CRM Report

Hyeonho Lee

2018년 8월 25일

Load Data

```
library(dplyr)
mailorder = read.csv('D:/second semester/crm/mailorder.csv')
```

패키지 실행과 데이터 로드

Question 1.

2000개의 estimation sample과 validation sample로 분리

```
mailorder1 = head(mailorder, 2000)
mailorder2 = tail(mailorder, 2000)
```

```
set.seed(1234)
ind = sample(1:nrow(mailorder2), nrow(mailorder2), replace = FALSE)
mailorder2_1 = mailorder2[ind <= 500,]

paste0('500명 중 실제로 구매한 인원은 ', sum(mailorder2_1$purchase == 1), '명')
```

```
## [1] "500명 중 실제로 구매한 인원은 40명"
```

Question 2.

2X2X2 RFM codes 에 따라 estimation and validation sample을 분류

```
mailorder1$R = cut(mailorder1$recency, breaks = c(min(mailorder1$recency), 12,
                   max(mailorder1$recency)),include.lowest = T, labels = c(2,1))
mailorder1$F = cut(mailorder1$frequency, breaks = c(min(mailorder1$frequency)-1, 2,
                   max(mailorder1$frequency)),rigth = T, labels = c(1, 2))
mailorder1$M = cut(mailorder1$monetary, breaks = c(min(mailorder1$monetary)-1, 208,
                   max(mailorder1$monetary)),rigth = T, labels = c(1, 2))
mailorder2$R = cut(mailorder2$recency, breaks = c(min(mailorder2$recency), 12,
                   max(mailorder2$recency)),include.lowest = T, labels = c(2,1))
mailorder2$F = cut(mailorder2$frequency, breaks = c(min(mailorder2$frequency)-1, 2,
                   max(mailorder2$frequency)),rigth = T, labels = c(1, 2))
mailorder2$M = cut(mailorder2$monetary, breaks = c(min(mailorder2$monetary)-1, 208,
                   max(mailorder2$monetary)),rigth = T, labels = c(1, 2))
```

분류하는 과정에서 factor형 변수로 변환 되었기 때문에 factor를 풀어주는 코드

```
mailorder1$R = varhandle::unfactor(mailorder1$R)
mailorder1$F = varhandle::unfactor(mailorder1$F)
mailorder1$M = varhandle::unfactor(mailorder1$M)
mailorder2$R = varhandle::unfactor(mailorder2$R)
mailorder2$F = varhandle::unfactor(mailorder2$F)
mailorder2$M = varhandle::unfactor(mailorder2$M)
```

mailorder1 즉 R,F,M을 기준으로 estimation sample의 평균 mean_purchase 계산

```
mean_pur = mailorder1 %>% select(R, F, M, purchase) %>% group_by(R, F, M) %>%
  summarise(mean_purchase = mean(purchase))%>% arrange(desc(mean_purchase))
```

Table 1: mean_pur

R	F	M	mean_purchase
2	2	1	0.1751825
2	2	2	0.1672131
2	1	2	0.0891473
2	1	1	0.0636132
1	2	2	0.0611354
1	2	1	0.0504202
1	1	2	0.0465116
1	1	1	0.0290698

mailorder2(validation sample에 R,F,M 부여)의 R,F,M이 mailorder1의 R,F,M과 같을 때, mailorder2에 mean_purchase를 추가

```
for( i in 1:nrow(mailorder2))
{
  for( j in 1:nrow(mean_pur))
  {
    if(mean_pur$R[j] == mailorder2$R[i] &
       mean_pur$F[j] == mailorder2$F[i] &
       mean_pur$M[j] == mailorder2$M[i])
    {
      mailorder2$mean_purchase[i] = mean_pur$mean_purchase[j]
    }
  }
}
```

mean_purchase가 추가된 mailorder2를 purchase와 mean_purchase만 선택하고, mean_purchase를 기준으로 내림차순한 상위 500개의 purchase확률

```
mailorder2 %>% select(purchase, mean_purchase) %>%
  arrange(desc(mean_purchase)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##    mean
## 1 0.152
```

```
paste0('500명 중 실제로 구매한 인원은 ', mailorder2 %>% select(purchase, mean_purchase) %>%
  arrange(desc(mean_purchase)) %>% head(500) %>% filter(purchase == 1) %>% nrow(), '명')
```

```
## [1] "500명 중 실제로 구매한 인원은 76명"
```

Question 1의 방법에 비해 22명이 증가하였다. 무작위로 선택하는 방법보다 estimation sample의 다른 변수를 사용한 Q2의 방법이 효율적이라는 것을 알 수 있다. Q2는 3개의 변수를 구간을 나눠 그룹을 만들었다. 그룹 간 구매확률이 높은 상위 500명을 validation sample에서 선택하면, marketing target을 선정하는데 있어 효율적임을 알 수 있다.

Question 3.

5X5X5 RFM codes 에 따라 estimation and validation sample을 분류

```
mailorder1$R = cut(mailorder1$recency, breaks = c(min(mailorder1$recency),
                    4, 8, 12, 16, max(mailorder1$recency)),include.lowest = T,
                    labels = c(5,4,3,2,1))
mailorder1$F = cut(mailorder1$frequency, breaks = c(min(mailorder1$frequency)-1,
                    1, 2, 5, 9, max(mailorder1$frequency)),rigth = T,
                    labels = c(1,2,3,4,5))
mailorder1$M = cut(mailorder1$monetary, breaks = c(min(mailorder1$monetary)-1,
                    113, 181, 242, 299, max(mailorder1$monetary)),rigth = T,
                    labels = c(1,2,3,4,5))
mailorder2$R = cut(mailorder2$recency, breaks = c(min(mailorder2$recency),
                    4, 8, 12, 16, max(mailorder2$recency)),include.lowest = T,
                    labels = c(5,4,3,2,1))
mailorder2$F = cut(mailorder2$frequency, breaks = c(min(mailorder2$frequency)-1,
                    1, 2, 5, 9, max(mailorder2$frequency)),rigth = T,
                    labels = c(1,2,3,4,5))
mailorder2$M = cut(mailorder2$monetary, breaks = c(min(mailorder2$monetary)-1,
                    113, 181, 242, 299, max(mailorder2$monetary)),rigth = T,
                    labels = c(1,2,3,4,5))
```

분류하는 과정에서 factor형 변수로 변환 되었기 때문에 factor를 풀어주는 코드

```
mailorder1$R = varhandle::unfactor(mailorder1$R)
mailorder1$F = varhandle::unfactor(mailorder1$F)
mailorder1$M = varhandle::unfactor(mailorder1$M)
mailorder2$R = varhandle::unfactor(mailorder2$R)
mailorder2$F = varhandle::unfactor(mailorder2$F)
mailorder2$M = varhandle::unfactor(mailorder2$M)
```

mailorder1 즉 R,F,M을 기준으로 estimation sample의 평균 mean_purchase의 계산

```
mean_pur = mailorder1 %>% select(R, F, M, purchase) %>% group_by(R, F, M) %>%
  summarise(mean_purchase = mean(purchase))%>% arrange(desc(mean_purchase))
```

mean_pur의 상위 5, 하위 5 총 10개의 mean_purchase

Table 2: top 5 of mean_pur

R	F	M	mean_purchase
4	5	4	0.6
4	1	5	0.5
4	4	2	0.5
4	5	2	0.5
5	5	2	0.4

Table 3: bottom 5 of mean_pur

R	F	M	mean_purchase
5	3	4	0
5	4	1	0
5	4	2	0
5	4	4	0
5	5	4	0

mailorder2(validation sample에 R,F,M 부여)의 R,F,M이 mailorder1의 R,F,M과 같을 때, mailorder2에 mean_purchase를 추가

```
for( i in 1:nrow(mailorder2))
{
  for( j in 1:nrow(mean_pur))
  {
    if(mean_pur$R[j] == mailorder2$R[i] &
       mean_pur$F[j] == mailorder2$F[i] &
       mean_pur$M[j] == mailorder2$M[i])
    {
      mailorder2$mean_purchase[i] = mean_pur$mean_purchase[j]
    }
  }
}
```

mean_purchase가 추가된 mailorder2를 purchase와 mean_purchase만 선택하고, mean_purchase를 기준으로 내림차순하여 한 상위 500개의 purchase확률

```
mailorder2 %>% select(purchase, mean_purchase) %>%
  arrange(desc(mean_purchase)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##    mean
## 1 0.124
```

```
paste0('500명 중 실제로 구매한 인원은 ', mailorder2 %>% select(purchase, mean_purchase) %>%
  arrange(desc(mean_purchase)) %>% head(500) %>% filter(purchase == 1) %>% nrow(), '명')
```

```
## [1] "500명 중 실제로 구매한 인원은 62명"
```

Question 2(Q2)의 방법에 비해 14명이 감소하였다. Question 3은 Q2에 비해 구간의 폭을 좁혀서 더 많은 그룹을 만들었다. 더 많은 그룹이 생겼기 때문에, marketing target의 인원이 적어져 비용은 낮아지나, 구매확률이 높다고 판단한 상위 500명 중 실제로 구매했던 인원은 감소하였다. 즉, 적절한 기준으로 구간을 나누는 것이 효율적임을 알 수 있다.

Question 4.

mailorder1의 purchase를 반응변수로, recency, frequency, monetary를 설명변수로 만든 회귀식

```
model = lm(purchase~recency+frequency+monetary, mailorder1)
```

Table 4: Coefficients of model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0921429	0.0170722	5.397253	0.0000001
recency	-0.0043872	0.0007437	-5.899205	0.0000000
frequency	0.0087193	0.0019768	4.410772	0.0000108
monetary	0.0000650	0.0000689	0.944382	0.3450888

회귀식을 토대로 mailorder2(validation)의 purchase를 예측

```
pred = predict(model, mailorder2)
```

mailorder2에 예측값 할당

```
mailorder2$predict = pred
```

Table 5: mailorder2

	purchase	recency	frequency	monetary	predict
2001	0	16	1	298	0.0500453
2002	0	10	2	232	0.0807961
2003	0	20	4	220	0.0535820
2004	1	10	1	223	0.0714915
2005	0	6	1	279	0.0926822
2006	0	10	7	273	0.1270591

predict를 기준으로 내림차순한 상위 500개의 purchase 평균값

```
mailorder2 %>% select(purchase, recency, frequency, monetary, predict) %>%
  arrange(desc(predict)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##   mean
## 1 0.16
```

```
paste0('500명 중 실제로 구매한 인원은 ', mailorder2 %>% select(purchase, predict) %>%
  arrange(desc(predict)) %>% head(500) %>% filter(purchase == 1) %>% nrow(), '명')
```

```
## [1] "500명 중 실제로 구매한 인원은 80명"
```

위의 방법 중 실제로 구매한 인원을 가장 잘 맞추었다. 구간을 나누는 방법도 좋은 방법이지만, 각각 변수의 변동으로 구매여부의 변동을 예측하는 경우에 구매여부를 더 정확하게 예측할 수 있음을 보인다.

Question 5.

변수 선택

mailorder1의 purchase를 반응변수하고 남은 모든 변수로 만든 회귀모형

```
model = lm(purchase~., mailorder1)
```

Table 6: Coefficients of model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1507787	0.1162289	1.297256	0.1946932
id	-0.0000070	0.0000104	-0.670004	0.5029329
genderM	0.0504923	0.0132971	3.797248	0.0001507
monetary	0.0000791	0.0000688	1.150009	0.2502782
recency	-0.0034143	0.0012265	-2.783869	0.0054222
frequency	0.0133079	0.0047998	2.772602	0.0056127
duration	-0.0011409	0.0009982	-1.142927	0.2532060

단계적 회귀를 사용한 변수선택

```
model = step(model, direction = 'both')
```

Table 7: Coefficients of model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0900783	0.0136960	6.576984	0.0000000
genderM	0.0502545	0.0132588	3.790284	0.0001549
recency	-0.0045233	0.0007420	-6.095975	0.0000000
frequency	0.0094932	0.0016994	5.586106	0.0000000

구매여부 예측과 할당

```
pred = predict(model, mailorder2)
mailorder2$predict = pred
```

상위 500명의 실제 구매확률과 실제 구매 인원

```
mailorder2 %>% arrange(desc(predict)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##    mean
## 1 0.182
```

```
paste0(mailorder2 %>% select(purchase, predict) %>% arrange(desc(predict)) %>%
         head(500) %>% filter(purchase == 1) %>% nrow(), '명')
```

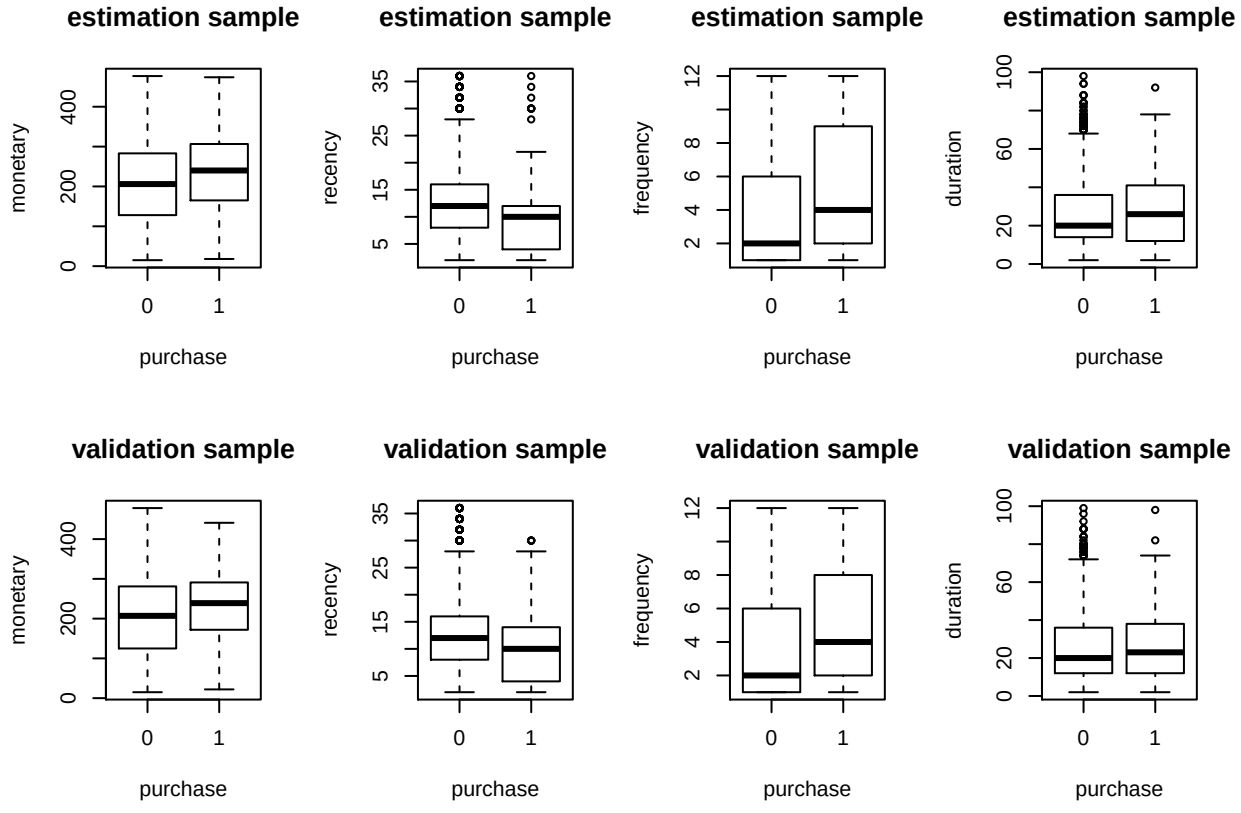
```
## [1] "91명"
```

임의로 변수를 선택하지 않고, 단계적 회귀알고리즘을 사용하여 AIC를 낮추는 변수를 선택하면, 더 좋은 회귀모델을 만들 수 있음을 알 수 있다.

구매여부에 대한 주요변수 4개의 boxplot

```
par(mfrow = c(2, 4))
boxplot(monetary~purchase, mailorder1, xlab = 'purchase', ylab = 'monetary',
        main = 'estimation sample')
boxplot(recency~purchase, mailorder1, xlab = 'purchase', ylab = 'recency',
        main = 'estimation sample')
boxplot(frequency~purchase, mailorder1, xlab = 'purchase', ylab = 'frequency',
        main = 'estimation sample')
boxplot(duration~purchase, mailorder1, xlab = 'purchase', ylab = 'duration',
        main = 'estimation sample')

boxplot(monetary~purchase, mailorder2, xlab = 'purchase', ylab = 'monetary',
        main = 'validation sample')
boxplot(recency~purchase, mailorder2, xlab = 'purchase', ylab = 'recency',
        main = 'validation sample')
boxplot(frequency~purchase, mailorder2, xlab = 'purchase', ylab = 'frequency',
        main = 'validation sample')
boxplot(duration~purchase, mailorder2, xlab = 'purchase', ylab = 'duration',
        main = 'validation sample')
```



duration의 경우 구매여부에 따라 변화가 boxplot의 차이가 거의 없으므로 유의하지 않을 가능성이 크다고 판단한다.

차이가 있는 3개의 변수와 성별을 넣고, poly함수를 사용하여 변수를 제곱한 것을 추가로 넣어준다.

```
model = lm(purchase ~ gender + poly(monetary,2) + poly(recency,2) + poly(frequency,2),
           mailorder1)
model = step(model, direction = 'both')
```

Table 8: Coefficients of model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0671052	0.0071079	9.4408901	0.0000000
genderM	0.0498091	0.0132362	3.7630922	0.0001727
poly(recency, 2)1	-1.6391300	0.2682582	-6.1102689	0.0000000
poly(recency, 2)2	0.8014183	0.2681568	2.9886183	0.0028367
poly(frequency, 2)1	1.4725484	0.2681778	5.4909401	0.0000000
poly(frequency, 2)2	0.0839266	0.2680194	0.3131363	0.7542098

Coefficients를 보고 판단하기에 poly함수는 recency만 적용하는 것이 효율적이라고 판단

구매여부 예측과 할당

```
pred = predict(model, mailorder2)
mailorder2$predict = pred
mailorder2 %>% arrange(desc(predict)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##    mean
## 1 0.186
```

이 모델의 경우 18.6%의 정확도를 보임

차이가 있는 3개의 변수와 성별을 넣고, poly함수를 사용하여 변수를 제곱한 것을 추가로 넣어준다.

```
model = lm(purchase ~ gender + monetary + poly(recency,2) + frequency, mailorder1)
```

Table 9: Coefficients of model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0192378	0.0144503	1.331308	0.1832399
genderM	0.0507051	0.0132576	3.824600	0.0001350
monetary	0.0000741	0.0000687	1.079357	0.2805592
poly(recency, 2)1	-1.6371864	0.2681539	-6.105398	0.0000000
poly(recency, 2)2	0.7931490	0.2681839	2.957482	0.0031381
frequency	0.0082417	0.0019692	4.185326	0.0000297

```
pred = predict(model, mailorder2)
mailorder2$predict = pred
mailorder2 %>% arrange(desc(predict)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##   mean
## 1 0.19
```

이 모델의 경우 19%의 정확도를 보인다.

다른 분류모형

로지스틱 회귀모형

```
model = glm(purchase ~ gender + monetary + poly(recency,2) + frequency, mailorder1,
            family = 'binomial')
pred = predict(model, mailorder2, type = 'response')
mailorder2$predict = pred
result1 = mailorder2 %>% arrange(desc(predict)) %>% head(500) %>%
  summarise(mean = mean(purchase))
```

랜덤포레스트 분류모형(randomForest의 경우 poly함수 사용불가)

```
set.seed(1234)
model = randomForest::randomForest(purchase ~ gender + monetary + recency + frequency,
                                   mailorder1, ntree = 200)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
pred = predict(model, mailorder2)
mailorder2$predict = pred
result2 = mailorder2 %>% arrange(desc(predict)) %>% head(500) %>%
  summarise(mean = mean(purchase))
```

svm모형

```
model = e1071::svm(purchase ~ gender + monetary + poly(recency,2) + frequency,
                   mailorder1, kernel = 'linear', cost = 9)
pred = predict(model, mailorder2)
mailorder2$predict = pred
result3 = mailorder2 %>% arrange(desc(predict)) %>% head(500) %>%
  summarise(mean = mean(purchase))
```

naiveBayes(poly 사용불가)

```
model = e1071::naiveBayes(as.factor(purchase) ~ gender + monetary + recency + frequency,
                          mailorder1)
pred = predict(model, mailorder2)
mailorder2$predict = varhandle::unfactor(pred)
result4 = mailorder2 %>% arrange(desc(predict)) %>% head(500) %>%
  summarise(mean = mean(purchase))
```

각각의 예측확률 TABLE

```
result = rbind(result1, result2, result3, result4)
result = cbind(c('로지스틱회귀', '랜덤포레스트', 'SVM', 'NaiveBayes'), result)
colnames(result) = c('Model', 'Probability')
```

Table 10: A caption

Model	Probability
로지스틱회귀	0.192
랜덤포레스트	0.168
SVM	0.162
NaiveBayes	0.088

로지스틱 회귀모형이 19.2%로 가장 높은 확률을 보인다. 500명 중 96명이 구매했다는 점을 알 수 있다.

성능이 더 좋은 모델을 얻기 위해 데이터를 더 볼 필요성을 느꼈다.

estimation sample와 validation sampled의 purchase횟수

Table 11: purchase of mailorder1

Var1	Freq
0	1837
1	163

Table 12: purchase of mailorder2

Var1	Freq
0	1838
1	162

purchase이 매우 불균형 인 것을 알 수 있다. 이런 불균형 데이터의 경우 처리하는 방법이 여러가지가 있는데, 그 중 한가지인 Random Over-Sampling을 적용하였다.

purchase을 0과 1인 경우로 나누어 1의 데이터를 2배 over-sampling하여 mailorder3로 bind

```
mailorder1_1 = mailorder1 %>% filter(purchase == 0)
mailorder1_2 = mailorder1 %>% filter(purchase == 1)

mailorder3 = rbind(mailorder1_1,mailorder1_2)
mailorder3 = rbind(mailorder3,mailorder1_2)
```

Table 13: purchase of mailorder3

Var1	Freq
0	1837
1	326

동일한 결과를 보기위한 seed번호를 설정하고, 비복원 추출로 resampling을 실행

```
set.seed(488983)
ind = sample(1:nrow(mailorder3), nrow(mailorder3), replace = F)

mailorder4 = mailorder3[ind <= 721,]
```

```
knitr::kable(table(mailorder4$purchase), caption = 'purchase of mailorder4')
```

Table 14: purchase of mailorder4

Var1	Freq
0	609
1	112

```r
model = lm(purchase ~ gender + monetary + recency + frequency, mailorder4)
pred = predict(model, mailorder2)
mailorder2$predict = pred
mailorder2 %>% arrange(desc(predict)) %>% head(500) %>% summarise(mean = mean(purchase))
```

```
##   mean
## 1  0.2
```

resampling한 mailorder4로 회귀분석을 진행한 결과 20%의 정확도를 볼 수 있었다. 즉 500명 중 100명이 실제로 구매를 한 이력이 있다는 점을 알 수 있다.

이 점을 보아 불균형한 데이터의 경우 높은 예측 모형을 만들기 위해서는 모델선택보다 추가데이터를 수집하거나, 기존의 데이터의 가공에 노력을 기울이면 좋은 성능을 기대할 수 있음을 알 수있다. 그러나 교차검증을 하지 않았기 때문에 과적합을 걱정해야 한다. 결론은 4000개의 데이터를 2000:2000으로 나누어 모델을 검정하는 것보다 전체 데이터에 대해서 k-fold 교차검증을 사용하여 모델링을 한다면 더 좋은 모델을 만들 수 있을 것이라고 생각한다.

회귀스플라인과 스무딩스플라인을 혼합해서 사용하는 경우, 19.8%라는 높은 예측력을 가진 모델이 생성되나, 마찬가지로 과적합에 대한 여부, 해석의 어려움으로 인해 다루지 않았다.