Ridge Problem

Jarkko Schaad

2024-04-07

Downloading dataset Student-performance and splitting into test_set & train_set

```
library(tidyverse)
```

```
## Warning: Paket 'ggplot2' wurde unter R Version 4.3.3 erstellt
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.3.3 erstellt
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
download_and_unzip <- function(download_url, dest_dir, zip_file_name) {
  # Ensure the destination directory exists
  if (!dir.exists(dest_dir)) {
    dir.create(dest_dir)
  }

  # Define the zip file path
  zip_file_path <- file.path(dest_dir, zip_file_name)

  # Download the zip file
  download.file(url = download_url, destfile = zip_file_path, method = "auto")

  # Unzip the file
  unzip(zipfile = zip_file_path, exdir = dest_dir)
}

split_data <- function(data, split_ratio = 0.8) {
  # Splitting the data into train and test sets
  set.seed(123) # For reproducibility
  training_sample <- sample(nrow(data), size = floor(nrow(data) * split_ratio))
```

```r
  train_set <- data[training_sample, ]
  test_set <- data[-training_sample, ]

  # Return a list containing the train and test datasets
  return(list(train_set = train_set, test_set = test_set))
}

# Given dataset details
download_url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/00320/student.zip"
dest_dir <- "student-performance"
zip_file_name <- "student-performance.zip"
data_file <- "student-mat.csv"

# Download and unzip the dataset
download_and_unzip(download_url, dest_dir, zip_file_name)

# Define the path to the dataset CSV file
data_file_path <- file.path(dest_dir, data_file)

# Read the dataset
data <- read.csv(data_file_path, sep = ";")
glimpse(data)
```

```
## Rows: 395
## Columns: 33
## $ school     <chr> "GP", "GP", "GP", "GP", "GP", "GP", "GP", "GP", "GP", "GP",~
## $ sex        <chr> "F", "F", "F", "F", "F", "M", "M", "F", "M", "M", "F", "F",~
## $ age        <int> 18, 17, 15, 15, 16, 16, 16, 17, 15, 15, 15, 15, 15, 15, 15,~
## $ address    <chr> "U", "U", "U", "U", "U", "U", "U", "U", "U", "U", "U", "U",~
## $ famsize    <chr> "GT3", "GT3", "LE3", "GT3", "GT3", "LE3", "LE3", "GT3", "LE~
## $ Pstatus    <chr> "A", "T", "T", "T", "T", "T", "T", "A", "A", "T", "T", "T",~
## $ Medu       <int> 4, 1, 1, 4, 3, 4, 2, 4, 3, 3, 4, 2, 4, 4, 2, 4, 4, 3, 3, 4,~
## $ Fedu       <int> 4, 1, 1, 2, 3, 3, 2, 4, 2, 4, 4, 1, 4, 3, 2, 4, 4, 3, 2, 3,~
## $ Mjob       <chr> "at_home", "at_home", "at_home", "health", "other", "servic~
## $ Fjob       <chr> "teacher", "other", "other", "services", "other", "other", ~
## $ reason     <chr> "course", "course", "other", "home", "home", "reputation", ~
## $ guardian   <chr> "mother", "father", "mother", "mother", "father", "mother",~
## $ traveltime <int> 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 3, 1, 2, 1, 1, 1, 3, 1, 1,~
## $ studytime  <int> 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 3, 1, 2, 3, 1, 3, 2, 1, 1,~
## $ failures   <int> 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0,~
## $ schoolsup  <chr> "yes", "no", "yes", "no", "no", "no", "no", "yes", "no", "n~
## $ famsup     <chr> "no", "yes", "no", "yes", "yes", "yes", "no", "yes", "yes",~
## $ paid       <chr> "no", "no", "yes", "yes", "yes", "yes", "no", "no", "yes", ~
## $ activities <chr> "no", "no", "no", "yes", "no", "yes", "no", "no", "no", "ye~
## $ nursery    <chr> "yes", "no", "yes", "yes", "yes", "yes", "yes", "yes", "yes~
## $ higher     <chr> "yes", "yes", "yes", "yes", "yes", "yes", "yes", "yes", "ye~
## $ internet   <chr> "no", "yes", "yes", "yes", "no", "yes", "yes", "no", "yes",~
## $ romantic   <chr> "no", "no", "no", "yes", "no", "no", "no", "no", "no", "no"~
## $ famrel     <int> 4, 5, 4, 3, 4, 5, 4, 4, 4, 5, 3, 5, 4, 5, 4, 4, 3, 5, 5, 3,~
## $ freetime   <int> 3, 3, 3, 2, 3, 4, 4, 1, 2, 5, 3, 2, 3, 4, 5, 4, 2, 3, 5, 1,~
## $ goout      <int> 4, 3, 2, 2, 2, 2, 4, 4, 2, 1, 3, 2, 3, 3, 2, 4, 3, 2, 5, 3,~
## $ Dalc       <int> 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1,~
## $ Walc       <int> 1, 1, 3, 1, 2, 2, 1, 1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 1, 4, 3,~
## $ health     <int> 3, 3, 3, 5, 5, 5, 3, 1, 1, 5, 2, 4, 5, 3, 3, 2, 2, 4, 5, 5,~
## $ absences   <int> 6, 4, 10, 2, 4, 10, 0, 6, 0, 0, 0, 4, 2, 2, 0, 4, 6, 4, 16,~
## $ G1         <int> 5, 5, 7, 15, 6, 15, 12, 6, 16, 14, 10, 10, 14, 10, 14, 14, ~
## $ G2         <int> 6, 5, 8, 14, 10, 15, 12, 5, 18, 15, 8, 12, 14, 10, 16, 14, ~
## $ G3         <int> 6, 6, 10, 15, 10, 15, 11, 6, 19, 15, 9, 12, 14, 11, 16, 14,~
```

```
# Now data is ready to be passed to the split_data function

datasets <- split_data(data)
train_set <- datasets$train_set
test_set <- datasets$test_set
```

Creating Ridge

```
library(glmnet)
```

```
## Lade nötiges Paket: Matrix
```

```
##
## Attache Paket: 'Matrix'
```

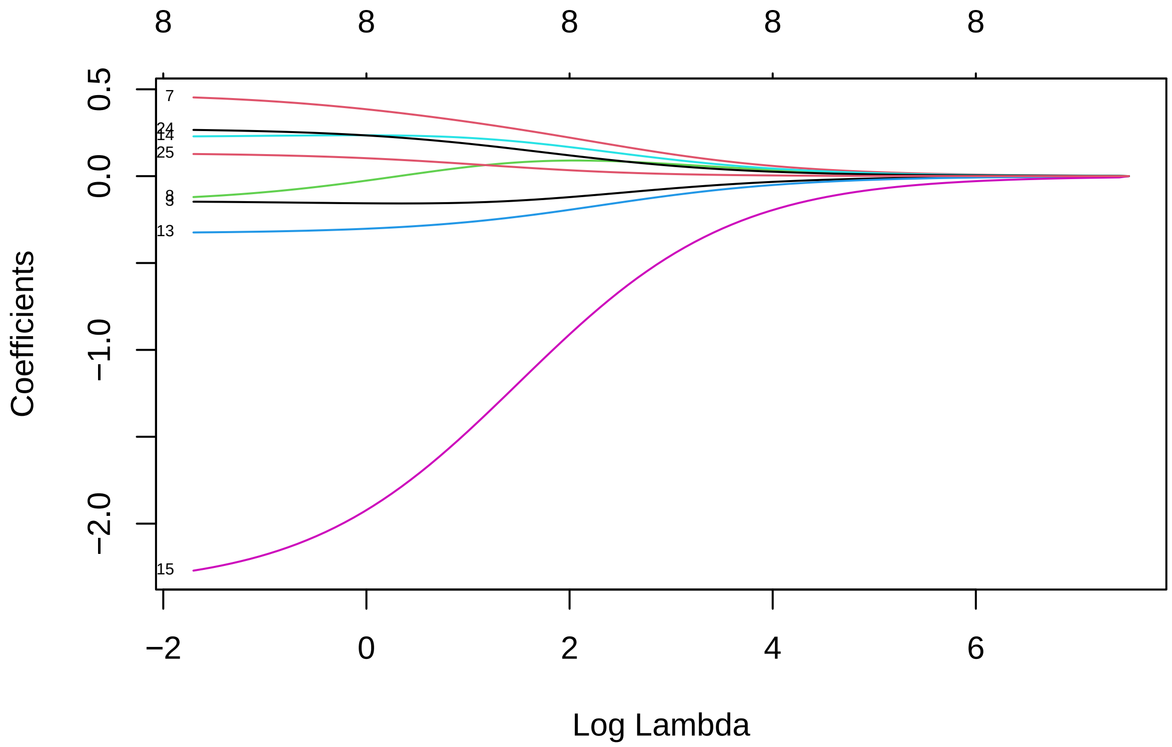
```
## Die folgenden Objekte sind maskiert von 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-8
```

```
ridge <- glmnet(as.matrix(train_set[,c(1:25)]), train_set$G3, alpha = 0)
```

```
## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt
```

```
plot(ridge, xvar = "lambda", label = TRUE)
```

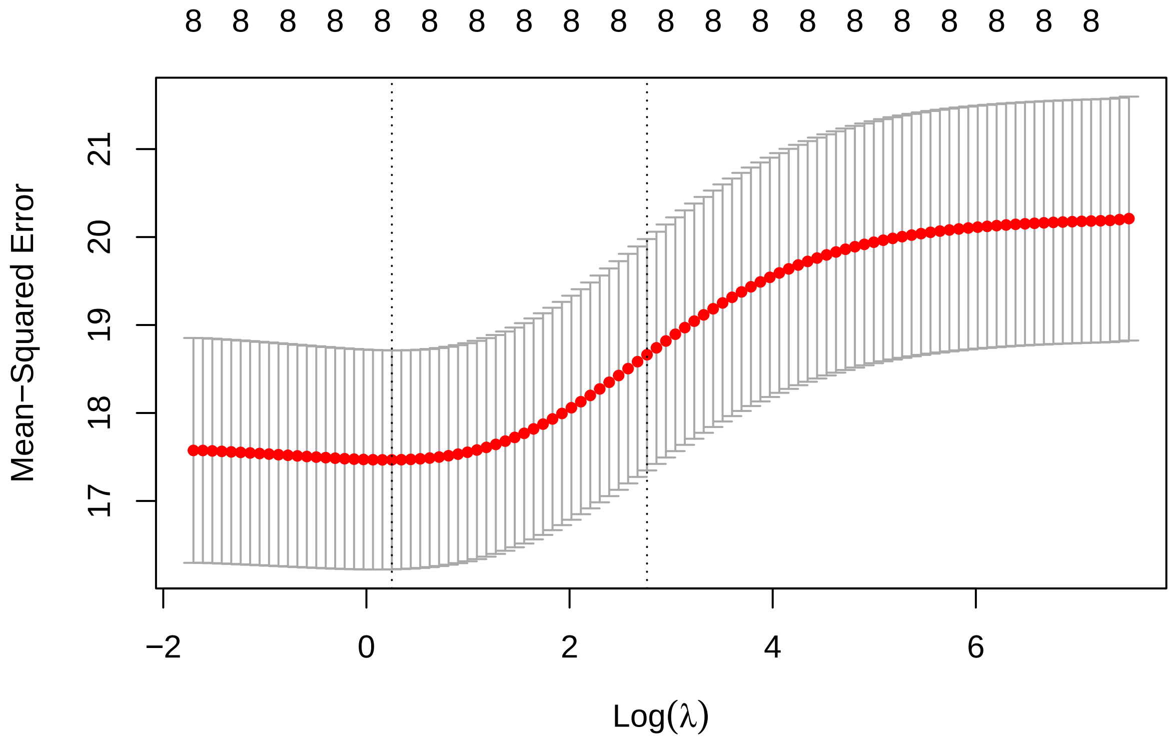


Ridge Plot looks good. Alle coefficients reach 0 at the same point. But when I proceed to check the Model with the optimal Lambda, a lot of coefficients are deleted from the Ridge-Model

```
ridge.cv <- cv.glmnet(as.matrix(train_set[,c(1:25)]), train_set$G3,
                      type.measure = "mse", nfolds = 5, alpha = 0)
```

```
## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in storage.mode(xd) <- "double": NAs durch Umwandlung erzeugt

## Warning in cbind2(1, newx) %*% nbeta: NAs durch Umwandlung erzeugt

## Warning in cbind2(1, newx) %*% nbeta: NAs durch Umwandlung erzeugt

## Warning in cbind2(1, newx) %*% nbeta: NAs durch Umwandlung erzeugt

## Warning in cbind2(1, newx) %*% nbeta: NAs durch Umwandlung erzeugt

## Warning in cbind2(1, newx) %*% nbeta: NAs durch Umwandlung erzeugt
```

```
plot(ridge.cv)
```



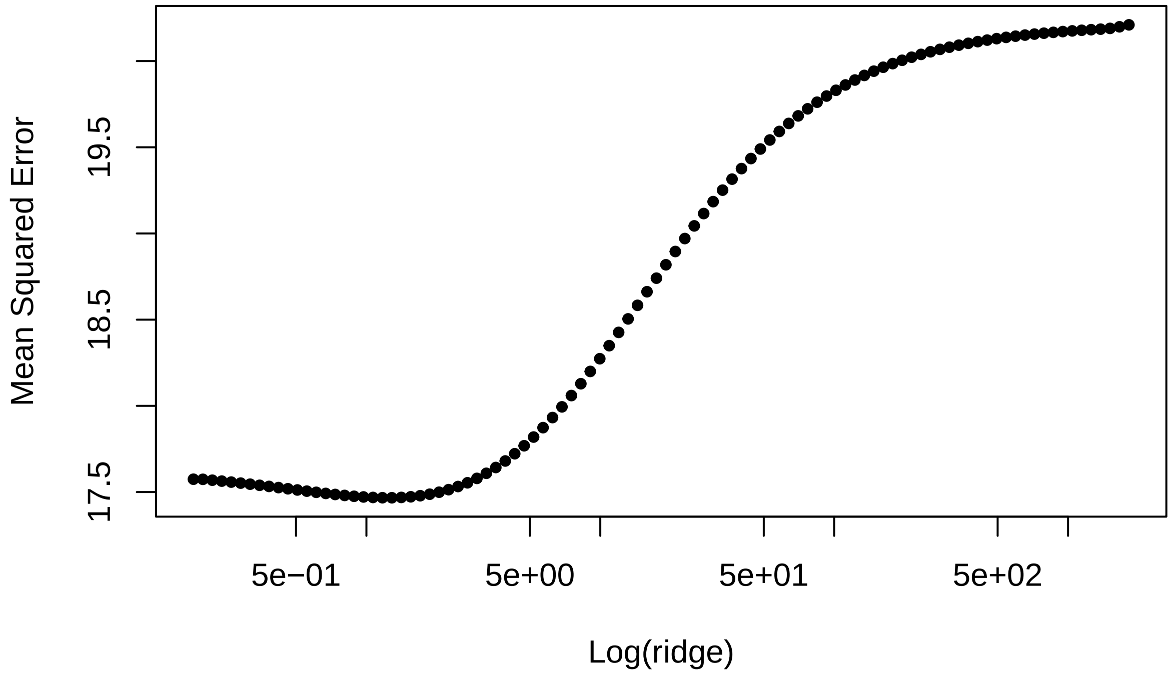
```
print(paste0("Optimal lambda that minimizes cross-validated MSE: ",
             ridge.cv$lambda.min))
```

```
## [1] "Optimal lambda that minimizes cross-validated MSE: 1.28476232029456"
```

```
print(paste0("Optimal lambda using one-standard-error-rule: ",
             ridge.cv$lambda.1se))
```

```
## [1] "Optimal lambda using one-standard-error-rule: 15.8391503753317"
```

```
plot(ridge.cv$lambda, ridge.cv$cvm, type = "n", log = "x", xlab = "Log(ridge)",
     ylab = "Mean Squared Error")
points(ridge.cv$lambda, ridge.cv$cvm, pch = 20)  # Add the points
```



```
# Print Ridge coefficients
print(coef(ridge.cv, s = "lambda.min"))
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 11.510338147
## school        .
## sex           .
## age         -0.157040623
## address       .
## famsize       .
## Pstatus       .
## Medu         0.368899824
## Fedu        -0.005361391
## Mjob          .
## Fjob          .
## reason        .
## guardian      .
## traveltime  -0.295648783
## studytime    0.234390812
## failures    -1.826843270
## schoolsup     .
## famsup        .
## paid          .
## activities    .
## nursery       .
## higher       .
## internet     .
## romantic     .
## famrel       0.225084897
## freetime     0.096245083
```