kNN

KWONHYUNJIN

2018년 9월 18일

#caret 데이터 학습 라이브러리 #class라이브러리 : train 데이터의 각 행에 대한 범주인 팩터 벡터 #gmodel 패키지: CrossTable(교차표)를 그려주는 패키지 #kknn: knn패키지

#데이터 불러오기

```
bc <- read.xlsx("New_version_breast_cancer.xlsx",1)
str(bc)
```

```
## 'data.frame':    277 obs. of  19 variables:
##  $ NA.          : Factor w/ 277 levels "1","10","100",..: 1 112 201 212 223 234 245 256 267 2 ...
##  $ age          : num  4 5 5 4 4 5 5 4 4 4 ...
##  $ tumor.size   : num  4 4 8 8 7 6 9 3 1 9 ...
##  $ inv.nodes    : num  1 1 1 1 2 2 1 1 1 6 ...
##  $ deg.malig    : num  3 1 2 3 2 2 3 2 2 2 ...
##  $ irradiat     : num  0 0 0 1 0 1 0 0 0 1 ...
##  $ Class        : Factor w/ 2 levels "norecur","recur": 2 1 2 1 2 1 1 1 1 1 ...
##  $ lt40         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ge40         : num  0 1 1 0 0 0 1 0 0 1 ...
##  $ premeno      : num  1 0 0 1 1 1 0 1 1 0 ...
##  $ node.capse   : num  1 0 0 1 1 0 0 0 0 1 ...
##  $ no.node.capse: num  0 1 1 0 0 1 1 1 1 0 ...
##  $ breast.left  : num  0 0 1 0 1 0 1 1 0 0 ...
##  $ breast.right : num  1 1 0 1 0 1 0 0 1 1 ...
##  $ quad.cen     : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ quad.Ll      : num  0 0 1 1 0 0 0 0 0 0 ...
##  $ quad.Lu      : num  1 0 0 0 0 1 1 1 0 1 ...
##  $ quad.Rl      : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ quad.Ru      : num  0 0 0 0 1 0 0 0 0 0 ...
```

#set.seed란 랜덤한 값을 시작하기 전에 사용하면 이후에도 같은 값으로 랜덤값을 갖는다. #seed(123)은 동일한 랜덤값을 계속해서 받기위해 #nrow ㄴ 주어진 데이터 프레임 또는 벡터의 행의 수 또는 길이를 반환하는 함수 #sample 함수는 데이터에서 무작위로 샘플을 추출해주는 함수

```
set.seed(123)
bc_shuffle <- bc[sample(nrow(bc)), ]
head(bc_shuffle)
```

```
##     NA. age tumor.size inv.nodes deg.malig irradiat   Class lt40 ge40
## 80   80   3          1         1         2        0 norecur    0    0
## 218 218   3          5         1         3        1   recur    0    0
## 113 113   4          8         4         2        1 norecur    0    0
## 242 242   5          6         1         1        0 norecur    0    0
## 257 257   5          1         1         2        0 norecur    0    1
## 13   13   5          7         1         1        0 norecur    0    1
##     premeno node.capse no.node.capse breast.left breast.right quad.cen
## 80        1          0             1           0            1        1
## 218       1          0             1           1            0        0
## 113       1          1             0           0            1        0
## 242       1          0             1           1            0        0
## 257       0          0             1           1            0        1
## 13        0          0             1           0            1        1
##     quad.Ll quad.Lu quad.Rl quad.Ru
## 80        0       0       0       0
## 218       0       1       0       0
## 113       0       1       0       0
## 242       1       0       0       0
## 257       0       0       0       0
## 13        0       0       0       0
```

#현재 bc 데이터 프레임에 id라는 컬럼이 필요없다.(그냥 shuffle된거 확인용) #bc_shuffle에서 1번컬럼은 제외하고 나머지 컬럼을 bc2에 할당한다.

```
bc2 <- bc_shuffle[-1]
head(bc2)
```

```
##     age tumor.size inv.nodes deg.malig irradiat   Class lt40 ge40 premeno
## 80    3          1         1         2        0 norecur    0    0       1
## 218   3          5         1         3        1   recur    0    0       1
## 113   4          8         4         2        1 norecur    0    0       1
## 242   5          6         1         1        0 norecur    0    0       1
## 257   5          1         1         2        0 norecur    0    1       0
## 13    5          7         1         1        0 norecur    0    1       0
##     node.capse no.node.capse breast.left breast.right quad.cen quad.Ll
## 80           0             1           0            1        1       0
## 218          0             1           1            0        0       0
## 113          1             0           0            1        0       0
## 242          0             1           1            0        0       1
## 257          0             1           1            0        1       0
## 13           0             1           0            1        1       0
##     quad.Lu quad.Rl quad.Ru
## 80        0       0       0
## 218       1       0       0
## 113       1       0       0
## 242       0       0       0
## 257       0       0       0
## 13        0       0       0
```

#normalize라는 정규화함수를 사용자 정의 함수로 생성한다. #class(label)에 해당하는 행 을 찾음 -> 정규화 시킬때 범주형 데이터를 제외하기 위해서

```
normalize <- function(x) {
  return ( (x-min(x)) / (max(x) - min(x))  )
}

ncol <- which(colnames(bc2) == "Class")
ncol
```

```
## [1] 6
```

```
head(bc2[-ncol])
```

```
##     age tumor.size inv.nodes deg.malig irradiat lt40 ge40 premeno
## 80    3          1         1         2        0    0    0       1
## 218   3          5         1         3        1    0    0       1
## 113   4          8         4         2        1    0    0       1
## 242   5          6         1         1        0    0    0       1
## 257   5          1         1         2        0    0    1       0
## 13    5          7         1         1        0    0    1       0
##     node.capse no.node.capse breast.left breast.right quad.cen quad.Ll
## 80           0             1           0            1        1       0
## 218          0             1           1            0        0       0
## 113          1             0           0            1        0       0
## 242          0             1           1            0        0       1
## 257          0             1           1            0        1       0
## 13           0             1           0            1        1       0
##     quad.Lu quad.Rl quad.Ru
## 80        0       0       0
## 218       1       0       0
## 113       1       0       0
## 242       0       0       0
## 257       0       0       0
## 13        0       0       0
```

#lapply(list or vector,function): 함수를 하나하나에 사용하려 할때 #(굉장히 자주 쓰이는 함수) #as.data.frame: dataframe으로 변환하는 함수 #factor인 label을 제외하고 normalize한다.

```
bc_n <- as.data.frame(lapply(bc2[-ncol],normalize))
head(bc_n)
```

```
##   age tumor.size inv.nodes deg.malig irradiat lt40 ge40 premeno node.capse
## 1 0.2        0.0       0.0       0.5        0    0    0       1          0
## 2 0.2        0.4       0.0       1.0        1    0    0       1          0
## 3 0.4        0.7       0.5       0.5        1    0    0       1          1
## 4 0.6        0.5       0.0       0.0        0    0    0       1          0
## 5 0.6        0.0       0.0       0.5        0    0    1       0          0
## 6 0.6        0.6       0.0       0.0        0    0    1       0          0
##   no.node.capse breast.left breast.right quad.cen quad.Ll quad.Lu quad.Rl
## 1             1           0            1        1       0       0       0
## 2             1           1            0        0       0       1       0
## 3             0           0            1        0       0       1       0
## 4             1           1            0        0       1       0       0
## 5             1           1            0        1       0       0       0
## 6             1           0            1        1       0       0       0
##   quad.Ru
## 1       0
## 2       0
## 3       0
## 4       0
## 5       0
## 6       0
```

#Train 과 Test 를 9:1로 나눈다. #bc_n은 shuffle 을 한 bc2 값에서 label 값을 빼고 정규화 시킨값

```
train_num<-round(0.9*nrow(bc_n),0)
bc_train<-bc_n[1:train_num,]
bc_test<-bc_n[(train_num+1):nrow(bc_n),]
```

#bc2는 bc를 shuffle 한 값 #bc2에서 train 과 test 갯수에 맞게 label 값을 가져와서 bc_oo_label 이라는 변수에 담는다. #결국 bc_train 은 factor bc_train_label 은 해당 factor의 label 을 가지고 있다.

```
bc_train_label <- bc2[1:train_num,ncol]
bc_test_label <- bc2[(train_num+1):nrow(bc_n),ncol]
```

데이터 훈련

#k_n값은 주로 훈련데이터의 제곱근/홀수 #class : train 데이터의 각 행에 대한 범주인 팩터 벡터 #근처에 k갯수의 값을 고르고 다수결에 따라 해당 label 결정 #bc_train 과 bc_test 는 둘다 Label 이 없는 값 #bc_test_pred 은 bc_test 한거에 대한 Label 을 예측한 값을 가지고 있음

```
bc_test_pred <- knn(train=bc_train, test=bc_test, cl= bc_train_label, k = 15 )
bc_test_pred
```

```
##  [1] norecur norecur norecur norecur norecur norecur norecur norecur
##  [9] norecur norecur norecur norecur norecur recur   norecur norecur
## [17] norecur norecur norecur recur   recur   norecur norecur norecur
## [25] norecur norecur recur   norecur
## Levels: norecur recur
```

#예측값(result1)과 실제값(bc_test_label : 정답)의 교차표 생성

```
CrossTable(x=bc_test_label,y=bc_test_pred)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  28
##
##
##                | bc_test_pred
## bc_test_label |   norecur |     recur | Row Total |
## --------------|-----------|-----------|-----------|
##       norecur |        15 |         0 |        15 |
##               |     0.357 |     2.143 |           |
##               |     1.000 |     0.000 |     0.536 |
##               |     0.625 |     0.000 |           |
##               |     0.536 |     0.000 |           |
## --------------|-----------|-----------|-----------|
##         recur |         9 |         4 |        13 |
##               |     0.412 |     2.473 |           |
##               |     0.692 |     0.308 |     0.464 |
##               |     0.375 |     1.000 |           |
##               |     0.321 |     0.143 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |        24 |         4 |        28 |
##               |     0.857 |     0.143 |           |
## --------------|-----------|-----------|-----------|
##
##
```

#confusionMatrix(예측값, 결과값) 함수를 이용하면 정확도 확인

```
confusionMatrix(table(bc_test_label, bc_test_pred))
```

```
## Confusion Matrix and Statistics
##
##              bc_test_pred
## bc_test_label norecur recur
##       norecur      15     0
##       recur         9     4
##
##                Accuracy : 0.6786
##                  95% CI : (0.4765, 0.8412)
##     No Information Rate : 0.8571
##     P-Value [Acc > NIR] : 0.996079
##
##                   Kappa : 0.3226
##  Mcnemar's Test P-Value : 0.007661
##
##             Sensitivity : 0.6250
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.3077
##              Prevalence : 0.8571
##          Detection Rate : 0.5357
##    Detection Prevalence : 0.5357
##       Balanced Accuracy : 0.8125
##
##        'Positive' Class : norecur
##
```