Assignment 1

Manasa Chelukala

10/22/2022

PART-A

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

Ans. In general, regularization refers to making something acceptable or regular.This is why we use it in machine learning applications.Regularization in machine learning involves reducing the coefficients towards zero or regularizing them.In other words, regularization inhibits overfitting by preventing an elaborate or flexible model from being learned.

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.

Ans. An analysis of the model's performance across the training dataset is based on a Loss function.Loss functions express the difference between model predictions and actual problem situations.

The Two common loss functions for regression models are Mean Absolute Error & Mean Squared Errors.

The Two common loss functions for classification models are Log Loss & Hinge loss.

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.

Ans.Since there are so many hyper parameters in such a small data set, we cannot fully trust the above model.Therefore, although the training error may be small, this model isn't optimal when used with test data.

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

Ans. Lambda is a hyperparameter that determines the penalty level in regularized linear models.

PART-B

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.3
```

```r
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
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.1.3
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

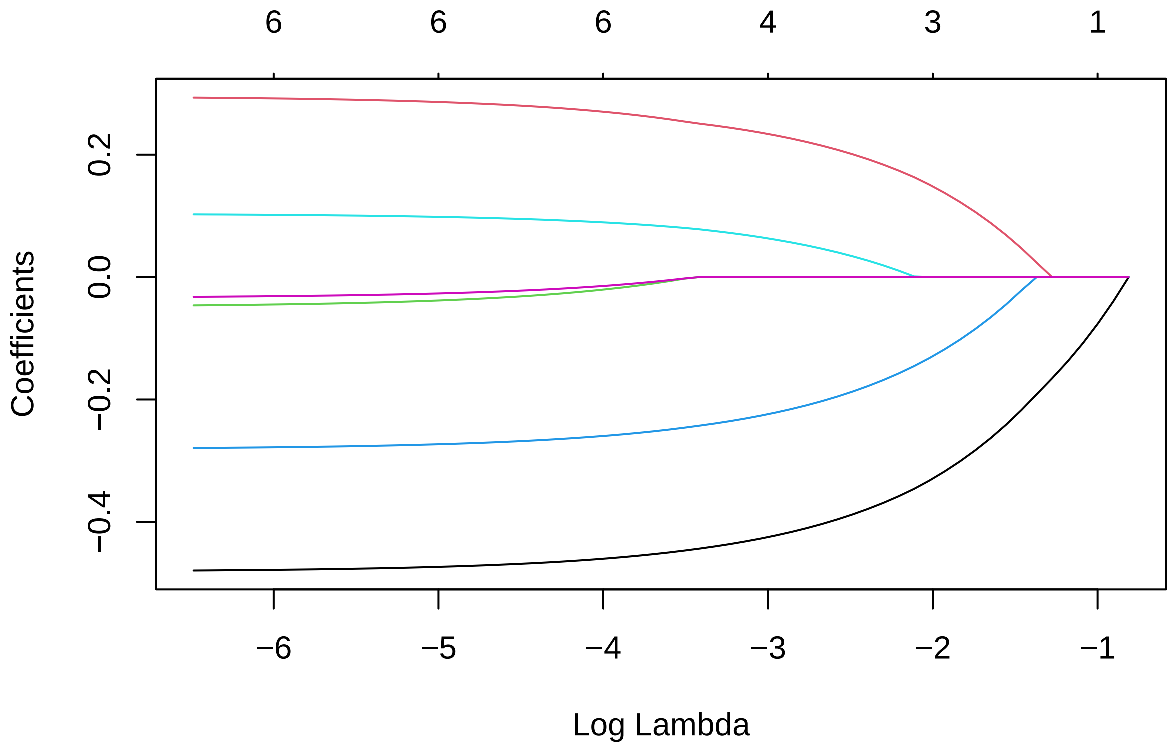
```r
library(caret)
```

```
## Loading required package: ggplot2
```

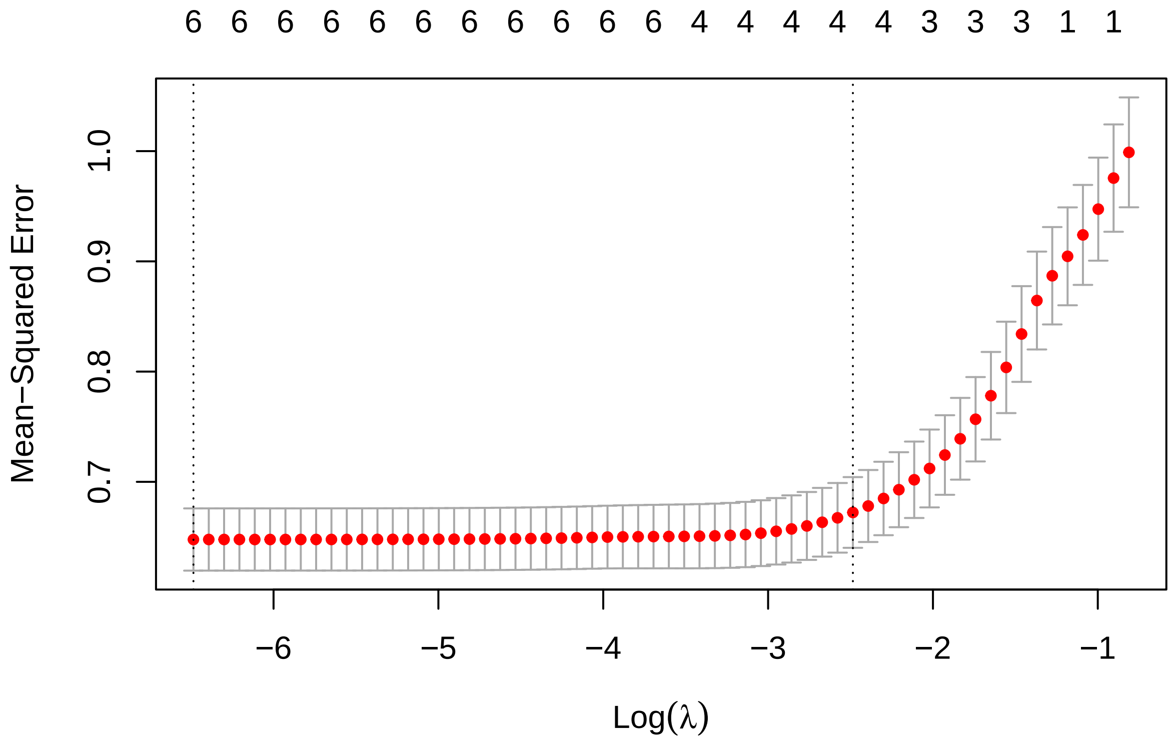
```
## Loading required package: lattice
```

```r
Carseats_Filtered <- Carseats %>% select("Sales", "Price", "Advertising", "Population", "Age", "Income"
```

```r
Normalized_Carseats_Filtered <- scale(Carseats_Filtered)
X <- as.matrix(Normalized_Carseats_Filtered[, c('Price','Advertising','Population','Age','Income','Educa
Y <- Normalized_Carseats_Filtered[,'Sales']
fit.lasso <- glmnet(X,Y,alpha = 1)
plot(fit.lasso,xvar = "lambda")
```

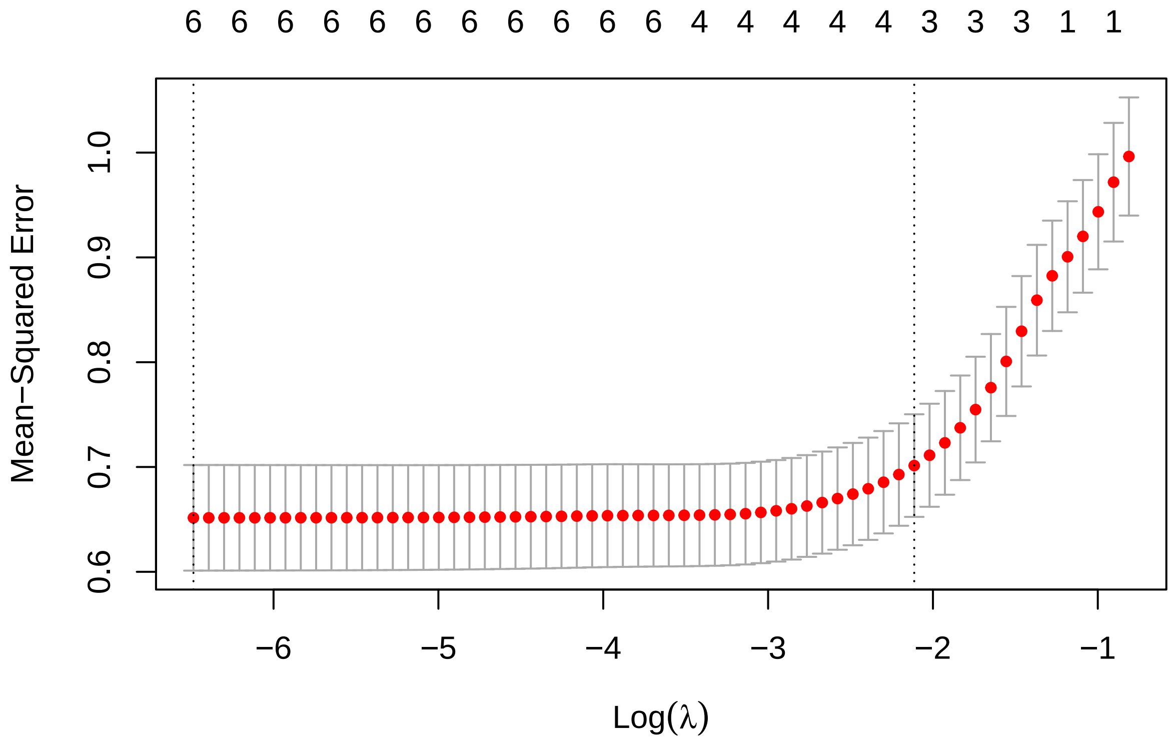


```
plot(cv.glmnet(X,Y,alpha = 1))
```



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?

```
CV_Fit <- cv.glmnet(X,Y, alpha = 1)
plot(CV_Fit)
```



```
lambda <- CV_Fit$lambda.min
lambda
```

```
## [1] 0.001524481
```

The best value of lambda for our lasso model is 0.0015.

QB2. What is the coefficient for the price (normalized) attribute in the best model (i.e. model with the optimal lambda)?

```
coef(fit.lasso, s = lambda)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  9.866665e-17
## Price       -4.793834e-01
## Advertising  2.932098e-01
## Population  -4.624934e-02
## Age         -2.792202e-01
## Income       1.024459e-01
## Education   -3.223128e-02
```

The coefficient for the “Price” (normalized) is -4.79x10-1.

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?

```
coef(fit.lasso, s = 0.01)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  9.798009e-17
## Price       -4.696889e-01
## Advertising  2.815718e-01
## Population  -3.323443e-02
## Age         -2.693300e-01
## Income       9.585212e-02
## Education   -2.330455e-02
```

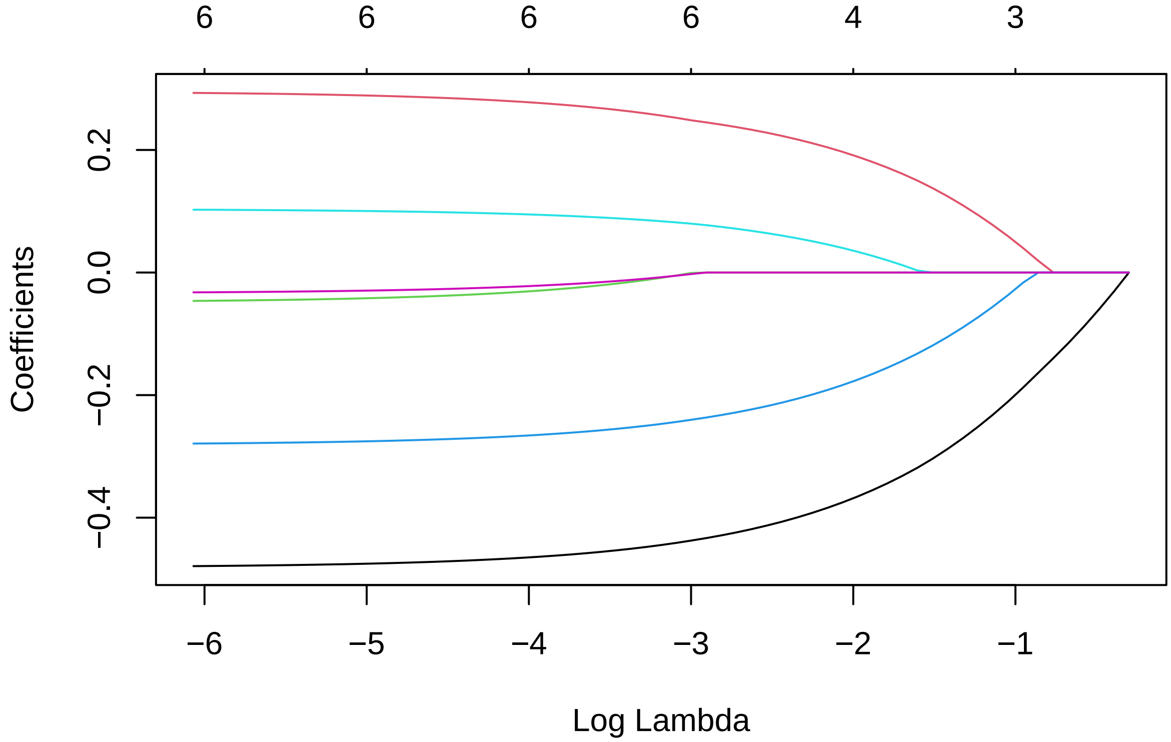
```
coef(fit.lasso, s = 0.1)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  9.803050e-17
## Price       -3.691394e-01
## Advertising  1.839178e-01
## Population   .
## Age         -1.684796e-01
## Income       1.925921e-02
## Education    .
```

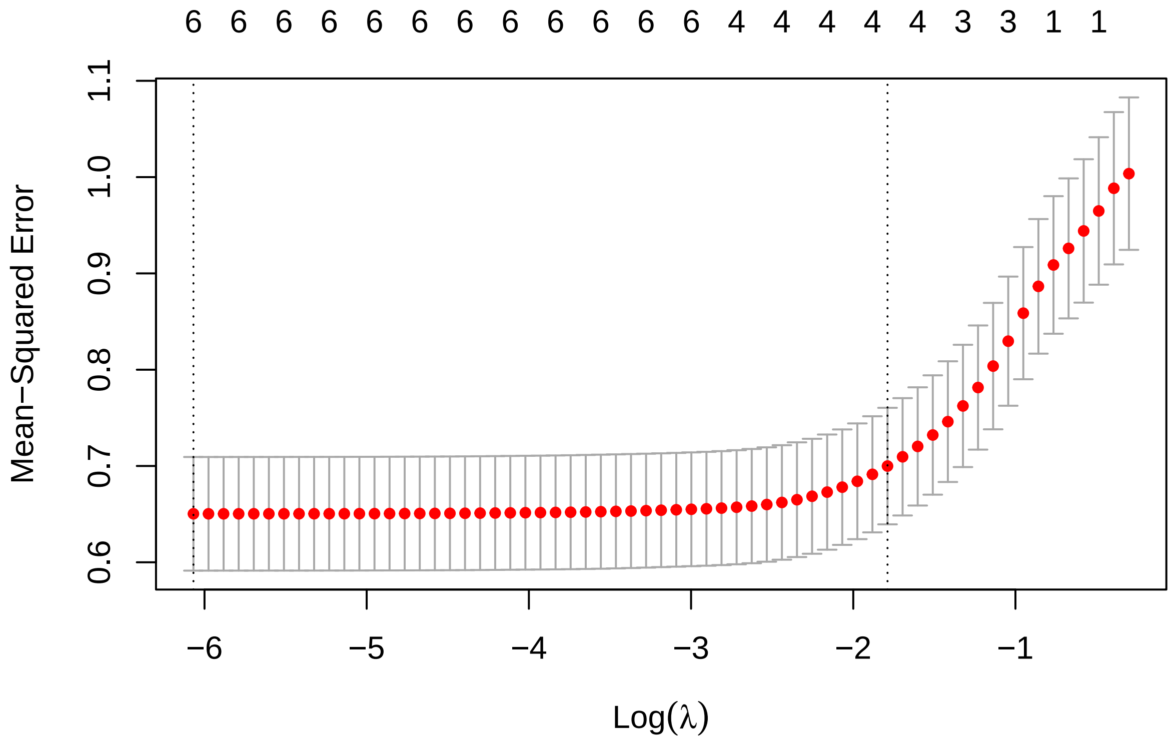
When the Lambda value is adjusted from 0.0015 to 0.01, we do not lose any qualities.However, we lose the properties of Population and Education attribute values when the Lambda value is adjusted from 0.01 to 0.1.We may anticipate to lose more properties as the lambda value increases.

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

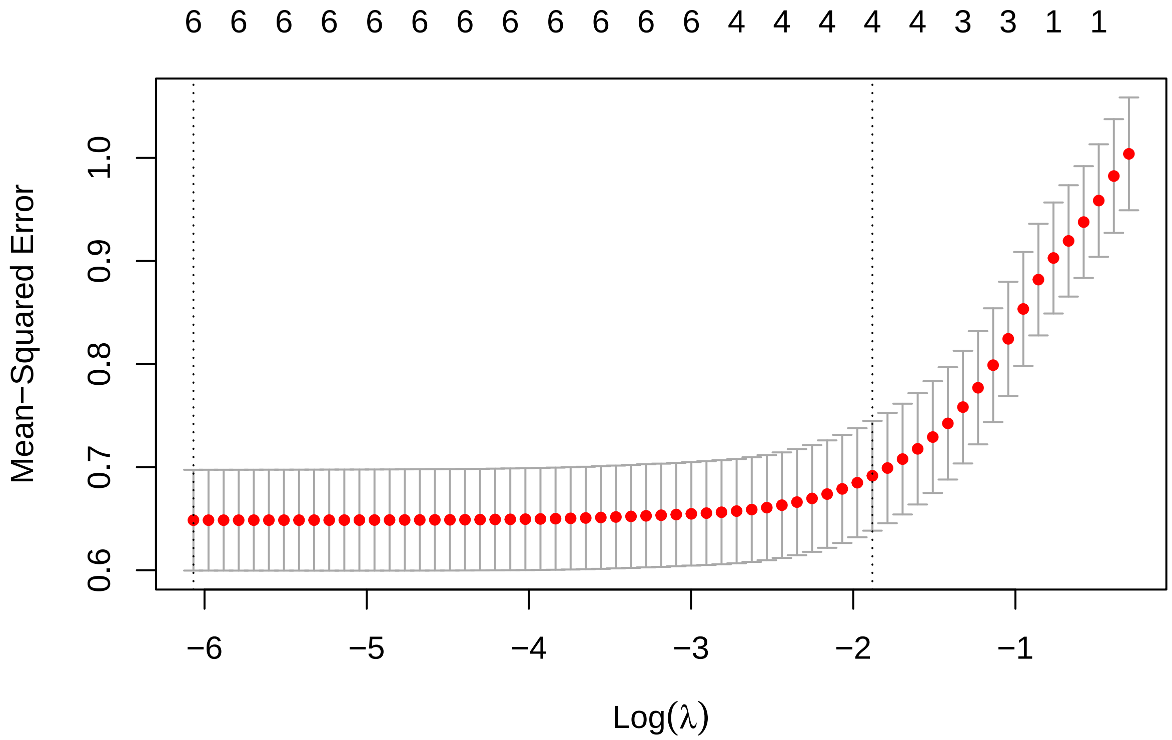
```
fit.elastic <- glmnet(X,Y,alpha = 0.6)
plot(fit.elastic,xvar = "lambda")
```



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



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



```
elastic <- CV_Fit_elastic$lambda.min
elastic
```

```
## [1] 0.002315083
```

The best value of Lambda for an elastic model with alpha set to 0.6 is 0.0023.