White_RF_Final.R

nebojsahrnjez

2021-12-01

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(moments)
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:purrr':
##
##     some

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(ggplot2)
library(ggrepel)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
#Libraries from exploratory analysis

library(cvTools)
```

```
## Loading required package: lattice

## Loading required package: robustbase
```

```
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:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(ordinalForest)
#Libraries for this script
options(warn=-1)
#Supresses warnings added in after code was complete, code would not compile
#due to the warnings from ordinalForest

white <- read_csv("winequality-white.csv")
```

```
## Rows: 4898 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar, chlo...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
sum(is.na(white))
```

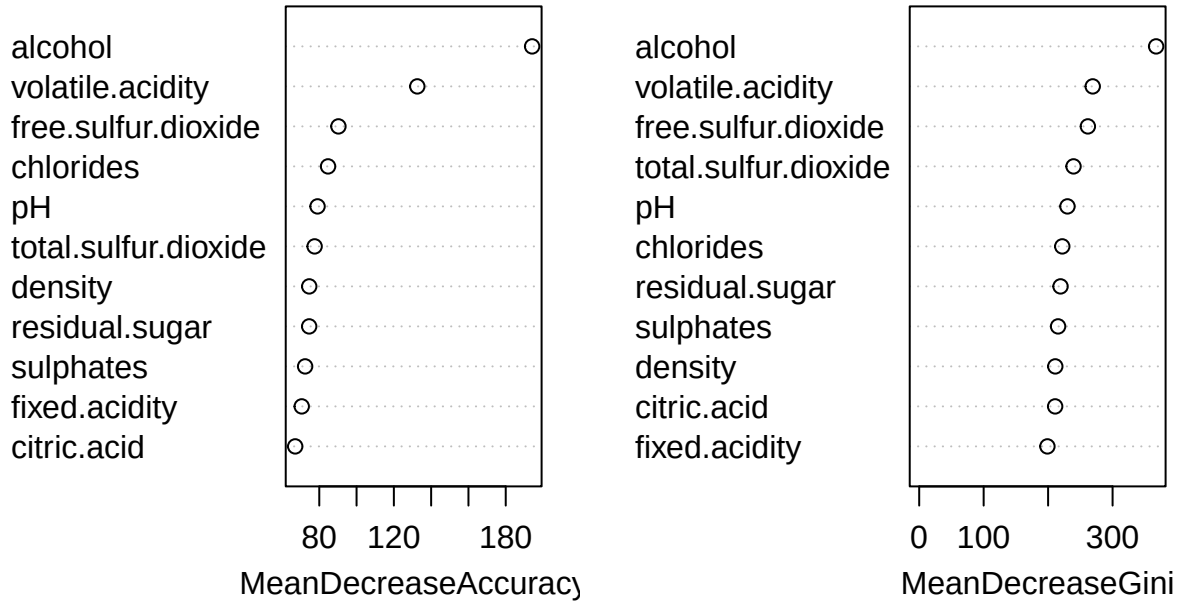
```
## [1] 0
```

```r
white <- na.omit(white)
#Reading in the data

dfw <- as.data.frame(white)

dfw <- dfw[-2782,]
dfw2 <- subset(dfw, select = -density)

dfw$quality <- as.factor(dfw$quality)
dfw2$quality <- as.factor(dfw$quality)

names(dfw) <- make.names(names(dfw))
names(dfw2) <- make.names(names(dfw2))

#Creating the dataframes to be used

RF.results <- data.frame(matrix(ncol=2,nrow=0,
                  dimnames=list(NULL, c("Model", "Classification Accuracy %"))))

#Empty data frame for results

set.seed(100)

test <- sample(1:nrow(dfw), size = nrow(dfw)/5)
train <- (-test)


dfw.train <- dfw[train,]
dfw.test <- dfw[test,]
```

```r
#Training and Test sets for dfw

w.rf <- randomForest(quality~., data = dfw.train, mtry =11, ntree = 500,
                     importance = TRUE)

w.rf.pred <- predict(w.rf, newdata = dfw.test)

w.rf.class <- mean(w.rf.pred == dfw.test$quality)

RF.results[1,] <- c("dfw random forest w/ training/test set",
                    round(w.rf.class*100,digits=2))

importance(w.rf)
```

```
##                               3        4         5        6         7        8
## fixed.acidity        -0.19815025 16.81047  49.12067 43.99115  40.36680 38.47280
## volatile.acidity     -0.67255545 38.57130  80.71581 69.78645  80.77170 60.40624
## citric.acid          -0.13691553 26.85008  38.56879 45.26911  43.83755 29.35741
## residual.sugar       -2.29805753 19.28787  43.74502 47.71863  41.98899 29.74113
## chlorides            -0.26524194 20.16473  49.35813 38.76396  62.78233 39.76878
## free.sulfur.dioxide   2.40319665 36.62505  50.27702 55.24694  48.82240 41.80909
## total.sulfur.dioxide  0.09725676 19.18628  43.09484 40.26648  52.54856 35.54459
## density              -0.50891343 16.82717  33.01243 44.98235  42.89418 29.87978
## pH                   -0.19067965 19.58999  49.93915 43.76638  52.04474 32.85690
## sulphates            -1.70188109 19.60654  41.42780 45.41766  43.54630 33.74976
## alcohol              -5.26920772 27.96764 124.06227 68.11730 133.39012 96.16808
##                               9 MeanDecreaseAccuracy MeanDecreaseGini
## fixed.acidity        -1.0010015             70.59406         198.8900
## volatile.acidity      1.0010015            132.57249         269.1800
## citric.acid          -1.0010015             67.09409         210.8973
## residual.sugar        0.0000000             74.58575         219.2771
## chlorides             0.0000000             84.70852         221.9345
## free.sulfur.dioxide  -0.2425499             90.27665         261.6894
## total.sulfur.dioxide -1.0010015             77.50284         239.4119
## density               1.0010015             74.61638         211.1060
## pH                   -1.4170505             79.06876         230.0347
## sulphates             0.0000000             72.40811         215.4684
## alcohol              -1.0010015            194.14783         367.5649
```

```r
varImpPlot(w.rf)
```

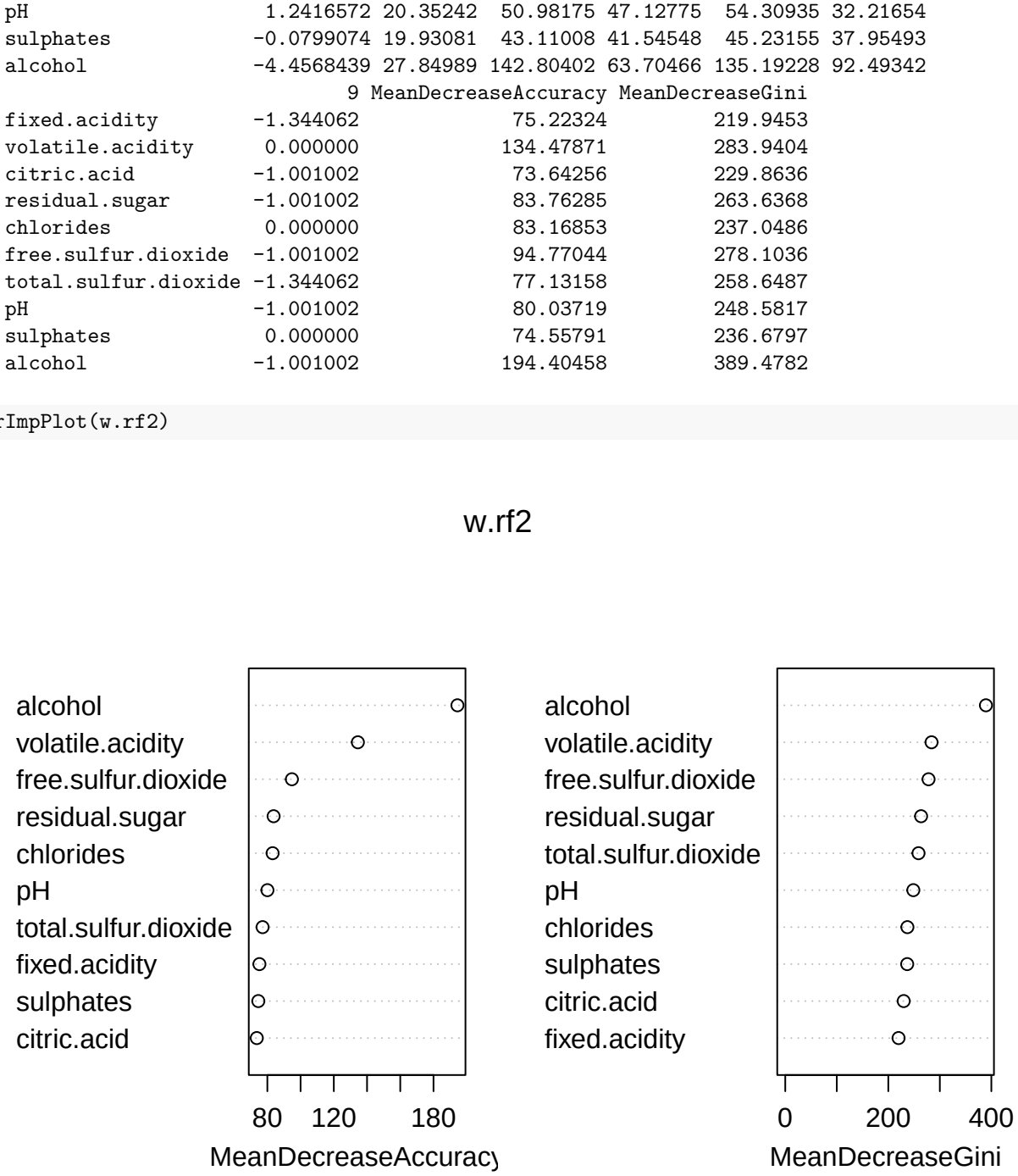



```
#dwf random forest with training/test set

dfw.train2 <- dfw2[train,]
dfw.test2 <- dfw2[test,]

#Training and Test sets for dfw2

w.rf2 <- randomForest(quality~., data = dfw.train2, mtry =10, ntree = 500,
                      importance = TRUE)

w.rf.pred2 <- predict(w.rf2, newdata = dfw.test2)

w.rf.class2 <- mean(w.rf.pred2 == dfw.test2$quality)

RF.results[2,] <- c("dfw2 random forest w/ training/test set",
                    round(w.rf.class2*100,digits=2))

importance(w.rf2)
```

```
##                              3        4         5        6         7        8
## fixed.acidity         1.1212146 15.16496  49.87448 44.43943  47.30505 40.94027
## volatile.acidity     -0.7658610 42.15082  82.13579 72.04561  80.64637 62.00883
## citric.acid           0.6298817 27.53278  39.81413 49.35675  48.00140 29.51876
## residual.sugar       -1.4511069 20.57610  47.25491 51.80525  50.37756 35.08008
## chlorides             0.8998934 22.03711  56.22455 38.22210  60.95854 36.17587
```

```
## free.sulfur.dioxide   1.9906456 40.06023  53.65823 62.00072  48.59247 40.79102
## total.sulfur.dioxide  0.2961856 18.51994  43.58027 40.60753  53.64248 36.25914
## pH                    1.2416572 20.35242  50.98175 47.12775  54.30935 32.21654
## sulphates            -0.0799074 19.93081  43.11008 41.54548  45.23155 37.95493
## alcohol              -4.4568439 27.84989 142.80402 63.70466 135.19228 92.49342
##                              9 MeanDecreaseAccuracy MeanDecreaseGini
## fixed.acidity        -1.344062             75.22324         219.9453
## volatile.acidity      0.000000            134.47871         283.9404
## citric.acid          -1.001002             73.64256         229.8636
## residual.sugar       -1.001002             83.76285         263.6368
## chlorides             0.000000             83.16853         237.0486
## free.sulfur.dioxide  -1.001002             94.77044         278.1036
## total.sulfur.dioxide -1.344062             77.13158         258.6487
## pH                   -1.001002             80.03719         248.5817
## sulphates             0.000000             74.55791         236.6797
## alcohol              -1.001002            194.40458         389.4782
```

```
varImpPlot(w.rf2)
```




```
#dwf2 random forest with training/test set

k <- 10 #number of folds

folds <- cvFolds(nrow(dfw), K=k)
folds2 <- cvFolds(nrow(dfw2), K=k)
```

```
w.rf.cv.class <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))
w.rf.cv.class2 <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))

#preparing both datasets for cross-validation

for(i in 1:k){
  tr.rf <- dfw[folds$subsets[folds$which != i],]
  te.rf <- dfw[folds$subsets[folds$which == i],]

  w.rf.cv <- randomForest(quality~., data = tr.rf, mtry =11, ntree = 500,
                          importance = TRUE)

  w.rf.pred.cv <- predict(w.rf.cv, newdata = te.rf)

  w.rf.cv.class[i] <- mean(w.rf.pred.cv == te.rf$quality)

}

w.rf.cv.class
```

```
##               1
##  [1,] 0.7122449
##  [2,] 0.6816327
##  [3,] 0.6693878
##  [4,] 0.7061224
##  [5,] 0.7102041
##  [6,] 0.6836735
##  [7,] 0.7489796
##  [8,] 0.6584867
##  [9,] 0.6666667
## [10,] 0.6707566
```

```
w.rf.cv.class <- mean(w.rf.cv.class)
print(paste("The average outputs correctly predicted is",
            round(w.rf.cv.class*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 69.08 %"
```

```
RF.results[3,] <- c("dfw Random Forest w/ 10-fold CV",
                    round(w.rf.cv.class*100,digits=2))

#dfw random forest with cross-validation

for(i in 1:k){
  tr.rf <- dfw2[folds2$subsets[folds2$which != i],]
  te.rf <- dfw2[folds2$subsets[folds2$which == i],]

  w.rf.cv2 <- randomForest(quality~., data = tr.rf, mtry =10, ntree = 500,
                          importance = TRUE)

  w.rf.pred.cv2 <- predict(w.rf.cv2, newdata = te.rf)
```

```
  w.rf.cv.class2[i] <- mean(w.rf.pred.cv2 == te.rf$quality)

}

w.rf.cv.class2
```

```
##               1
##  [1,] 0.6816327
##  [2,] 0.7040816
##  [3,] 0.6938776
##  [4,] 0.7020408
##  [5,] 0.7244898
##  [6,] 0.7061224
##  [7,] 0.7122449
##  [8,] 0.6421268
##  [9,] 0.7116564
## [10,] 0.7137014
```

```
w.rf.cv.class2 <- mean(w.rf.cv.class2)
print(paste("The average outputs correctly predicted is",
            round(w.rf.cv.class2*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 69.92 %"
```

```
RF.results[4,] <- c("dfw Random Forest w/ 10-fold CV",
                    round(w.rf.cv.class2*100,digits=2))

#dfw2 random forest with cross-validation

w.ordfor <- ordfor("quality", data = dfw.train, mtry = 11,
                   nsets = 100, ntreeperdiv = 10, ntreefinal = 500,
                   nbest = 1, npermtrial = 50)

w.ordfor.pred <- predict(w.ordfor, newdata = dfw.test)

w.ordfor.class <- mean(w.ordfor.pred$ypred==dfw.test$quality)

RF.results[5,] <- c("dfw Ordinal Forest w/ training/test set",
                    round(w.ordfor.class*100,digits=2))


head(sort(w.ordfor$varimp, decreasing = TRUE), 4)
```

```
##             alcohol    volatile.acidity free.sulfur.dioxide      residual.sugar
##          0.05266326          0.02736551          0.01563804          0.01519016
```

```
#dfw ordinal forest with training/test set

w.ordfor2 <- ordfor("quality", data = dfw.train2, mtry = 10,
                    nsets = 100, ntreeperdiv = 10, ntreefinal = 500,
                    nbest = 1, npermtrial = 50)
```

```
w.ordfor.pred2 <- predict(w.ordfor2, newdata = dfw.test2)

w.ordfor.class2 <- mean(w.ordfor.pred2$ypred==dfw.test2$quality)

RF.results[6,] <- c("dfw2 Ordinal Forest w/ training/test set",
                    round(w.ordfor.class2*100,digits=2))

head(sort(w.ordfor2$varimp, decreasing = TRUE), 4)
```

```
##             alcohol    volatile.acidity free.sulfur.dioxide           chlorides
##          0.05138017          0.02879295          0.01648423          0.01446572
```

```
#dfw2 ordinal forest with training/test set

w.ordfor.cv.class <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))
w.ordfor.cv.class2 <- matrix(NA,k,1, dimnames=list(NULL, paste(1)))

#preparing both datasets for cross-validation

for(i in 1:k){
  tr.of <- dfw[folds$subsets[folds$which != i],]
  te.of <- dfw[folds$subsets[folds$which == i],]

  w.ordfor.cv <- ordfor("quality", data = tr.of, mtry = 11,
                        nsets = 100, ntreeperdiv = 10, ntreefinal = 500,
                        nbest = 1, npermtrial = 50)

  w.ordfor.pred.cv <- predict(w.ordfor.cv, newdata = te.of)

  w.ordfor.cv.class[i] <- mean(w.ordfor.pred.cv$ypred == te.of$quality)

}

w.ordfor.cv.class
```

```
##               1
##  [1,] 0.7000000
##  [2,] 0.6693878
##  [3,] 0.6653061
##  [4,] 0.6877551
##  [5,] 0.7183673
##  [6,] 0.6714286
##  [7,] 0.7122449
##  [8,] 0.6728016
##  [9,] 0.6809816
## [10,] 0.6400818
```

```
w.ordfor.cv.class <- mean(w.ordfor.cv.class)
print(paste("The average outputs correctly predicted is",
            round(w.ordfor.cv.class*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 68.18 %"
```

```
RF.results[7,] <- c("dfw Ordinal Forest w/ 10-fold CV",
                    round(w.ordfor.cv.class*100,digits=2))

#dfw ordinal forest with 10-fold cross validation

for(i in 1:k){
  tr.of <- dfw2[folds2$subsets[folds2$which != i],]
  te.of <- dfw2[folds2$subsets[folds2$which == i],]

  w.ordfor.cv2 <- ordfor("quality", data = tr.of, mtry = 10,
                         nsets = 100, ntreeperdiv = 10, ntreefinal = 500,
                         nbest = 1, npermtrial = 50)

  w.ordfor.pred.cv2 <- predict(w.ordfor.cv2, newdata = te.of)

  w.ordfor.cv.class2[i] <- mean(w.ordfor.pred.cv2$ypred == te.of$quality)

}

w.ordfor.cv.class2
```

```
##               1
##  [1,] 0.6714286
##  [2,] 0.6857143
##  [3,] 0.6857143
##  [4,] 0.6938776
##  [5,] 0.7102041
##  [6,] 0.6857143
##  [7,] 0.7020408
##  [8,] 0.6543967
##  [9,] 0.7157464
## [10,] 0.7137014
```

```
w.ordfor.cv.class2 <- mean(w.ordfor.cv.class2)
print(paste("The average outputs correctly predicted is",
            round(w.ordfor.cv.class2*100,digits =2),"%",sep=" "))
```

```
## [1] "The average outputs correctly predicted is 69.19 %"
```

```
RF.results[8,] <- c("dfw2 Ordinal Forest w/ 10-fold CV",
                    round(w.ordfor.cv.class2*100,digits=2))

#dfw2 ordinal forest with 10-fold cross validation

RF.results
```

```
##                                    Model Classification.Accuracy..
## 1   dfw random forest w/ training/test set                    68.23
## 2  dfw2 random forest w/ training/test set                    68.13
## 3          dfw Random Forest w/ 10-fold CV                    69.08
## 4          dfw Random Forest w/ 10-fold CV                    69.92
## 5  dfw Ordinal Forest w/ training/test set                    67.93
```

```
## 6 dfw2 Ordinal Forest w/ training/test set                    68.23
## 7         dfw Ordinal Forest w/ 10-fold CV                    68.18
## 8        dfw2 Ordinal Forest w/ 10-fold CV                    69.19
```