

# ADM Assignment-1

Rachana Kurra

2023-03-03

## PART-A

QA1. What is the main purpose of regularization when training predictive models?

Generally, Regularization methods are used in machine learning to avoid underfitting or overfitting for unseen data. Regularization strives to optimize the training set's performance to prevent underfitting, but it also penalizes the model when it gets complicated which is known as overfitting and generalizes well i.e., it improves a model's performance by simplifying it.

Techniques: Lasso and Ridge Regressions were used for regularization.

QA2. What is the role of a loss function in a predictive model? And name two common loss functions for regression models and two common loss functions for classification models

The role of the loss function can define how good a prediction is by measuring the penalty. It generally commutes the distance between the current output and the expected output of the algorithm. The process of reducing the loss is achieved by changing the parameters until the lowest loss is encountered. Common loss functions for regression models are: MSE (mean squared error) and MAE(mean absolute error) And for classification models: Binary cross-entropy and categorical cross-entropy loss.

QA3. Consider the following scenario. You are building a classification model with many hyper parameters on a relatively small dataset. You will see that the training error is extremely small. Can you fully trust this model? Discuss the reason.

If the errors in the training set are extremely small, it means that the model is overfitting and it has all the noise data and outliers. Considering the given example, the model with many hyperparameters is built using a small data set. So, there is a high chance of overfitting. When overfitting occurs, the model may achieve good training accuracy, but it will perform poorly on unseen data. Hence, to address this issue, we need to evaluate the model on a validation set or using cross-validation techniques, which will provide an estimate of the model's generalization performance. If the model performs well on the validation set and the variance is less between train error and validation error then it might be trustworthy or reliable.

QA4. What is the role of the lambda parameter in regularized linear models such as Lasso or Ridge regression models?

The lambda parameter is also known as the regularization parameter/ hyperparameter for regularized linear models which prevents overfitting by adding a penalty to the cost function that the model is trying to minimize. The tuning parameter lambda is chosen by cross validation. Increasing or decreasing lambda provides a tradeoff between underfitting and overfitting. In lasso, the penalty has the effect of forcing coefficients to zero, the penalty is the sum of the absolute values of the coefficients In ridge, when the lambda increases, the penalty grows the coefficients will get close to zero but non-zero coefficients. The shrinkage penalty is lambda times the sum of squares of the coefficients so coefficients that get too large are penalized.

## PART-B

```

#loading the libraries
library(ISLR)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-6

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

summary(Carseats)

##      Sales          CompPrice         Income        Advertising
## Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000
## 1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000
## Median : 7.490   Median :125   Median : 69.00   Median : 5.000
## Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635
## 3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000
## Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000
##      Population          Price       ShelveLoc          Age        Education
## Min.   : 10.0   Min.   : 24.0   Bad   : 96   Min.   :25.00   Min.   :10.0
## 1st Qu.:139.0   1st Qu.:100.0   Good  : 85   1st Qu.:39.75   1st Qu.:12.0
## Median :272.0   Median :117.0   Medium:219   Median :54.50   Median :14.0
## Mean   :264.8   Mean   :115.8   Medium:219   Mean   :53.32   Mean   :13.9
## 3rd Qu.:398.5   3rd Qu.:131.0   Medium:219   3rd Qu.:66.00   3rd Qu.:16.0
## Max.   :509.0   Max.   :191.0   Medium:219   Max.   :80.00   Max.   :18.0
##      Urban          US
## No    :118   No   :142
## Yes   :282   Yes  :258
##
##
```

```

#Considering the input attributes and scaling them and also converting them to matrix formart

Carseats_Filtered = Carseats %>% select ( "Price",
"Advertising","Population","Age","Income","Education")%>% scale(center = TRUE, scale = TRUE) %>% as.matrix()

#storing input attributes and response variable into X,Y in matrix formats
X = (Carseats_Filtered)
Y = Carseats %>% select("Sales") %>% as.matrix()

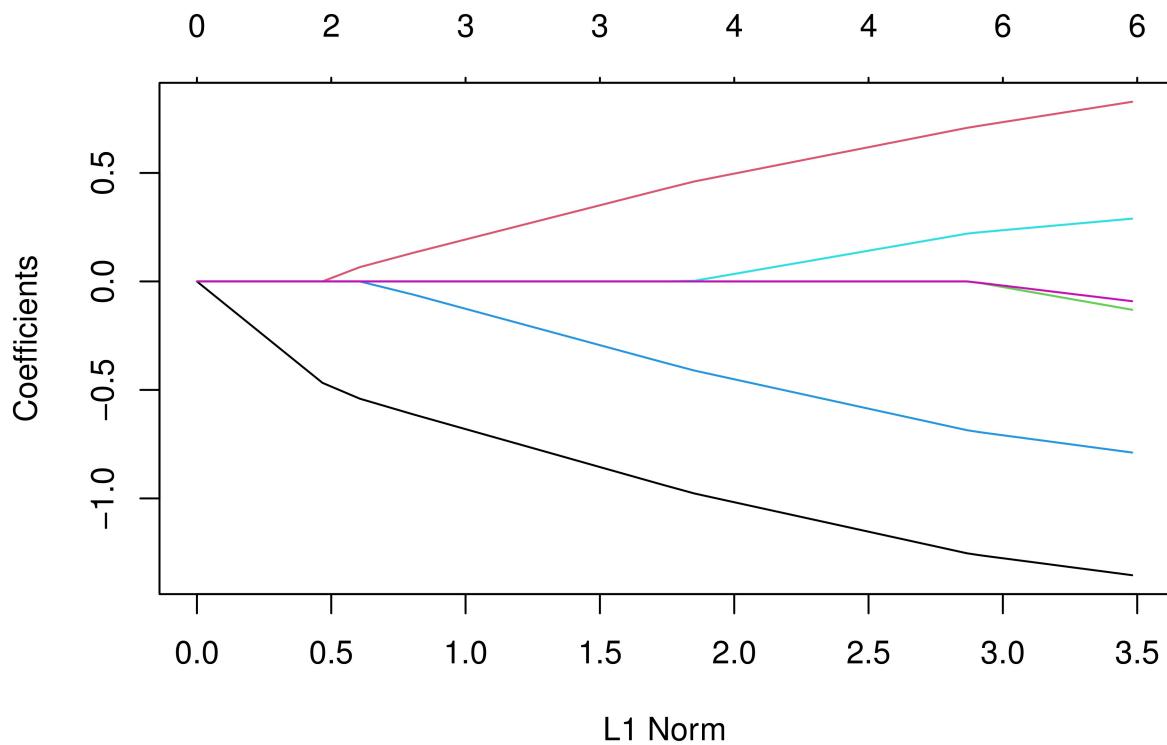
# Question-1: apply lasso regression model on the data to predict sales based on all other attributes.

Model = glmnet(X, Y)
summary(Model)

##          Length Class      Mode
## a0          62   -none-   numeric
## beta        372   dgCMatrix S4
## df           62   -none-   numeric
## dim          2    -none-   numeric
## lambda       62   -none-   numeric
## dev.ratio    62   -none-   numeric
## nulldev      1    -none-   numeric
## npasses      1    -none-   numeric
## jerr          1    -none-   numeric
## offset        1    -none-   logical
## call          3    -none-   call
## nobs          1    -none-   numeric

plot(Model)

```

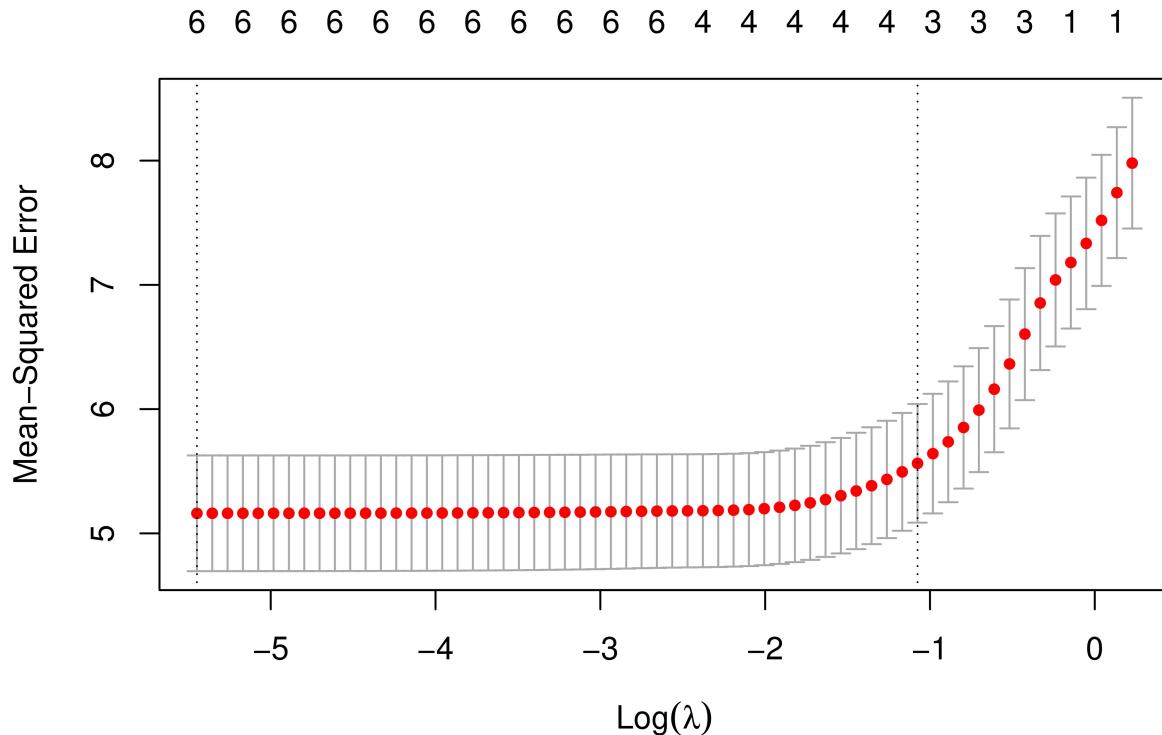


```
#Performing k-fold cross-validation to find optimal lambda value
cv_Model <- cv.glmnet(X, Y, alpha = 1)
```

```
#finding optimal lambda value that minimizes test MSE
best_lambda <- cv_Model$lambda.min
best_lambda
```

```
## [1] 0.004305309
```

```
#plot of test MSE by lambda value
plot(cv_Model)
```



From the above results, we can say that increase of variance in the dependent variable (sales) is explained by regularization for the given attributes and we got the best lambda as 0.004305

```
#question-2: coefficient for the price attribute in the best model
cv_Model$lambda.1se
```

```
## [1] 0.3411883

coef(cv_Model, s= "lambda.min")

## 7 x 1 sparse Matrix of class "dgCMatrix"
##           s1
## (Intercept) 7.49632500
## Price      -1.35383399
## Advertising 0.82805813
## Population -0.13061347
## Age        -0.78854992
## Income     0.28931898
## Education -0.09102484
```

The coefficient of price attribute is -1.35 for the best value of lambda

```
#QUESTION-3: No. of attributes remaining in the model if lambda is set to 0.01, and what is lambda if it is
# when lambda sets to 0.01

coef(cv_Model, s=0.01)
```

```

## 7 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept) 7.49632500
## Price       -1.34732839
## Advertising 0.82025640
## Population  -0.12187511
## Age         -0.78190585
## Income      0.28488681
## Education   -0.08502705

```

*#Lambda when set to 0.1.*

```
coef(cv_Model, s=0.1)
```

```

## 7 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept) 7.4963250
## Price       -1.2447750
## Advertising 0.7007231
## Population .
## Age         -0.6775428
## Income      0.2139222
## Education   .

```

*#lambda when set to 0.2*

```
coef(cv_Model, s=0.2)
```

```

## 7 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept) 7.4963250
## Price       -1.1338813
## Advertising 0.6013226
## Population .
## Age         -0.5669483
## Income      0.1264650
## Education   .

```

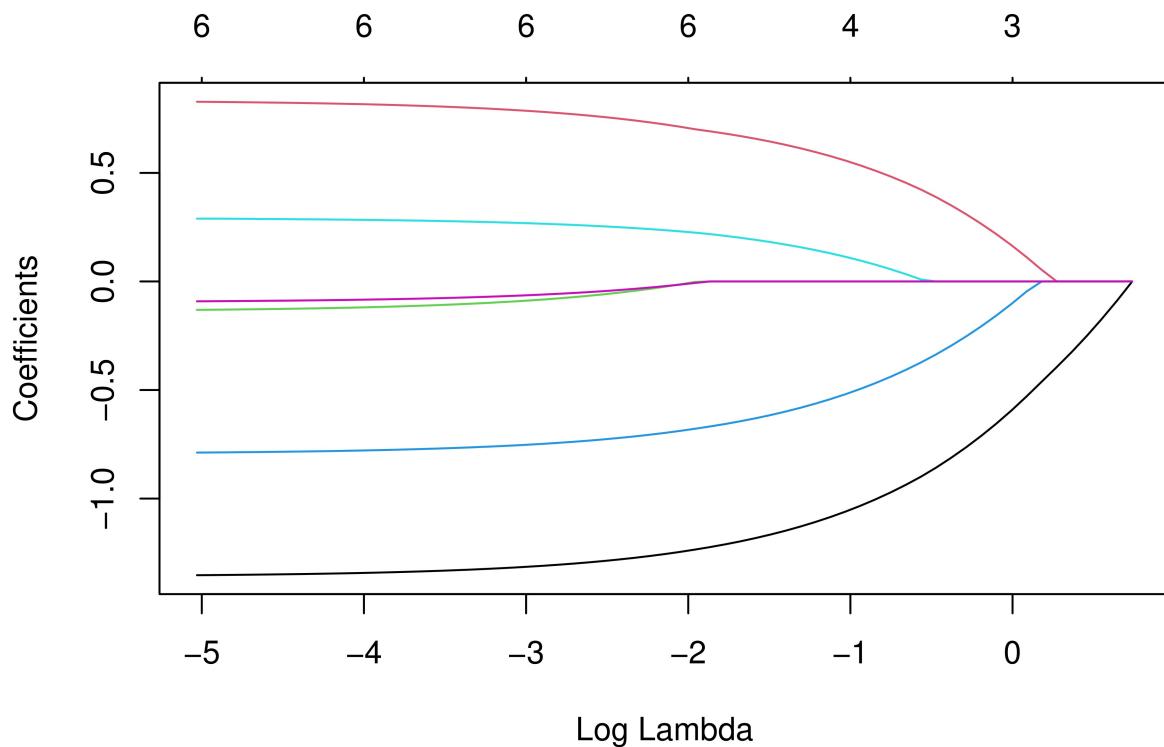
*From the results, we can observe that change in lambda value to 0.01 doesn't make any difference with the coefficients but the 2 coefficients are forced to 0 if increase the lambda value to 0.1 and 0.2. Hence, we can say that increase in lambda value will shrink the coefficients to zero.*

*#question-4: building elastic-net model with alpha set to 0.6*

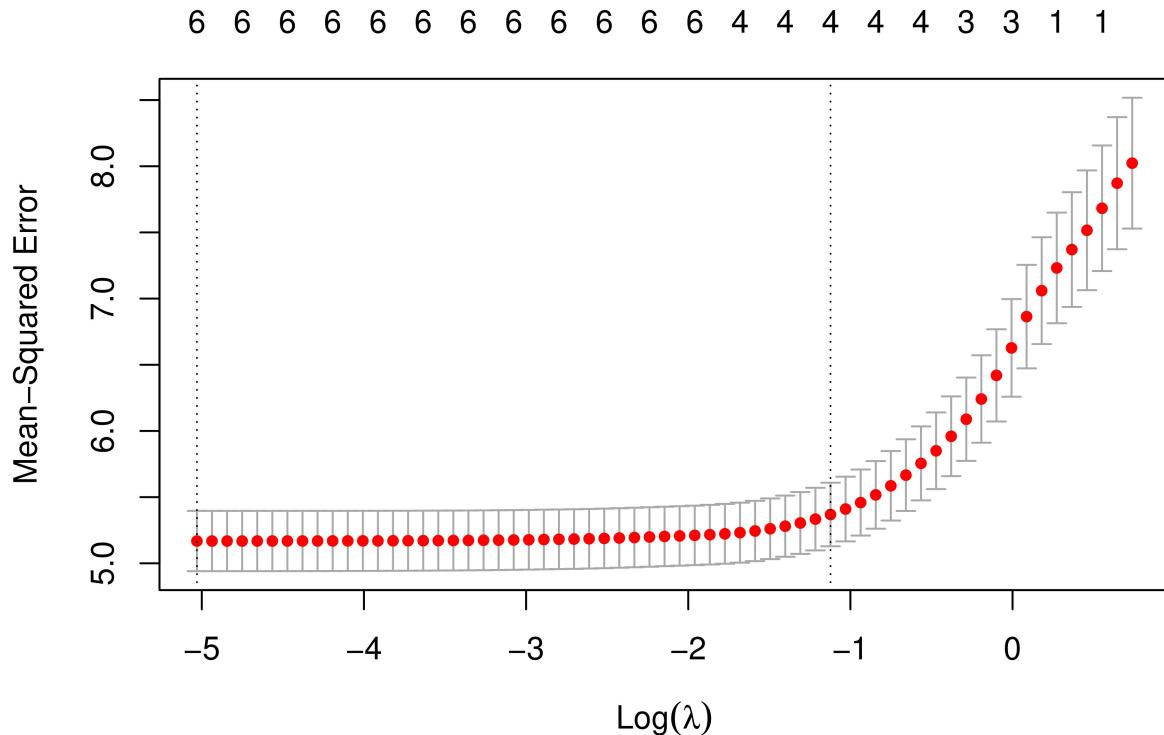
```

fit.elnet= glmnet(X,Y, alpha=0.6)
plot(fit.elnet, xvar="lambda")

```



```
plot(cv.glmnet(X,Y, alpha=0.6))
```



```

fit.elnet <- cv.glmnet(X, Y, alpha = 0.6)

#optimal lambda value
bestlambda <- fit.elnet$lambda.min
bestlambda

## [1] 0.0199663

print(fit.elnet)

##
## Call: cv.glmnet(x = X, y = Y, alpha = 0.6)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.0200     51   5.184 0.5527       6
## 1se 0.5686     15   5.712 0.5919       4

summary(fit.elnet)

##          Length Class  Mode
## lambda     63    -none- numeric
## cvm        63    -none- numeric

```

```
## cvsd      63    -none- numeric
## cvup      63    -none- numeric
## cvlo      63    -none- numeric
## nzero     63    -none- numeric
## call       4     -none- call
## name       1     -none- character
## glmnet.fit 12    elnet  list
## lambda.min  1    -none- numeric
## lambda.1se   1    -none- numeric
## index      2    -none- numeric
```

*Best lambda value for elastic-net model is 0.00654*