Exercício Aplicado 01 - MAE 0501

Vítor Garcia Comissoli

2024-10-04

9)

a)

```
train <- sample(nrow(College), nrow(College) * 0.7) # 70%
test <- setdiff(seq_len(nrow(College)), train) # 30%

# Inicializando a lista de Erros quadráticos médios
mse <- list()
```

b)

```
fit <- lm(Apps ~ ., data = College[train, ])
summary(fit)
```

```
##
## Call:
## lm(formula = Apps ~ ., data = College[train, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3743.5  -472.6   -24.1   352.4  7006.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -379.80795  487.16917  -0.780 0.435965
## PrivateYes  -566.10088  169.65133  -3.337 0.000907 ***
## Accept         1.67324    0.04611  36.289  < 2e-16 ***
## Enroll        -0.70141    0.22526  -3.114 0.001947 **
## Top10perc     54.54696    6.74767   8.084 4.36e-15 ***
## Top25perc    -19.67832    5.51775  -3.566 0.000395 ***
## F.Undergrad   -0.00631    0.04048  -0.156 0.876175
## P.Undergrad    0.06128    0.04869   1.259 0.208708
## Outstate      -0.09823    0.02386  -4.118 4.44e-05 ***
## Room.Board     0.12324    0.06037   2.041 0.041701 *
## Books         -0.10736    0.29901  -0.359 0.719713
## Personal       0.07798    0.08102   0.962 0.336249
## PhD           -9.99759    5.76370  -1.735 0.083402 .
## Terminal      -1.74893    6.27165  -0.279 0.780460
## S.F.Ratio     25.11457   15.34634   1.637 0.102330
## perc.alumni    2.88724    4.92217   0.587 0.557738
## Expend         0.09018    0.01599   5.641 2.76e-08 ***
```

```
## Grad.Rate      9.30337    3.59282   2.589 0.009880 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1070 on 525 degrees of freedom
## Multiple R-squared:  0.931,  Adjusted R-squared:  0.9288
## F-statistic: 416.8 on 17 and 525 DF,  p-value: < 2.2e-16
```

```
(mse$lm <- mean((predict(fit, College[test, ]) - College$Apps[test])^2))
```

```
## [1] 1057306
```

c)

```
mm <- model.matrix(Apps ~ ., data = College[train, ])
fit2 <- cv.glmnet(mm, College$Apps[train], alpha = 0)
fit2
```

```
##
## Call:  cv.glmnet(x = mm, y = College$Apps[train], alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure      SE Nonzero
## min  378.4   100 1986606  864330      17
## 1se 1527.5    85 2780866 1432292      17
```

```
p <- predict(fit2, model.matrix(Apps ~ ., data = College[test, ]), s = fit2$lambda.min)
(mse$ridge <- mean((p - College$Apps[test])^2))
```

```
## [1] 860956.6
```

d)

```
mm <- model.matrix(Apps ~ ., data = College[train, ])
fit3 <- cv.glmnet(mm, College$Apps[train], alpha = 1)
fit3
```

```
##
## Call:  cv.glmnet(x = mm, y = College$Apps[train], alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure     SE Nonzero
## min   13.0    62 1416444 393004      14
## 1se  405.7    25 1774392 574615       3
```

```
p <- predict(fit3, model.matrix(Apps ~ ., data = College[test, ]), s = fit3$lambda.min)
(mse$las <- mean((p - College$Apps[test])^2))
```

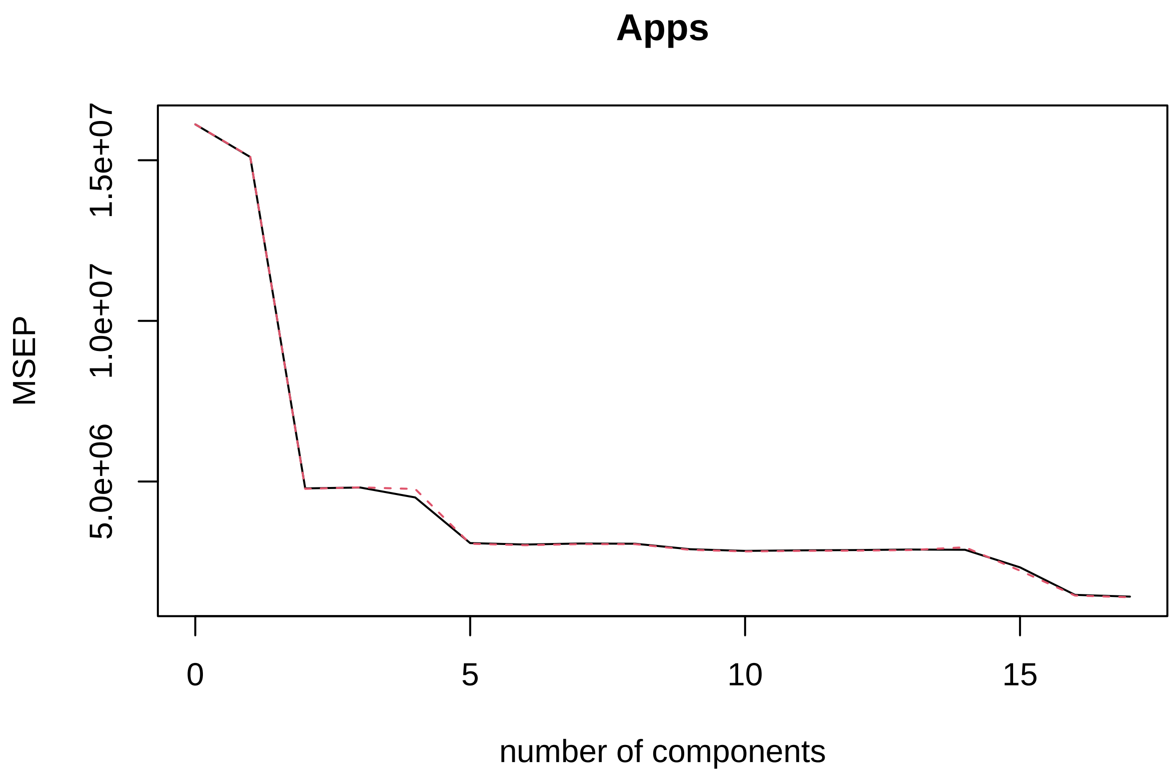
```
## [1] 1007319
```

e)

```
fit4 <- pcr(Apps ~ ., data = College[train, ], scale = TRUE, validation = "CV")
summary(fit4)
```

```
## Data:    X dimension: 543 17
##  Y dimension: 543 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4015     3886     2187     2194     2122     1756     1743
## adjCV         4015     3887     2185     2195     2183     1750     1737
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1752     1750     1701      1685      1690      1693      1698
## adjCV     1747     1746     1695      1680      1685      1688      1693
##        14 comps  15 comps  16 comps  17 comps
## CV         1695      1526      1213      1190
## adjCV      1718      1492      1204      1182
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      31.237    56.85    63.83    69.77    75.49    80.31    84.24    87.53
## Apps    6.558    71.22    71.28    72.18    82.02    82.47    82.48    82.51
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       90.48     92.96     94.99     96.72     97.79     98.63     99.34
## Apps    83.65     84.10     84.16     84.20     84.25     84.40     91.53
##       16 comps  17 comps
## X        99.83     100.0
## Apps     92.90      93.1
```

```
validationplot(fit4, val.type = "MSEP")
```




```
p <- predict(fit4, College[test, ], ncomp = 16)
(mse$pcr <- mean((p - College$Apps[test])^2))
```

```
## [1] 1157794
```

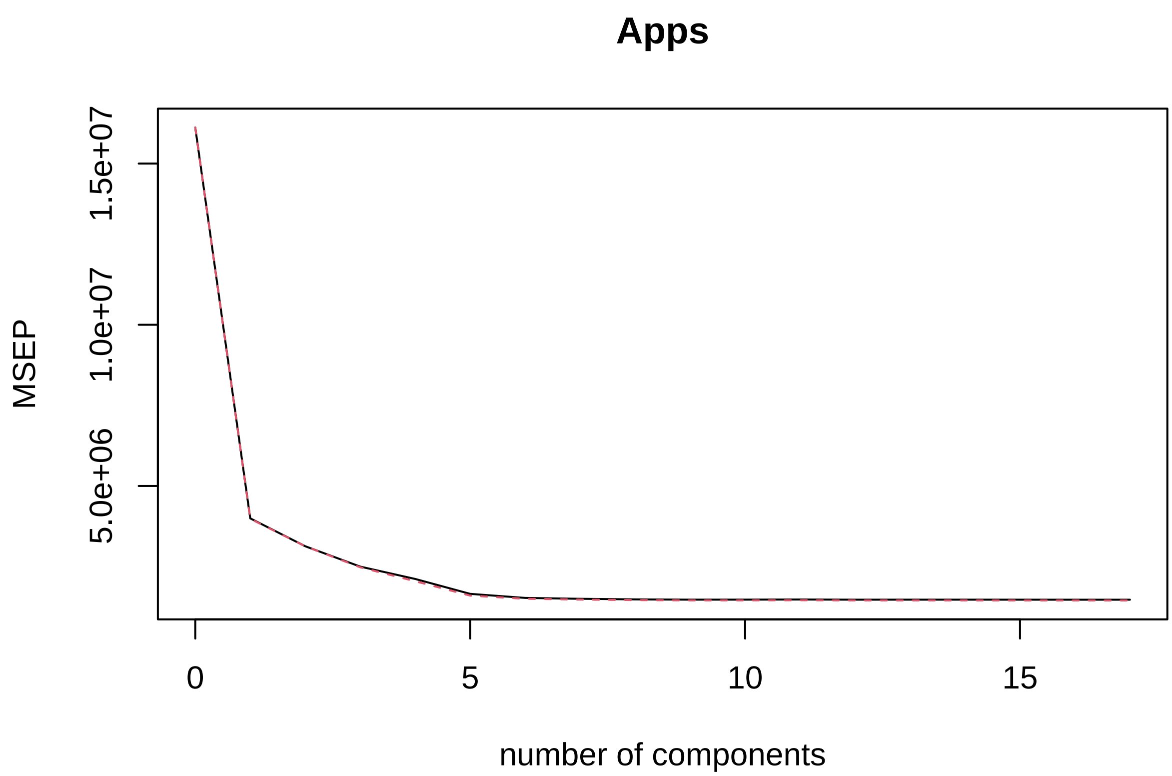
f)

```
fit5 <- plsr(Apps ~ ., data = College[train, ], scale = TRUE, validation = "CV")
summary(fit5)
```

```
## Data:    X dimension: 543 17
##  Y dimension: 543 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4015     1999     1769     1581     1454     1286     1236
## adjCV         4015     1995     1769     1574     1430     1265     1225
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1225     1217     1214      1214      1216      1214      1214
## adjCV     1215     1208     1205      1205      1206      1204      1205
##        14 comps  15 comps  16 comps  17 comps
## CV         1214      1214      1213      1213
## adjCV      1204      1204      1204      1204
```

```
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X       26.19    44.82    62.27    64.34    67.41    72.63    77.05    79.60
## Apps    76.18    82.74    86.71    91.23    92.71    92.94    92.99    93.04
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       82.73     86.53     88.52     90.73     92.85     94.79     97.16
## Apps    93.07     93.08     93.09     93.10     93.10     93.10     93.10
##       16 comps  17 comps
## X        98.92     100.0
## Apps     93.10      93.1
```

```
validationplot(fit5, val.type = "MSEP")
```

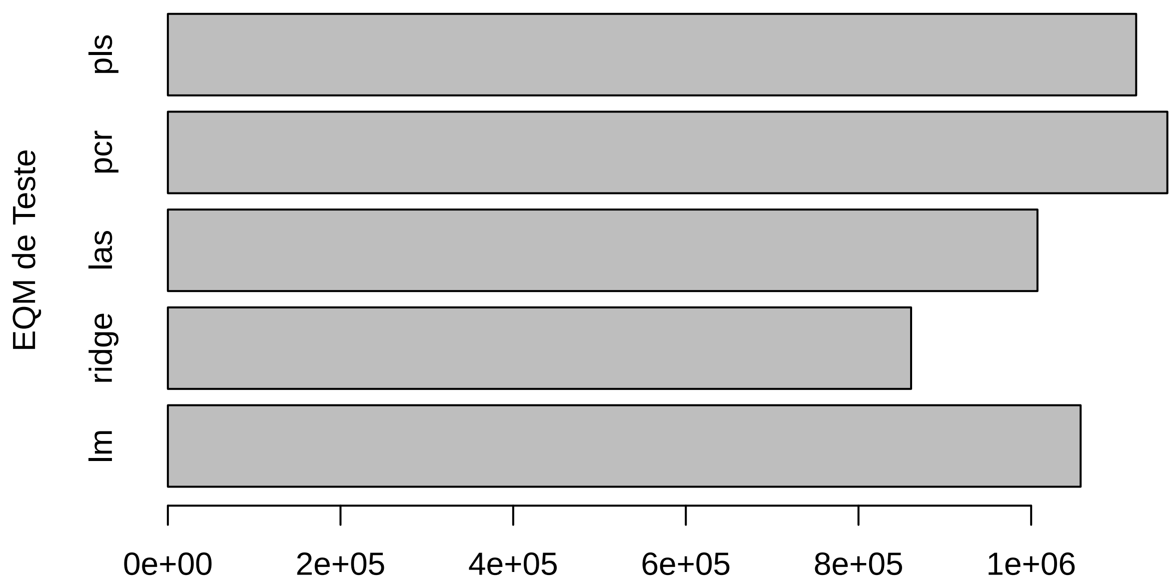



```
p <- predict(fit5, College[test, ], ncomp = 6)
(mse$pls <- mean((p - College$Apps[test])^2))
```

```
## [1] 1121669
```

g)

```
barplot(unlist(mse), ylab = "EQM de Teste", horiz = TRUE)
```



Dado as altos valores de R^2 obtidos nos modelos, pode-se dizer que eles preveem sim, com uma boa acurrária, o número de aplicações recebidas pelas universidades.

Tanto Ridge quanto Lasso foram os modelos que levaram a um menor erro de teste, sendo o menor deles oriundo regressão Ridge (dada a seed fixada na realização do exercício), assim sendo esse o modelo que eu escolheria.