64037 - Assignment 1

Manaswini

2022-10-31

##1: What is the main purpose of regularization when training predictive models?

The main purpose of regularization when training predictive models is that it penalizes the model for being too complex and will regularize/shrink the coefficients towards zero. This is done to prevent over fitting. It is a set of techniques to limit the models capabilities to learn.

##2: 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 main role of a loss function in a predictive model is that it measures the penalty. The loss function computes distance between the output of the algorithm and the expected output.Two common loss functions for regression models are Sum Square of Error (SSE) and Mean absolute error (MAE). The two common loss functions for classification models are binary-cross-entropy and negative log likelihood.

##3 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.

When building a model with many hyper parameters on a small data set, and the training error is very small, you can not fully trust the model. This is because there is not enough data in the data set. With a small data set, the training accuracy will increase as it will try to fit to every data point, which means the training error will be very small. Model can not be trusted because it may be overfit.

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

The roll of the lambda parameter in regularized linear models like lasso and ridge is that lambda balances between minimizing the sum square of the residuals. A higher lambda means that it gives heavier regularization.

#PART B

```
#Installing & loading the packages
library(ISLR)
```

```
#install.packages("dplyr")
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

#install.packages("glmnet")
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-4

#install.packages("caret")
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

#Selecting Sales, Price, Advertising, Population, age and income
Carseats_Filtered <- Carseats %>% select("Sales", "Price",
"Advertising","Population","Age","Income","Education")
Carseats_withoutsales <- Carseats %>% select( "Price",
"Advertising","Population","Age","Income","Education")
sales <- Carseats_Filtered <- Carseats %>% select("Sales")
preProcess(sales)

## Created from 400 samples and 1 variables
##
## Pre-processing:
##   - centered (1)
##   - ignored (0)
##   - scaled (1)

Y <- as.matrix(sales)
```

#QB1 Build a Lasso regression model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education"). What is the best value of lambda for such a lasso model?

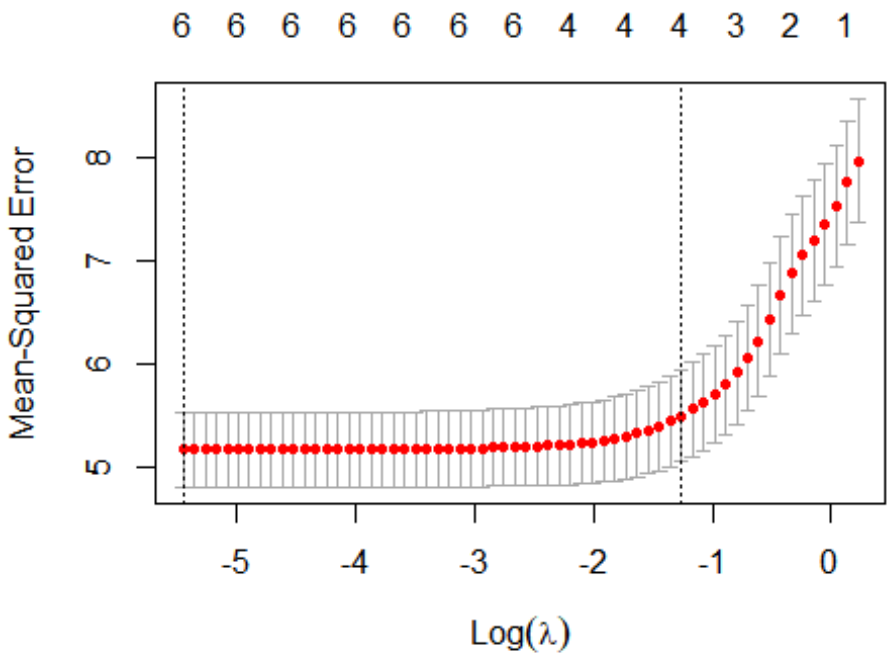
```
#Scaling the data
preProcess(Carseats_withoutsales)
```

```
## Created from 400 samples and 6 variables
##
## Pre-processing:
##   - centered (6)
##   - ignored (0)
##   - scaled (6)
```

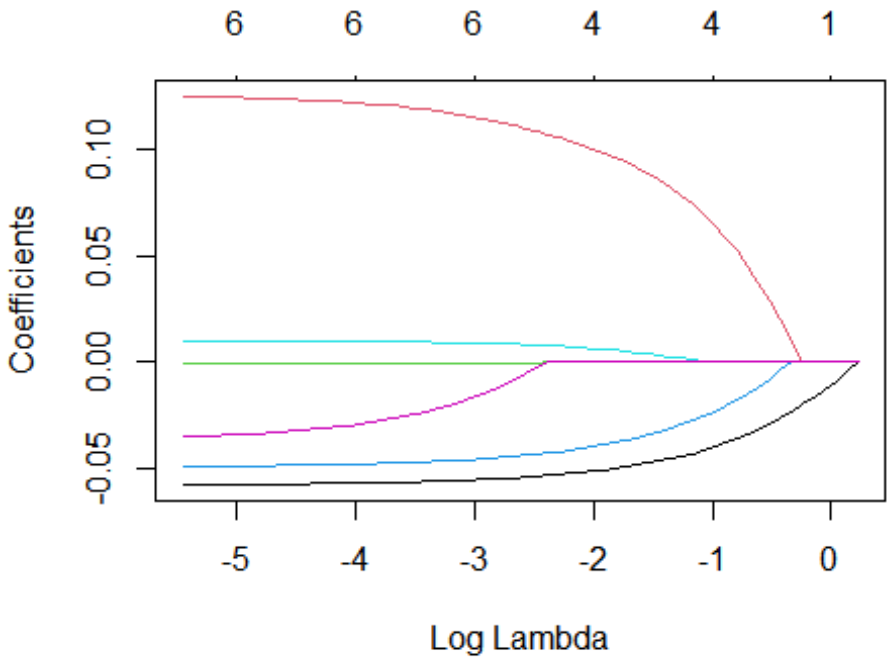
```
X <- data.matrix(Carseats_withoutsales[,c( "Price",
"Advertising","Population","Age","Income","Education")])
#perform k-fold validation to find optimal lambda
cv_model <- cv.glmnet(X,Y, alpha=1)
#Find optiomal lambda value that minimizes test
best_lambda <- cv_model$lambda.min
best_lambda
```

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

```
plot(cv_model)
```




```
fit.lasso <- glmnet(X,Y,alpha=1)
plot(fit.lasso, xvar="lambda")
```




```
##The best lambda value for such lasso model is .004305309
```

#What is the coefficient for the price (normalized) attribute in the best model (i.e. model with the optimal lambda)?

```
#Finding coefficients of the best model
best_model <- glmnet(X,Y,alpha = 1, lambda = best_lambda)
coef(best_model)

## 7 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 15.8946432052
## Price       -0.0571806037
## Advertising  0.1245169280
## Population  -0.0008863179
## Age         -0.0486751305
## Income       0.0103378849
## Education   -0.0347353427

##The best coefficient for the price normalized attribute in the best model w
ith the optimal lambda is -0.05718.
```

#QB3 How many attributes remain in the model if lambda is set to 0.01? How that number changes if lambda is increased to 0.1? Do you expect more variables to stay in the model (i.e., to have non-zero coefficients) as we increase lambda?

```
#Setting the lambda to 0.01
change <- glmnet(X,Y,alpha = 1, lambda = .01)
coef(change)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) 15.8120560645
## Price       -0.0569054918
## Advertising  0.1233407471
## Population  -0.0008269765
## Age         -0.0482649383
## Income       0.0101795876
## Education   -0.0324465388
```

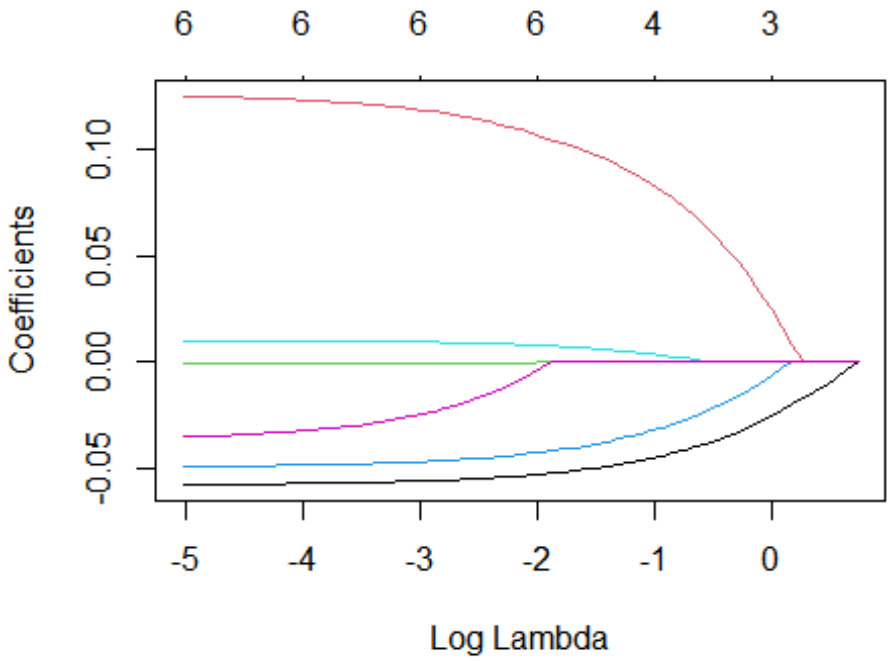
```
#Setting the lambda to 0.1
change2 <- glmnet(X,Y,alpha = 1, lambda = .1)
coef(change2)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) 14.59030452
## Price       -0.05257390
## Advertising  0.10536612
## Population   .
## Age         -0.04182286
## Income       0.00764389
## Education    .
```

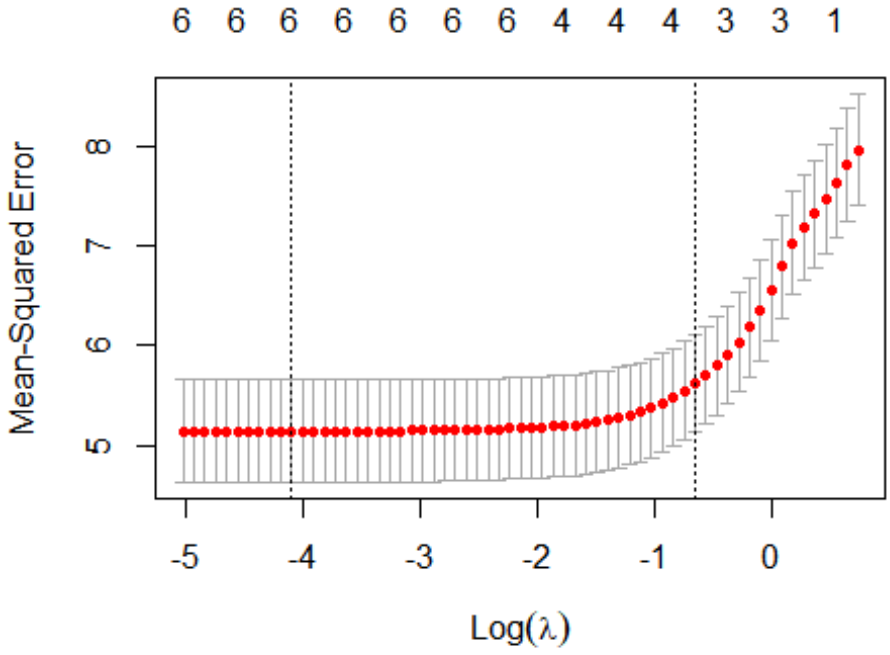
```
##If the lambda is set to .01, all the attributes remain which are price, adv
ertising, population, age, income, and education. When changing the lambda to
.1, the only attributes that remain are price, advertising, age, and income.
Increasing the lambda to .1 made less variables stay in the model.
```

#QB4 Build an elastic-net model with alpha set to 0.6. What is the best value of lambda for such a model?

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



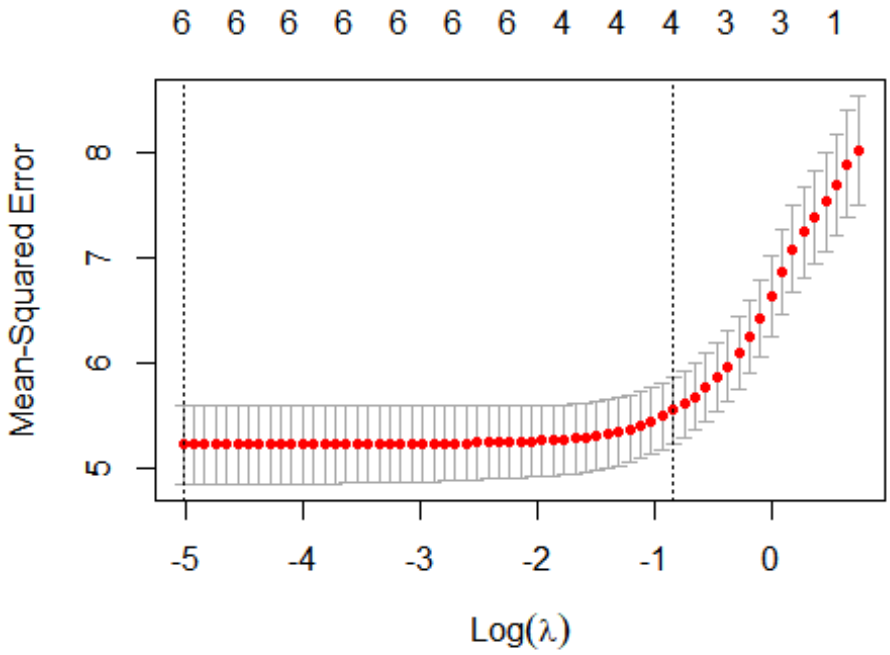

```
plot(cv.glmnet(X,Y,alpha=0.6))
```



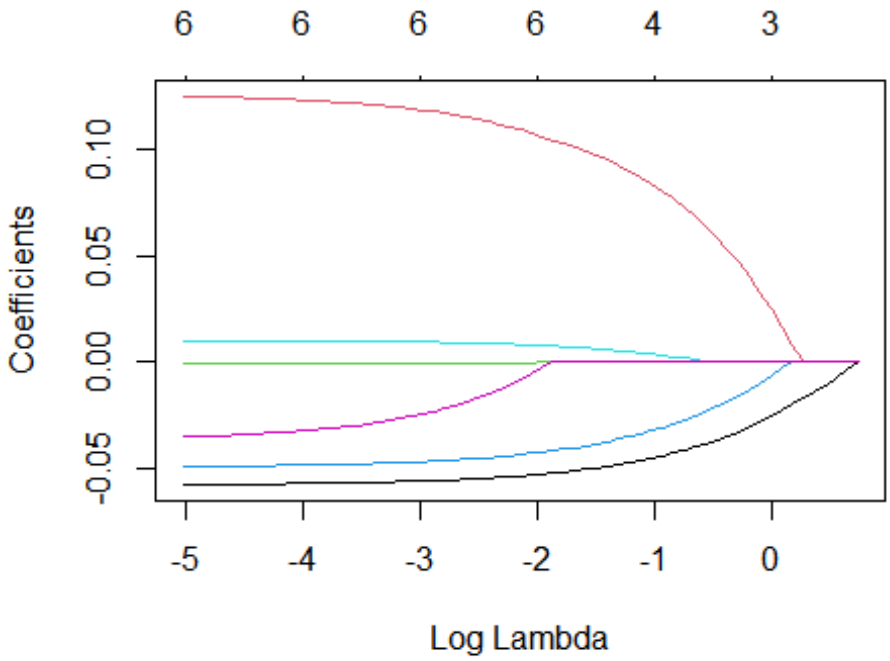
```r
#Performing k-fold validation to find optimal lambda
elastic <- cv.glmnet(X,Y, alpha=0.6)
#Finding optional lambda value that minimizes test
bestlambda <- elastic$lambda.min
bestlambda
```

```
## [1] 0.006538062
```

```r
plot(elastic)
```




```r
fit.elastic <- glmnet(X,Y,alpha=0.6)
plot(fit.elastic, xvar="lambda")
```




```
##The best lambda value for such elastic-net model is .00653
```