White_MLR_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(MASS)
```

```
##
## Attaching package: ’MASS’
```

```
## The following object is masked from ’package:dplyr’:
##
##     select
```

```
#Libraries for this script

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)

#Creating the dataframes to be used

MLR.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)

#Training and Test sets for dfw

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

#Create a test and training data set, 20/80 split

w.mlm.train <- lm(
  quality ~.,
  data = dfw.train
)
#Create linear regression with all predictors using the training dataset

w.mlm.predict <- predict(w.mlm.train, newdata = dfw.test)
w.mlm.predict.rounded <- round(w.mlm.predict, digits = 0)

#Round the predicted to a integer so it can be compared to the test set for
# classification

(con_mat <- table(w.mlm.predict.rounded, dfw.test$quality))
```

```
##
## w.mlm.predict.rounded   3   4   5   6   7   8
##                     4   0   2   1   0   0   0
##                     5   1  16 128  52   5   1
##                     6   3  18 157 357 131  20
##                     7   0   0   3  32  38  14
```

```r
w.mlm.acc <- round(mean(w.mlm.predict.rounded==dfw.test$quality)*100,digits = 2)

#Create the confusion matrix and calculate the proportion correct

MLR.results[1,] <- c("dfw MLR w/ Train/Test", w.mlm.acc)

#dfw train/test multi linear regression

test <- sample(1:nrow(dfw2), size = nrow(dfw2)/5)
train <- (-test)

dfw2.train <- dfw2[train,]
dfw2.test <- dfw2[test,]

#Create a test and training data set, 20/80 split

w.mlm.train2 <- lm(
  quality ~.,
  data = dfw2.train
)
#Create linear regression with all predictors using the training dataset

w.mlm.predict2 <- predict(w.mlm.train2, newdata = dfw2.test)
w.mlm.predict.rounded2 <- round(w.mlm.predict2, digits = 0)

#Round the predicted to a integer so it can be compared to the test set for
# classification

(con_mat <- table(w.mlm.predict.rounded2, dfw2.test$quality))
```

```
##
## w.mlm.predict.rounded2   3   4   5   6   7   8   9
##                      4   0   1   3   1   0   0   0
##                      5   0  12 135  59   2   0   0
##                      6   4  18 141 340 134  17   1
##                      7   0   1   5  52  46   5   2
```

```r
w.mlm.acc2 <- round(mean(w.mlm.predict.rounded2==dfw2.test$quality)*100,
                    digits = 2)

#Create the confusion matrix and calculate the proportion correct

MLR.results[2,] <- c("dfw2 MLR w/ Train/Test", w.mlm.acc2)

#dfw2 train/test multi linear regression

k <- 10 #number of folds

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

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

```
#Preparing both datasets for cross-validation

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

  w.mlm <- lm(quality~., data = tr.mlr)
  w.mlm.pred <- predict(w.mlm, newdata = te.mlr)

  w.mlm.cv.class[i] <- mean(round(w.mlm.pred, digits = 0)==te.mlr$quality)
}

w.mlm.cv.class
```

```
##                1
##  [1,] 0.4959184
##  [2,] 0.5040816
##  [3,] 0.5469388
##  [4,] 0.5285714
##  [5,] 0.5265306
##  [6,] 0.5061224
##  [7,] 0.5387755
##  [8,] 0.5235174
##  [9,] 0.5092025
## [10,] 0.4989775
```

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

```
## [1] "The average outputs correctly predicted is 51.79 %"
```

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

#dfw cross-validated Multiple Linear Regression

for(i in 1:k){
  tr.mlr2 <- dfw2[folds2$subsets[folds2$which != i],]
  te.mlr2 <- dfw2[folds2$subsets[folds2$which == i],]

  w.mlm2 <- lm(quality~., data = tr.mlr2)
  w.mlm.pred2 <- predict(w.mlm2, newdata = te.mlr2)

  w.mlm.cv.class2[i] <- mean(round(w.mlm.pred2, digits = 0)==te.mlr2$quality)
}

w.mlm.cv.class2
```

```
##                1
##  [1,] 0.5163265
##  [2,] 0.5244898
##  [3,] 0.5571429
##  [4,] 0.5142857
##  [5,] 0.4938776
##  [6,] 0.4897959
##  [7,] 0.5040816
##  [8,] 0.5480573
##  [9,] 0.5214724
## [10,] 0.4846626
```

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

```
## [1] "The average outputs correctly predicted is 51.54 %"
```

```
MLR.results[4,] <- c("dfw2 MLR w/ 10-fold CV", round(w.mlm.cv.class2*100,
                                                     digits=2))

#dfw2 cross-validated Multiple Linear Regression

dfw$quality <- factor(dfw$quality, ordered = TRUE)
dfw2$quality <- factor(dfw2$quality, ordered = TRUE)

#Making the response variable a factor for ordinal logistic regression

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

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

test <- sample(1:nrow(dfw2), size = nrow(dfw2)/5)
train <- (-test)

dfw2.train <- dfw2[train,]
dfw2.test <- dfw2[test,]

#Re-create training and test set with new factored response variable

w.olr <- polr(quality~., data = dfw.train, Hess = TRUE)

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

w.olr.pred <- as.numeric(as.character(unlist(w.olr.pred)))
dfw.test$quality <- as.numeric(as.character(unlist(dfw.test$quality)))

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

MLR.results[5,] <- c("dfw OLR w/ Training/Test", round(w.olr.class*100,
                                                       digits=2))

#dfw OLR w/ Training/Test Set
```

```r
w.olr2 <- polr(quality~., data = dfw2.train, Hess = TRUE)

w.olr.pred2 <- predict(w.olr2, newdata = dfw2.test)

w.olr.pred2 <- as.numeric(as.character(unlist(w.olr.pred2)))
dfw2.test$quality <- as.numeric(as.character(unlist(dfw2.test$quality)))

w.olr.class2 <- mean(w.olr.pred2 == dfw2.test$quality)

MLR.results[6,] <- c("dfw2 OLR w/ Training/Test", round(w.olr.class2*100,
                                                       digits=2))

#dfw2 OLR w/ Training/Test Set

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

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

#Re-create the folds and empty classification matrix for OLR

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

  w.olr.cv <- polr(quality~., data = tr.olr, Hess = TRUE)
  w.olr.cv.pred <- predict(w.olr.cv, newdata = te.olr)

  w.olr.cv.pred <- as.numeric(as.character(unlist(w.olr.cv.pred)))
  te.olr$quality <- as.numeric(as.character(unlist(te.olr$quality)))

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

w.olr.cv.class
```

```
##                1
##  [1,] 0.5448980
##  [2,] 0.5469388
##  [3,] 0.5285714
##  [4,] 0.5020408
##  [5,] 0.5346939
##  [6,] 0.4897959
##  [7,] 0.5061224
##  [8,] 0.5153374
##  [9,] 0.5562372
## [10,] 0.5337423
```

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

```
## [1] "The average outputs correctly predicted is 52.58 %"
```

```
MLR.results[7,] <- c("dfw OLR w/ 10-fold CV", round(w.olr.cv.class*100,
                                                    digits=2))

#dfw OLR w/ 10-fold CV

for(i in 1:k){
  tr.olr2 <- dfw2[folds2$subsets[folds2$which != i],]
  te.olr2 <- dfw2[folds2$subsets[folds2$which == i],]

  w.olr.cv2 <- polr(quality~., data = tr.olr2, Hess = TRUE)
  w.olr.cv.pred2 <- predict(w.olr.cv2, newdata = te.olr2)

  w.olr.cv.pred2 <- as.numeric(as.character(unlist(w.olr.cv.pred2)))
  te.olr2$quality <- as.numeric(as.character(unlist(te.olr2$quality)))

  w.olr.cv.class2[i] <- mean(w.olr.cv.pred2==te.olr2$quality)
}

w.olr.cv.class2
```

```
##                1
##  [1,] 0.4959184
##  [2,] 0.5346939
##  [3,] 0.5244898
##  [4,] 0.5612245
##  [5,] 0.5265306
##  [6,] 0.5265306
##  [7,] 0.5122449
##  [8,] 0.5235174
##  [9,] 0.5296524
## [10,] 0.5378323
```

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

```
## [1] "The average outputs correctly predicted is 52.73 %"
```

```
MLR.results[8,] <- c("dfw2 OLR w/ 10-fold CV", round(w.olr.cv.class2*100,
                                                    digits=2))

#dfw2 OLR w/ 10-fold CV


MLR.results
```

```
##                         Model Classification.Accuracy..
## 1     dfw MLR w/ Train/Test                     53.63
## 2    dfw2 MLR w/ Train/Test                     53.32
## 3     dfw MLR w/ 10-fold CV                     51.79
## 4    dfw2 MLR w/ 10-fold CV                     51.54
## 5  dfw OLR w/ Training/Test                     51.48
## 6 dfw2 OLR w/ Training/Test                     54.34
## 7     dfw OLR w/ 10-fold CV                     52.58
## 8    dfw2 OLR w/ 10-fold CV                     52.73
```