Weekly Assignment 5

Benedikt Sojka

2024-03-12

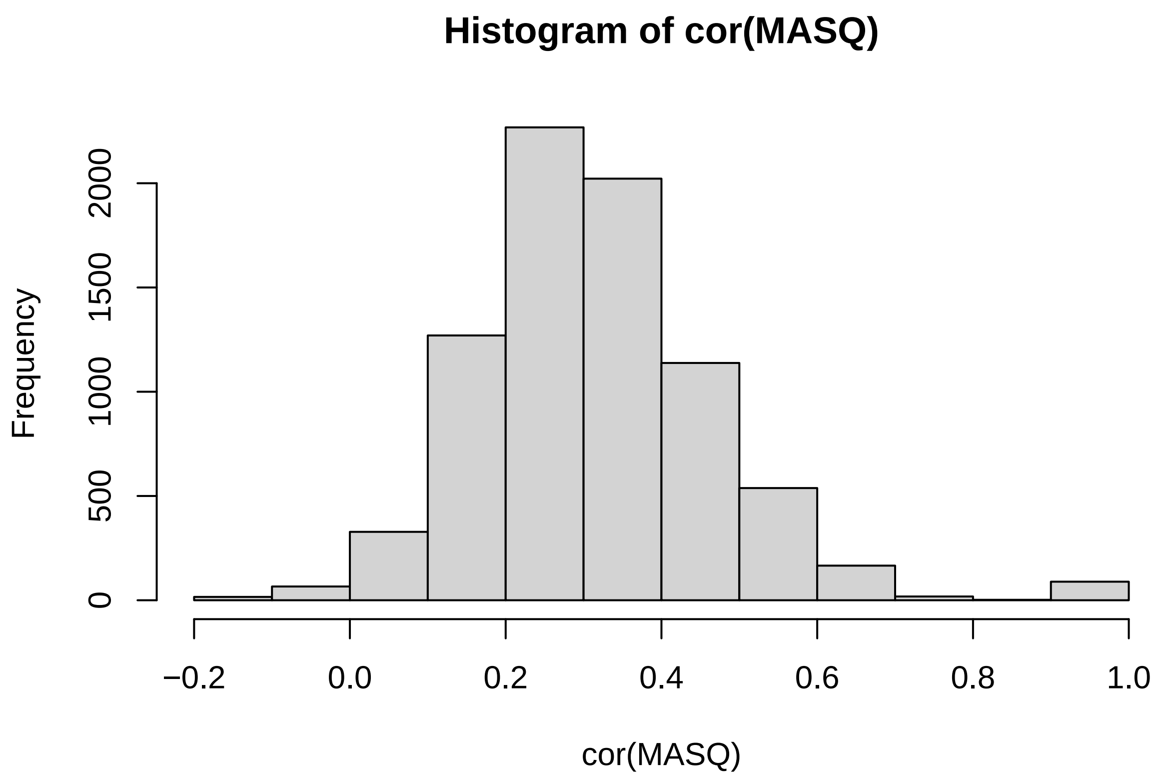
```r
train <- readRDS("masq_train.Rda")
test <- readRDS("masq_test.Rda")
```

(a) Inspect multicollinearity between the numeric MASQ items. What do you expect about relative per- formance of lasso, ridge and elastic net regression?

```r
colnames(train)
```

```
##   [1] "D_DEPDYS" "GENDER"   "Leeftijd" "DEMOG1"   "DEMOG2"   "DEMOG3"
##   [7] "DEMOG4"   "DEMOG5"   "DEMOG6"   "DEMOG7"   "DEMOG8"   "MASQ01"
##  [13] "MASQ02"   "MASQ03"   "MASQ04"   "MASQ05"   "MASQ06"   "MASQ07"
##  [19] "MASQ08"   "MASQ09"   "MASQ11"   "MASQ12"   "MASQ13"   "MASQ14"
##  [25] "MASQ15"   "MASQ16"   "MASQ17"   "MASQ18"   "MASQ19"   "MASQ20"
##  [31] "MASQ21"   "MASQ22"   "MASQ23"   "MASQ24"   "MASQ25"   "MASQ26"
##  [37] "MASQ27"   "MASQ28"   "MASQ29"   "MASQ30"   "MASQ31"   "MASQ32"
##  [43] "MASQ33"   "MASQ34"   "MASQ35"   "MASQ36"   "MASQ37"   "MASQ38"
##  [49] "MASQ39"   "MASQ40"   "MASQ41"   "MASQ42"   "MASQ43"   "MASQ44"
##  [55] "MASQ45"   "MASQ46"   "MASQ47"   "MASQ48"   "MASQ49"   "MASQ50"
##  [61] "MASQ51"   "MASQ52"   "MASQ53"   "MASQ54"   "MASQ55"   "MASQ56"
##  [67] "MASQ57"   "MASQ58"   "MASQ59"   "MASQ60"   "MASQ61"   "MASQ62"
##  [73] "MASQ63"   "MASQ64"   "MASQ65"   "MASQ66"   "MASQ67"   "MASQ68"
##  [79] "MASQ69"   "MASQ70"   "MASQ71"   "MASQ72"   "MASQ73"   "MASQ74"
##  [85] "MASQ75"   "MASQ76"   "MASQ77"   "MASQ78"   "MASQ79"   "MASQ80"
##  [91] "MASQ81"   "MASQ82"   "MASQ83"   "MASQ84"   "MASQ85"   "MASQ86"
##  [97] "MASQ87"   "MASQ88"   "MASQ89"   "MASQ90"
```

```r
MASQ = train[, 12:100]
hist(cor(MASQ))
```



The pairwise correlations follow a normal distribution around 0.3. All 3 techniques might be useful to handle this issue. Elastic net might be the best choice as it combines the benefits of lasso and ridge. A real conclusion cannot be drawn without further investigation of the methods performances.

(b) Pick three candidate procedures from ridge, elastic net, lasso, relaxed lasso.

I'll compare the three methods 'lasso', 'ridge', and 'elastic net with an alpha of 0.5.

(c) Select the most accurate model through 10-fold cross-validation on the training set.

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
x = model.matrix(D_DEPDYS ~ ., data = train)
x_test = model.matrix(D_DEPDYS ~ ., data = test)
y = train$D_DEPDYS

lasso = cv.glmnet(x, y, alpha = 1); lasso
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 1)
##
## Measure: Mean-Squared Error
##
##       Lambda Index Measure       SE Nonzero
## min 0.01140    34   0.1727 0.005496      37
## 1se 0.04604    19   0.1777 0.004382      17
```

```r
ridge = cv.glmnet(x, y, alpha = 0); ridge
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure       SE Nonzero
## min 0.3324    72  0.1725 0.003931     132
## 1se 1.7740    54  0.1764 0.003463     132
```

```r
elastic = cv.glmnet(x, y, alpha = 0.5); elastic
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 0.5)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure       SE Nonzero
## min 0.02078    35  0.1716 0.004714      43
## 1se 0.07644    21  0.1760 0.003892      19
```

```r
PRE_L = predict(lasso, s = "lambda.min", newx = x_test)
PRE_R = predict(ridge, s = "lambda.min", newx = x_test)
PRE_E = predict(elastic, s = "lambda.min", newx = x_test)

MSE_L = mean((PRE_L - test$D_DEPDYS)^2)
MSE_R = mean((PRE_R - test$D_DEPDYS)^2)
MSE_E = mean((PRE_E - test$D_DEPDYS)^2)

MSE_MAX = var(test$D_DEPDYS)

ACC_L = 1 - MSE_L/MSE_MAX; ACC_L
```

```
## [1] 0.3331148
```

```r
ACC_R = 1 - MSE_R/MSE_MAX; ACC_R
```

```
## [1] 0.3304663
```

```r
ACC_E = 1 - MSE_E/MSE_MAX; ACC_E
```

```
## [1] 0.333626
```

We see that lasso actually performs best among the 3 methods with an cross-validated R2 of 0.3344316.

(d) Compute the misclassification rate (MCR) on the test set.

```r
preds_L_1se = predict(lasso, newx = x[x_test, ], type = "response")
preds_L_min = predict(lasso, newx = x[x_test, ], type = "response",
                      s = "lambda.min")

tab_L_1se = prop.table(table(preds_L_1se > .5, y[x_test]))
tab_L_min = prop.table(table(preds_L_min > .5, y[x_test]))
tab_L_1se; tab_L_min
```

```
##
```

```
##                   0           1
##   FALSE 0.141621916 0.001009058
##   TRUE  0.001994920 0.855374105
```

```
## 
##                     0            1
##   FALSE 0.1416857073 0.0007538942
##   TRUE  0.0019311289 0.8556292697
```

```r
sum(diag(tab_L_1se)); sum(diag(tab_L_min))
```

```
## [1] 0.996996
```

```
## [1] 0.997315
```

The MCR is slightly higher for the lambda.min criteria

(e) Use the coef method to extract the selected variables and their coefficients from the best-performing model.

```r
L_coefs = coef(lasso, s = "lambda.min")
L_coefs[L_coefs[,1] != 0,]
```

```
##   (Intercept)       GENDERv      Leeftijd       DEMOG26       DEMOG32 
## -4.814769e-01  8.170714e-03  1.149901e-03 -7.256811e-02  1.328822e-02 
##       DEMOG34      DEMOG3NA       DEMOG53       DEMOG55       DEMOG62 
## -6.421318e-05 -3.518655e-02 -3.346312e-02  1.993973e-02  5.647645e-02 
##        MASQ01        MASQ02        MASQ03        MASQ05        MASQ13 
##  3.481755e-02 -1.548544e-02 -6.228674e-03  7.338993e-04  8.940010e-03 
##        MASQ14        MASQ16        MASQ18        MASQ21        MASQ22 
##  2.865788e-03  6.948313e-02  3.406095e-03  3.920310e-03  2.403443e-02 
##        MASQ24        MASQ29        MASQ30        MASQ31        MASQ33 
##  5.460323e-03  7.899327e-04  2.265849e-02  8.409185e-03  2.287192e-03 
##        MASQ37        MASQ38        MASQ41        MASQ43        MASQ54 
##  1.893545e-02  5.152496e-03  2.533319e-02  5.592111e-03  1.655632e-03 
##        MASQ59        MASQ60        MASQ62        MASQ70        MASQ76 
## -1.041407e-02  9.627503e-03  1.395614e-02  4.299627e-03  9.622140e-03 
##        MASQ78        MASQ89        MASQ90 
##  9.813473e-03  2.667092e-02  1.438215e-02
```

```r
Anhedonic_Depression = c(1, 14, 18, 21, 23, 26, 27, 30, 33, 35, 36, 39, 40, 44, 49, 53, 58, 66, 72, 78,
Anxious_Arousal = c(3, 19, 25, 45, 48, 52, 55, 57, 61, 67, 69, 73, 75, 79, 85, 87, 88)
General_Distress_Depression = c(6, 8, 10, 13, 16, 22, 24, 42, 47, 56, 64, 74)
General_Distress_Anxiety = c(2, 9, 12, 15, 20, 59, 63, 65, 77, 81, 82)
General_Distress_Mixed = c(4, 5, 17, 29, 31, 34, 37, 50, 51, 70, 76, 80, 83, 84, 90)
```