15

비정형데이터분석

텍스트 데이터 분석 사례(2)

통계·데이터과학과 장영재 교수

학습목차

1. 문서-단어행렬과 코사인 유사도
2. 군집분석
3. 분류 분석(classification)

01

문서-단어행렬과 코사인 유사도

1. 문서-단어행렬과 코사인 유사도

- 로이터 코퍼스의 기사들을 이용하여 문서-단어행렬을 작성하고 이를 통해 코사인 유사도를 산출
  - 문서-단어행렬을 작성하기 전에 두 리스트 Reut_topics와 Reut_content를 외환 관련 기사가 먼저 오고 금리 관련 기사가 나중에 오도록 다시 정렬

```
> Reut_topics <- Reut_topics[c(which(Reut_topics=="money-fx"), which
(Reut_topics=="interest"))]
> Reut_content <- Reut_content[c(which(Reut_topics=="money-fx"), which
(Reut_topics=="interest"))]
```

1. 문서-단어행렬과 코사인 유사도

1 문서-단어행렬의 작성

- 전처리된 로이터 코퍼스의 기사들은 Reut_content 리스트에 저장되어 있으므로 이 리스트를 이용하여 문서-단어행렬 Reut_DTM을 작성
  - 리스트 형태를 행렬로 변환

```
> Reut_DTM <- lapply(Reut_content, FUN = function(x, lev){table(factor(x, lev
, ordered = T))}, lev = Reut_lev )
> Reut_DTM <- matrix(unlist(Reut_DTM), nrow = length(Reut_DTM), byrow =
TRUE)
> colnames(Reut_DTM) <- Reut_lev
```

1. 문서-단어행렬과 코사인 유사도

2 코사인 유사도 계산

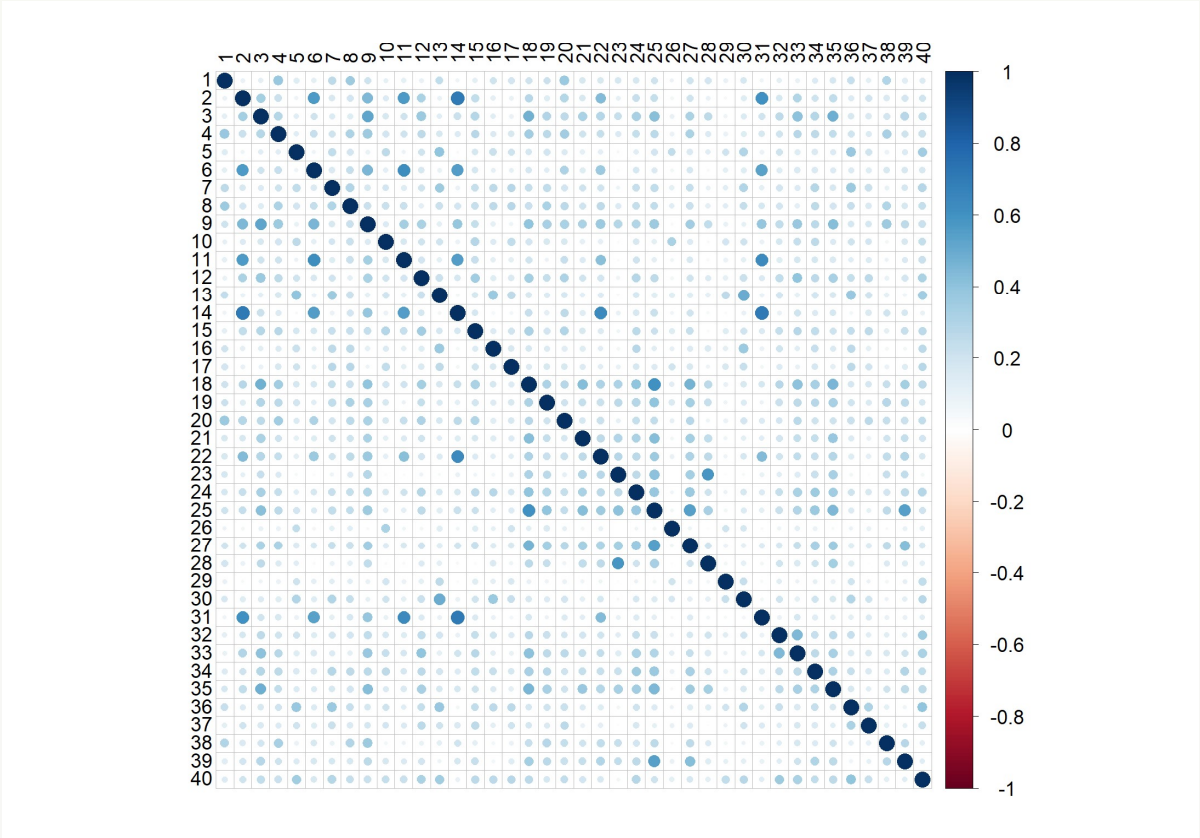
- 기사들간 단어 사용의 유사성을 살펴보기 위해 코사인 유사도 행렬 작성
  - 내적은 문서-단어행렬과 그 전치행렬의 곱이며 이 곱행렬에서 대각원소들은 기사 벡터들의 길이의 제곱에 해당

```
> Reut_DTMsqr <- Reut_DTM %*% t(Reut_DTM)
> Reut_CosSim <- Reut_DTMsqr / sqrt(diag(Reut_DTMsqr) %*% t(diag
(Reut_DTMsqr)))
```

→ 외환과 금리 관련 기사들 중에서 각각 20건의 기사를 랜덤추출하여 코사인 유사도 부분행렬을 시각화

1. 문서-단어행렬과 코사인 유사도

```
> set.seed(1)
> FXsample <- sort(sample.int(n=sum(Reut_topics=="money-fx"), size=20))
> INTsample <- sort(sample.int(n=sum(Reut_topics=="interest"), size=20))
> smpl <- c(FXsample, INTsample+sum(Reut_topics=="money-fx"))
> library(corrplot)
> corrplot(Reut_CosSim[smpl, smpl])
```


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

1. 문서-단어행렬과 코사인 유사도

- 코사인 유사도 행렬의 부분행렬 중 1~20 행과 열은 외환 관련 기사에 해당되고 21~40 행과 열은 금리 관련 기사에 해당

→ 1~20과 21~40 사이의 기사들이 대체로 코사인 유사도가 높음

02

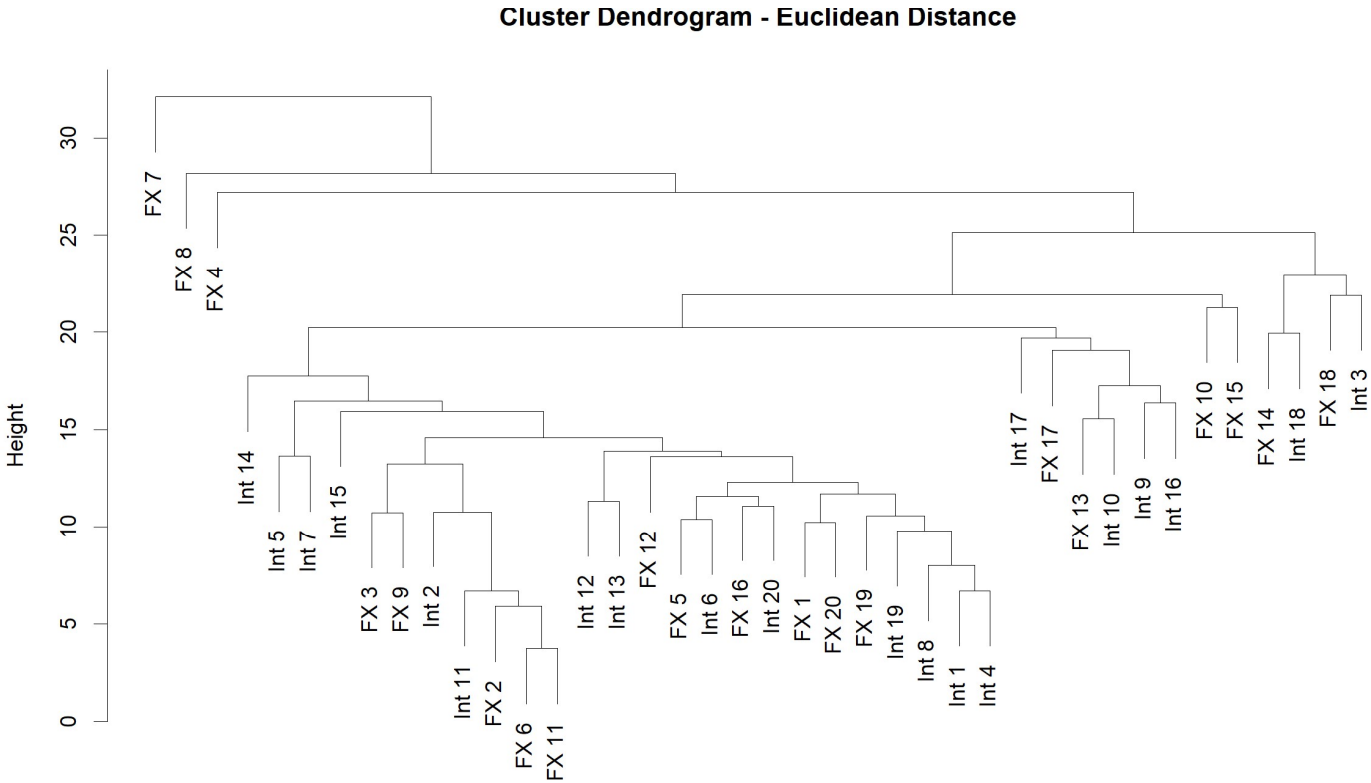
군집분석

2. 군집분석

1 유클리드 거리를 이용한 응집분석

- 유클리드 거리를 이용하여 작성한 거리행렬을 기준으로 군집분석 실시
  - 랜덤추출된 40건의 기사만으로 분석

```
> Reut_euclidean <- hclust(dist(Reut_DTM[smpl,]))
> Reut_euclidean$labels <- c(paste("FX", c(1:20)), paste("Int", c(1:20)))
# 기사 구별을 위해
> plot(Reut_euclidean, main = "Cluster Dendrogram - Euclidean Distance",
xlab="", sub="")
```



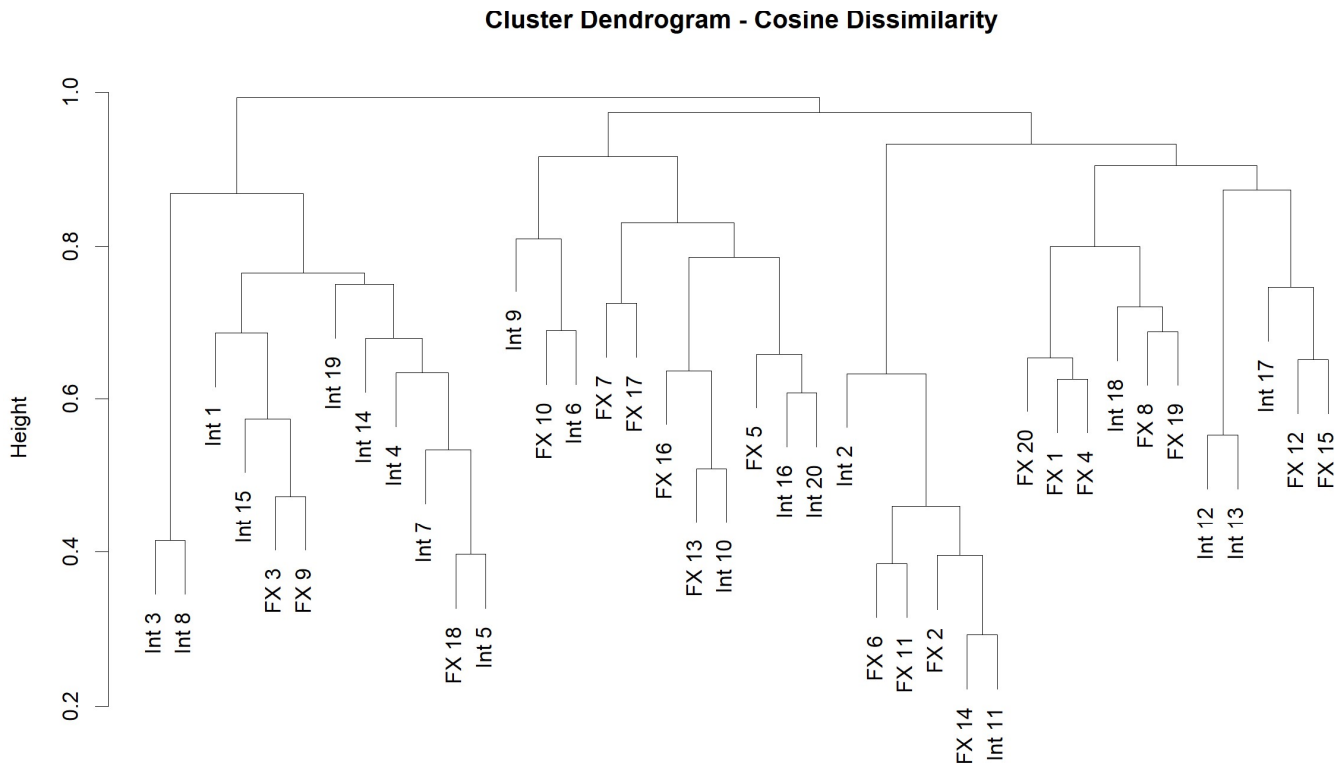
<그림> 유클리드 거리를 이용한 계층적 군집분석 결과(응집분석, 완전연결법)

2. 군집분석

2 코사인 비유사성 행렬을 이용한 응집분석

- 코사인 비유사성 행렬을 기준으로 거리를 측정하였을 때의 군집분석
  - 랜덤추출된 40건 기사의 코사인 비유사성 행렬을 구하고 행렬의 변수 타입을 dist로 변환하기 위해 as.dist( ) 함수를 이용

```
> Reut_clusters <- hclust(as.dist(1-Reut_CosSim[smpl, smpl]))
> Reut_clusters$labels <- c(paste("FX", c(1:20)), paste("Int", c(1:20)))
> plot(Reut_clusters, main = "Cluster Dendrogram - Cosine Dissimilarity", xlab="",
sub="")
```



<그림> 코사인 비유사성 행렬을 이용한 계층적 군집분석 결과(응집분석, 완전연결법)

03

분류분석(classification)

3. 분류분석(classification)

분류나무모형을 이용하여 기사들을 분류

- 통상적인 통계모형에 비해 많은 변수로 이루어져 있어 변수선택 과정이 필요

→ 두 주제의 기사들에서 출현빈도의 차이가 큰 단어들을 선택하기 위해 Reut_fx_int_mat[,1] - Reut_fx_int_mat[,2]에 abs( ) 함수를 적용하여 60이상인 단어의 열을 선택

→ 문서-단어행렬에서 선택된 행렬, 즉 Reut_selected 행렬에 포함되어 있는 단어들만으로 구성된 부분행렬을 작성

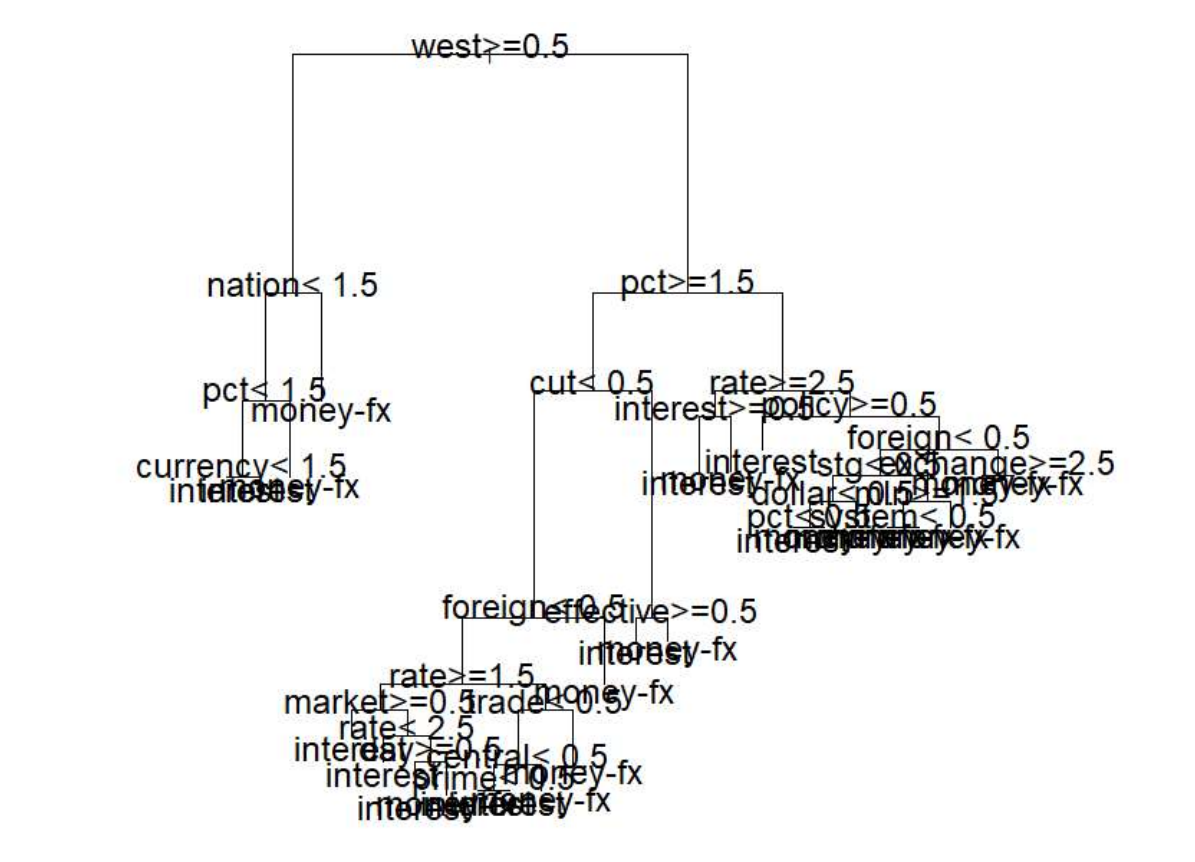
```
> Reut_selected <- Reut_fx_int_mat[abs(Reut_fx_int_mat[,1] - Reut_fx_int_mat[,2]) >= 60,]
> rownames(Reut_selected)
> Reut_selected_DTM <- Reut_DTM[, colnames(Reut_DTM) %in% rownames(Reut_selected)]
```

3. 분류분석(classification)

1 분류나무모형의 작성

- 부분행렬 Reut_selected_DTM을 이용하여 분류나무모형을 작성
  - minsplit = 30 : 집단 내의 기사 수가 30건 미만인 노드는 분할 제외
  - cp = -0.01 : 비용-복잡함수가 최소가 되는 점 이후에도 계속 분할
  - xval = 10 : 10겹-교차검증(10-fold CV) 실시

```
> library(rpart)
> ctrl <- rpart.control(minsplit = 30, cp = -0.01, xval = 10)
> fit_tree <- rpart(unlist(Reut_topics) ~ ., data = data.frame(Reut_selected_DTM),
method = "class", control = ctrl)
> plot(fit_tree)
> text(fit_tree)
```



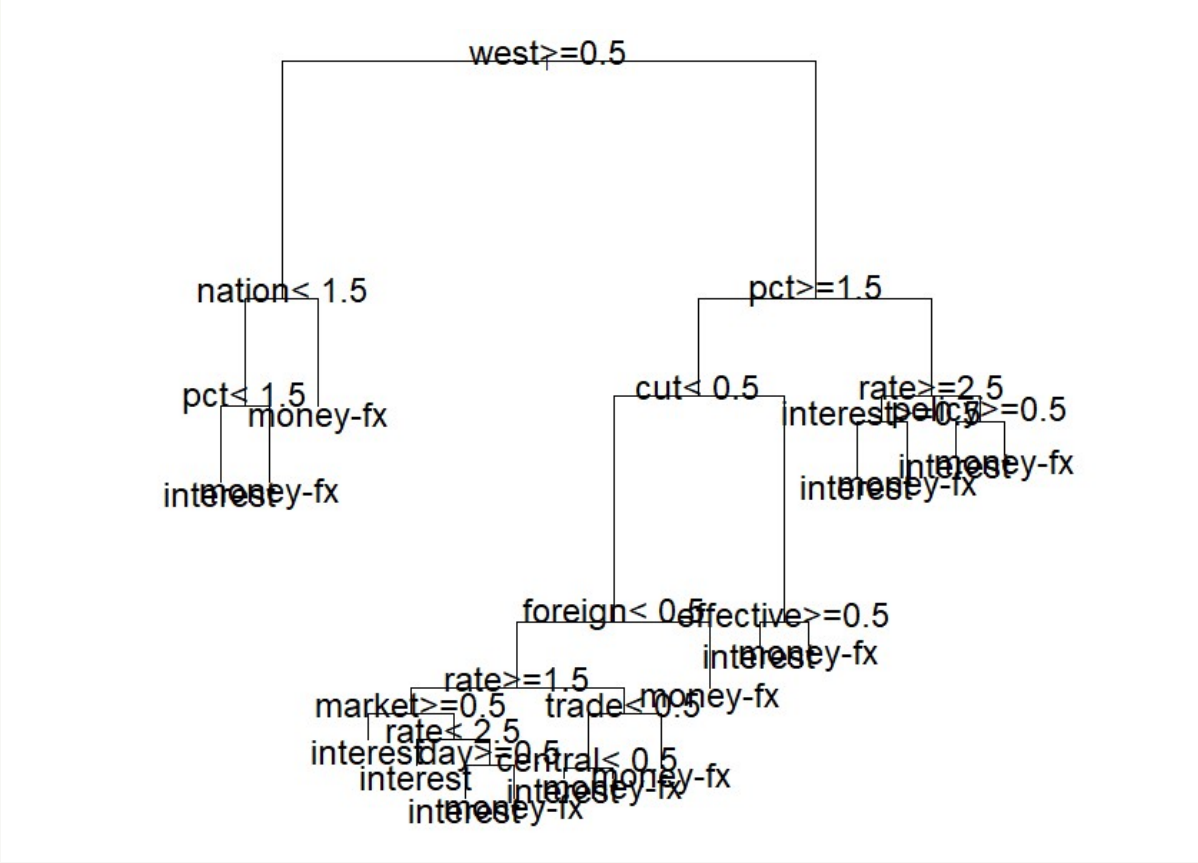
<그림> 분류나무모형(가지치기 수행 이전)

3. 분류분석(classification)

2 분류나무모형의 가지치기

- 분류모형이 지나치게 복잡한 경우에는 한 가지치기를 통해 필요 이상으로 분할된 노드들을 정리
  - 분류나무의 가지치기를 위해서 prune( ) 함수를 이용

```
> prune_tree <- prune(fit_tree, cp=0)
> length(fit_tree$frame$var)
[1] 51
> sum(fit_tree$frame$var == "<leaf>")
[1] 26
> length(prune_tree$frame$var)
[1] 33
> sum(prune_tree$frame$var == "<leaf>")
[1] 17
> plot(prune_tree)
> text(prune_tree)
```



<그림> 분류나무모형(가지치기 수행 후)

3. 분류분석(classification)

3 분류분석 결과

- 분류나무모형을 사용한 분류분석 결과의 오분류율 산출
  - predict( ) 함수로 예측치를 구하고 table( ) 함수로 교차표 작성

```
> pred <- predict(prune_tree, type="class")
> table(pred)
> confmat <- table(unlist(Reut_topics), pred)
> confmat
```

3. 분류분석(classification)

- 정오분류표를 이용하여 분류모형 성능을 평가(편의상 금리 기사를 양(+)으로 외환 기사를 음(-)으로 간주)

		예측 결과	
		interest	money-fx
실제 값	interest	138 (TP)	73 (FN)
	money-fx	43 (FP)	216 (TN)

3. 분류분석(classification)

→ 다음과 같이 정확도(precision), 민감도(sensitivity)와 특이도(specificity)를 산출하여 평가할 수 있음

$$\text{Precision} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{138 + 216}{138 + 43 + 73 + 216} = \frac{354}{470} = 0.753$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{138}{138 + 73} = 0.654$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{216}{216 + 43} = 0.834$$

→ 민감도는 0.654로, 특이도는 0.834로 나타나 외환 기사에 대한 분류가 금리 기사에 비해 상대적으로 더 정확하게 이루어졌음을 시사

실습하기

강의를 마쳤습니다.

끝

한 학기 동안
수고 많으셨습니다.