Bruce Campbell ST-617 Homework 2

Tue Jul 05 17:45:39 2016

Chapter 5

Problem 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.

```r
rm(list = ls())
library(ISLR)
attach(Default)
glm.fit = glm(default ~ income + balance, data = Default, family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = binomial,
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4725  -0.1444  -0.0574  -0.0211   3.7245
##
## 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 appbroach, 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. ii. Fit a multiple logistic regression model using only the training observations. 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. iv. Compute the validation set error, which is the fraction of the observations in the validation set that are misclassified.

```
set.seed(7)

train = sample(nrow(Default), floor(nrow(Default) * 2/3))
DF <- Default
DFTrain <- DF[train, ]
DFTest <- DF[-train, ]

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

```
## 
## Call:
## glm(formula = default ~ income + balance, family = binomial, 
##     data = DFTrain)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.2321  -0.1470  -0.0589  -0.0215   3.7152  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.129e+01  5.255e-01 -21.479   <2e-16 ***
## income       1.430e-05  6.074e-06   2.355   0.0185 *  
## balance      5.630e-03  2.792e-04  20.163   <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: 1940.3  on 6665  degrees of freedom
## Residual deviance: 1067.6  on 6663  degrees of freedom
## AIC: 1073.6
## 
## Number of Fisher Scoring iterations: 8
```

```
glm.probs = predict(glm.fit, DFTest, type = "response")

contrasts(DFTest$default)
```

```
##     Yes
## No    0
## Yes   1
```

```r
glm.pred = rep("No ", nrow(DFTest))
glm.pred[glm.probs > 0.5] = " Yes"
TB <- table(glm.pred, DFTest$default)
ACC_Validation = (TB[2] + TB[3])/length(DFTest$default)

modelsDF <- data.frame(iteration = 1, Accuracy = 1 - ACC_Validation)
```

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.

```r
attach(Default)
modelsDF <- data.frame(iteration = numeric(), Accuracy = numeric())

for (i in 1:3) {
    train = sample(nrow(Default), floor(nrow(Default) * 2/3))
    DF <- Default
    DFTrain <- DF[train, ]
    DFTest <- DF[-train, ]

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

    glm.probs = predict(glm.fit, DFTest, type = "response")

    glm.pred = rep("No ", nrow(DFTest))
    glm.pred[glm.probs > 0.5] = " Yes"
    TB <- table(glm.pred, DFTest$default)
    ACC_Validation = (TB[2] + TB[3])/length(DFTest$default)
    modelsDF <- rbind(modelsDF, data.frame(iteration = i, Accuracy = 1 - ACC_Validation))
}
library(pander)
pander(modelsDF)
```

iteration	Accuracy
1	0.02759
2	0.02789
3	0.02579

The validation test set error rates are all very similar. This indicated the model is stable with respect to the random split into test and training sets and that the validation approach may be vaible in this instance althugh we'd probably want to do more iterations to confirm this.

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.

```r
attach(Default)
modelsDFAug <- data.frame(iteration = numeric(), Accuracy = numeric())

for (i in 1:3) {
    train = sample(nrow(Default), floor(nrow(Default) * 2/3))
    DF <- Default
    DFTrain <- DF[train, ]
    DFTest <- DF[-train, ]

    glm.fit = glm(default ~ income + balance + student, data = DFTrain, family = binomial)
    summary(glm.fit)

    glm.probs = predict(glm.fit, DFTest, type = "response")

    glm.pred = rep("No ", nrow(DFTest))
    glm.pred[glm.probs > 0.5] = " Yes"
    TB <- table(glm.pred, DFTest$default)
    ACC_Validation = (TB[2] + TB[3])/length(DFTest$default)
    modelsDFAug <- rbind(modelsDFAug, data.frame(iteration = i, Accuracy = 1 -
        ACC_Validation))
}
library(pander)
pander(modelsDFAug)
```

iteration	Accuracy
1	0.02729
2	0.02579
3	0.02729

Including the student status as a predictor did not appear to change the validation set error rates - the change is the number of errors for each of the three runs is :

```r
diff <- (modelsDFAug$Accuracy - modelsDF$Accuracy) * nrow(DFTest)
pander(diff)
```

-1, *-7* and *5*

To make a more precise statement we'd run more iterations and compare the errors using a statistical test such as a t-test.

To make a