logistic_regression_and_naive_bayes

juhwan

6/6/2021

서론

우리는 통신회사의 고객 이탈 데이터를 가지고, 로지스틱 회귀와 나이브 베이즈 분류를 이용해 이탈 고객을 예측해 보고자 한다.

본론

데이터 전처리

데이터를 불러오고, 종속변수의 값을 0과 1로 변환시킨다.
이 때 Yes는 1로, No는 0으로 분류했다.

```
df = read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
df$Churn = ifelse(df$Churn == 'Yes', 1, 0)
```

```
df = df[,-1]         # 고객 아이디는 필요가 없다.
df = na.omit(df)     # NA값이 있는 고객을 모두 없앤다.
head(df)              # 데이터를 확인한다.
```

```
##   gender SeniorCitizen Partner Dependents tenure PhoneService    MultipleLines
## 1 Female             0     Yes         No      1           No No phone service
## 2   Male             0      No         No     34          Yes               No
## 3   Male             0      No         No      2          Yes               No
## 4   Male             0      No         No     45           No No phone service
## 5 Female             0      No         No      2          Yes               No
## 6 Female             0      No         No      8          Yes              Yes
##   InternetService OnlineSecurity OnlineBackup DeviceProtection TechSupport
## 1             DSL             No          Yes               No          No
## 2             DSL            Yes           No              Yes          No
## 3             DSL            Yes          Yes               No          No
## 4             DSL            Yes           No              Yes         Yes
## 5     Fiber optic             No           No               No          No
## 6     Fiber optic             No           No              Yes          No
##   StreamingTV StreamingMovies       Contract PaperlessBilling
## 1          No              No Month-to-month              Yes
## 2          No              No       One year               No
## 3          No              No Month-to-month              Yes
## 4          No              No       One year               No
## 5          No              No Month-to-month              Yes
## 6         Yes             Yes Month-to-month              Yes
##               PaymentMethod MonthlyCharges TotalCharges Churn
## 1          Electronic check          29.85        29.85     0
## 2              Mailed check          56.95      1889.50     0
## 3              Mailed check          53.85       108.15     1
## 4 Bank transfer (automatic)          42.30      1840.75     0
## 5          Electronic check          70.70       151.65     1
## 6          Electronic check          99.65       820.50     1
```

Logistic regression

모델 생성

```
logreg1 = glm(Churn ~ ., family = binomial, data = df)  # 전체 값으로 회귀식을 뽑는다.
summary(logreg1)      # 유의미한 값을 찾는다.
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9180  -0.6791  -0.2855   0.7282   3.4300
##
## Coefficients: (7 not defined because of singularities)
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       1.165e+00  8.151e-01   1.430  0.15284
## genderMale                       -2.183e-02  6.480e-02  -0.337  0.73619
```

```
## SeniorCitizen                           2.168e-01  8.453e-02   2.564  0.01033 *
## PartnerYes                             -3.840e-04  7.783e-02  -0.005  0.99606
## DependentsYes                          -1.485e-01  8.973e-02  -1.655  0.09796 .
## tenure                                 -6.059e-02  6.236e-03  -9.716  < 2e-16 ***
## PhoneServiceYes                         1.715e-01  6.487e-01   0.264  0.79153
## MultipleLinesNo phone service                  NA         NA      NA       NA
## MultipleLinesYes                        4.484e-01  1.773e-01   2.530  0.01142 *
## InternetServiceFiber optic              1.747e+00  7.981e-01   2.190  0.02855 *
## InternetServiceNo                      -1.786e+00  8.073e-01  -2.213  0.02691 *
## OnlineSecurityNo internet service              NA         NA      NA       NA
## OnlineSecurityYes                      -2.054e-01  1.787e-01  -1.150  0.25031
## OnlineBackupNo internet service                NA         NA      NA       NA
## OnlineBackupYes                         2.604e-02  1.754e-01   0.148  0.88197
## DeviceProtectionNo internet service            NA         NA      NA       NA
## DeviceProtectionYes                     1.474e-01  1.764e-01   0.836  0.40339
## TechSupportNo internet service                 NA         NA      NA       NA
## TechSupportYes                         -1.805e-01  1.806e-01  -0.999  0.31759
## StreamingTVNo internet service                 NA         NA      NA       NA
## StreamingTVYes                          5.905e-01  3.263e-01   1.810  0.07035 .
## StreamingMoviesNo internet service             NA         NA      NA       NA
## StreamingMoviesYes                      5.993e-01  3.267e-01   1.834  0.06658 .
## ContractOne year                       -6.608e-01  1.076e-01  -6.142 8.15e-10 ***
## ContractTwo year                       -1.357e+00  1.764e-01  -7.691 1.46e-14 ***
## PaperlessBillingYes                     3.424e-01  7.450e-02   4.596 4.31e-06 ***
## PaymentMethodCredit card (automatic)   -8.779e-02  1.141e-01  -0.770  0.44156
## PaymentMethodElectronic check           3.045e-01  9.450e-02   3.222  0.00127 **
## PaymentMethodMailed check              -5.759e-02  1.149e-01  -0.501  0.61627
## MonthlyCharges                         -4.034e-02  3.176e-02  -1.270  0.20392
## TotalCharges                            3.289e-04  7.063e-05   4.657 3.20e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8143.4  on 7031  degrees of freedom
## Residual deviance: 5826.3  on 7008  degrees of freedom
## AIC: 5874.3
##
## Number of Fisher Scoring iterations: 6
```

```
# 사용할 변수는 별 두 개 이상 기준으로 했다.
# tenure, Contract, PaperlessBilling, TotalCharges
logreg1Null = glm(Churn ~ 1, family = binomial, data = df) # 가장 기초의 회귀선을 뽑는다.
anova(logreg1Null, logreg1, test = 'LRT')                  # 모델을 서로 비교한다.
```

```
## Analysis of Deviance Table
##
## Model 1: Churn ~ 1
## Model 2: Churn ~ gender + SeniorCitizen + Partner + Dependents + tenure +
##     PhoneService + MultipleLines + InternetService + OnlineSecurity +
##     OnlineBackup + DeviceProtection + TechSupport + StreamingTV +
##     StreamingMovies + Contract + PaperlessBilling + PaymentMethod +
##     MonthlyCharges + TotalCharges
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      7031     8143.4
## 2      7008     5826.3 23   2317.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# P 값을 보면 우리의 회귀식이 확실히 더 잘 설명한다. (의미가 있다.)

# 아까 뽑아준 유의미한 변수들로 독립변수를 선택한다.
logreg1_Modify = glm(Churn ~ tenure + Contract + PaperlessBilling + TotalCharges,
family = binomial, data = df)
anova(logreg1Null, logreg1_Modify, test = 'LRT')  # 아까 만든 NULL모델과 비교한다.
```

```
## Analysis of Deviance Table
##
## Model 1: Churn ~ 1
## Model 2: Churn ~ tenure + Contract + PaperlessBilling + TotalCharges
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      7031     8143.4
## 2      7026     6242.8  5   1900.5 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coef(logreg1_Modify)  # 계수를 가져와 본다.
```

```
##         (Intercept)              tenure    ContractOne year    ContractTwo year
##       -0.2431033837       -0.0886584444       -1.1190162237       -2.2452553117
## PaperlessBillingYes        TotalCharges
##        0.7079626151        0.0007413371
```

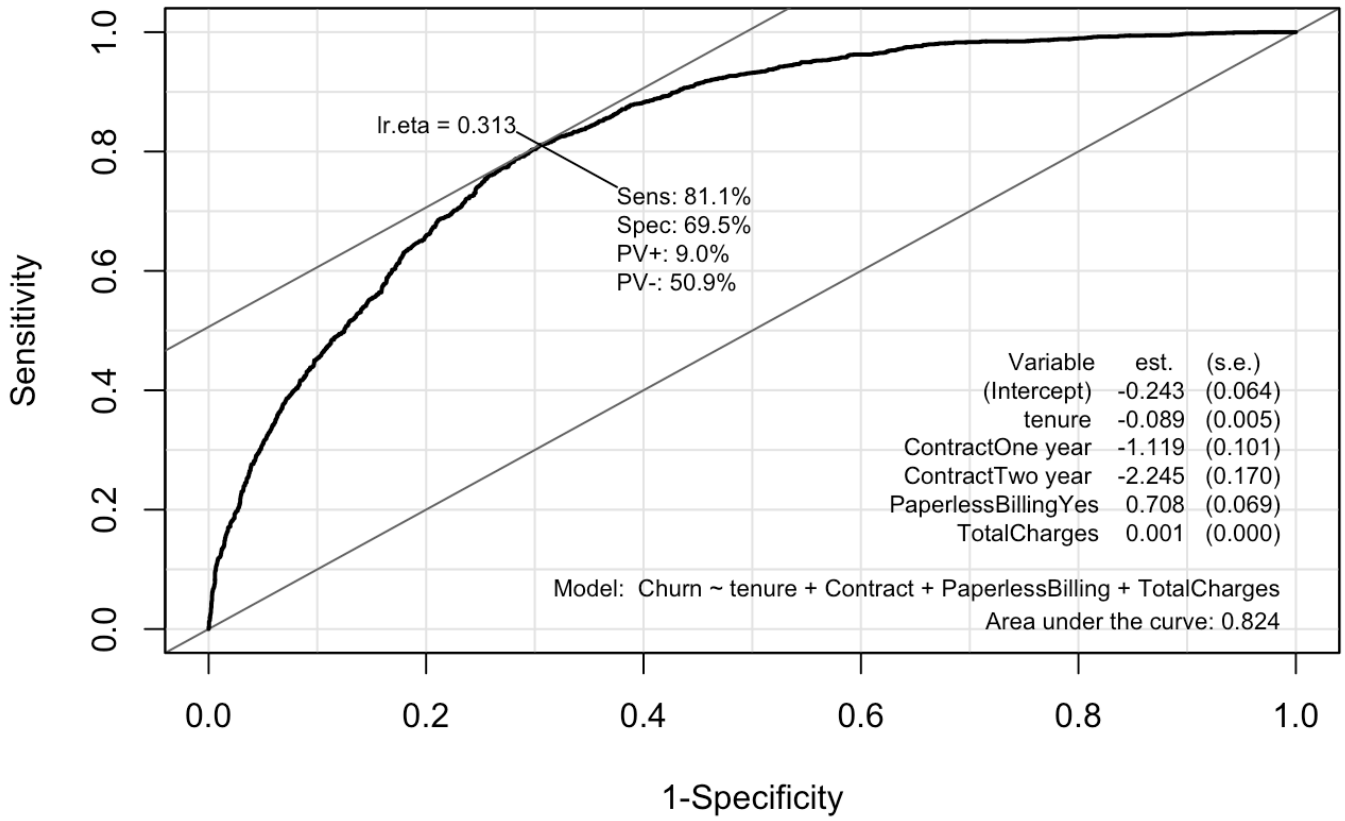
```
exp(coef(logreg1_Modify)) # 'odds가 이 숫자배수 만큼 증가함'
```

```
##         (Intercept)              tenure    ContractOne year    ContractTwo year
##           0.7841904           0.9151581           0.3266009           0.1059005
## PaperlessBillingYes        TotalCharges
##           2.0298515           1.0007416
```

부분 결론 : 즉 tenure이 1증가 할때 odds는 0.9배 됨, Contract가 One year는 odds가 0.3배가 됨, Contract가 Two year는 odds가 0.1배가 됨, PaperlessBilling이 Yes면 odds가 2배가 됨, TotalCharges는 1증가할 때 odds가 1.0007배 됨.

모델 성능 평가

```
graph1 = ROC(form = Churn ~ tenure + Contract + PaperlessBilling + TotalCharges, d
ata = df, plot = 'ROC')
```




```
graph1$res[round(graph1$res$lr.eta, 3) == 0.313,]  # 3번째 값이 가장 .313에 가깝다.
```

```
##                        sens      spec        pvp       pvn    lr.eta
## 0.312789435700227 0.8111289 0.6947511 0.08959391 0.5097025 0.3127894
## 0.312889356627246 0.8111289 0.6949448 0.08957117 0.5095438 0.3128894
## 0.312985192098337 0.8111289 0.6951385 0.08954845 0.5093851 0.3129852
## 0.313395551333727 0.8105939 0.6951385 0.08977936 0.5095500 0.3133956
## 0.313454656772414 0.8105939 0.6953322 0.08975659 0.5093912 0.3134547
```

```
graph1$AUC   # 그래프를 적분한 값, 즉 면적 값이다. 여기서 이 값은 분류기의 정확도를 의미한다.
```

```
## [1] 0.8240617
```

```
confusionMatrix(
  as.factor(ifelse(predict(logreg1_Modify, type = "response")>0.313,1,0)),
  as.factor(logreg1_Modify$y),
  positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3589  353
##          1 1574 1516
##
##                Accuracy : 0.726
##                  95% CI : (0.7154, 0.7364)
##     No Information Rate : 0.7342
##     P-Value [Acc > NIR] : 0.9425
##
##                   Kappa : 0.419
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8111
##             Specificity : 0.6951
##          Pos Pred Value : 0.4906
##          Neg Pred Value : 0.9105
##              Prevalence : 0.2658
##          Detection Rate : 0.2156
##    Detection Prevalence : 0.4394
##       Balanced Accuracy : 0.7531
##
##        'Positive' Class : 1
##
```

```
logreg1_Modify_F1 = 2*0.8111*0.4906 /(0.8111+0.4906)   # 0.611
logreg1_Modify_F1   # F1 값은 Sensitivity와 Pos Pred Value의 조화평균이다.
```

```
## [1] 0.6113938
```

여러 지표들을 볼수 있다. 민감도는 0.81, 특이도는 0.6951, F1값은 0.611이다. 하지만 No Information Rate에 비해 나은 것이 별로 없는 것으로 보인다. 때문에 좋은 모델이라고 할 수는 없다.

Naive bayes classification

분할

```
# 연속형 자료를 n개의 범주형 자료로 바꿔줄 수 있는 함수를 정의한다.
mygroup = function (y,k=4){
  count = length(y)
  z = rank(y,ties.method = "min")
  return(floor((z-1)/(count/k))+1)
}

df$MonthlyCharges_group = mygroup(df$MonthlyCharges, 10)  # MonthlyCharges 값이 낮으면: 1 ~ 높으면: 10
table(df$MonthlyCharges_group)                            # 그룹이 10개로 나눠짐
```

```
##
##   1   2   3   4   5   6   7   8   9  10
## 714 702 698 699 706 709 700 704 698 702
```

```
df$tenure_group = mygroup(df$tenure, 15)                  # tenure 값을 15개로 범주화
table(df$tenure_group)                                    # 확인
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
## 613 438 419 489 401 507 490 407 511 454 484 488 485 484 362
```

```
df$TotalCharges_group = mygroup(df$TotalCharges, 10)      # TotalCharges 값을 10로 범주화
table(df$TotalCharges_group)                              # 확인
```

```
##
##   1   2   3   4   5   6   7   8   9  10
## 705 702 703 703 703 704 703 703 703 703
```

```
df = df[, -c(5, 18, 19)]   # 쓸모없어진 연속형 자료를 삭제

# train, test 분할
train_idx = sample(7032, 7032*3/4)  # 샘플 생성
df_train = df[train_idx,]
df_test = df[-train_idx,]
df_train_labels = df[train_idx,]$Churn
df_test_labels = df[-train_idx,]$Churn

df_train = df_train[,-17]           # 훈련 데이터에 Churn 삭제

prop.table(table(df_train_labels))  # 테스트와 훈련 세트의 비율이 비슷한 지 확인
```

```
## df_train_labels
##         0         1
## 0.7358741 0.2641259
```

```
prop.table(table(df_test_labels))   # 확인
```

```
## df_test_labels
##         0         1
## 0.7292378 0.2707622
```

비슷하다. 잘 나뉘어졌다. 골고루

모델 생성 및 평가

```
df_classifier = naiveBayes(df_train, df_train_labels)   # 나이브 베이즈 분류
df_test_pred = predict(df_classifier, df_test)          # 위에서 생성된 모델로 분류해 본다
.
CrossTable(df_test_pred, df_test_labels,                # 크로스 테이블을 본다.
           prop.chisq = F, prop.c = F, prop.r = FALSE,
           dnn = c('predicted', 'actual'))
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
## ==================================
##              actual
## predicted        0       1   Total
## ----------------------------------
## 0              886      86     972
##              0.504   0.049
## ----------------------------------
## 1              396     390     786
##              0.225   0.222
## ----------------------------------
## Total         1282     476    1758
## ==================================
```

```
confusionMatrix(as.factor(df_test_pred),          # 더 깔끔하게 본다. 여러 지표를
                as.factor(df_test_labels),
                positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 886  86
##          1 396 390
##
##                Accuracy : 0.7258
##                  95% CI : (0.7043, 0.7466)
##     No Information Rate : 0.7292
##     P-Value [Acc > NIR] : 0.6378
##
##                   Kappa : 0.4237
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8193
##             Specificity : 0.6911
##          Pos Pred Value : 0.4962
##          Neg Pred Value : 0.9115
##              Prevalence : 0.2708
##          Detection Rate : 0.2218
##    Detection Prevalence : 0.4471
##       Balanced Accuracy : 0.7552
##
##        'Positive' Class : 1
##
```

P값이 유의하므로, 모델은 설명력이 있다. 정확도는 0.74 하지만 No Information Rate도 0.7304인 것으로 보아 뛰어난 예측을 하는 것은 아니다. 민감도는 0.8143 특이도는 0.7157이다.

라플라스 적용

```
df_classifier_laplace = naiveBayes(df_train, df_train_labels, laplace = 1)
df_test_pred_laplace = predict(df_classifier_laplace, df_test)

CrossTable(df_test_pred_laplace, df_test_labels,
           prop.chisq = F, prop.c = F, prop.r = FALSE,
           dnn = c('predicted', 'actual'))
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
## ==================================
##              actual
## predicted         0       1   Total
## ----------------------------------
## 0               883      86     969
##              0.502   0.049
## ----------------------------------
## 1               399     390     789
##              0.227   0.222
## ----------------------------------
## Total          1282     476    1758
## ==================================
```

```
confusionMatrix(as.factor(df_test_pred_laplace),
                as.factor(df_test_labels),
                positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 883  86
##          1 399 390
##
##                Accuracy : 0.7241
##                  95% CI : (0.7026, 0.7449)
##     No Information Rate : 0.7292
##     P-Value [Acc > NIR] : 0.696
##
##                   Kappa : 0.4211
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8193
##             Specificity : 0.6888
##          Pos Pred Value : 0.4943
##          Neg Pred Value : 0.9112
##              Prevalence : 0.2708
##          Detection Rate : 0.2218
##    Detection Prevalence : 0.4488
##       Balanced Accuracy : 0.7540
##
##        'Positive' Class : 1
##
```

라플라스를 적용해도 별 차이가 없는 결과가 나왔다. 오히려 실제 0인걸 1로 예측한 값이 2개 늘어났다. 미세하게 안 좋아졌다.

결론

우선 로지스틱 회귀의 결과 민감도는 0.8111, 특이도는 0.6951, F1값은 0.611이다. 또한 P값은 매우 작으므로 이번 회귀분석 모델이 유의미한 결과라고 볼 수 있다. 하지만 정보가 없을 때의 기본 rate가 0.7342인데, Accuracy가 0.726인 것으로 보아 특별히 좋은 모델은 아니다.

나이브 베이즈 회귀의 결과도 마찬가지로 정보가 없는 기본 모델에 비해 특별히 나을 것이 거의 없는 것으로 보인다. 또한 라플라스를 1 적용해도 별다른 차이가 없는 것으로 보아 변수가 0이기 때문에 좋은 결과가 나오지 않는 것은 아닌 것으로 보인다. 라플라스 적용 기준 민감도는 0.8151, 특이도는 0.6966, Accuracy는 0.7287, F1 값은 0.6193396이며 P값은 2e-16이하이다.