**ASSIGNMENT-4**

**STAT LEARNING LAB: BOOTSTRAP AND CROSS VALIDATION**

**Name: Sowmya Patlolla**

**1.Change the value of the seed for (Validation Set Approach) part 1 to the last 4 numbers of your student ID. Report the test rates for a Poly fit quadratic and cubic using the seed.**

**A.** Last 4 digits of the id number is **8256.**

**Source Code:**

#loading (ISLR2)

#setting the seed with last 4 digits of ID number

Set.seed(8256)

#sample for training and testing data set

Train <- sample (392,196)

Attach (Auto)

#fitting the data for poly Quadratic

Lm. fit2 <- lm(mpg ~ poly(horsepower, 2), data = Auto, subset = train)

#finding the test error

Mean (mpg – predict (lm. fit2, Auto)) [-train] ^2)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

We can see output for the MSE for poly fit Quadratic is 20.78461.

**#Cubic**

The last 4 digits of the id number is **8256**.

#loading (ISLR2)

#setting the seed with last 4 digits of ID number

Set.seed(8256)

#sample for training and testing data

set Train <- sample (392,196)

Attach (Auto)

#fitting the data for Cubic

Lm. fit2 <- lm(mpg ~ poly(horsepower, 3), data = Auto, subset = train)

#finding the test error

Mean (mpg – predict (lm. fit2, Auto))[-train]^2)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

We can see output for the MSE for poly fit Quadratic is 20.73093

**2) Change the ratio of train/test to a new percentage. (original was 50/50) Select 3 new ratios and compute and report the test performance for each using the best performing of the two Poly fit values from task 1.**

**A.** We want to explore three new ratios: 60/40, 70/30, and 80/20.

Following is the source code to create a 60/40 split:

**Source Code:**

#loading the library ISLR2 Library (ISLR2)

#setting the seed with last 4 digits of ID number

Set.seed(8256)

#Diving the sample for training and testing data set in the ratio 6:4

Train <- sample (392,235)

#fitting the data for poly Quadratic

lm. fit2 <- lm(mpg ~ poly(horsepower, 2), data = Auto, subset = train)

#finding the test error

Mean (mpg – predict (lm. fit2, Auto)) [-train] ^2)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The test error is 20.7846.

**70/30, split:**

Following is the source code to create a 70/30 split:

**Source Code**

#loading the library ISLR2 Library (ISLR2)

#setting the seed with last 4 digits of ID number

Set.seed(8256)

#Diving the sample for training and testing data set in the ratio 7:3

Train <- sample (392,274)

#fitting the data for poly Quadratic

lm. fit2 <- lm(mpg ~ poly(horsepower, 2), data = Auto, subset = train)

#finding the test error

Mean (mpg – predict (lm. fit2, Auto))[-train]^2)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The test error is 21.126.

**# 80/20 split:**

Following is the source code to create 80/20 split:

**Source Code**

#loading the library ISLR2 Library (ISLR2)

#setting the seed with last 4 digits of ID number

Set.seed(8256)

#Diving the sample for training and testing data set in the ratio 8:2

Train <- sample (392,314)

#fitting the data for poly Quadratic

lm. fit2 <- lm(mpg ~ poly(horsepower, 2), data = Auto, subset = train)

#finding the test error

Mean (mpg – predict (lm. fit2, Auto)) [-train] ^2)

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The test error is 23.77071.

Using the seed 8256 and a test rate of 22.0635, cubic proved to be the best Poly fit value based on task 1 results.

This Poly fit value now allows us to calculate test performance for each of the three additional ratios:

for 60/40 split: Test performance for Quadratic: 20.78461

for 70/30 split: Test performance for Quadratic: 21.12631

for 80/20 split: Test performance for Quadratic: 23.77071

As a result, the test results using the best Poly fit value from task 1 for each of the three new ratios are as stated above.

**3) The LOOCV used in the lab compares poly order using horsepower for mpg prediction. Compute and report the overall performance of Poly for orders 1:8 for any one of the other features in the Auto dataset (acceleration, cylinders, displacement, weight) to mpg.**

**Source Code:**

#loading library

Library(boot)

cv.error <- rep (0,8)

#using for loop to find test error for poly order (1 to 8)

For (I in 1:8)

{

Glm.fit <- glm(mpg ~ poly(acceleration, i), data = Auto)

cv.error[i] <-cv.glm(auto,glm.fit)$delta[1]

}

cv.error

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**4) Select a different feature form Auto than you did in task 3 and perform a 5 fold and 10 fold k-fold cross validation.**

A) Performing 5 – fold cross validation

**Source Code:**

cv.error.5 <- rep (0,5)

for (I in 1:5)

{

glm.fit <- glm(mpg ~ poly(displacement, i), data = Auto)

cv.error.5[i] <- cv.glm(Auto, glm.fit, K=5)$delta[1]

}

cv.error.5

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

b) Performing 10-fold cross validation

**Source Code:**

cv.error.10 <- rep (0,10)

for (I in 1:10)

{

glm.fit <- glm(mpg ~ poly(displacement,i), data = Auto)

cv.error.10[i] <- cv.glm(Auto, glm.fit, K=10)$delta[1]

}

cv.error.10

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**5) Compute the last bootstrap exercise (the quadratic fit for horsepower) from the lab 3 more times, using a new number of samples {250, 500, 2500}. The goal of this task will be to compare the error estimates of the increasing number of samples. Report your observations about the error as the number of bootstrap sets increases.**

A)

**Source Code**

boot.fn <- function(data, index)

coef(

lm(mpg ~ horsepower + l(horsepower^2),

data = data, subset = index)

)

Set.seed(8256)

boot (Auto, boot. fn,250)

summary (

lm(mpg ~ horsepower + l(horsepower^2),data = Auto)

)

$coef

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

b)500

**Source Code**

boot.fn <- function(data, index)

coef(lm(mpg ~ horsepower + l(horsepower^2),

data = data, subset = index))

Set.seed(8256)

Boot (Auto, boot. fn,500)

Summary (lm(mpg ~ horsepower + l(horsepower^2), data = Auto)

)

$coef

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**c) 2500, Source Code**

boot.fn <- function(data, index)

coef( lm(mpg ~ horsepower + l(horsepower^2),

data = data, subset = index))

Set.seed(8256)

boot (Auto, boot. fn,2500)

summary (lm(mpg ~ horsepower + l(horsepower^2),data = Auto)

)

$scoef

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

As an outcome, we can see from the output that as the number of bootstraps resamples rises, the standard error of the bootstrap statistic t1\* drops. The standard error was 2.04 for the first analysis with 250 samples; it dropped to 2.02 for the second analysis with 500 samples; and it dropped even further to 2.06 for the third analysis with 2500 resamples.