Exercise 4.7: Problem 11 (part e), Exercise 5.4: Problem 5

Section 23 - Group 6 Project Groups (Bosan Hsu, Fan Liu, Jimeng Yin, Michael Liu, Richard Wang, Zhuoqian Zhang

11. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Autodata set. for context: (b) Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatter plots and boxplots may be useful tools to answer this question. Describe your findings.

(e) Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

```
library(ISLR)
attach(Auto)
Auto = na.omit(Auto)
mpg01 <- ifelse(mpg > median(mpg), 1, 0)
Auto <- data.frame(Auto, mpg01)
set.seed(100)
train <- sample(nrow(Auto),nrow(Auto)/2,replace = FALSE)
Auto.train <- Auto[train,]
Auto.test <- Auto[-train,]
mpg01.test <- mpg01[-train]
library(MASS)
qda.fit = qda(mpg01~ log(displacement) + log(weight) + year, data = Aut
o, subset = train)
qda.pred = predict(qda.fit, Auto.test)
table(qda.pred$class, mpg01.test)

##    mpg01.test
##      0  1
##   0 92  3
##   1 12 89

mean(qda.pred$class != mpg01.test)

## [1] 0.07653061
```

the test error of the model obtained is 7.65%

5. In Chapter 4, we used logistic regression to predict the probability of default using income and balance on the Default data set. We will now 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 uses income and balance to predict default.

```
library(ISLR)
attach(Default)

set.seed(1)
fit.glm = glm(default ~ income + balance, family = "binomial")
summary(fit.glm)

##
## Call:
## glm(formula = default ~ income + balance, family = "binomial")
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

(b) Using the validation set approach, estimate the test error of this model. In order to do this, you must perform the following steps:

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

```
set.seed(1)
train = sample(nrow(Default), nrow(Default)*0.7, replace = F)
train.data = Default[train,]
test.data = Default[-train,]
```

ii. Fit a multiple logistic regression model using only the train-ing observations.

```
glm.fit = glm(default ~ income + balance, data = train.data, family = "
binomial")
summary(glm.fit)

##
## Call:
## glm(formula = default ~ income + balance, family = "binomial",
##     data = train.data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.167e+01  5.214e-01 -22.379  < 2e-16 ***
## income       2.560e-05  6.012e-06   4.258 2.06e-05 ***
## balance      5.574e-03  2.678e-04  20.816  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2030.3  on 6999  degrees of freedom
## Residual deviance: 1079.6  on 6997  degrees of freedom
## AIC: 1085.6
##
## Number of Fisher Scoring iterations: 8
```

iii. Obtain a prediction of default status for each individual in the validation set by computing the posterior probability of default for that individual, and classifying the individual to the default category if the posterior probability is greater than 0.5.

```
glm.probs = predict(glm.fit, data = test.data, type="response")
glm.pred = ifelse(glm.probs>0.5,1,0)
```

iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

```
mean(glm.pred != test.data)

## [1] 0.9875
```

The fraction of the observations in the validation set that are misclassified is 1.28%.

(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.

```
compute_validation_set_error <- function(seed) {
    set.seed(seed)
    train = sample(nrow(Default), nrow(Default)*0.7, replace = F)
    train.data = Default[train,]
    test.data = Default[-train,]
    glm.fit = glm(default ~ income + balance, data = train.data, family
 = "binomial")
    glm.probs = predict(glm.fit, data = test.data, type="response")
    glm.pred = ifelse(glm.probs>0.5,1,0)
    error = mean(glm.pred != test.data)
    return(error)
}
seeds = c(2, 3, 4)
errors = sapply(seeds, compute_validation_set_error)
print(errors)

## [1] 0.9863333 0.9885833 0.9886667
```

There is a variation in error rates, but the number is not very large. The variation may caused by the training and test dataset we include. Since the variation is not very large, the model perform well in the dataset.

(d) Now consider a logistic regression model that predicts the probability of default using income, balance, and a dummy variable for student. Estimate the test error for this model using the validation set approach. Comment on whether or not including a dummy variable for student leads to a reduction in the test error rate.

```
set.seed(1)
train = sample(nrow(Default), nrow(Default)*0.7, replace = F)
train.data = Default[train,]
test.data = Default[-train,]
glm.fit = glm (default~income+balance+student, data = train.data, famil
y = binomial)
glm.probs = predict(glm.fit, data = test.data, type="response")
glm.pred = ifelse(glm.probs>0.5,1,0)
mean(glm.pred != test.data)

## [1] 0.9874167
```

There is no significant change in test error rate when we include a dummy variable. But we need to try more times to make sure about this conclusion.