SML Assignment 3

Shubhang Periwal 19201104

4/18/2020

```
library(mlbench)
```

```
## Warning: package 'mlbench' was built under R version 3.6.3
```

```
data("Satellite")
# this will re-order alphabetically class labels and remove spacing
Satellite$classes <- gsub(" ", "_", Satellite$classes)
Satellite$classes <- factor( as.character(Satellite$classes) )
# to have the same initial split
set.seed(777222)
D <- nrow(Satellite)
keep <- sample(1:D, 5500)
test <- setdiff(1:D, keep)
dat <- Satellite[keep,]
dat_test <- Satellite[test,]
```

Multinomial Logistic Regression

```
library(nnet)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
fitLog <- multinom(classes ~ .,data = dat,maxit=300, trace=FALSE) #multi with maximum number of iterati
fitCt <- randomForest(classes ~ ., data = dat,maxit=300, trace=FALSE) # randomforest
```

```
#  Random Forest
predValCt <- predict(fitCt, type = "class", newdata = dat_test)
tabValCt <- table(dat_test$classes, predValCt)
tabValCt
```

```
##                       predValCt
##                        cotton_crop damp_grey_soil grey_soil red_soil
##   cotton_crop                   96              0         1        0
##   damp_grey_soil                 1             63        17        2
##   grey_soil                      0              6       172        2
##   red_soil                       0              0         3      221
##   vegetation_stubble             0              1         0        4
##   very_damp_grey_soil            0             11         4        0
##                       predValCt
```

```
##                       vegetation_stubble very_damp_grey_soil
##   cotton_crop                          0                   0
##   damp_grey_soil                       0                  17
##   grey_soil                            0                   1
##   red_soil                             1                   0
##   vegetation_stubble                  87                   7
##   very_damp_grey_soil                  5                 213
```

```
accCt <- sum(diag(tabValCt))/sum(tabValCt)
#
# Multinomial Regression
predValLog <- predict(fitLog, type = "class", newdata = dat_test)
tabValLog <- table(dat_test$classes, predValLog)
tabValLog
```

```
##                       predValLog
##                        cotton_crop damp_grey_soil grey_soil red_soil
##   cotton_crop                   89              0         1        0
##   damp_grey_soil                 0             47        19        2
##   grey_soil                      0              6       173        1
##   red_soil                       0              0         4      217
##   vegetation_stubble             5              1         1        2
##   very_damp_grey_soil            0             21         4        0
##                       predValLog
##                        vegetation_stubble very_damp_grey_soil
##   cotton_crop                           7                   0
##   damp_grey_soil                        1                  31
##   grey_soil                             0                   1
##   red_soil                              4                   0
##   vegetation_stubble                   72                  18
##   very_damp_grey_soil                   8                 200
```

```
accLog <- sum(diag(tabValLog))/sum(tabValLog)
```

```
# print accuracy
acc <- c(random_forest = accCt, multinomial = accLog)
acc
```

```
## random_forest   multinomial
##     0.9112299     0.8534759
```

```
# use the method that did best on the validation data
# to predict the test data
best <- names( which.max(acc) )
switch(best,
       random_forest = {
         predTestCt <- predict(fitCt, type = "class", newdata = dat_test)
         tabTestCt <- table(dat_test$classes, predTestCt)
         accBest <- sum(diag(tabTestCt))/sum(tabTestCt)
       },
       multinomial = {
         predTestLog <- predict(fitLog, type = "class", newdata = dat_test)
```

```
        tabTestLog <- table(dat_test$classes, predTestLog)
        accBest <- sum(diag(tabTestLog))/sum(tabTestLog)
      }
)
best
```

```
## [1] "random_forest"
```

```
accBest
```

```
## [1] 0.9112299
```

```
# replicate the process a number of times
R <- 100
out <- matrix(NA, R, 4)
colnames(out) <- c("val_random_forest", "val_logistic", "best", "test")
out <- as.data.frame(out)

for ( r in 1:R ) {

  # split the data
  keep <- sample(1:D, 5500)
  test <- setdiff(1:D, keep)

  dat_test <- as.data.frame(Satellite[test,])


  train <- sample(keep, size = 0.7*5500)                    # 70% of data points are used as training
  val <- sample( setdiff(keep, train) )     # 30% of data points are used as validation data
  dat <- as.data.frame(Satellite[train,])
  dat_val <- as.data.frame(Satellite[val,])

  # fit classifiers to only the training data
  fitCt <- randomForest(classes ~ ., data = dat,trace=FALSE)        # Random Forest
  fitLog <- multinom(classes ~ ., data =  dat,trace=FALSE)    # multinomial logistic regression

  # classify the validation data observations
  predValCt <- predict(fitCt, type = "class", newdata = dat_val)    # Random forest
  tabValCt <- table(dat_val$classes, predValCt)
  tabValCt
  accCt <- sum(diag(tabValCt))/sum(tabValCt)
  #
  predValLog <- predict(fitLog, type = "class", newdata = dat_val)  # logistic regression
  tabValLog <- table(dat_val$classes, predValLog)
  tabValLog
  accLog <- sum(diag(tabValLog))/sum(tabValLog)

  # accuracy
  acc <- c(random_Forest = accCt, multinomial = accLog)
  out[r,1] <- accCt
  out[r,2] <- accLog
```

```
  # use the method that did best on the validation data
  # to predict the test data
  best <- names( which.max(acc) )
  switch(best,
         random_Forest = {
           predTestCt <- predict(fitCt, type = "class", newdata = dat_test)
           tabTestCt <- table(dat_test$classes, predTestCt)
           accBest <- sum(diag(tabTestCt))/sum(tabTestCt)
         },
         multinomial = {
           predTestLog <- predict(fitLog, type = "class", newdata = dat_test)
           tabTestLog <- table(dat_test$classes, predTestLog)
           accBest <- sum(diag(tabTestLog))/sum(tabTestLog)
         }
  )
  out[r,3] <- best
  out[r,4] <- accBest

}
```

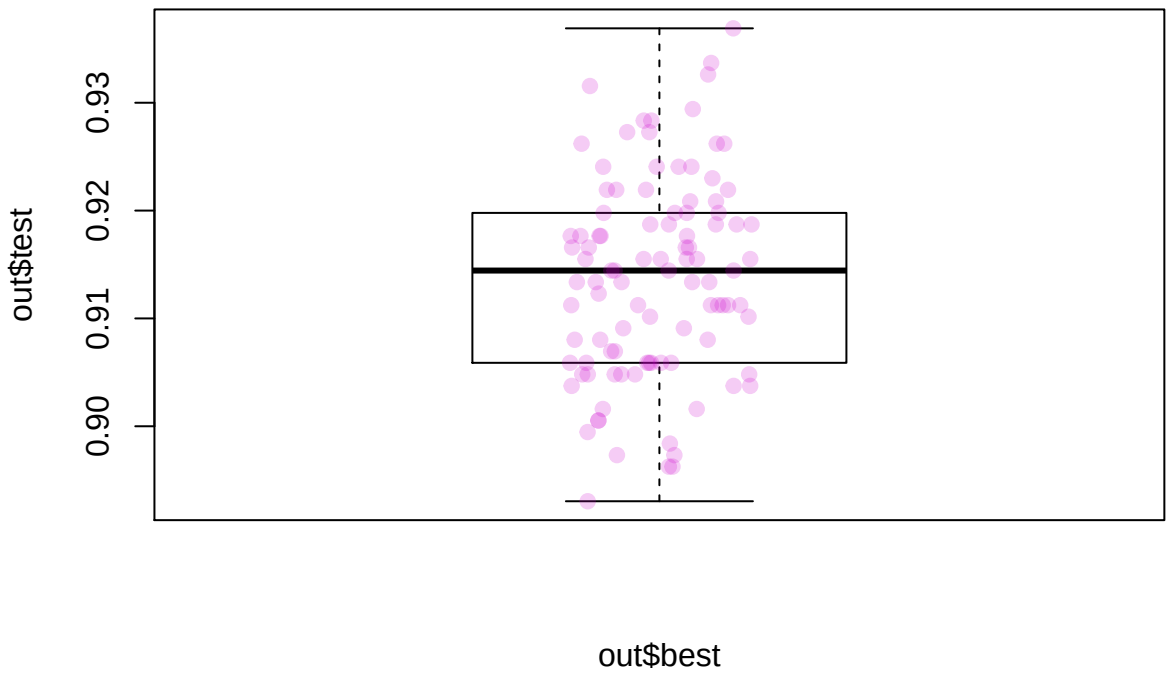
```
# check out the error rate summary statistics
table(out[,3])
```

```
##
## random_Forest
##           100
```

```
tapply(out[,4], out[,3], summary)
```

```
## $random_Forest
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8930  0.9059  0.9144  0.9138  0.9198  0.9369
```

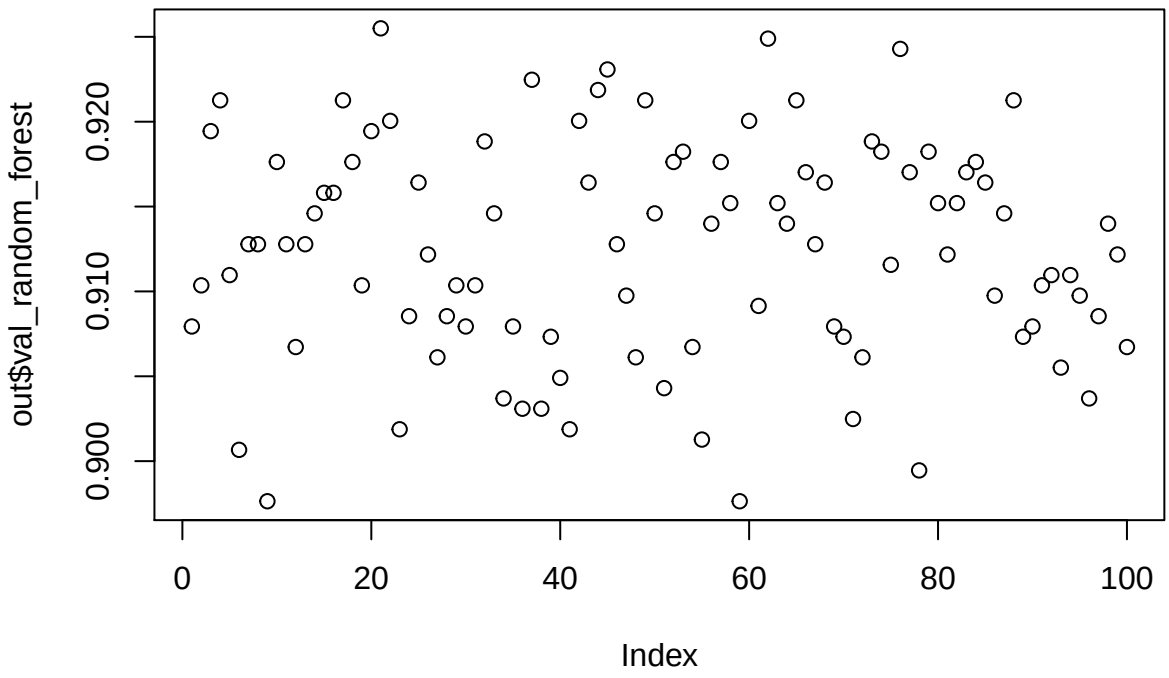
```
boxplot(out$test ~ out$best)
stripchart(out$test ~ out$best, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 19, col = adjustcolor("magenta3", 0.2))
```




```
mean(out$val_random_forest)
```

```
## [1] 0.9124228
```

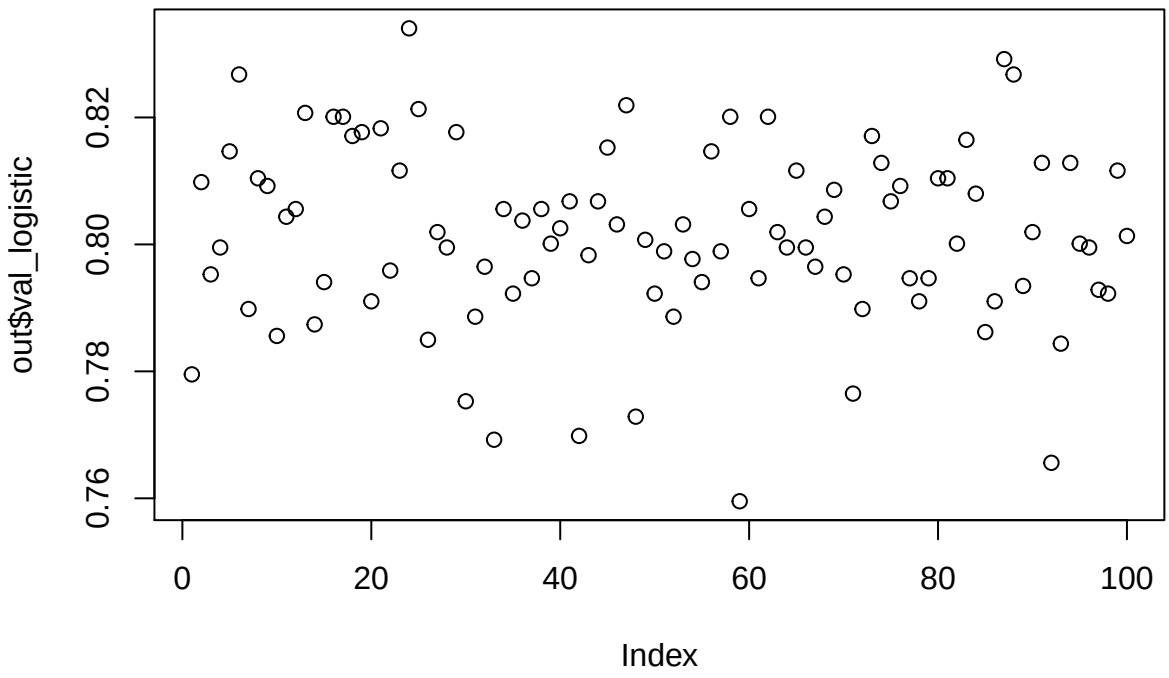
```
plot(out$val_random_forest)
```




```
mean(out$val_logistic)
```

```
## [1] 0.801593
```

```
plot(out$val_logistic)
```




```
var(out$val_random_forest)
```

```
## [1] 4.19048e-05
```

```
var(out$val_logistic)
```

```
## [1] 0.0002012216
```