2017 빅콘테스트

(대출관련 연체,상환 예측 분석)

변수선택을 위한 stepwise 실행

```
data <- read.csv("Data_set.csv")
test <- read.csv("Test_set.csv")

# 테스트 데이터, 트레이닝 데이터 나누기
split.data = function(data, p = 0.7, s = 666){
  set.seed(s)
  index = sample(1:dim(data)[1])
  train = data[index[1:floor(dim(data)[1] * p)], ]
  test = data[index[((ceiling(dim(data)[1] * p)) + 1):dim(data)[1]], ]
  return(list(train = train, test = test))
}
allset=split.data(data,p=0.7)
trainset = allset$train
testset = allset$test

glm <- glm(TARGET~ ., data = trainset)
sglm <- step(glm, direction = "both")
```

결과 중 AIC 값이 가장 작은 변수들만 선택

```
## Step:  AIC=-34625.21
## TARGET ~ BNK_LNIF_CNT + SPART_LNIF_CNT + ECT_LNIF_CNT + TOT_LNIF_AMT +
##     TOT_CLIF_AMT + BNK_LNIF_AMT + CPT_LNIF_AMT + CRDT_OCCR_MDIF +
##     SPTCT_OCCR_MDIF + CRDT_CARD_CNT + CTCD_OCCR_MDIF + CB_GUIF_AMT +
##     ACTL_FMLY_NUM + CRDT_LOAN_CNT + CRLN_OVDU_RATE + LT1Y_CLOD_RATE +
##     STRT_CRDT_GRAD + LT1Y_PEOD_RATE + CNTT_LAMT_CNT + LT1Y_CTLT_CNT +
##     FMLY_PLPY_CNT + AGE + SEX + TEL_MBSP_GRAD + ARPU + MON_TLFE_AMT +
##     MOBL_FATY_PRC + NUM_DAY_SUSP + CRMM_OVDU_AMT + TLFE_UNPD_CNT +
##     LT1Y_MXOD_AMT + PAYM_METD + LINE_STUS + MOBL_PRIN
##
##                       Df Deviance    AIC
## <none>                     2503.7 -34625
## - CNTT_LAMT_CNT        1   2503.8 -34625
## + LT1Y_SLOD_RATE       1   2503.6 -34625
## + AVG_STLN_RATE        1   2503.6 -34624
```

```
## + PREM_OVDU_RATE     1   2503.6 -34624
## + CUST_ID            1   2503.7 -34624
## - ACTL_FMLY_NUM      1   2503.8 -34624
## - STRT_CRDT_GRAD     1   2503.8 -34624
## + CB_GUIF_CNT        1   2503.7 -34624
## + TOT_REPY_AMT       1   2503.7 -34624
## + LT1Y_STLN_AMT      1   2503.7 -34624
## + MIN_CNTT_DATE      1   2503.7 -34624
## + TOT_CRLN_AMT       1   2503.7 -34624
## + AVG_CALL_FREQ      1   2503.7 -34624
## + SVINS_MON_PREM     1   2503.7 -34624
## + TEL_CNTT_QTR       1   2503.7 -34624
## + CRLN_30OVDU_RATE   1   2503.7 -34624
## + AVG_CALL_TIME      1   2503.7 -34623
## + FMLY_CLAM_CNT      1   2503.7 -34623
## + MAX_MON_PREM       1   2503.7 -34623
## + AUTR_FAIL_MCNT     1   2503.7 -34623
## + MATE_JOB_INCM      1   2503.7 -34623
## + GDINS_MON_PREM     1   2503.7 -34623
## + FMLY_TOT_PREM      1   2503.7 -34623
## + CUST_JOB_INCM      1   2503.7 -34623
## + HSHD_INFR_INCM     1   2503.7 -34623
## + FYCM_PAID_AMT      1   2503.7 -34623
## + TOT_PREM           1   2503.7 -34623
## + CPT_LNIF_CNT       1   2503.7 -34623
## + FMLY_SVINS_MNPREM  1   2503.7 -34623
## + STLN_REMN_AMT      1   2503.7 -34623
## + LTST_CRDT_GRAD     1   2503.7 -34623
## + CBPT_MBSP_YN       1   2503.7 -34623
## + CUST_FMLY_NUM      1   2503.7 -34623
## + FMLY_GDINS_MNPREM  1   2503.7 -34623
## - LT1Y_CLOD_RATE     1   2503.8 -34623
## - CRDT_LOAN_CNT      1   2503.9 -34622
## - TOT_CLIF_AMT       1   2503.9 -34621
## - NUM_DAY_SUSP       1   2504.0 -34618
## - CPT_LNIF_AMT       1   2504.1 -34615
## - BNK_LNIF_AMT       1   2504.1 -34615
## - SEX                1   2504.1 -34615
## - MON_TLFE_AMT       1   2504.2 -34613
## - FMLY_PLPY_CNT      1   2504.3 -34611
## + OCCP_NAME_G       17   2503.1 -34609
## + LAST_CHLD_AGE     14   2503.3 -34609
## - CRLN_OVDU_RATE     1   2504.4 -34606
## - TLFE_UNPD_CNT      1   2504.4 -34606
## - ARPU               1   2504.5 -34605
## + MATE_OCCP_NAME_G  17   2503.3 -34601
## - BNK_LNIF_CNT       1   2504.8 -34598
## - TEL_MBSP_GRAD      4   2505.2 -34591
## - ECT_LNIF_CNT       1   2505.2 -34585
```

```
## - LT1Y_CTLT_CNT      1   2505.4 -34579
## - CRMM_OVDU_AMT      1   2505.6 -34573
## - CB_GUIF_AMT        1   2505.6 -34573
## - MOBL_FATY_PRC      1   2505.9 -34564
## - TOT_LNIF_AMT       1   2506.3 -34554
## - LINE_STUS          1   2506.5 -34549
## - MOBL_PRIN          1   2507.6 -34519
## - CRDT_OCCR_MDIF     1   2507.7 -34515
## - SPTCT_OCCR_MDIF    1   2507.8 -34512
## - LT1Y_MXOD_AMT      1   2510.1 -34448
## - AGE               10   2513.2 -34379
## - CTCD_OCCR_MDIF     1   2519.0 -34201
## - CRDT_CARD_CNT      1   2529.2 -33915
## - LT1Y_PEOD_RATE     8   2530.7 -33889
## - SPART_LNIF_CNT     1   2533.2 -33805
## - PAYM_METD          4   2558.5 -33114
```

선택된 변수를 활용하여 랜덤포레스트 분석

```
library(ranger)

## Warning: package 'ranger' was built under R version 3.4.2

DATA03 <- read.csv("Data_set03.csv")

# 파생변수 추가 (여러 파생 변수 중 가장 좋은 값의 변수들만 사용)
# TOT_SPTCT : 대출정보 현재 총 금액 X 대출정보 최근 개설일로부터 현재까지 유지기간
[2 산업분류-신용대출]

str(DATA03)

## 'data.frame':    100233 obs. of  76 variables:
##  $ CUST_ID          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ TARGET           : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ BNK_LNIF_CNT     : int  1 1 0 0 4 1 0 1 2 0 ...
##  $ CPT_LNIF_CNT     : int  0 0 1 2 0 0 1 0 0 1 ...
##  $ SPART_LNIF_CNT   : int  0 0 3 4 0 1 2 0 0 1 ...
##  $ ECT_LNIF_CNT     : int  0 0 2 2 0 1 1 0 0 0 ...
##  $ TOT_LNIF_AMT     : int  9001 24001 15001 6001 21001 141001 12001 3001 2
73001 9001 ...
##  $ TOT_CLIF_AMT     : int  9001 0 9001 3001 15001 27001 3001 3001 273001 9
001 ...
##  $ BNK_LNIF_AMT     : int  9001 24001 0 0 21001 111001 0 3001 273001 0 ...
##  $ CPT_LNIF_AMT     : int  0 0 3001 3001 0 0 9001 0 0 9001 ...
##  $ CRDT_OCCR_MDIF   : int  1 0 1 1 1 1 121 1 37 1 ...
##  $ SPTCT_OCCR_MDIF  : int  0 0 25 25 0 1 121 0 0 1 ...
```

```
##  $ CRDT_CARD_CNT    : int  2 2 4 4 1 4 2 2 5 3 ...
##  $ CTCD_OCCR_MDIF   : int  13 121 121 61 97 121 121 121 121 25 ...
##  $ CB_GUIF_CNT      : int  3 0 0 0 0 0 1 0 0 0 ...
##  $ CB_GUIF_AMT      : int  420001 0 0 0 0 0 6001 0 0 0 ...
##  $ OCCP_NAME_G      : Factor w/ 18 levels "*","1 차산업 종사자",..: 7 15 17
 18 7 4 17 8 17 11 ...
##  $ CUST_JOB_INCM    : int  5400 5500 0 0 4800 4400 0 0 0 4700 ...
##  $ HSHD_INFR_INCM   : int  7700 8100 4900 10100 4800 7700 7700 10300 7600
12400 ...
##  $ ACTL_FMLY_NUM    : int  4 4 4 2 4 2 5 3 4 4 ...
##  $ CUST_FMLY_NUM    : int  1 2 1 1 1 2 3 1 1 1 ...
##  $ LAST_CHLD_AGE    : int  24 29 34 0 14 0 19 24 9 14 ...
##  $ MATE_OCCP_NAME_G : Factor w/ 18 levels "*","1 차산업 종사자",..: 17 17 3
 5 17 11 3 12 16 8 ...
##  $ MATE_JOB_INCM    : int  0 0 0 0 0 3300 4400 5000 5400 7500 ...
##  $ CRDT_LOAN_CNT    : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ MIN_CNTT_DATE    : int  0 0 0 0 0 0 200106 0 0 0 ...
##  $ TOT_CRLN_AMT     : int  0 0 0 0 0 0 4000000 0 0 0 ...
##  $ TOT_REPY_AMT     : int  0 0 0 0 0 0 4000000 0 0 0 ...
##  $ CRLN_OVDU_RATE   : int  0 0 0 0 0 0 81 0 0 0 ...
##  $ CRLN_30OVDU_RATE : int  0 0 0 0 0 0 14 0 0 0 ...
##  $ LT1Y_CLOD_RATE   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ STRT_CRDT_GRAD   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LTST_CRDT_GRAD   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREM_OVDU_RATE   : int  12 13 2 4 0 25 71 4 3 15 ...
##  $ LT1Y_PEOD_RATE   : Factor w/ 9 levels "0","10 미만","20 미만",..: 3 1 1 1
 1 1 2 1 2 2 ...
##  $ AVG_STLN_RATE    : int  0 0 0 0 95 0 94 0 0 99 ...
##  $ STLN_REMN_AMT    : int  0 0 0 0 2000000 0 3000000 0 0 0 ...
##  $ LT1Y_STLN_AMT    : int  0 0 0 0 0 0 2000000 0 0 3000000 ...
##  $ LT1Y_SLOD_RATE   : int  0 0 0 0 0 0 10 0 0 30 ...
##  $ GDINS_MON_PREM   : int  190000 0 0 0 0 100000 0 0 0 300000 ...
##  $ SVINS_MON_PREM   : int  0 0 0 0 0 0 200000 0 60000 0 ...
##  $ FMLY_GDINS_MNPREM: int  190000 110000 0 0 0 190000 300000 0 0 300000 ...
##  $ FMLY_SVINS_MNPREM: int  0 0 0 0 0 0 200000 0 60000 0 ...
##  $ MAX_MON_PREM     : int  190000 0 100000 0 300000 190000 200000 60000 60
000 340000 ...
##  $ TOT_PREM         : int  20000000 7000000 11000000 4000000 4000000 10000
00 11000000 4000000 2000000 36000000 ...
##  $ FMLY_TOT_PREM    : int  20000000 36000000 11000000 4000000 4000000 6000
000 33000000 4000000 2000000 36000000 ...
##  $ CNTT_LAMT_CNT    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LT1Y_CTLT_CNT    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AUTR_FAIL_MCNT   : int  10 0 0 0 0 0 1 0 0 21 ...
##  $ FYCM_PAID_AMT    : int  0 300000 0 0 500000 300000 800000 0 0 300000 ...
##  $ FMLY_CLAM_CNT    : int  0 2 0 0 2 1 0 0 0 1 ...
##  $ FMLY_PLPY_CNT    : int  0 5 2 1 1 1 0 1 0 2 ...
```

```
##  $ AGE              : Factor w/ 12 levels "*","20","25",..: 8 8 10 5 7 7 6
 10 6 6 ...
##  $ SEX              : Factor w/ 3 levels "*","1","2": 2 2 3 2 2 2 3 2 3 2
 ...
##  $ AVG_CALL_TIME    : int  450 81 139 1118 396 268 744 309 314 0 ...
##  $ AVG_CALL_FREQ    : int  493 22 17 0 354 179 535 221 179 0 ...
##  $ TEL_MBSP_GRAD    : Factor w/ 5 levels "","E","Q","R",..: 1 1 1 1 5 3 5
4 4 1 ...
##  $ ARPU             : int  30000 30000 30000 30000 50000 60000 50000 10000
 60000 0 ...
##  $ MON_TLFE_AMT     : int  80000 40000 40000 80000 80000 80000 120000 7000
0 70000 0 ...
##  $ CBPT_MBSP_YN     : Factor w/ 2 levels "N","Y": 1 1 2 1 2 1 1 2 2 1 ...
##  $ MOBL_FATY_PRC    : int  800000 500000 500000 900000 800000 400000 90000
0 900000 0 800000 ...
##  $ TEL_CNTT_QTR     : int  20111 20143 20103 20144 20131 20154 20143 20021
 20131 20133 ...
##  $ NUM_DAY_SUSP     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CRMM_OVDU_AMT    : int  0 0 0 540000 130000 0 120000 0 0 0 ...
##  $ TLFE_UNPD_CNT    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LT1Y_MXOD_AMT    : int  0 0 0 630000 90000 0 290000 0 0 0 ...
##  $ PAYM_METD        : Factor w/ 5 levels "","G","K","O",..: 4 4 4 2 2 4 2
4 4 4 ...
##  $ LINE_STUS        : Factor w/ 2 levels "S","U": 2 2 2 1 2 2 2 2 2 2 ...
##  $ MOBL_PRIN        : int  580000 90000 120000 320000 410000 170000 720000
 40000 0 0 ...
##  $ CRMM_MOBL        : num  0.00 0.00 0.00 4.86e+11 1.04e+11 ...
##  $ CRDT_LT1Y        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CRDT_CTCD        : int  26 242 484 244 97 484 242 242 605 75 ...
##  $ TOT_SPTCT        : int  0 0 375025 150025 0 141001 1452121 0 0 9001 ...
##  $ LT1Y_CR30        : int  0 0 0 0 0 0 14 0 0 0 ...
##  $ CRMM_LT1Y        : int  0 0 0 1170000 220000 0 410000 0 0 0 ...
##  $ CRMM_LT1Y_02     : int  0 0 0 630000 130000 0 290000 0 0 0 ...
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
DATA03 <- select(DATA03, TARGET, CNTT_LAMT_CNT,ACTL_FMLY_NUM,STRT_CRDT_GRAD,

                 LT1Y_CLOD_RATE, CRDT_LOAN_CNT, TOT_CLIF_AMT,NUM_DAY_SUSP,
```

```
                CPT_LNIF_AMT, BNK_LNIF_AMT, SEX, MON_TLFE_AMT, FMLY_PLPY_CNT,

                CRLN_OVDU_RATE, TLFE_UNPD_CNT, ARPU, BNK_LNIF_CNT, TEL_MBSP_
GRAD,
                ECT_LNIF_CNT, LT1Y_CTLT_CNT, CB_GUIF_AMT, MOBL_FATY_PRC,

                TOT_LNIF_AMT, LINE_STUS, MOBL_PRIN, CRDT_OCCR_MDIF, SPTCT_OC
CR_MDIF,
                AGE, CTCD_OCCR_MDIF, CRDT_CARD_CNT, LT1Y_PEOD_RATE,
                SPART_LNIF_CNT, PAYM_METD,CRMM_OVDU_AMT,LT1Y_MXOD_AMT,TOT_SP
TCT)
str(DATA03)

## 'data.frame':    100233 obs. of  36 variables:
##  $ TARGET         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ CNTT_LAMT_CNT  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ACTL_FMLY_NUM  : int  4 4 4 2 4 2 5 3 4 4 ...
##  $ STRT_CRDT_GRAD : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LT1Y_CLOD_RATE : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CRDT_LOAN_CNT  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ TOT_CLIF_AMT   : int  9001 0 9001 3001 15001 27001 3001 3001 273001 900
1 ...
##  $ NUM_DAY_SUSP   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CPT_LNIF_AMT   : int  0 0 3001 3001 0 0 9001 0 0 9001 ...
##  $ BNK_LNIF_AMT   : int  9001 24001 0 0 21001 111001 0 3001 273001 0 ...
##  $ SEX            : Factor w/ 3 levels "*","1","2": 2 2 3 2 2 2 3 2 3 2 ...
##  $ MON_TLFE_AMT   : int  80000 40000 40000 80000 80000 80000 120000 70000
70000 0 ...
##  $ FMLY_PLPY_CNT  : int  0 5 2 1 1 1 0 1 0 2 ...
##  $ CRLN_OVDU_RATE : int  0 0 0 0 0 0 81 0 0 0 ...
##  $ TLFE_UNPD_CNT  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ARPU           : int  30000 30000 30000 30000 50000 60000 50000 10000 6
0000 0 ...
##  $ BNK_LNIF_CNT   : int  1 1 0 0 4 1 0 1 2 0 ...
##  $ TEL_MBSP_GRAD  : Factor w/ 5 levels "","E","Q","R",..: 1 1 1 1 5 3 5 4
4 1 ...
##  $ ECT_LNIF_CNT   : int  0 0 2 2 0 1 1 0 0 0 ...
##  $ LT1Y_CTLT_CNT  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CB_GUIF_AMT    : int  420001 0 0 0 0 0 6001 0 0 0 ...
##  $ MOBL_FATY_PRC  : int  800000 500000 500000 900000 800000 400000 900000
900000 0 800000 ...
##  $ TOT_LNIF_AMT   : int  9001 24001 15001 6001 21001 141001 12001 3001 273
001 9001 ...
##  $ LINE_STUS      : Factor w/ 2 levels "S","U": 2 2 2 1 2 2 2 2 2 2 ...
##  $ MOBL_PRIN      : int  580000 90000 120000 320000 410000 170000 720000 4
0000 0 0 ...
##  $ CRDT_OCCR_MDIF : int  1 0 1 1 1 1 121 1 37 1 ...
##  $ SPTCT_OCCR_MDIF: int  0 0 25 25 0 1 121 0 0 1 ...
```

```
##  $ AGE            : Factor w/ 12 levels "*","20","25",..: 8 8 10 5 7 7 6 1
0 6 6 ...
##  $ CTCD_OCCR_MDIF : int  13 121 121 61 97 121 121 121 121 25 ...
##  $ CRDT_CARD_CNT  : int  2 2 4 4 1 4 2 2 5 3 ...
##  $ LT1Y_PEOD_RATE : Factor w/ 9 levels "0","10 미만","20 미만",..: 3 1 1 1 1
 1 2 1 2 2 ...
##  $ SPART_LNIF_CNT : int  0 0 3 4 0 1 2 0 0 1 ...
##  $ PAYM_METD      : Factor w/ 5 levels "","G","K","O",..: 4 4 4 2 2 4 2 4
4 4 ...
##  $ CRMM_OVDU_AMT  : int  0 0 0 540000 130000 0 120000 0 0 0 ...
##  $ LT1Y_MXOD_AMT  : int  0 0 0 630000 90000 0 290000 0 0 0 ...
##  $ TOT_SPTCT      : int  0 0 375025 150025 0 141001 1452121 0 0 9001 ...
```

```
split.data = function(data, p = 0.7, s = 555){
  set.seed(s)
  num = nrow(data)
  index = sample(1:num, p * num)
  train = data[index,]
  test = data[-index,]
  return(list(train = train, test = test))
}

allset = split.data(DATA03, p=0.7)
trainset = allset$train
testset = allset$test

dim(trainset); dim(testset)
```

```
## [1] 70163    36
```

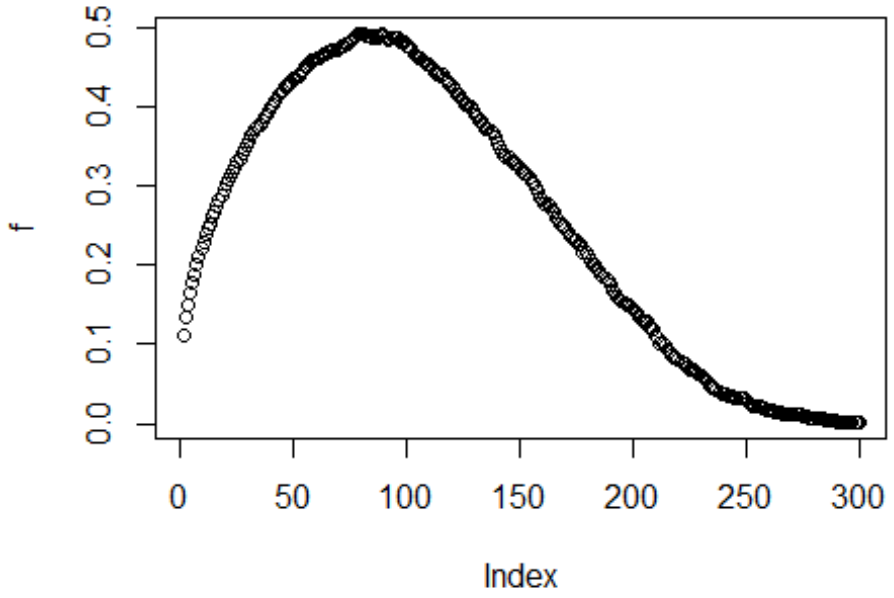
```
## [1] 30070    36
```

```
Model_y = ranger(factor(TARGET)~., trainset, importance = "impurity", probabi
lity = TRUE)
Model_y
```

```
## Ranger result
##
## Call:
##  ranger(factor(TARGET) ~ ., trainset, importance = "impurity",      probab
ility = TRUE)
##
## Type:                             Probability estimation
## Number of trees:                  500
## Sample size:                      70163
## Number of independent variables:  35
## Mtry:                             5
## Target node size:                 10
```

```
## Variable importance mode:         impurity
## OOB prediction error:             0.03013898

Model_y$variable.importance

##   CNTT_LAMT_CNT   ACTL_FMLY_NUM  STRT_CRDT_GRAD  LT1Y_CLOD_RATE
##       40.289645      101.577563       23.834279        8.435753
##   CRDT_LOAN_CNT    TOT_CLIF_AMT    NUM_DAY_SUSP    CPT_LNIF_AMT
##       29.497218      135.512381       69.186053      131.525682
##    BNK_LNIF_AMT             SEX    MON_TLFE_AMT   FMLY_PLPY_CNT
##      116.422495       48.139699      169.127266      102.473461
##  CRLN_OVDU_RATE   TLFE_UNPD_CNT            ARPU    BNK_LNIF_CNT
##       33.313854       11.458714      137.813139       64.157599
##   TEL_MBSP_GRAD    ECT_LNIF_CNT   LT1Y_CTLT_CNT     CB_GUIF_AMT
##       77.849121       77.666369       31.388751       73.174349
##   MOBL_FATY_PRC    TOT_LNIF_AMT       LINE_STUS       MOBL_PRIN
##      134.334460      195.186098       20.135221      209.615406
##  CRDT_OCCR_MDIF SPTCT_OCCR_MDIF             AGE  CTCD_OCCR_MDIF
##      170.995143      179.939760      156.059639      209.326563
##   CRDT_CARD_CNT  LT1Y_PEOD_RATE  SPART_LNIF_CNT       PAYM_METD
##      239.388374      174.994030      125.772648      174.030869
##   CRMM_OVDU_AMT   LT1Y_MXOD_AMT       TOT_SPTCT
##      228.988706      238.803576      206.334455

pred <- predict(Model_y, testset)
pred

## Ranger prediction
##
## Type:                             Probability estimation
## Sample size:                      30070
## Number of independent variables:  35

str(pred)

## List of 5
##  $ predictions              : num [1:30070, 1:2] 0.949 1 0.958 0.538 0.882
 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "0" "1"
##  $ num.trees                : num 500
##  $ num.independent.variables: num 35
##  $ num.samples              : int 30070
##  $ treetype                 : chr "Probability estimation"
##  - attr(*, "class")= chr "ranger.prediction"

table(pred$predictions)
```

```
##
##                     0 4.19009813650899e-06 5.78034682080925e-06


…

# 결과값이 많아서 생략

predVal <- pred$predictions
head(predVal)

##              0           1
## [1,] 0.9487841 0.0512158730
## [2,] 0.9996667 0.0003333333
## [3,] 0.9576651 0.0423349206
## [4,] 0.5376016 0.4623984127
## [5,] 0.8815492 0.1184507937
## [6,] 0.9530405 0.0469595238

predVal <- predVal[,2]
head(predVal)

## [1] 0.0512158730 0.0003333333 0.0423349206 0.4623984127 0.1184507937
## [6] 0.0469595238

min(predVal)

## [1] 0

max(predVal)

## [1] 0.8728341

range <- seq(min(predVal)+0.001, max(predVal)-0.001, length = 300)
f <- numeric()
for (i in 1:300) {
  threshold <- range[i]
  pred_label <- ifelse(predVal > threshold, "1","0")
  tab <- table(testset$TARGET, pred_label)
  p <- tab[2,2]/(tab[2,2]+tab[1,2])
  r <- tab[2,2]/(tab[2,2]+tab[2,1])
  f[i] <- 2*p*r/(p+r)
}

which.max(f)

## [1] 80

plot(f)
```




```
f

##   [1] 0.111299067 0.134264402 0.150365138 0.165031281 0.176842704
##   [6] 0.189135425 0.199983728 0.210871441 0.220072333 0.229290921
##  [11] 0.238203581 0.246657066 0.253091684 0.261622101 0.267982104
##  [16] 0.275166192 0.281579593 0.287922950 0.294440094 0.302736218
##  [21] 0.309375433 0.313875870 0.318697296 0.323977307 0.330227585
##  [26] 0.334841629 0.340227674 0.346084387 0.352328122 0.358050120
##  [31] 0.363349131 0.367507604 0.371804237 0.375046555 0.378511769
##  [36] 0.382830626 0.387900356 0.391068196 0.396234138 0.400665004
##  [41] 0.402877698 0.408453742 0.414730381 0.417314095 0.421123900
##  [46] 0.424604085 0.427438171 0.429314421 0.432211538 0.434909800
##  [51] 0.436821323 0.439280360 0.442635462 0.446627035 0.450223273
##  [56] 0.452659574 0.454667743 0.458401305 0.458735854 0.462443946
##  [61] 0.461451247 0.463309353 0.465562337 0.466391509 0.468694097
##  [66] 0.469240048 0.471663620 0.471023428 0.471910112 0.475197472
##  [71] 0.476982097 0.477698772 0.479295729 0.478346457 0.480928690
##  [76] 0.484848485 0.489726027 0.491191710 0.492157546 0.492631579
##  [81] 0.491687301 0.490350250 0.489884393 0.488694384 0.487463127
##  [86] 0.489004845 0.488135593 0.489142857 0.491727588 0.488768397
##  [91] 0.487500000 0.484368817 0.486011191 0.487055016 0.486729277
##  [96] 0.487644152 0.482098251 0.482526316 0.481292517 0.478111588
## [101] 0.476396709 0.475109170 0.468047598 0.465219318 0.462019660
## [106] 0.460567823 0.459373581 0.456382002 0.453874539 0.454629630
## [111] 0.447970135 0.443817583 0.441496921 0.442115293 0.439423077
## [116] 0.440776699 0.436612824 0.434139122 0.428286853 0.426639960
```

```
## [121] 0.420040486 0.416581372 0.415678185 0.409776391 0.407543216
## [126] 0.404019038 0.401709402 0.399570354 0.394808004 0.388888889
## [131] 0.387308534 0.381687810 0.382059801 0.377295492 0.373392957
## [136] 0.373318386 0.368658399 0.366723260 0.360842345 0.353008596
## [141] 0.348676640 0.343188406 0.338963308 0.336842105 0.335877863
## [146] 0.330977621 0.328994083 0.327790974 0.326774001 0.322541966
## [151] 0.317880795 0.316616314 0.315533981 0.311355311 0.304855562
## [156] 0.302282542 0.298636927 0.291952589 0.285893417 0.283018868
## [161] 0.278481013 0.277742549 0.273711012 0.269820972 0.267605634
## [166] 0.259020619 0.255979315 0.253082414 0.249837345 0.250162866
## [171] 0.242463958 0.236998025 0.233663366 0.231328486 0.230616302
## [176] 0.227242525 0.226214238 0.217012726 0.216252518 0.210242588
## [181] 0.205544287 0.202033898 0.200950441 0.196319018 0.194121668
## [186] 0.187928669 0.184319120 0.183195592 0.179434092 0.174636175
## [191] 0.168289291 0.166085136 0.160951714 0.158374212 0.155898876
## [196] 0.152112676 0.150916784 0.149717514 0.147100424 0.145789101
## [201] 0.143160879 0.139303483 0.134094151 0.130092924 0.130092924
## [206] 0.130185980 0.122214234 0.122214234 0.119510439 0.111352133
## [211] 0.108616944 0.100436681 0.100583090 0.097810219 0.093635699
## [216] 0.090842491 0.086637298 0.082474227 0.082535004 0.079940785
## [221] 0.077094144 0.075667656 0.074294205 0.069992554 0.067114094
## [226] 0.067114094 0.067114094 0.064227035 0.062780269 0.061331339
## [231] 0.058426966 0.054094666 0.051204819 0.046792453 0.045317221
## [236] 0.045317221 0.042360061 0.040909091 0.037936267 0.037965072
## [241] 0.037965072 0.033485540 0.033485540 0.033511043 0.032036613
## [246] 0.032036613 0.032036613 0.032036613 0.032036613 0.030534351
## [251] 0.027522936 0.024502297 0.022988506 0.023006135 0.023006135
## [256] 0.023006135 0.019969278 0.019969278 0.016936105 0.016949153
## [261] 0.015420200 0.015420200 0.013888889 0.013888889 0.013888889
## [266] 0.012355212 0.012355212 0.012355212 0.012355212 0.012355212
## [271] 0.010819165 0.010819165 0.010819165 0.010827533 0.009287926
## [276] 0.007745933 0.007745933 0.006201550 0.006201550 0.006201550
## [281] 0.006201550 0.006201550 0.006201550 0.006201550 0.004654771
## [286] 0.004654771 0.004654771 0.003105590 0.003105590 0.001554002
## [291] 0.001554002 0.001554002 0.001554002 0.001554002 0.001554002
## [296] 0.001554002 0.001554002 0.001554002 0.001554002 0.001554002

thre <- (max(predVal)*which.max(f))/300
pred_label <- ifelse(predVal > thre, "1","0")
table(pred_label)

## pred_label
##     0     1
## 28517  1553

mR1 <- table(testset$TARGET, pred_label)
mR1

##    pred_label
##         0     1
```

```
##   0 27930   854
##   1   587   699
```

```
mR1[1,1];mR1[1,2]
```

```
## [1] 27930
```

```
## [1] 854
```

```
mR1[2,1];mR1[2,2]
```

```
## [1] 587
```

```
## [1] 699
```

```
# precision & recall
# 예측한 실제 연체자 수 / 예측한 연체자 전체 수 = TP/(FP+TP)
p <- mR1[2,2]/(mR1[2,2]+mR1[1,2])  # precision
# 예측한 실제 연체자 수 / 실제 연체자 전체 수 = TP/(FN+TP)
r <- mR1[2,2]/(mR1[2,1]+mR1[2,2]) # recall

# F-measure
f_m <- 2*p*r/(p+r)
f_m
```

```
## [1] 0.4924269
```

```
# 나온 모델을 바탕으로 Test 데이터로 예측
```

```
Rtest <- read.csv("Test_set.csv")

Rpred <- predict(Model_y, Rtest)

str(Rpred)

RpredVal <- Rpred$predictions[,2]

answer <- ifelse(RpredVal > thre, "1","0")



write.csv(answer, file = "answer.csv")
```