Predictive Modeling Exercises Week 4

Amol Gote

08/02/2020

# 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.

# 5a - Fit a logistic regression model that uses income and balance to predict default.

regression\_model <- glm(default ~ income + balance, data = Default, family = "binomial")  
summary(regression\_model)

##   
## 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

# 5b Using the validation set approach, estimate the test error of this model.

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

sample\_set<-sample(nrow(Default),nrow(Default)/2)  
training\_dataset <- Default[sample\_set,]  
test\_dataset <- Default[-sample\_set,]

# ii Fit a multiple logistic regression model using only the training observations.

regression\_model <- glm(default ~ income + balance, data = training\_dataset, family = "binomial")  
summary(regression\_model)

##   
## Call:  
## glm(formula = default ~ income + balance, family = "binomial",   
## data = training\_dataset)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1437 -0.1322 -0.0518 -0.0187 3.5563   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.168e+01 6.377e-01 -18.315 <2e-16 \*\*\*  
## income 1.684e-05 7.458e-06 2.259 0.0239 \*   
## balance 5.754e-03 3.386e-04 16.991 <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: 1333.49 on 4999 degrees of freedom  
## Residual deviance: 706.03 on 4997 degrees of freedom  
## AIC: 712.03  
##   
## 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.

probablity <- predict(regression\_model, test\_dataset, type = "response")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5,"Yes","No")

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

validation\_error\_rate <- mean(prediction!=test\_dataset$default)  
validation\_error\_rate

## [1] 0.0306

matrix <- table(test\_dataset$default,prediction, dnn=c("Observed","Predicted"))  
matrix

## Predicted  
## Observed No Yes  
## No 4796 19  
## Yes 134 51

Validation set error rate is: 0.0306. or 3.06%

# 5c 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.

sample\_set<-sample(nrow(Default),nrow(Default)\*0.6)  
training\_dataset <- Default[sample\_set,]  
test\_dataset <- Default[-sample\_set,]  
regression\_model <- glm(default ~ income + balance, data = training\_dataset, family = "binomial")  
probablity <- predict(regression\_model, test\_dataset, type = "response")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5,"Yes","No")  
validation\_error\_rate1 <- mean(prediction!=test\_dataset$default)  
validation\_error\_rate1

## [1] 0.02825

sample\_set<-sample(nrow(Default),nrow(Default)\*0.7)  
training\_dataset <- Default[sample\_set,]  
test\_dataset <- Default[-sample\_set,]  
regression\_model <- glm(default ~ income + balance, data = training\_dataset, family = "binomial")  
probablity <- predict(regression\_model, test\_dataset, type = "response")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5,"Yes","No")  
validation\_error\_rate2 <- mean(prediction!=test\_dataset$default)  
validation\_error\_rate2

## [1] 0.02666667

sample\_set<-sample(nrow(Default),nrow(Default)\*0.8)  
training\_dataset <- Default[sample\_set,]  
test\_dataset <- Default[-sample\_set,]  
regression\_model <- glm(default ~ income + balance, data = training\_dataset, family = "binomial")  
probablity <- predict(regression\_model, test\_dataset, type = "response")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5,"Yes","No")  
validation\_error\_rate3 <- mean(prediction!=test\_dataset$default)  
validation\_error\_rate3

## [1] 0.0225

Following is the data split in training and test data ration along with validation error rate/misclassification

1. Tranining : Test -> 60 : 60, validation error rate 0.02825. or 2.825%   
2. Tranining : Test -> 70 : 30, validation error rate 0.0266667. or

2.6666667%   
3. Tranining : Test -> 80 : 20, validation error rate 0.0225. or 2.25%

Validation error rate is different for different sample splits of taining and test dataset. This indicates that validation error rate varies by which observations are in the training/validation sets.

# 5d 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.

sample\_set<-sample(nrow(Default),nrow(Default)\*0.8)  
training\_dataset <- Default[sample\_set,]  
test\_dataset <- Default[-sample\_set,]  
regression\_model <- glm(default ~ income + balance + student, data = training\_dataset, family = "binomial")  
probablity <- predict(regression\_model, test\_dataset, type = "response")  
probablity<-predict(regression\_model,test\_dataset,type="response")  
prediction<-ifelse(probablity > 0.5,"Yes","No")  
mean(prediction!=test\_dataset$default)

## [1] 0.0285

Including the the dummy variable for student did not lead to any reduction in validation set error rate.

# 9 We will now consider the Boston housing data set, from the MASS library.

# 9a Based on this data set, provide an estimate for the population mean of medv. Call this estimate ˆμ.

est\_pop\_mean\_medv <- mean(Boston$medv)  
est\_pop\_mean\_medv

## [1] 22.53281

Estimate for the population mean of medv: 22.5328063.

# 9b Provide an estimate of the standard error of ˆμ. Interpret this result.

est\_se\_medv <- sd(Boston$medv)/sqrt(nrow(Boston))  
est\_se\_medv

## [1] 0.4088611

The estimate of the standard error: 0.4088611.

#9c Now estimate the standard error of ˆμ using the bootstrap. How does this compare to your answer from (b)?

boot.fn = function (data ,index ){  
 mean\_medv <- mean(data[index])  
 return (mean\_medv)  
}  
boot(Boston$medv, boot.fn, 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 22.53281 -0.01513379 0.4101586

Standard error obtained using bootstrap is 0.4101586 which is close to 0.4088611 which was obtained earlier.

# 9d Based on your bootstrap estimate from (c), provide a 95% confidence interval for the mean of medv. Compare it to the results obtained using t.test(Boston$medv).

t.test(Boston$medv)

##   
## One Sample t-test  
##   
## data: Boston$medv  
## t = 55.111, df = 505, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 21.72953 23.33608  
## sample estimates:  
## mean of x   
## 22.53281

ci.mean\_medv <- c(22.53281 - (2 \* 0.4107397), 22.53281 + (2 \* 0.4107397))   
ci.mean\_medv

## [1] 21.71133 23.35429

Bootstrap confidence interval for the mean of medv is very close to that obtained using t.test(Boston$medv)

# 9e Based on this data set, provide an estimate, ˆμmed, for the median value of medv in the population.

est\_pop\_median\_medv <- median(Boston$medv)  
est\_pop\_median\_medv

## [1] 21.2

# 9f We now would like to estimate the standard error of ˆμmed. Unfortunately, there is no simple formula for computing the standard error of the median. Instead, estimate the standard error of the median using the bootstrap. Comment on your findings.

boot.fn = function (data ,index ){  
 median\_medv <- median(data[index])  
 return (median\_medv)  
}  
boot(Boston$medv, boot.fn, 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 21.2 0.00775 0.3725201

Estimated median value using bootstrap is 21.2 which is same as 9e above with standard error rate of 0.3725201 which is smaller compared to median value.

# 9g Based on this data set, provide an estimate for the tenth percentile of medv in Boston suburbs. Call this quantity ˆμ0.1. (You can use the quantile() function.)

est\_10th\_percentile\_medv <- quantile(Boston$medv, c(0.1))  
est\_10th\_percentile\_medv

## 10%   
## 12.75

# 9h Use the bootstrap to estimate the standard error of ˆμ0.1. Comment on your findings.

boot.fn = function (data ,index ){  
 quantile\_percentile <- quantile(data[index], c(0.1))  
 return (quantile\_percentile)  
}  
boot(Boston$medv, boot.fn, 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston$medv, statistic = boot.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 12.75 -0.0165 0.510454

Estimated tenth percentile of medv using bootstrap is 12.75 which is same as 9g above with standard error rate of 0.510454 which is smaller compared to the 10th percentile, this is an indication that estimate is representing population accurately