

# CRM Report

Hyeonho Lee

2018년 8월 25일

## Load Data

```
# library(dplyr)
# 데이터 조작에 필요한 패키지
mailorder = read.csv('D:/second semester/crm/mailorder.csv')
# load data
```

## Question 1.

```
mailorder1 = head(mailorder, 2000)
mailorder2 = tail(mailorder, 2000)
# 2000개의 estimation sample과 validation sample로 분리

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.

```
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)), right = T, labels = c(1, 2))
mailorder1$M = cut(mailorder1$monetary, breaks = c(min(mailorder1$monetary)-1, 208,
max(mailorder1$monetary)), right = 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)), right = T, labels = c(1, 2))
mailorder2$M = cut(mailorder2$monetary, breaks = c(min(mailorder2$monetary)-1, 208,
max(mailorder2$monetary)), right = T, labels = c(1, 2))
# 2X2X2 RFM codes 에 따라 estimation and validation sample을 분류

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)
# 분류하는 과정에서 factor형 변수로 변환 되었기 때문에 factor를 풀어주는 코드

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

knitr::kable(mean_pur, caption = 'A caption')

```

Table 1: A caption

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

```

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

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]
    }
  }
}

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

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

##      mean
## 1 0.152

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

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.

```
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)),right = 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)),right = 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)),right = 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)),right = T,
  labels = c(1,2,3,4,5))
# 5X5X5 RFM codes 에 따라 estimation and validation sample을 분류

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)
# 분류하는 과정에서 factor형 변수로 변환 되었기 때문에 factor를 풀어주는 코드

mean_pur = mailorder1 %>% select(R, F, M, purchase) %>% group_by(R, F, M) %>%
  summarise(mean_purchase = mean(purchase))%>% arrange(desc(mean_purchase))
# mailorder1 즉 R,F,M을 기준으로 estimation sample의 평균 mean_purchase의 계산

knitr::kable(head(mean_pur,5), caption = 'A caption')
```

Table 2: A caption

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

```
knitr::kable(tail(mean_pur,5), caption = 'A caption')
```

Table 3: A caption

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

```
# mean_pur의 상위 5, 하위 5 총 10개의 mean_purchase

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]
    }
  }
}

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

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

##      mean
## 1 0.124

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

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.

```
model = lm(purchase~recency+frequency+monetary, mailorder1)
# mailorder1의 purchase를 반응변수로, recency, frequency, monetary를 설명변수로 만든 회귀식
```

```
knitr::kable(summary(model)$coeff, caption = 'A caption')
```

Table 4: A caption

	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

```
# model의 추정치, 표준오차, t value, p-value
```

```
pred = predict(model, mailorder2)
# 회귀식을 토대로 mailorder2(validation)의 purchase를 예측
```

```
mailorder2$predict = pred
# mailorder2에 예측값 할당
```

```
knitr::kable(mailorder2 %>% select(purchase, recency, frequency, monetary, predict)
              %>% head(), caption = 'A caption')
```

Table 5: A caption

	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

```
# mailorder2의 5X5
```

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

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

```
# predict를 기준으로 내림차순한 상위 500개의 purchase 평균값
```

```
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.

```
mailorder = read.csv('D:/second semester/crm/mailorder.csv')
mailorder1 = head(mailorder, 2000)
```

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

Question 5.

```
# 변수 선택
model = lm(purchase~., mailorder1)
# mailorder1의 purchase를 제외한 모든 변수로 만든 회귀모형

knitr::kable(summary(model)$coef, caption = 'A caption')
```

Table 6: A caption

	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

```
# 회귀모형 Coefficients
```

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

```
## Start:  AIC=-5254.42
## purchase ~ id + gender + monetary + recency + frequency + duration
##
##           Df Sum of Sq  RSS    AIC
## - id       1   0.03233 143.58 -5256.0
## - duration  1   0.09409 143.65 -5255.1
## - monetary  1   0.09526 143.65 -5255.1
## <none>                        143.55 -5254.4
## - frequency 1   0.55370 144.10 -5248.7
## - recency   1   0.55821 144.11 -5248.7
## - gender    1   1.03857 144.59 -5242.0
##
## Step:  AIC=-5255.97
## purchase ~ gender + monetary + recency + frequency + duration
##
##           Df Sum of Sq  RSS    AIC
## - duration  1   0.09646 143.68 -5256.6
## - monetary  1   0.09681 143.68 -5256.6
## <none>                        143.58 -5256.0
## + id       1   0.03233 143.55 -5254.4
## - recency   1   0.55134 144.13 -5250.3
## - frequency  1   0.55922 144.14 -5250.2
## - gender    1   1.03898 144.62 -5243.5
##
## Step:  AIC=-5256.63
## purchase ~ gender + monetary + recency + frequency
```

```
##
##           Df Sum of Sq   RSS   AIC
## - monetary    1    0.10048 143.78 -5257.2
## <none>                143.68 -5256.6
## + duration    1    0.09646 143.58 -5256.0
## + id           1    0.03470 143.65 -5255.1
## - gender       1    1.07067 144.75 -5243.8
## - frequency    1    1.27754 144.96 -5240.9
## - recency      1    2.67461 146.35 -5221.7
##
## Step:   AIC=-5257.23
## purchase ~ gender + recency + frequency
##
##           Df Sum of Sq   RSS   AIC
## <none>                143.78 -5257.2
## + monetary    1    0.10048 143.68 -5256.6
## + duration    1    0.10012 143.68 -5256.6
## + id           1    0.03639 143.74 -5255.7
## - gender       1    1.03486 144.81 -5244.9
## - frequency    1    2.24780 146.03 -5228.2
## - recency      1    2.67686 146.46 -5222.3
```

# 단계적 회귀를 사용한 변수선택

```
knitr::kable(summary(model)$coef, caption = 'A caption')
```

Table 7: A caption

	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

# 단계적 회귀를 사용한 회귀모형의 *Coefficients*

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

```
mailorder2$predict = pred
```

# 구매여부 예측과 할당

```
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')
```

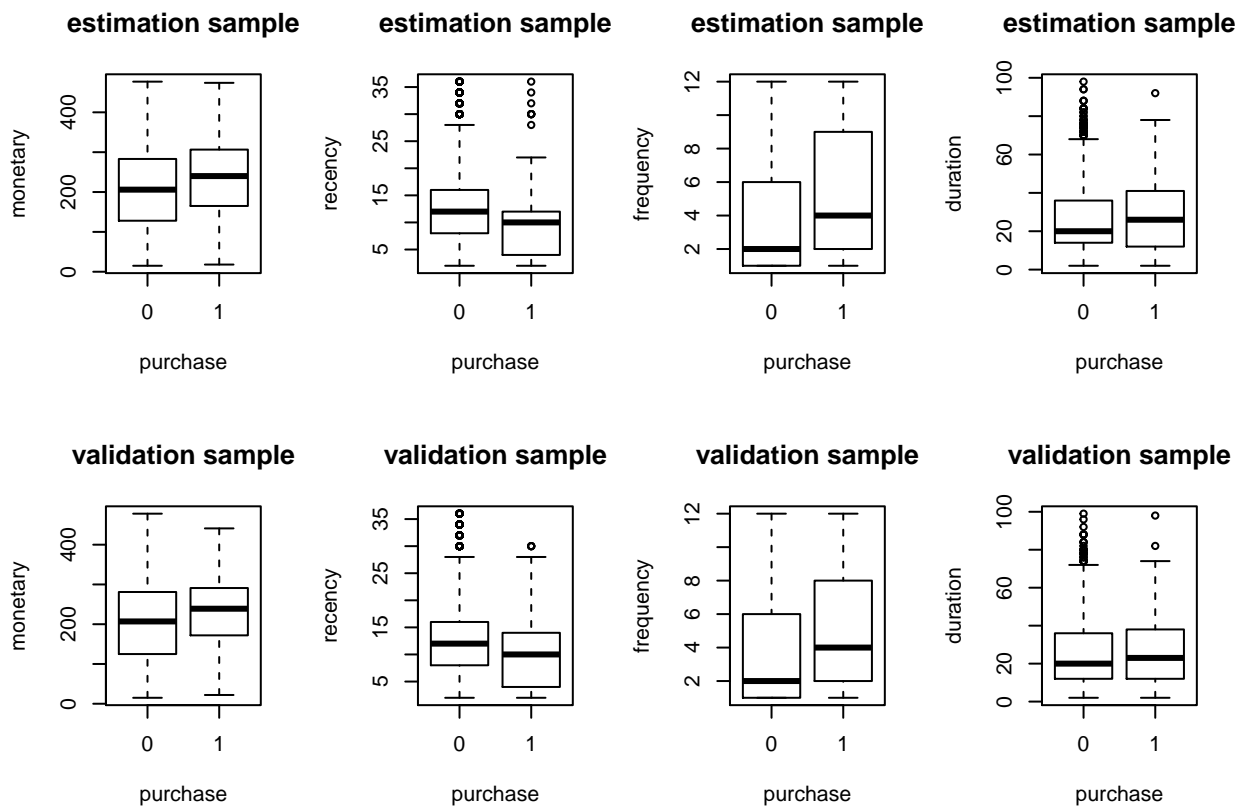
```
## [1] "91명"
```

# 상위 500명의 실제 구매확률과 실제 구매 인원

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.

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

model = lm(purchase ~ gender + poly(monetary,2) + poly(recency,2) + poly(frequency,2), mailorder1)
# 차이가 있는 3개의 변수와 성별을 넣고, poly함수를 사용하여 변수를 제곱한 것을 추가로 넣어준다.
model = step(model, direction = 'both')

## Start: AIC=-5261.3
## purchase ~ gender + poly(monetary, 2) + poly(recency, 2) + poly(frequency,
## 2)
##
##              Df Sum of Sq  RSS   AIC
## - poly(monetary, 2)    2    0.2140 143.13 -5262.3
```



```
## <none> 142.91 -5261.3
## - gender 1 1.0608 143.98 -5248.5
## - poly(frequency, 2) 2 1.3879 144.30 -5246.0
## - poly(recency, 2) 2 3.2862 146.20 -5219.8
##
## Step: AIC=-5262.31
## purchase ~ gender + poly(recency, 2) + poly(frequency, 2)
##
## Df Sum of Sq RSS AIC
## <none> 143.13 -5262.3
## + poly(monetary, 2) 2 0.2140 142.91 -5261.3
## - gender 1 1.0165 144.15 -5250.2
## - poly(frequency, 2) 2 2.1711 145.30 -5236.2
## - poly(recency, 2) 2 3.3226 146.45 -5220.4
knitr::kable(summary(model)$coef, caption = 'A caption')
```

Table 8: A caption

	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

# 단계별 회귀를 사용한 후 *Coefficients*

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

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

coef를 보고 판단하기에 poly함수는 recency만 적용하는 것이 효율적이라고 판단

또한 이 모델의 경우 18.6%의 정확도를 보임

```
model = lm(purchase ~ gender + monetary + poly(recency,2) + frequency, mailorder1)
# 차이가 있는 3개의 변수와 성별을 넣고, poly함수를 사용하여 변수를 제공한 것을 추가로 넣어준다.
knitr::kable(summary(model)$coef, caption = 'A caption')
```

Table 9: A caption

	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

```
# Coefficients
```

```
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

```
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))
# 로지스틱 회귀모형
```

```
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))
# 랜덤포레스트 분류모형 (randomForest의 경우 poly함수 사용불가)
```

```
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))
# svm모형
```

```
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))
# naiveBayes(poly 사용불가)
```

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

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.

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

```
table(mailorder1$purchase)
```

```
##
##      0      1
## 1837  163
```

```
table(mailorder2$purchase)
```

```
##
##      0      1
## 1838  162
```

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

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

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

```
table(mailorder3$purchase)
```

```
##
##      0      1
## 1837  326
```

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

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

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

동일한 결과를 보기위한 seed번호를 설정하고, 비복원 추출로 resampling을 실행

```
table(mailorder4$purchase)
```

```
##
##      0      1
## 609 112
```

```
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.