Test Split

```
mo <- fread('~/R/mobile/train.csv')
mo$price_range <- as.factor(mo$price_range)
mo_obs <- nrow(mo)
mo_idx <- sample(mo_obs, size = trunc(0.70 * mo_obs))
mo_trn <- mo[mo_idx, ]
mo_test <- mo[-mo_idx, ]

Y_test <- mo_test[,price_range]
```

Finding optimal hyperparameter values with K-Fold CV:

```
CV_accuracies = c()
# calculates CV accuracy for Random Forest with various number of trees
for (num_trees in seq(from = 10, to = 210, by = 10)){

  k = 5
  #Randomly shuffle the data
  mo_trn_cross <- mo_trn[sample(nrow(mo_trn)),]

  #Create K equally size folds
  folds <- cut(seq(1,nrow(mo_trn_cross)),breaks=k,labels=FALSE)

  accuracies <- c()

  #Perform K-fold cross validation
  for(i in 1:k){
    #Segement data by fold using which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    testData <- mo_trn_cross[testIndexes, ]
    trainData <- mo_trn_cross[-testIndexes, ]
    Y_CV <- testData$price_range

    num_features <- sqrt(length(colnames(mo_trn)) -1) # number of variables randomly chosen at e
ach node - sqrt of # of features
    rf_classifier <- randomForest(price_range ~ ., data = trainData, ntree = num_trees, mtry = n
um_features, importance = TRUE )
    Y_test_hat <- predict(rf_classifier, newdata = testData, type = "class")

    accuracy <- mean(Y_test_hat == Y_CV)

    accuracies <- c(accuracies, accuracy)
  }
  CV_accuracy <- mean(accuracies)
  CV_accuracies <- c(CV_accuracies, CV_accuracy)



}
```

Plotting Accuracy vs Number of Trees:



Variable Importance:

```
num_features <- sqrt(length(colnames(mo_trn)) -1)
rf_classifier <- randomForest(price_range ~ ., data = mo_trn, ntree = 150, mtry = num_features,
 importance = TRUE )

Y_test_hat <- predict(rf_classifier, newdata = mo_test, type = "class")

mean(Y_test_hat == Y_test)
```

```
## [1] 0.8733333
```

```
cm <- table(observed=Y_test_hat, predicted=Y_test)
cm
```

```
##          predicted
## observed   0   1   2   3
##        0 135   8   0   0
##        1  13 133  22   0
##        2   0  10 120   7
##        3   0   0  16 136
```

```
vi <- importance(rf_classifier, type = 2)

vi
```

```
##               MeanDecreaseGini
## battery_power        77.391165
## blue                  6.703439
## clock_speed          30.338408
## dual_sim              7.226297
## fc                   26.072046
## four_g                7.435071
## int_memory           40.025239
## m_dep                27.243303
## mobile_wt            41.810411
## n_cores              24.326710
## pc                   30.818795
## px_height            58.534474
## px_width             62.066756
## ram                 494.409537
## sc_h                 30.944804
## sc_w                 29.880913
## talk_time            32.972265
## three_g               5.942866
## touch_screen          7.645306
## wifi                  7.228656
```