project1\_regularization

EMVP

2023-03-14

#1. Please use the random seed 123 to divide the data into 75% training and 25% testing.   
  
#~Install Packages  
if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

if (!requireNamespace("glmnet")) install.packages('glmnet')

## Loading required namespace: glmnet

if (!requireNamespace("caTools")) install.packages('caTools')

## Loading required namespace: caTools

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

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 4.1-6

library(caTools)  
  
#~Read File  
data("Boston", package = "MASS")  
str(Boston) # the response variable is 'medv' and the other 13 are used to fit the model

## '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 ...

#~Set seed and split data  
set.seed(123)  
training.samples <- Boston$medv %>% createDataPartition(p = 0.75, list = FALSE)  
training <- Boston[training.samples, ]  
testing <- Boston[-training.samples, ]

#2. Please first find the best Ridge Regression model using the training data. Please (a) find the best λ value through cross-validation and display this value; (b) display the coefficients of the fitted model; and (c) make prediction on the testing data, and report the RMSE and the Coefficient of Determination R^2.   
x <- model.matrix(medv~., training)[,-1]  
y <- training$medv  
cv <- cv.glmnet(x, y, alpha = 0)  
cv$lambda.min

## [1] 0.6490823

#0.6490823 is the best lambda for ridge  
  
model <- glmnet(x, y, alpha = 0, lambda = cv$lambda.min) #alpha=0 for ridge  
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

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

## RMSE Rsquare  
## 1 6.635525 0.6626213

#6.635525 for RMSE and 0.6626213 for Rsquare (coef of determination)

#3 Please first find the best LASSO model using the training data. Please (a) find the best λ value through cross-validation and display this value; (b) display the coefficients of the fitted model; and (c) make prediction on the testing data, and report the RMSE and the Coefficient of Determination R^2.   
  
cv <- cv.glmnet(x, y, alpha = 1)  
cv$lambda.min

## [1] 0.02943509

#0.02943509 is the best lambda for LASSO  
  
model <- glmnet(x, y, alpha = 1, lambda = cv$lambda.min) # alpha=1: lasso  
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

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

## RMSE Rsquare  
## 1 6.472275 0.6749717

#6.472275 for RMSE and 0.6749717 for Rsquare (coef of determination)

#4. Please first find the best Elastic Net model using the training data. Please (a) find the best tuning parameters values through cross-validation and display these values; (b) display the coefficients of the fitted model; and (c) make prediction on the testing data, and report the RMSE and the Coefficient of Determination R^2.   
  
model <- train(  
 medv ~., data = training, method = "glmnet",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
)  
  
model$bestTune

## alpha lambda  
## 72 0.8 0.006927914

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

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

## RMSE Rsquare  
## 1 6.433159 0.6781032

#6.433159 for RMSE and 0.6781032 for Rsquare (coef of determination)