KANGWON NATIONAL UNIVERSITY · 강원대학교 ·

KNU

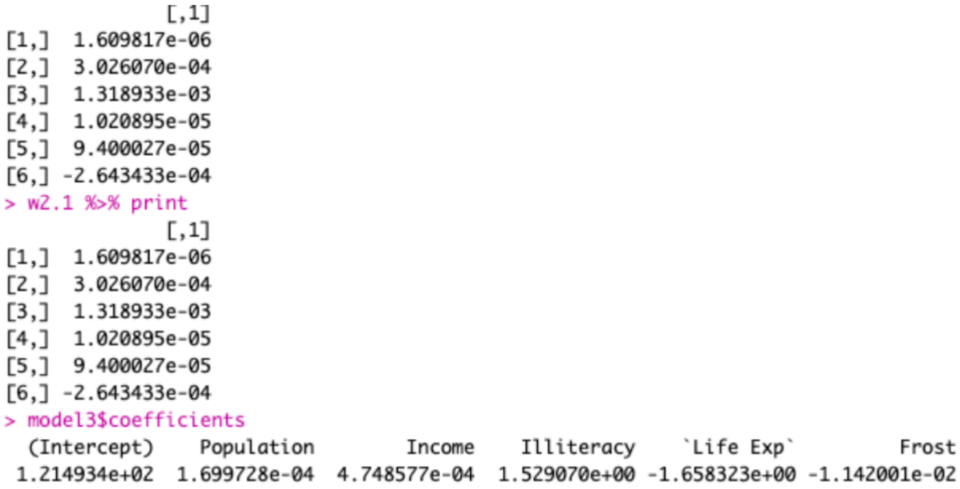
9주차 과제

과목	머신러닝을이용한재난설계
담당교수	이두호
학번	201720970
학과	소프트웨어·미디어·산업공학부
이름	권대한

201720970_권대한

1. Population, Income, Illiteracy, Life Exp, Frost 를 입력변수로 하고, Murder를 출력변수로 하여 경사하강법을 이용해 다중회귀분석을 실시하라. 단, 학습률과 종료조건은 본인이 정할 것!

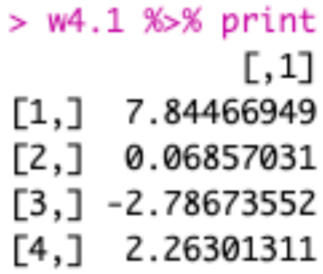
```
state <- as.data.frame(state.x77)

model3 <- lm(Murder ~ Population + Income + Illiteracy + `Life Exp` + Frost,data = state)

x2.1 <- cbind(1, state$Population,   state$Income, state$Illiteracy, state$`Life Exp`, state$Frost)

x2.1 <- as.matrix(x2.1)

y2.1 <- as.matrix(state$Murder)

n2 <- length(y2.1)

cost <- function(X, y, w)   (t((y - X%*%w)) %*% (y - X %*% w)) / (2*n2)

alpha <- 0.00000001

num_iters <- 500

cost_history.2 <- double(num_iters)

w_history.2 <- list(num_iters)

w2.1 <- matrix(c(0, 0, 0, 0, 0, 0), nrow = 6)

for (i in 1:num_iters) {

  grad <- (t(x2.1) %*% (x2.1 %*% w2.1 - y2.1)) / n2

  w2.1 <- w2.1 - alpha * grad

  w2.1 %>% print

  cost_history.2[i] <- cost(x2.1, y2.1, w2.1)

  w_history.2[[i]] <- w2.1

}

w2.1 %>% print

model3$coefficients
```



```
              [,1]
[1,]  1.609817e-06
[2,]  3.026070e-04
[3,]  1.318933e-03
[4,]  1.020895e-05
[5,]  9.400027e-05
[6,] -2.643433e-04
> w2.1 %>% print
              [,1]
[1,]  1.609817e-06
[2,]  3.026070e-04
[3,]  1.318933e-03
[4,]  1.020895e-05
[5,]  9.400027e-05
[6,] -2.643433e-04
> model3$coefficients
  (Intercept)    Population        Income    Illiteracy    `Life Exp`         Frost 
 1.214934e+02  1.699728e-04  4.748577e-04  1.529070e+00 -1.658323e+00 -1.142001e-02 
```

1. mpg, cyl, wt를 입력 변수로, vs 를 출력 변수로 하는 로지스틱 회귀 모델을 만들어라.

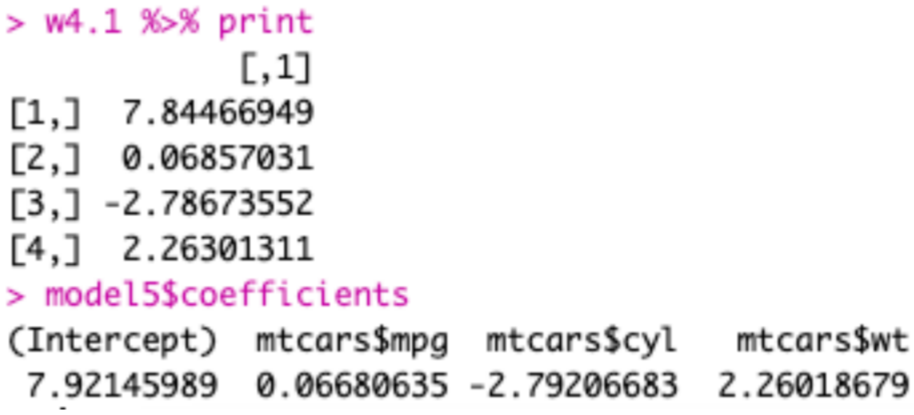
```
mtcars <- mtcars
x4.1 <- cbind(1, mtcars$mpg, mtcars$cyl, mtcars$wt) %>% as.matrix
y4.1 <- mtcars$vs %>% as.matrix
n4 <- length(y4.1)
# glm model
model5 <- glm(mtcars$vs ~ mtcars$mpg + mtcars$cyl + mtcars$wt, family = "binomial")
alpha <- 0.01
num_iters <- 100000
cost_history.4 <- double(num_iters)
w_history.4 <- list(num_iters)
w4.1 <- matrix(c(8, 1, -1, 3), nrow = 4)
for (i in 1:num_iters) {
  grad <- t(x4.1) %*% (1 / (1 + exp(-x4.1 %*% w4.1)) - y4.1) / n4
  w4.1 <- w4.1 - alpha * grad
  cost_history.4[i] <- cost(x4.1, y4.1, w4.1)
  w_history.4[[i]] <- w4.1
}
w4.1 %>% print
model5$coefficients
```



```
> w4.1 %>% print
            [,1]
[1,]  7.84466949
[2,]  0.06857031
[3,] -2.78673552
[4,]  2.26301311
```

2. glm함수를 이용하여 결과를 비교하라.

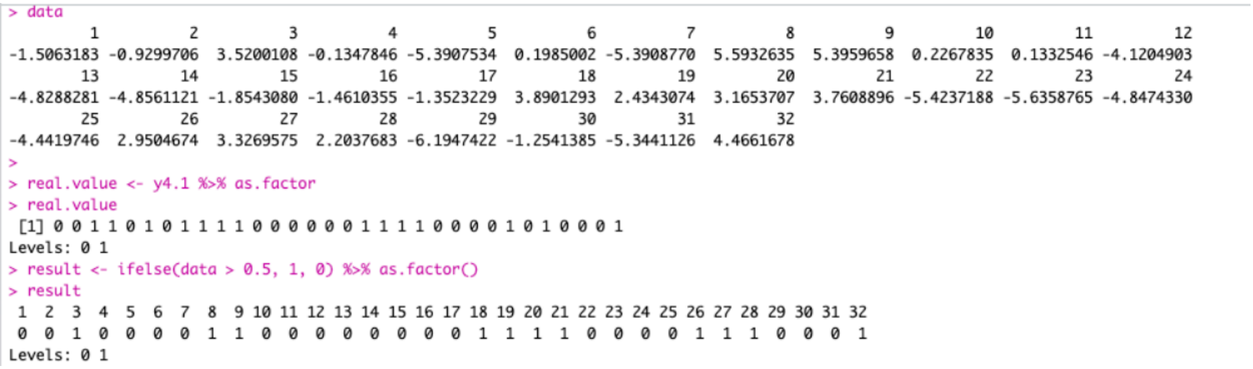
```
model5 <- glm(mtcars$vs ~ mtcars$mpg + mtcars$cyl + mtcars$wt, family = "binomial")

w4.1 %>% print

model5$coefficients
```



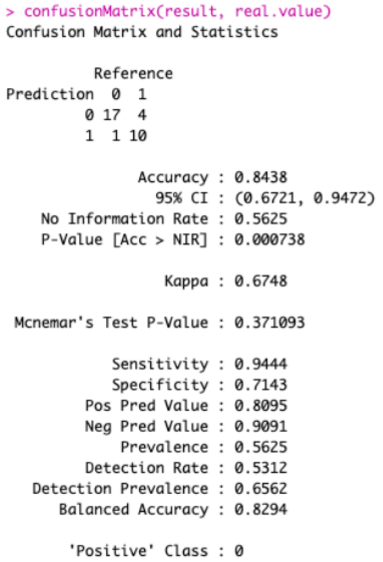
```
> w4.1 %>% print
            [,1]
[1,]  7.84466949
[2,]  0.06857031
[3,] -2.78673552
[4,]  2.26301311
> model5$coefficients
(Intercept)  mtcars$mpg  mtcars$cyl   mtcars$wt 
 7.92145989  0.06680635 -2.79206683  2.26018679 
```

3. Confusion matrix를 작성하고, 모델의 정확도를 계산하라.

```
data <- predict.glm(model5, newdata = x4.1 %>% as.data.frame())

real.value <- y4.1 %>% as.factor

result <- ifelse(data > 0.5, 1, 0) %>% as.factor()

confusionMatrix(result, real.value)
```



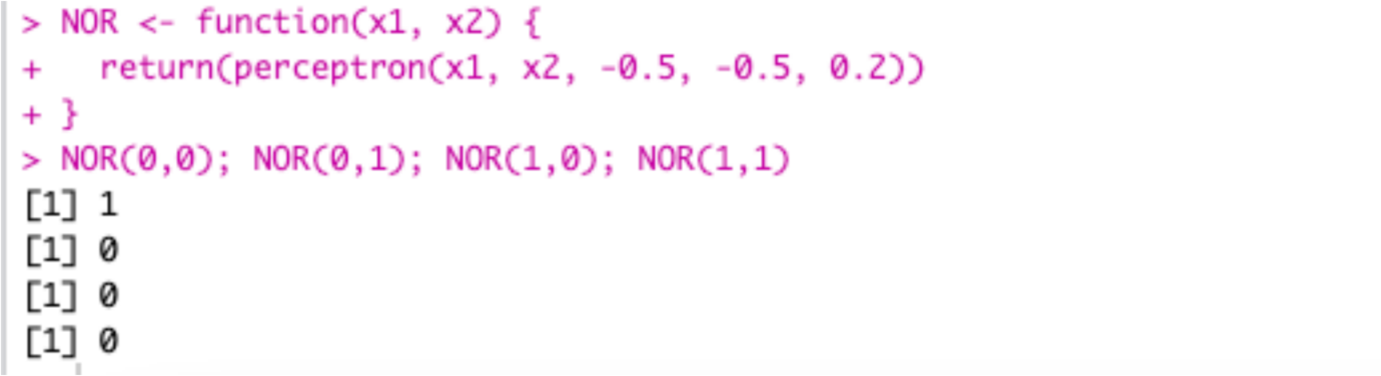
```
> data
         1          2          3          4          5          6          7          8          9         10         11         12 
-1.5063183 -0.9299706  3.5200108 -0.1347846 -5.3907534  0.1985002 -5.3908770  5.5932635  5.3959658  0.2267835  0.1332546 -4.1204903 
        13         14         15         16         17         18         19         20         21         22         23         24 
-4.8288281 -4.8561121 -1.8543080 -1.4610355 -1.3523229  3.8901293  2.4343074  3.1653707  3.7608896 -5.4237188 -5.6358765 -4.8474330 
        25         26         27         28         29         30         31         32 
-4.4419746  2.9504674  3.3269575  2.2037683 -6.1947422 -1.2541385 -5.3441126  4.4661678 
> 
> real.value <- y4.1 %>% as.factor
> real.value
 [1] 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1
Levels: 0 1
> result <- ifelse(data > 0.5, 1, 0) %>% as.factor()
> result
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 
 0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  0  0  1  1  1  1  0  0  0  0  1  1  1  0  0  0  1 
Levels: 0 1
```



```
> confusionMatrix(result, real.value)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 17  4
         1  1 10
                                          
               Accuracy : 0.8438          
                 95% CI : (0.6721, 0.9472)
    No Information Rate : 0.5625          
    P-Value [Acc > NIR] : 0.000738        
                                          
                  Kappa : 0.6748          
                                          
 Mcnemar's Test P-Value : 0.371093        
                                          
            Sensitivity : 0.9444          
            Specificity : 0.7143          
         Pos Pred Value : 0.8095          
         Neg Pred Value : 0.9091          
             Prevalence : 0.5625          
         Detection Rate : 0.5312          
   Detection Prevalence : 0.6562          
      Balanced Accuracy : 0.8294          
                                          
       'Positive' Class : 0               
```

1. NOR operation 퍼셉트론을 구현하시오.

```
perceptron <- function(x1, x2, w1, w2, b) {
  if(w1*x1 + w2*x2 + b <= 0) {
    return(0)
  } else {
    return(1)
  }
}
NOR <- function(x1, x2) {
  return(perceptron(x1, x2, -0.5, -0.5, 0.2))
}
NOR(0,0); NOR(0,1); NOR(1,0); NOR(1,1)
```



```
> NOR <- function(x1, x2) {
+   return(perceptron(x1, x2, -0.5, -0.5, 0.2))
+ }
> NOR(0,0); NOR(0,1); NOR(1,0); NOR(1,1)
[1] 1
[1] 0
[1] 0
[1] 0
```