Assignment1_Vamsi_Nalluri

QA1. What is the main purpose of regularization when training predictive models? 10 points

Models accuracy may be different from training to test due to variety of reasons. For example, when the model learns finer details or noise in the data apart from patterns, then the training accuracy will be higher than the test accuracy as the model has not seen the new data. This is called over fitting and it happens , for example, due to many features/variables in the data set.

Regularization is used to penalize the model coefficients and it can be done by changing the lambda value or penalty term. It controls the coefficient values either by decreasing the values or dropping the variables while minimizing the loss. When the variables are dropped model's complexity is reduced and as such over fitting can be eliminated.

Further, increasing the penalty value or lambda value to a large extent may make the model underfit.

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. 10 points

Loss function calculates the difference between the model's output with that of expected output of the model or a variable. If the difference is larger, then the loss function penalizes the model in order to make the difference smaller as the objective is to make the difference smaller.

Commonly used regression loss functions are Mean Absolute Error (MAE) and Root Mean squared Error (RMSE).

Commonly used classification loss functions are Binary Cross Entropy and Categorical Cross Entropy.

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. 10 points

No, we cannot trust the model. When the data set is too small , there is a possibility that model may follow the data too closely , learning too little patterns. And it may learn the noise , which is not required , as well. Noise is stochastic and that is difficult to predict. Hence it performs very well on the training data set and may perform very badly on the test data as it hasn't seen the new data and the required patterns are not learnt well.

QA4. What is the role of the lambda parameter in regularized linear models such as Lasso or Ridge regression models? 10 points

Regularization penalizes the variable coefficients in the model to avoid over fitting using the penalty parameter or lambda. In Lasso, when we increase the lambda value it drops variables that are not significant to the model and also reduces the value of the coefficients of the remaining variables in the model, while minimizing the overall loss function. This way the model will get rid of complexity by reducing the number of features in the data set, eliminating the over fitting scenario. On the contrary, ridge model only decreases the coefficient value while keeping all the variables in the model.

We need to make a note that while increasing the lamda value, one needs to make sure that the model which is otherwise optimal or over fit would not under fit the model as lambda is goes too small.

R Studio Output

```
#Clear the Workspace
rm(list = ls())
```

Install the required packages and libraries , as required, at appropriate chunks.

```
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 4.1-3

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice
```

Loading the Data from ISLR library (Carseats)

```
Carseats_Filtered <- Carseats %>% select("Sales", "Price", "Advertising","Pop
ulation","Age","Income","Education")
```

Seperating Sales(Dependent Variable) and Other Data (Independent Variables)

Converting the Data into Matrix form : A required step for Glmnet model to work

---

```
Carseats_Sales <- Carseats_Filtered$Sales
Carseats_Other <- data.matrix(Carseats_Filtered[, c(-1)])
```

Data Pre-Processing: Standardizing the data and displaying the Summary of the Data

---

```
preProc <- preProcess(Carseats_Other, method=c("center", "scale"))
Carseats_Scaled <- predict(preProc,Carseats_Other)
summary(Carseats_Scaled)

##      Price            Advertising        Population            Age
##  Min.   :-3.87702   Min.   :-0.9977   Min.   :-1.72918   Min.   :-1.74827
```

```
##  1st Qu.:-0.66711   1st Qu.:-0.9977   1st Qu.:-0.85387   1st Qu.:-0.83779
##  Median : 0.05089   Median :-0.2459   Median : 0.04858   Median : 0.07268
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.64219   3rd Qu.: 0.8067   3rd Qu.: 0.90693   3rd Qu.: 0.78255
##  Max.   : 3.17633   Max.   : 3.3630   Max.   : 1.65671   Max.   : 1.64673
##      Income            Education
##  Min.   :-1.70290   Min.   :-1.48825
##  1st Qu.:-0.92573   1st Qu.:-0.72504
##  Median : 0.01224   Median : 0.03816
##  Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.79834   3rd Qu.: 0.80137
##  Max.   : 1.83458   Max.   : 1.56457
```

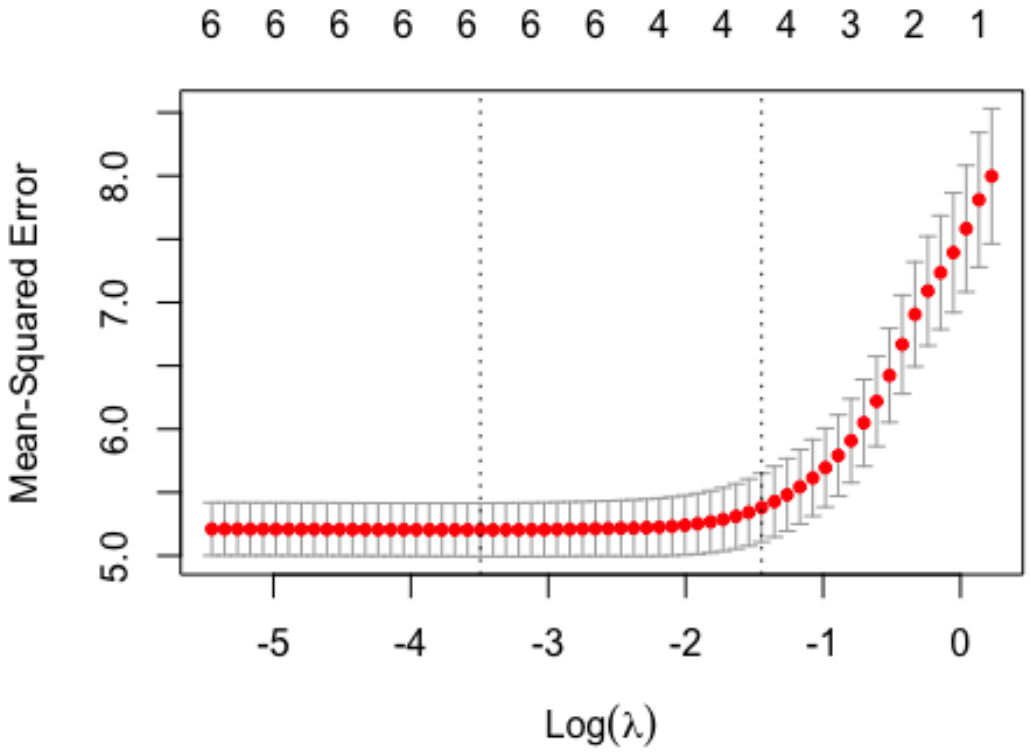
Build the Lasso Regression Model by default its Lasso Model (if aplha = 1 is not mentioned). By default k fold cross validation is k=10

```
Model <- cv.glmnet(Carseats_Scaled,Carseats_Sales , alpha = 1)

Lambda_Best <- Model$lambda.min
Lambda_Best

## [1] 0.0303731

#produce plot of test MSE by lambda value
plot(Model)
```



1.Optimal Lambda value is 0.004305309

Finding the Coefficients for the Best model with the Optimal Lambda

---

```
Model_Best <- glmnet(Carseats_Scaled,Carseats_Sales, alpha = 1, lambda = Lamb
da_Best)
coef(Model_Best)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)  7.49632500
## Price       -1.32402896
## Advertising  0.79227709
## Population  -0.09058924
## Age         -0.75813259
## Income       0.26903731
## Education   -0.06356931
```

2.Coefficent of Price is -1.35384596

Finding the remaining variables with lambda of 0.01 and 0.1

```
Model_Best1 <- glmnet(Carseats_Scaled,Carseats_Sales, alpha = 1, lambda = 0.0
1)
coef(Model_Best1)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)  7.49632500
## Price       -1.34733223
## Advertising  0.82026088
## Population  -0.12187685
## Age         -0.78190633
## Income       0.28488631
## Education   -0.08502707
```

```
Model_Best2 <- glmnet(Carseats_Scaled,Carseats_Sales, alpha = 1, lambda = 0.1
)
coef(Model_Best2)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  7.4963250
## Price       -1.2447745
## Advertising  0.7007230
## Population   .
## Age         -0.6775428
## Income       0.2139222
## Education    .
```

3.With Lambda 0.01 all the variables are remaining But with 0.1 population & Education variables are going out.

As lambda increases variables will drop.

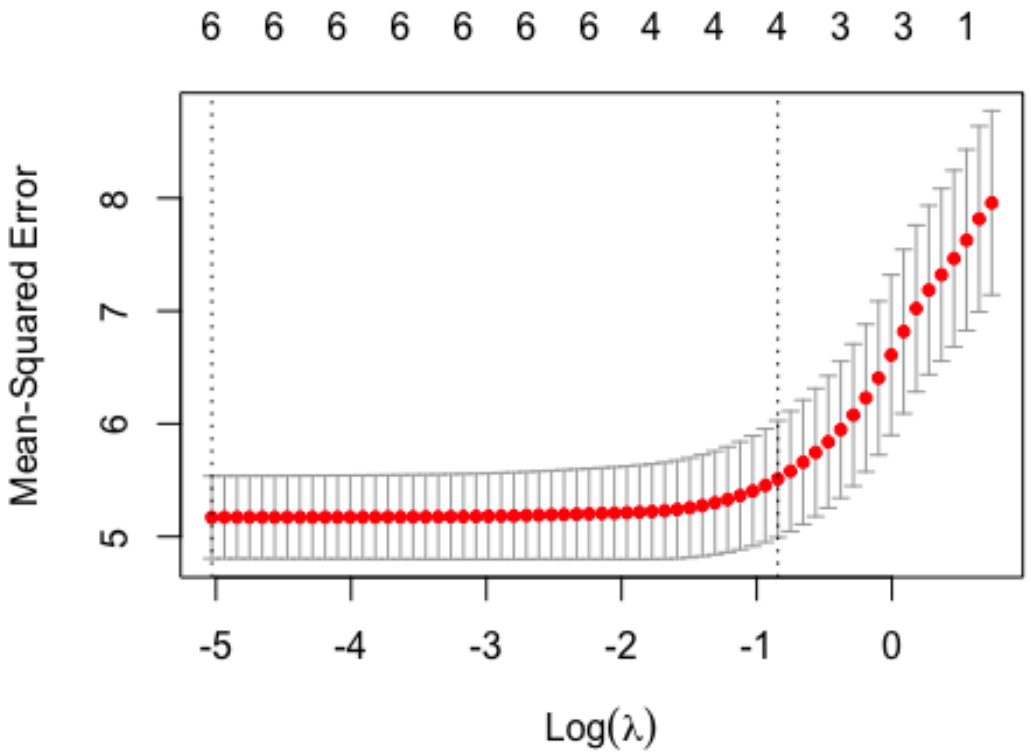
Best lambda for Elastic Model (alpha=0.6)

```
Model1 <- cv.glmnet(Carseats_Scaled,Carseats_Sales , alpha = 0.6)

Lambda_Best1 <- Model1$lambda.min
Lambda_Best1
```

```
## [1] 0.006538062

#produce plot of test MSE by lambda value
plot(Model1)
```



4.Optimal Lambda for elastic model is 0.006538062