# 20152410 배형준 Data Mining midterm

## 목차

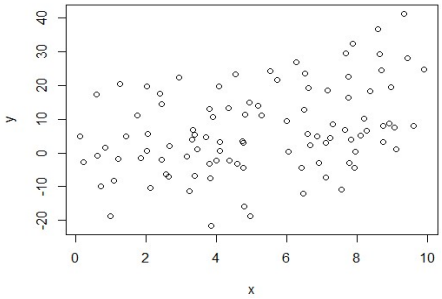
# 본문

## 1. Bootstrap

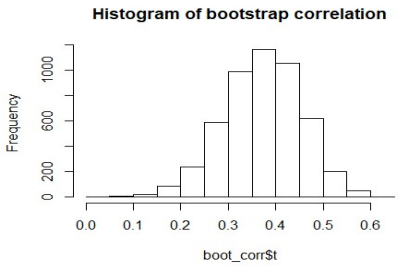
Use the attached “2020Sdata.txt” to answer the following questions.

### (a) Bootstrap bias and standard error

Calculate the correlation coeffcient of x and y with a bootstrap bias estimate and a bootstrap standard error.




붓스트랩 상관계수를 구하기 위해 read.csv 함수를 이용하여 2020Sdata.txt 를 읽었습니다. 해당 데이터는 x, y 두 개의 변수, 100 개의 관측치로 이루어져 있습니다. 오른쪽의 산점도를 보면, 두 변수의 양의 상관관계를 가지고 있지만 그 정도가 크게 강하지 않은 것을 확인할 수 있습니다.




붓스트랩 관련 패키지인 boot 를 불러온 뒤, boot 함수의 statistic 인수에 들어갈 상관계수 함수인 my_cor 를 function 객체로 저장하였습니다. `boot(data1, statistic=my_cor, R=5000)` 코드를 이용하여 붓스트랩 데이터 5000 개에 대해 상관계수를 구했고, 붓스트랩 상관계수의 히스토그램은 다음과 같습니다.

위의 boot 객체를 실행시켰을 때 원 데이터의 상관계수는 0.3731081, bootstrap bias 는 0.001005584, bootstrap standard error 는 0.08127118 로 계산되었습니다.

### (b) Construct 95% bootstrap confidence intervals.

boot.ci 함수를 이용하여 95% 신뢰구간을 구했을 때, 다음과 같이 4 개의 신뢰구간을 구할 수 있었습니다.

95% Normal CI : ( 0.2148, 0.5334 )　　95% Basic (residual) CI : ( 0.2220, 0.5413 )

95% Percentile CI : ( 0.2049, 0.5242 )　　95% BCa CI : ( 0.2018, 0.5220 )

cor.test 를 이용해 구한 95% CI ( 0.1907, 0.5306 )와는 다른 양상인 4 개의 붓스트랩 CI 를 확인할 수 있습니다.

## 2. Hitters data

Use “Hitters” data in ISLR package to answer the following questions. Consider a binary response whether a player’s salary is greater than or equal to 750.

### (a) Naive Bayes classifier using 10 fold CV

Use all predictors and construct a naive Bayes classifier. Report a 10-foldCV classification error.

Hitters 데이터의 결측값을 제거하여 data2 에 저장하였습니다. dplyr 패키지의 mutate 함수와 ifelse 함수를 이용해 Salary 변수의 값이 750 이상인지 아닌지로 binary_salary 변수를 만들었습니다. caret 패키지의 train 함수를 method=’nb’로 설정하여 10 fold CV naiveBayes 를 학습하였고 CV accuracy 는 아래와 같습니다. 최소 77%의 정확도에서 최대 92%의 정확도를 가지는 것으로 보아 validation accuracy 의 편차가 큰 편이라고 생각합니다. 평균 정확도는 약 83%입니다.

| 평균 정확도 | **82.891738%** |
|---|---|
| **validation set** | **accuracy** |
| fold01 | 80.769230% |
| fold02 | 84.615380% |
| fold03 | 76.923080% |
| fold04 | 88.461540% |
| fold05 | 77.777780% |
| fold06 | 88.888890% |
| fold07 | 76.923080% |
| fold08 | 80.769230% |
| fold09 | 92.307690% |
| fold10 | 81.481480% |

## (b) Logistic regression using 10 fold CV

Use all predictors and construct a logit model classifier. Report a 10-fold CV classification error.

(a)와 마찬가지로 caret 패키지의 train 함수를 사용하였고 method='glm', family='binomial'로 설정하여 10 fold CV logistic regression 을 학습하였고 CV accuracy 는 아래와 같습니다. 최소 65%의 정확도에서 최대 85%의 정확도를 가지는 것으로 보아 (a)의 naiveBayes 보다도 더 큰 validation accuracy 편차를 가집니다. 또한 평균 정확도는 약 77%입니다.

| 평균 정확도 | **77.150996%** |
|---|---|
| **validation set** | **accuracy** |
| fold01 | 73.076920% |
| fold02 | 73.076920% |
| fold03 | 73.076920% |
| fold04 | 84.615380% |
| fold05 | 74.074070% |
| fold06 | 81.481480% |
| fold07 | 65.384620% |
| fold08 | 80.769230% |
| fold09 | 80.769230% |
| fold10 | 85.185190% |

Validation set 에 대한 평균 정확도는 naiveBayes 가 약 83%로 logistic regression 의 약 77%보다 6%p 높은 성능을 보입니다. 주어진 데이터에 대해선 naiveBayes 가 더 잘 적합되는 모습을 보입니다. 하지만 testset 이 없어 test accuracy 를 구해보지 않았기 때문에 해당 데이터에 대해 어느 모델의 성능이 더 우수하다고 판단하기에는 무리가 있다고 생각합니다.

## 3. Contingency table

Suppose a categorical response Y can take on 1, 2, or 3. We have two categorical predictors X1 and X2. X1 has two levels, denoted by “A” and “B”. X2 has three levels, denoted by “a”, “b”, and “c”. A training data set consists of 50 observations as follow:

### (a) Bayes classifier

Construct Bayes classifier table for Y hat.

Predictor 인 X1, X2 모두 범주형 변수이기 때문에 nnet 패키지의 multinom 함수를 이용하여 Bayes classifier 모델을 학습할 수 있었습니다. Y 와 Yhat 의 교차표는 다음과 같습니다.

| | | Yhat | | |
|---|---|---|---|---|
| | | **1** | **2** | **3** |
| **Y** | **1** | 8 | 1 | 1 |
| | **2** | 3 | 15 | 2 |
| | **3** | 4 | 3 | 13 |

confusionMatrix 함수를 이용하여 train accuracy : 72%를 구할 수 있었습니다.

### (b) Misclassification rate using LOOCV

Calculate the LOOCV classification error rate for Yhat.

반복문을 이용하여 LOOCV 를 구현할 수 있었고 error_rate 벡터에 각 반복에 대해 정확히 예측했는지 여부를 저장하였습니다. Classification error rate 를 계산하는게 목적이었기 때문에 예측이 정확하였다면 0, 예측이 틀렸다면 1 로 기록했습니다. LOOCV 학습을 마친 후 mean(error_rate)로 오분류율을 계산했을 때 0.28 로, 28%의 오분류율을 보였습니다.

# Appendix : R codes

## 1. Bootstrap

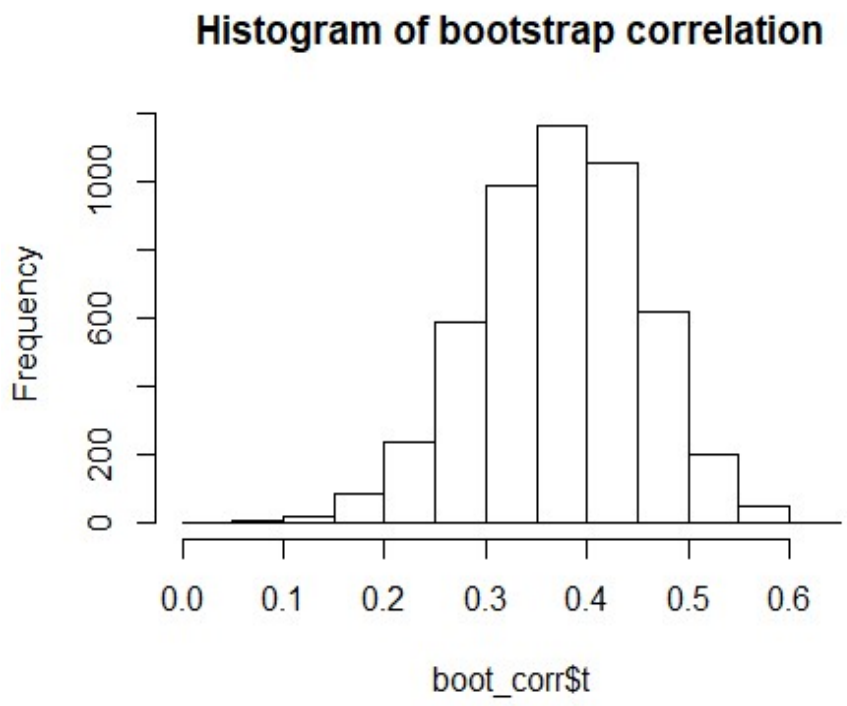
```
data1 = read.csv('./중간고사/MID2020Sdata.txt', header=TRUE, sep=' ')
student = 20152410
```

### (a) Bootstrap bias and standard error

```
library(boot)

my_cor = function(dataset, index){
  data = dataset[index, ]
  correlation = boot::corr(data)
  return(correlation)
}

set.seed(student)
boot_corr = boot(data1, statistic=my_cor, R=5000)
hist(boot_corr$t, main='Histogram of bootstrap correlation')
```



```
boot_corr
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data1, statistic = my_cor, R = 5000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.3731081 -0.001005584  0.08127118
```

## (b) Construct 95% bootstrap confidence intervals.

```
boot_corr_ci = boot.ci(boot_corr, conf=0.95)
```

```
## Warning in boot.ci(boot_corr, conf = 0.95): bootstrap variances needed for
## studentized intervals
```

```
boot_corr_ci
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_corr, conf = 0.95)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.2148,  0.5334 )   ( 0.2220,  0.5413 )
##
## Level     Percentile            BCa
## 95%   ( 0.2049,  0.5242 )   ( 0.2018,  0.5220 )
## Calculations and Intervals on Original Scale
```

## 2. Hitters data

```
library(ISLR)
data2 = na.omit(ISLR::Hitters)
```

## (a) Naive Bayes classifier using 10 fold CV

```
library(dplyr)
library(e1071)
library(caret)
library(klaR)


data2_bs = data2 %>%
  mutate(binary_salary = as.factor(ifelse(Salary >= 750, 1, 0))) %>%
  dplyr::select(-Salary)

set.seed(student)
fit_nb = train(form=binary_salary ~.,
               data=data2_bs,
               method='nb',
               trControl=trainControl(method='cv', number=10))
#model_naiveBayes = naiveBayes(binary_salary ~ ., data=data2_bs)

fit_nb$resample
```

```
##     Accuracy     Kappa Resample
## 1  0.8076923 0.5695364   Fold01
## 2  0.8461538 0.6413793   Fold02
## 3  0.7692308 0.5031847   Fold03
## 4  0.8846154 0.7417219   Fold04
## 5  0.7777778 0.4705882   Fold05
## 6  0.8888889 0.6966292   Fold06
## 7  0.7692308 0.4620690   Fold07
## 8  0.8076923 0.4881890   Fold08
## 9  0.9230769 0.8206897   Fold09
## 10 0.8148148 0.5768025   Fold10
```

```
mean(fit_nb$resample$Accuracy) # final CV accuracy of naiveBayes
```

```
## [1] 0.8289174
```

## (b) Logistic regression using 10 fold CV

```
set.seed(student)
fit_logit = train(form=binary_salary ~.,
                  data=data2_bs,
                  method='glm',
                  family='binomial',
                  trControl=trainControl(method='cv', number=10))

#model_logit = glm(binary_salary ~., data=data2_bs, family='binomial')

fit_logit$resample
```

```
##     Accuracy      Kappa Resample
## 1  0.7307692  0.2834646   Fold01
## 2  0.7307692  0.2834646   Fold02
## 3  0.7307692  0.2086957   Fold03
## 4  0.8461538  0.6090226   Fold04
## 5  0.7407407  0.2921348   Fold05
## 6  0.8148148  0.4398340   Fold06
## 7  0.6538462 -0.0173913   Fold07
## 8  0.8076923  0.4881890   Fold08
## 9  0.8076923  0.4347826   Fold09
## 10 0.8518519  0.6470588   Fold10
```

```
mean(fit_logit$resample$Accuracy)
```

```
## [1] 0.77151
```

## 3. Contingency table

```
Y = as.factor(c(rep(1, 10), rep(2, 20), rep(3, 20)))
X1_1 = c(rep('A', 4+1), rep('B', 1+4))
X1_2 = c(rep('A', 2+1+7), rep('B', 1+8+1))
X1_3 = c(rep('A', 2+4+2), rep('B', 9+1+2))
X1 = c(X1_1, X1_2, X1_3)
X2_1 = c(rep('a', 4), rep('b', 1), rep('b', 1), rep('c', 4))
X2_2 = c(rep('a', 2), rep('b', 1), rep('c', 7), rep('a', 1), rep('b', 8), rep('c',
1))
X2_3 = c(rep('a', 2), rep('b', 4), rep('c', 2), rep('a', 9), rep('b', 1), rep('c',
2))
X2 = c(X2_1, X2_2, X2_3)

data3 = data.frame('Y'=Y, 'X1'=X1, 'X2'=X2)
```

## (a) Bayes classifier

```
library(nnet)

model_multinom = multinom(Y ~ X1 * X2, data=data3)
```

```
## # weights:  21 (12 variable)
## initial  value 54.930614
## iter  10 value 36.095348
## iter  20 value 34.650147
## iter  30 value 34.622521
## final  value 34.622498
## converged
```

```
model_multinom
```

```
## Call:
## multinom(formula = Y ~ X1 * X2, data = data3)
##
## Coefficients:
##   (Intercept)      X1B       X2b      X2c    X1B:X2b   X1B:X2c
## 2  -0.6925762 7.720857 0.6918639 12.32132  -5.640256 -20.73594
## 3  -0.6928817 9.919240 2.0785024 11.06879 -11.304294 -20.98814
##
## Residual Deviance: 69.245
## AIC: 93.245
```

```r
Yhat = predict(model_multinom, data3[, c('X1', 'X2')])
confusionMatrix(table(data3[, 'Y'], Yhat))
```

```
## Confusion Matrix and Statistics
##
##    Yhat
##      1  2  3
##   1  8  1  1
##   2  3 15  2
##   3  4  3 13
##
## Overall Statistics
##
##                Accuracy : 0.72
##                  95% CI : (0.5751, 0.8377)
##     No Information Rate : 0.38
##     P-Value [Acc > NIR] : 1.119e-06
##
##                   Kappa : 0.5758
##
##  Mcnemar's Test P-Value : 0.3916
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.5333   0.7895   0.8125
## Specificity            0.9429   0.8387   0.7941
## Pos Pred Value         0.8000   0.7500   0.6500
## Neg Pred Value         0.8250   0.8667   0.9000
## Prevalence             0.3000   0.3800   0.3200
## Detection Rate         0.1600   0.3000   0.2600
## Detection Prevalence   0.2000   0.4000   0.4000
## Balanced Accuracy      0.7381   0.8141   0.8033
```

## (b) Misclassification rate using LOOCV

```r
error_rate = c()

for (i in 1:length(data3[, 'Y'])){
  model_multinomial = multinom(Y ~ X1 * X2, data=data3[-i, ])
  pred = predict(model_multinomial, newdata=data3[i, ])
  error_rate[i] = ifelse(data3[i, 'Y']==pred, 0, 1)
}

mean(error_rate)
```

```
## [1] 0.28
```