Apurva Shrivastava
925009508
ISEN 613-600

Assignment #4

Problem 1

This question should be answered using the Default data set. In Chapter 4 on classification, we used logistic regression to predict the probability of default using income and balance. Now we will estimate the test error of this logistic regression model using the validation set approach. Do not forget to set a random seed before beginning your analysis.

(a) Fit a logistic regression model that predicts default using income and balance.

```
> library(ISLR)
> summary(Default)
 default    student       balance           income
 No :9667   No :7056   Min.   :   0.0   Min.   :  772
 Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
                       Median : 823.6   Median :34553
                       Mean   : 835.4   Mean   :33517
                       3rd Qu.:1166.3   3rd Qu.:43808
                       Max.   :2654.3   Max.   :73554

> set.seed(1)
> glm.fit <- glm(default ~ income + balance, data = Default, family = binomial)
> 
```

(b) Using the validation set approach, estimate the test error of this model. You need to perform the following steps:

i. Split the sample set into a training set and a validation set.

```
>
> train_data <- sample(dim(Default)[1], dim(Default)[1]/2)
> 
```

ii. Fit a logistic regression model using only the training data set.

```
>
> glm.fit <- glm(default ~ income + balance, data = Default, family = binomial, subset = train_data)
> 
```

iii. Obtain a prediction of default status for each individual in the validation set using a threshold of 0.5. iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

```
> glm.pred <- rep("No", dim(Default)[1]/2)
> glm.probs <- predict(glm.fit, Default[-train_data, ], type = "response")
> glm.pred[glm.probs > 0.5] <- "Yes"
> mean(glm.pred != Default[-train_data, ]$default)
[1] 0.0286
> 
```

Apurva Shrivastava
925009508
ISEN 613-600

This shows there is 2.86% error rate from the validation set approach.

(c) Repeat the process in (b) three times, using three different splits of the observations into a training set and a validation set. Comment on the results obtained.

1.

```
> train_data <-sample(dim(Default)[1], dim(Default)[1]/2)
> glm.fit <- glm(default ~ income + balance, data = Default, family = binomial, subset = train_data)
> glm.pred <- rep("No", dim(Default)[1]/2)
> glm.probs <- predict(glm.fit, Default[-train_data, ], type = "response")
> glm.pred[glm.probs > 0.5] <- "Yes"
> mean(glm.pred != Default[-train_data, ]$default)
[1] 0.0236
> 
```

2.

```
> train_data <-sample(dim(Default)[1], dim(Default)[1]/2)
> glm.fit <- glm(default ~ income + balance, data = Default, family = binomial, subset = train_data)
> glm.pred <- rep("No", dim(Default)[1]/2)
> glm.probs <- predict(glm.fit, Default[-train_data, ], type = "response")
> glm.pred[glm.probs > 0.5] <- "Yes"
> mean(glm.pred != Default[-train_data, ]$default)
[1] 0.028
> 
```

3.

```
> train_data <-sample(dim(Default)[1], dim(Default)[1]/2)
> glm.fit <- glm(default ~ income + balance, data = Default, family = binomial, subset = train_data)
> glm.pred <- rep("No", dim(Default)[1]/2)
> glm.probs <- predict(glm.fit, Default[-train_data, ], type = "response")
> glm.pred[glm.probs > 0.5] <- "Yes"
> mean(glm.pred != Default[-train_data, ]$default)
[1] 0.0268
> 
```

We see that error rate does not change considerable and the average error rate across three different splits is 2.613%.

(d) Consider another logistic regression model that predicts default using income, balance and student (qualitative). Estimate the test error for this model using the validation set approach. Does including the qualitative variable student lead to a reduction of test error rate?

```
> train_data <-sample(dim(Default)[1], dim(Default)[1]/2)
> glm.fit <- glm(default ~ income + balance + student, data = Default, family = binomial, subset = train_data)
> glm.pred <- rep("No", dim(Default)[1]/2)
> glm.probs <- predict(glm.fit, Default[-train_data, ], type = "response")
> glm.pred[glm.probs > 0.5] <- "Yes"
> mean(glm.pred != Default[-train_data, ]$default)
[1] 0.0264
> 
```

We see that the error rate after adding the “student” dummy variable is 2.64% which is approximately the same as without using the dummy variable. Hence, adding a dummy variable does not seem to reduce the error rate.

Apurva Shrivastava
925009508
ISEN 613-600

Problem 2

This question requires performing cross validation on a simulated data set.

(a) Generate a simulated data set as follows:

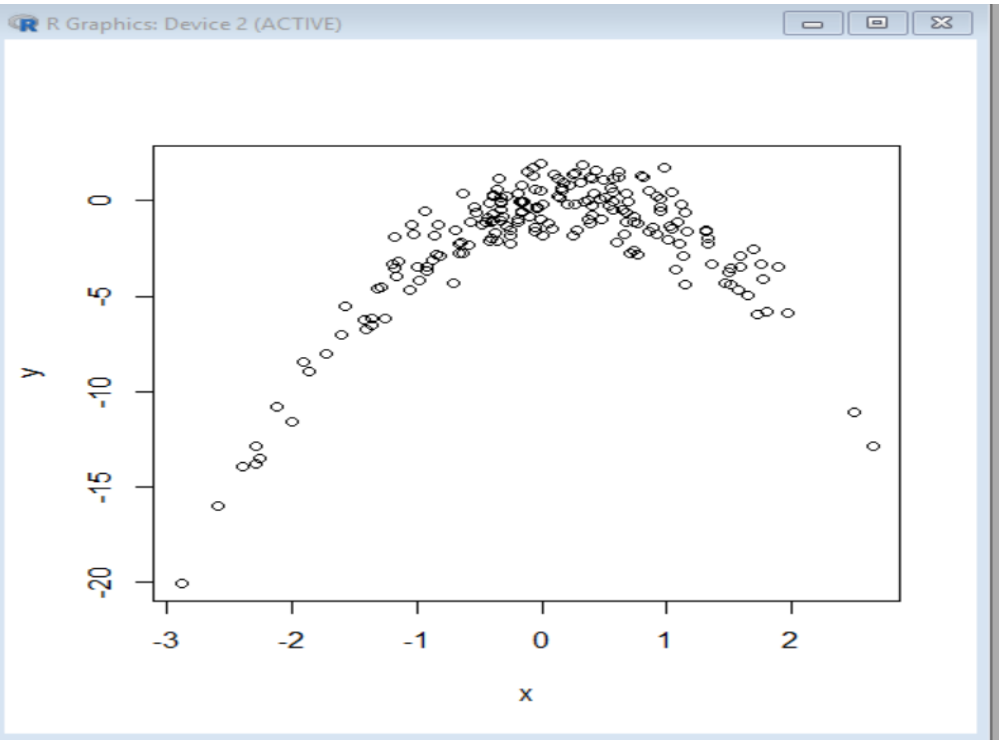
set.seed(1) x=rnorm(200) y=x-2*x^2+rnorm(200) In this data set, what is n and what is p? Write out the model used to generate the data in equation form (i.e., the true model of the data).

```
> set.seed(1)
> y <- rnorm(200)
> x <- rnorm(200)
>
> y <- x -2 * x ^2 + rnorm(200)
>
> plot(x,y)
>
```

Model: y = x-2*x^2 + ε

n= 200; **p =** 2

(b) Create a scatter plot of Y vs X. Comment on what you find.



This shows that x and y have a curved relationship.

Apurva Shrivastava
925009508
ISEN 613-600

(c) Consider the following four models for the data set:

i. $Y = \beta 0 + \beta 1X + \epsilon$

```
> library(boot)
> data_set <- data.frame(x,y)
> fit.glm_1 <- glm(y~x)
> cv.glm(data_set,fit.glm_1)$delta[1]
[1] 6.037638
> |
```

ii. $Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \epsilon$

```
> fit.glm_2 <- glm(y~poly(x,2))
> cv.glm(data_set,fit.glm_2)$delta[1]
[1] 1.040922
> |
```

iii. $Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \beta 3X\ 3 + \epsilon$

```
> fit.glm_3 <- glm(y~poly(x,3))
> cv.glm(data_set,fit.glm_3)$delta[1]
[1] 1.039049
> |
```

iv. $Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \beta 3X\ 3 + \beta 4X\ 4 + \epsilon$

```
> fit.glm_4 <- glm(y~poly(x,4))
> cv.glm(data_set,fit.glm_4)$delta[1]
[1] 1.028604
> |
```

Compute the LOOCV errors that result from fitting these models.

Model	**Error Rate**
$Y = \beta 0 + \beta 1X + \epsilon$	6.037
$Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \epsilon$	1.040
$Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \beta 3X\ 3 + \epsilon$	1.039
$Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \beta 3X\ 3 + \beta 4X\ 4 + \epsilon$	1.028

(d) Repeat (c) using another random seed, and report your results. Are your results the same as what you got in (c)? Why?

Apurva Shrivastava
925009508
ISEN 613-600

```
> set.seed(10)
> fit.glm_1 <- glm(y~x)
> cv.glm(data_set,fit.glm_1)$delta[1]
[1] 6.037638
> fit.glm_2 <- glm(y~poly(x,2))
> cv.glm(data_set,fit.glm_2)$delta[1]
[1] 1.040922
> fit.glm_3 <- glm(y~poly(x,3))
> cv.glm(data_set,fit.glm_3)$delta[1]
[1] 1.039049
> fit.glm_4 <- glm(y~poly(x,4))
> cv.glm(data_set,fit.glm_4)$delta[1]
[1] 1.028604
>
```

As we can see the results are same as in (c) since LOOCV evaluates n folds of a single observation.

(e) Which of the models in (c) had the smallest LOOCV error? Is this what you expected? Explain your answer.

The model: $Y = \beta0 + \beta1X + \beta2X\ 2 + \beta3X\ 3 + \beta4X\ 4 + \epsilon$ has the smallest LOOCV error as higher degree models have better accuracy. We can see as we increase the power the error rate starts decreasing.

(f) Now we use 5-fold CV for the model selection. Compute the CV errors that result from fitting the four models. Which model has the smallest CV error? Are the results consistent with LOOCV?

```
>
> cv.error_4 = rep(0,4)
> for(i in 1:4){
+ fit = glm(y ~poly(x,i))
+ cv.error_4[i] = cv.glm(data_set,fit,K=5)$delta[1]
+ }
> cv.error_4
[1] 5.962872 1.028538 1.096130 1.019385
>
```

The smallest error in 5-fold CV model is for fourth-degree polynomial model ($Y = \beta0 + \beta1X + \beta2X\ 2 + \beta3X\ 3 + \beta4X\ 4 + \epsilon$), which has an error of 1.019. This is consistent with the LOOCV as in LOOCV also the smallest error was for four-degree polynomial model.

(g) Repeat (f) using 10-fold CV. Are the results the same as 5-fold CV?

Apurva Shrivastava
925009508
ISEN 613-600

```
>
> cv.error_4 = rep(0,4)
> for(i in 1:4){
+ fit = glm(y ~poly(x,i))
+ cv.error_4[i] = cv.glm(data_set,fit,K=10)$delta[1]
+ }
> cv.error_4
[1] 5.959594 1.032272 1.046880 1.015591
>
```

The results for 10-fold model are similar to 5-fold model. The minimum error is for the fourth-degree polynomial model ($Y = \beta 0 + \beta 1X + \beta 2X\ 2 + \beta 3X\ 3 + \beta 4X\ 4 + \epsilon$).