

LAB 14

ABDUL RAUF

2023-02-05

Contents

1	LASSO REGRESSION:	2
2	Load the Data	2
3	Fit the ridge regression model	2
4	Fit lasso regression model using k-fold cross-validation	3
5	Plot the test MSE's vs. lambda values	3
6	Coefficients of best model	3
7	Prediction for the response value of a new observation	4
8	Lasso trace plot	4
9	Find R-squared of model on training data	5

```
__NAME__:**ABDUL RAUF** \
__ENRL NO__:**GL6092** \
__ROLL NO__:**22DSMSA116** \
```

1 LASSO REGRESSION:

Lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model

To perform lasso regression, we'll use functions from the `glmnet()` package. This package requires the response variable to be a vector and the set of predictor variables to be of the class `data.matrix`.

use the `glmnet()` function to fit the ridge regression model and specify `alpha=1`.

Note that setting `alpha` equal to 0 is equivalent to using ridge regression and setting `alpha` to some value between 0 and 1 is equivalent to using an elastic net.

To determine what value to use for `lambda`, we'll perform k-fold cross-validation and identify the `lambda` value that produces the lowest test mean squared error (MSE). `cv.glmnet()` automatically performs k-fold cross validation using `k = 10` folds.

2 Load the Data

```
library(glmnet)
library(ISLR2)

#Define predictor and response variables
y <- mtcars$hp
x <- data.matrix(mtcars[, c('mpg', 'wt', 'drat', 'qsec')])
```

3 Fit the ridge regression model

```
model <- glmnet(x, y, alpha = 1)
summary(model)
```

```
##          Length Class      Mode
## a0         64    -none-   numeric
## beta       256  dgCMatrix S4
## df          64    -none-   numeric
## dim          2    -none-   numeric
## lambda      64    -none-   numeric
## dev.ratio   64    -none-   numeric
## nulldev      1    -none-   numeric
## npasses      1    -none-   numeric
## jerr         1    -none-   numeric
## offset       1    -none-   logical
## call         4    -none-    call
## nobs         1    -none-   numeric
```

4 Fit lasso regression model using k-fold cross-validation

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

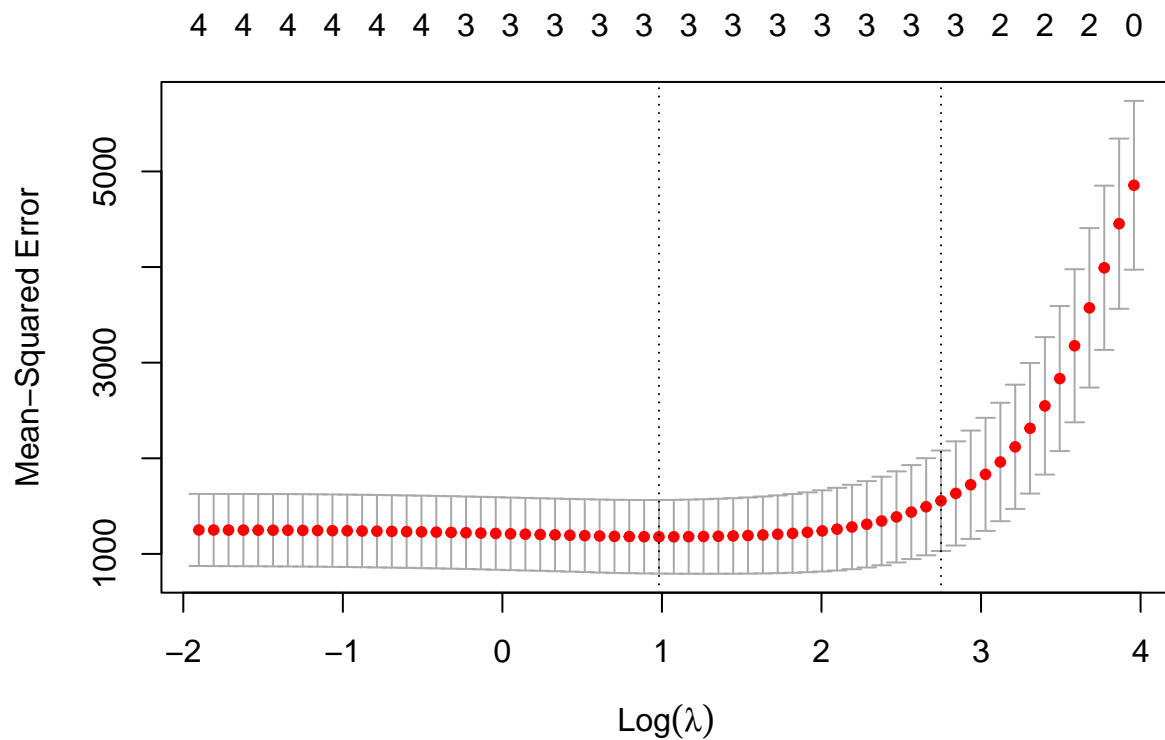
#display optimal lambda value
best_lambda
```

```
## [1] 2.668219
```

The lambda value that minimizes the test MSE turns out to be 2.431182.

5 Plot the test MSE's vs. lambda values

```
plot(cv_model)
```



6 Coefficients of best model

```
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                s0
## (Intercept) 484.20742
## mpg         -2.95796
## wt          21.37988
## drat         .
## qsec        -19.43425
```

the coefficient of the `drat` becomes exactly equal to 0 ,it get completely dropped the model.

7 Prediction for the response value of a new observation

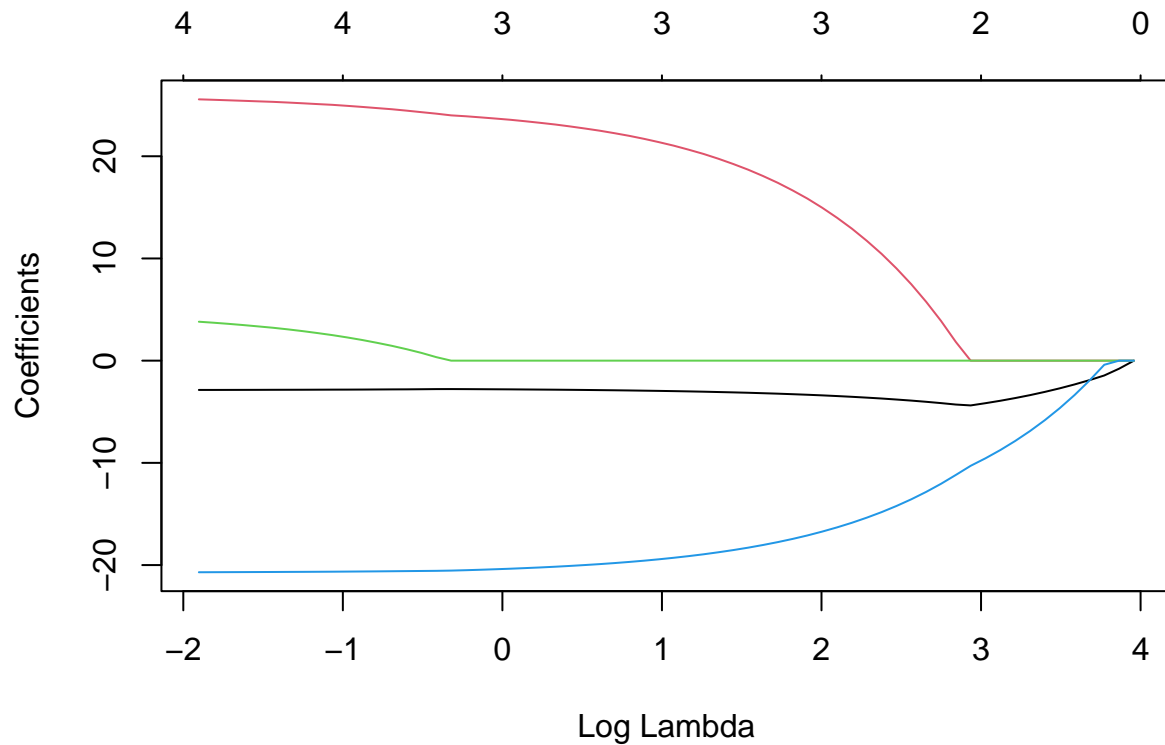
```
new = matrix(c(24, 2.5, 3.5, 18.5), nrow=1, ncol=4)
predict(best_model, s = best_lambda, newx = new)
```

```
##                s1
## [1,] 107.1325
```

based on the new input value as a test data , model predict that car have an hp value of 106.6893.

8 Lasso trace plot

```
plot(model,xvar = "lambda")
```



variables becomes exactly equal to 0 as lambda increases.

9 Find R-squared of model on training data

```
y_predicted <- predict(best_model, s = best_lambda, newx = x)

sst <- sum((y - mean(y))^2)
sse <- sum((y_predicted - y)^2)

rsq <- 1 - sse/sst
rsq
```

```
## [1] 0.803853
```

R-squared turns out to be 0.8047064 which means 80.47% of the variation in the response is explained by model.