gunsTestMachines

Thadryan Sweeney

May 14, 2018

```
data <- read.csv("rfImputedGunData.csv")

# import the go-to R package for ML
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2

## Warning: replacing previous import by 'plyr::ddply' when loading 'caret'

## Warning: replacing previous import by 'tidyr::%>%' when loading 'broom'

## Warning: replacing previous import by 'tidyr::gather' when loading 'broom'

## Warning: replacing previous import by 'tidyr::spread' when loading 'broom'

## Warning: replacing previous import by 'rlang::!!' when loading 'recipes'

## Warning: replacing previous import by 'rlang::expr' when loading 'recipes'

## Warning: replacing previous import by 'rlang::f_lhs' when loading 'recipes'

## Warning: replacing previous import by 'rlang::f_rhs' when loading 'recipes'

## Warning: replacing previous import by 'rlang::is_empty' when loading
## 'recipes'

## Warning: replacing previous import by 'rlang::lang' when loading 'recipes'

## Warning: replacing previous import by 'rlang::na_dbl' when loading
## 'recipes'

## Warning: replacing previous import by 'rlang::names2' when loading
## 'recipes'

## Warning: replacing previous import by 'rlang::quos' when loading 'recipes'

## Warning: replacing previous import by 'rlang::sym' when loading 'recipes'

## Warning: replacing previous import by 'rlang::syms' when loading 'recipes'
```

```
# set a random seed - this just means will be using the same random set each time for now
set.seed(8675309)

# sample the data (sample 1)
s1.data <- data[sample(nrow(data), 20000), ]

# create an idex of 70% entries in the dataframe based on race
partitionIndex = createDataPartition(s1.data$race, p = 0.7, list = FALSE)

# train will be the entries in the partition
train <- s1.data[ partitionIndex, ]

# test will be the opposite of the ones in the partition
test  <- s1.data[-partitionIndex, ]
```

```r
# train a random forest classifier
m <- train(race ~., method = "rpart", data = train)

# inspect the random forest model
m
```

```
## CART
##
## 14003 samples
##    10 predictor
##     5 classes: 'Asian/Pacific Islander', 'Black', 'Hispanic', 'Native American/Native Alaskan', 'Whi
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 14003, 14003, 14003, 14003, 14003, 14003, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.08952102  0.8338168  0.6723119
##   0.16272746  0.7835104  0.5603018
##   0.30537544  0.7104863  0.2642274
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.08952102.
```

```r
# apply the random forest model to the test using test as new data
test$m.pred <- predict(m, newdata = test)

# show the simple accuracy
m.simple.acc <- length(which(test$m.pred == test$race))/nrow(test)
m.simple.acc
```

```
## [1] 0.8147407
```

```r
plot(m$finalModel)
```