STSCI4740HW4

Nick Gembs

10/30/2022

library(ISLR)  
data=Default

#glimpse(data)

#1  
  
mylogit <- glm(default ~ income + balance, data = data, family = "binomial")  
  
summary(mylogit)

##   
## Call:  
## glm(formula = default ~ income + balance, family = "binomial",   
## data = data)  
##   
## 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

#2  
  
set.seed(1)  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 4090 132  
## Yes 753 25

mean(pred==testing$default)

## [1] 0.823

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.177

#3  
  
set.seed(2)  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 4292 141  
## Yes 545 22

mean(pred==testing$default)

## [1] 0.8628

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.1372

set.seed(3)  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 3948 121  
## Yes 897 34

mean(pred==testing$default)

## [1] 0.7964

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.2036

print("After running the logistic regression on 3 different samples, the max validation error rate was .2036 and the minimum was .1372. It appears that the model is significantly better than random guessing, but does have noticeable variance among trials")

## [1] "After running the logistic regression on 3 different samples, the max validation error rate was .2036 and the minimum was .1372. It appears that the model is significantly better than random guessing, but does have noticeable variance among trials"

#4  
  
set.seed(1)  
options(contrasts = c("contr.treatment", "contr.helmert")) # dummy  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance + student, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 4070 132  
## Yes 773 25

mean(pred==testing$default)

## [1] 0.819

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.181

options(contrasts = c("contr.treatment", "contr.helmert")) # dummy  
set.seed(2)  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance + student, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 4239 141  
## Yes 598 22

mean(pred==testing$default)

## [1] 0.8522

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.1478

options(contrasts = c("contr.treatment", "contr.helmert")) # dummy  
set.seed(3)  
  
train = sample(length(data$default), length(data$default)/2)  
  
training = data[train,]  
testing = data[-train,]  
  
mylogit <- glm(default ~ income + balance +student, data = training, family = "binomial")  
  
logit.pred=predict(mylogit, data=testing)  
pred = (exp(logit.pred))/(1+exp(logit.pred))  
  
for( i in 1:length(pred)){  
 if (pred[i] > .5){  
 pred[i] = "Yes"  
} else {  
 pred[i] = "No"  
}  
}  
  
table(pred,testing$default)

##   
## pred No Yes  
## No 3903 120  
## Yes 942 35

mean(pred==testing$default)

## [1] 0.7876

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.2124

print("After running the logistic regression on 3 different samples, the max validation error rate was .2124 and the minimum was .1478. It appears that the model is significantly better than random guessing, but does have noticeable variance among trials. It does not appear that the student variable was effective in predicting default. Including a dummy variable for student does not lead to a reduction in the test error rate")

## [1] "After running the logistic regression on 3 different samples, the max validation error rate was .2124 and the minimum was .1478. It appears that the model is significantly better than random guessing, but does have noticeable variance among trials. It does not appear that the student variable was effective in predicting default. Including a dummy variable for student does not lead to a reduction in the test error rate"

#5  
  
  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ✔ purrr 0.3.5   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()

ctrl <- trainControl(method = "cv", number = 5)  
  
options(contrasts = c("contr.treatment", "contr.helmert")) # dummy  
  
mylogit <- train(default ~ income + balance, data = data, method = "glm", family = "binomial", trControl = ctrl)  
  
print(mylogit)

## Generalized Linear Model   
##   
## 10000 samples  
## 2 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 8000, 8000, 8001, 8000, 7999   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9732997 0.4282539

logit.pred=predict(mylogit, data=testing)  
table(pred,testing$default)

##   
## pred No Yes  
## No 3903 120  
## Yes 942 35

mean(pred==testing$default)

## [1] 0.7876

cat("Testing Error Rate is:" , 1-mean(pred==testing$default) , "\n")

## Testing Error Rate is: 0.2124

mylogit <- train(default ~ income + balance + student, data = data, method = "glm", family = "binomial", trControl = ctrl)  
  
print(mylogit)

## Generalized Linear Model   
##   
## 10000 samples  
## 3 predictor  
## 2 classes: 'No', 'Yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 8000, 8000, 8000, 8001, 7999   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9730001 0.4206439

logit.pred=predict(mylogit, data=testing)  
table(pred,testing$default)

##   
## pred No Yes  
## No 3903 120  
## Yes 942 35

mean(pred==testing$default)

## [1] 0.7876

cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

## Testing Error Rate is: 0.2124

print("5-fold cross-validation yields the same results, adding dummy variable student does not reduce test error in predicting default.")

## [1] "5-fold cross-validation yields the same results, adding dummy variable student does not reduce test error in predicting default."

#LOOCV  
  
ctrl <- trainControl(method = "LOOCV")  
  
options(contrasts = c("contr.treatment", "contr.helmert")) # dummy  
  
mylogit <- train(default ~ income + balance, data = data, method = "glm", family = "binomial", trControl = ctrl)  
  
print(mylogit)  
logit.pred=predict(mylogit, data=testing)  
table(pred,testing$default)  
mean(pred==testing$default)  
  
cat("Testing Error Rate is:" , 1-mean(pred==testing$default) , "\n")  
  
mylogit <- train(default ~ income + balance + student, data = data, method = "glm", family = "binomial", trControl = ctrl)  
  
print(mylogit)  
logit.pred=predict(mylogit, data=testing)  
table(pred,testing$default)  
mean(pred==testing$default)  
  
cat("Testing Error Rate is:" , 1-mean(pred==testing$default))

library(ISLR2)

##   
## Attaching package: 'ISLR2'

## The following objects are masked from 'package:ISLR':  
##   
## Auto, Credit

df=Boston

#df

#a  
  
  
mu\_hat = mean(df$medv)  
mu\_hat

## [1] 22.53281

# b  
  
standard\_error = (sd(df$medv)/sqrt(length(df$medv)))  
  
standard\_error

## [1] 0.4088611

print("With standard error being .4088611, it can be inferred that the majority of the data for medv fall between .4088611 of the sample mean 22.53281")

## [1] "With standard error being .4088611, it can be inferred that the majority of the data for medv fall between .4088611 of the sample mean 22.53281"

# c  
set.seed(9)  
library(boot)

##   
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':  
##   
## melanoma

m <- function(medv,i){mean(df$medv[i])}  
  
# Calculate standard error using 100  
# bootstrapped samples  
boot = boot(df$medv, m, 100)  
boot

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = df$medv, statistic = m, R = 100)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 22.53281 -0.1033597 0.4143032

print("This answer is slightly larger than the result from b")

## [1] "This answer is slightly larger than the result from b"

# d  
  
t.test(df$medv)

##   
## One Sample t-test  
##   
## data: df$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

cat("Bootstrap Confidence Interval: " , c(mu\_hat - 2\*.4143032 , mu\_hat + 2\*.4143032))

## Bootstrap Confidence Interval: 21.7042 23.36141

print("Results are similar, bootstrap interval slightly wider")

## [1] "Results are similar, bootstrap interval slightly wider"

# e  
  
median = median(df$medv)  
median

## [1] 21.2

#f  
  
set.seed(9)  
library(boot)  
   
m <- function(medv,i){median(df$medv[i])}  
  
# Calculate standard error using 100  
# bootstrapped samples  
boot = boot(df$medv, m, 100)  
boot

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = df$medv, statistic = m, R = 100)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 21.2 -0.107 0.4002537

print("Standard error is .4002537. This is similar to the bootstrap standard error for mean, but slightly lower.")

## [1] "Standard error is .4002537. This is similar to the bootstrap standard error for mean, but slightly lower."

#g  
  
mu\_hat\_.01 = quantile(df$medv, probs = .1)  
  
(mu\_hat\_.01)

## 10%   
## 12.75

#h  
  
set.seed(9)  
library(boot)  
   
m <- function(medv,i){(quantile(df$medv[i], probs = .1))}  
  
# Calculate standard error using 100  
# bootstrapped samples  
boot = boot(df$medv, m, 100)  
boot

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = df$medv, statistic = m, R = 100)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 12.75 -0.043 0.5477696

print("Standard error is .5477696. This is higher than the bootstrap standard error for mean and median.")

## [1] "Standard error is .5477696. This is higher than the bootstrap standard error for mean and median."