Penalized Regression Practice

Installing packages

```
library(MASS)
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(caTools)
```

Reading data

```
data("Boston", package = "MASS")
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Split training and test data

```
set.seed(123)
training_samples <- Boston$medv %>%
  createDataPartition(p=0.75,list=FALSE)
train.data <- Boston[training_samples,]
test.data <- Boston[-training_samples,]
nrow(train.data)
```

```
## [1] 381
```

```
nrow(test.data)
```

```
## [1] 125
```

Ridge regression model - finding best lambda

```
x <- model.matrix(medv~.,train.data)[,-1]
y <- train.data$medv
cv <- cv.glmnet(x,y,alpha=0)
cv$lambda.min
```

```
## [1] 0.6490823
```

Best lambda for ridge regression is 0.6490823

Ridge regression model - coefficient of the fitted model

```
model <- glmnet(x,y,alpha=0,lambda=cv$lambda.min)
coef(model)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  25.994735184
## crim         -0.066393301
## zn            0.018062258
## indus        -0.058721267
## chas          1.738033967
## nox         -11.213169927
## rm            4.092962823
## age          -0.003856820
## dis          -0.849835447
## rad           0.118962860
## tax          -0.006818129
## ptratio      -0.842092676
## black         0.007931751
## lstat        -0.385975629
```

Ridge regression model - prediction, RMSE and R^2

```
x.test <- model.matrix(medv~., test.data)[,-1]
predictions <- model %>% predict(x.test) %>% as.vector()
data.frame(
  RMSE = RMSE(predictions,test.data$medv),
  Rsquare = R2(predictions, test.data$medv)
)
```

```
##       RMSE   Rsquare
## 1 6.635525 0.6626213
```

Lasso regression model - finding best lambda

```
cv <- cv.glmnet(x,y,alpha=1)
cv$lambda.min
```

```
## [1] 0.02943509
```

Lasso regression model - coefficients of fitted model

```
model <- glmnet(x,y,alph=1,lambda=cv$lambda.min)
coef(model)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  30.737740248
## crim         -0.070051875
## zn            0.023056204
## indus        -0.013502700
## chas          1.444465893
## nox         -14.898532459
## rm            4.083374808
## age           .
## dis          -1.028888982
## rad           0.210742993
## tax          -0.011072116
## ptratio      -0.917788678
## black         0.007918102
## lstat        -0.424325830
```

Lasso regression model - prediction, RMSE and R^2

```
x.test <- model.matrix(medv~., test.data)[,-1]
predictions <- model %>% predict(x.test) %>% as.vector()
data.frame(
  RMSE=RMSE(predictions,test.data$medv),
  Rsquare=R2(predictions,test.data$medv)
)
```

```
##       RMSE   Rsquare
## 1 6.472275 0.6749717
```

Elastic net model - best tuning parameter

```
model <- train(medv~., data=train.data, method="glmnet", trControl = trainControl("cv",number=10),tuneL
model$bestTune
```

```
##    alpha      lambda
## 72   0.8 0.006927914
```

Elastic net model - coefficients of fitted model

```
coef(model$finalModel, model$bestTune$lambda)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  32.15987652
## crim         -0.07747083
## zn            0.02640789
## indus        -0.01141485
## chas          1.48080976
## nox         -15.72944505
## rm            4.03988386
## age           .
## dis          -1.09833826
## rad           0.24664034
## tax          -0.01260015
## ptratio      -0.93092906
## black         0.00811210
## lstat        -0.42490149
```

Elastic net model - prediction, RMSE and R^2

```
x.test <- model.matrix(medv~., test.data)[,-1]
predictions <- model %>% predict(x.test)
data.frame(
  RMSE = RMSE(predictions, test.data$medv),
  Rsquare = R2(predictions, test.data$medv)
)
```

```
##       RMSE   Rsquare
## 1 6.433159 0.6781032
```