

데이터예측모델과 기계학습의 응용

IC-PBL 최종 보고서

3조(이윤주, 이윤지, 임지원, 정주은, 최인열)

1. 프로젝트

독서 어플 '리더스'를 사용하는 고객의 독서 관련 데이터 중 이용자가 스크랩한 문구 데이터를 바탕으로 '데이터 예측모델과 기계학습의 응용' 수업 시간에 배운 기계학습 모델과 텍스트 분석 방법을 활용하여 이용자에게 도서를 추천해주는 새로운 서비스 제안

이용자가 스크랩한 문구를 해시태그화하여 해당 해시태그를 바탕으로 팔로우 관계, 도서의 페이지 수, 유저의 직종, 유저의 관심있는 카테고리를 기준으로 정렬하여 다른 도서를 추천해주는 서비스 제안

2. 과정

① 해시태그화

[1-1] 스크랩 데이터를 이용한 스크랩 문구 해시태그화

```
library(KoNLP)
library(tidytext)
library(readxl)
library(tidyverse)
pkg_l <- c("tidyverse", "tidytext", "textdata")
purrr::map(pkg_l, require, ch = T)
library(stringr)
library(writexl)
library(dplyr)
library(reshape2)
library(arules)
library(arulesViz)
library(plyr)
library(dplyr)
```

▲ 라이브러리 부착

```
user_df <- read_xlsx("C:\\Users\\user\\Downloads\\01_user.xlsx")
suser_cat_df <- read_xlsx("C:\\Users\\user\\Downloads\\02_user_cat.xlsx")
follow_df <- read_xlsx("C:\\Users\\user\\Downloads\\03_follow.xlsx")
user_book_df <- read_xlsx("C:\\Users\\user\\Downloads\\04_user_book.xlsx")
book_df <- read_xlsx("C:\\Users\\user\\Downloads\\05_book.xlsx")
book_cat_df <- read_xlsx("C:\\Users\\user\\Downloads\\06_book_cat.xlsx")
cat_list_df <- read_xlsx("C:\\Users\\user\\Downloads\\07_cat.xlsx")
scrap <- read_xlsx("C:\\Users\\user\\Downloads\\08_scrap.xlsx")
```

▲ 데이터 불러오기

```
scrap %>% select(user_id, content, book_id) -> scrap
```

▲ 스크랩 데이터에서 필요하지 않은 열 제거

```
table(is.na(scrap))  
scrap %>% na.omit(scrap) -> scrap
```

▲ 스크랩 데이터에서 결측치 확인 후 결측치가 있는 행 제거

```
scrap %>% unnest_tokens(pos, content, token = SimplePos09) %>% group_by(user_id) %>% mutate(pos_order = 1:n()) -> scrap  
scrap %>% filter(str_detect(pos, "/n")) %>% mutate(pos_done = str_remove(pos, "/.*$")) -> n_done  
n_done %>% select(user_id, book_id, page, pos_done) -> scrap
```

▲ 스크랩 데이터의 content 토큰화 후 명사 추출

```
scrap %>% select(pos_done, user_id, book_id) -> scrap
```

▲ 단어, 유저 아이디, 북 아이디 순으로 열 순서 변경

```
scrap[order(scrap$pos_done) , ] -> scrap2
```

▲ 정렬 기준을 단어로 하여 새롭게 정렬 후, 새로운 데이터 프레임 생성

```
gsub("[[:punct:]]", "", scrap2$pos_done) -> scrap2$pos_done
```

▲ 스크랩 데이터 내 특수문자 제거

```
scrap2 %>% filter(str_length(pos_done) > 1) -> scrap2
```

▲ 데이터 내 한 글자 이하의 단어 제거

```
scrap2[duplicated(scrap2$pos_done),] -> scrap2
```

▲ 한 명만 스크랩한 문구의 행 제거

```
write_xlsx(scrap2, path = "scrap2.xlsx")
```

▲ 명사가 아닌 데이터를 수동으로 제거하고자 엑셀로 저장

```
remove_word <- c("빨강머리앤은", "오멜라스를", "정반합적",  
"65", "91", "그래서", "그저", "그냥",  
"아직", "언젠가", "놓치고", "다른",  
"따라서", "무슨", "박완서朴婉緒",  
"살다보니까", "성연", "량을", "어떤",  
"윤세경李珮瑩", "이런", "이영민李榮敏디자이너",  
"이세령李瑟玲", "생각", "임용한林容漢배낭",  
"전병식金炳植", "승연趙升衍시크하다",  
"조영길鄭榮吉애", "좋은", "챗터16", "챗터18",  
"최낙언崔洛彦", "콩트갈은", "함재봉咸在鳳한국사람",  
"— — — — —", "— — — — —", "— — — — —", "— — — — —",  
"— 어두워진다는", "알프레드", "— 안소", "— 사랑",  
"— 내가", "— 내", "— 난", "π π π π π", "π π π π π",  
"π π π π π", "π π π π π", "π π π π π", "π π π π π",  
"힘줄내라", "는", "힘들이", "힘들", "힘들", "힘들", "힘들",  
"힘들", "힘들", "힘들", "힘들", "힘들", "힘들", "힘들",  
"힐링이", "힐링을", "히틀러와", "히틀러를", "히틀러는",  
"히치하이킹을", "히치하이커", "히브리서heb",  
"히로시마에", "흰머리가", "희한한", "희진을",  
"희정님이", "희소한", "희석화", "희생자들이",  
"희박해", "희박한", "희미해지", "희미한", "희망찬",  
"희망적", "희망적", "희망적", "희망적"
```

```
remove_word <- as.data.frame(remove_word)
```

```
word <- read_excel("remove.xlsx")
```

	A
1	pos_done
2	01
3	08
4	00
5	011
6	02
7	03
8	04
9	05
10	06

```
colnames(word) <- "remove_word"
word
remove_word <- rbind(remove_word, word)
rm(word)
names(remove_word) <- c("pos_done")
```

```
anti_join(scrap2, remove_word, by = "pos_done") -> scrap3
```

▲ remove_word에 있던 데이터를 제거하여 조인

```
scrap3[duplicated(scrap3$pos_done),] -> scrap3
```

▲ 한 명만 스크랩한 문구 다시 제거

```
scrap3[order(scrap3$pos_done) , ] -> scrap3
```

▲ 문구를 기준으로 다시 정렬

```
scrap3[duplicated(scrap3$pos_done),] -> scrap3
```

▲ 한 명만 스크랩한 문구가 남아있어 다시 제거

	pos_done	user_id	book_id
1	20세기	163354282	1924965
2	21세기	343041958	110011
3	가게	1117834	16693
4	가게	203158186	872787
5	가게	87278182	522675
6	가게	87278182	522675
7	가게	67639072	502121
8	가게	188690770	410
9	가게	40507114	1475628
10	가게	178517674	19632
11	가게	8866120	64
12	가게	48855124	390911
13	가게	48855124	4787
14	가게	343041958	30
15	가게	29475154	79451
16	가게	8673616	1791
17	가게	8673616	1791
18	가게	8673616	1791
19	가게	8673616	1791
20	가게	8673616	1566
21	가게	8673616	1566
22	가게	208218820	920525
23	가게	208218820	920525
24	가게	383786643	838535

▲ 코드 실행 후 결과 화면

[1-2] 해시태그 단어 선택 시나리오

```
scrap3 %>% filter(pos_done == "가치") -> scrap4
```

```
> scrap4
# A tibble: 1,156 × 4
  user_id book_id page pos_done
  <dbl>   <dbl> <dbl> <chr>
1 69249442     918   326 가치
2 69249442    303741 168 가치
3 69249442    904339 168 가치
4 69249442    174150   95 가치
5 69249442     90876  202 가치
6 69249442    552825   81 가치
7 69249442    966054   63 가치
8 69249442    306336  186 가치
9 69249442     7444   98 가치
10 227983798    60884  162 가치
# ... with 1,146 more rows
```

▲ 스크랩한 문구 해시태그 중 '가치'라는 단어를 골랐다고 가정

```
scrap4[!duplicated(scrap4),] -> scrap4
```

▲ 완전히 중복되는 값을 제외

② 전처리

[2-1] 읽은 도서들의 평균 페이지 데이터 프레임 생성

```
book_df %>% select(id,page) -> book_page
user_book_df <- left_join(user_book_df,book_page,by = c("book_id" = "id"))
tapply(user_book_df$page, user_book_df$user_id, mean,na.rm=T) -> user_pagelength
user_pagelength <- as.data.frame(user_pagelength)
user_pagelength_row <- rownames(user_pagelength)
user_pagelength_row <- as.data.frame(user_pagelength_row)
user_pagelength <- cbind(user_pagelength_row,user_pagelength)
rownames(user_pagelength) <- NULL
colnames(user_pagelength) <- c("user_id","avg_bookpage")
user_pagelength
```

```
> user_pagelength
  user_id avg_bookpage
1    10936    353.36607
2    22042    436.29268
3    29446    376.76533
4    73870    376.96748
5   414454    377.68939
6   532918    344.49038
7   536620    352.81000
8   592150    372.84483
9   629170    338.48760
10  917926    365.28000
```

▲ 각 유저별 읽은 책의 평균 페이지 데이터 프레임 생성

[2-2] 스크랩 데이터 전처리

* 현재 이용자의 user_id가 290503174라고 가정하고 짤 추천 시스템 시나리오 코드입니다.

```
scrap5 <- scrap4 %>% filter(user_id != 290503174)
```

▲ 스크랩 데이터에서 현재 이용자 자신의 데이터 제거

```
my_book <- user_book_df %>% filter(user_id == 290503174) %>% select(book_id)
scrap5 <- anti_join(scrap5, my_book, by=c("book_id"="book_id"))
scrap6 <- scrap5
```

▲ 현재 이용자 책장의 도서 목록과 스크랩 데이터를 비교하여, 현재 이용자의 책장에 이미 있는 도서는 제외

	pos_done	user_id	book_id
1	가치	69249442	303741
2	가치	69249442	904339
3	가치	69249442	174150
4	가치	69249442	90876
5	가치	69249442	552825
6	가치	69249442	966054
7	가치	69249442	306336
8	가치	69249442	7444
9	가치	227963798	60884
10	가치	112233364	4167
11	가치	112233364	629832
12	가치	115487422	1198350
13	가치	115487422	17143
14	가치	132668404	131070
15	가치	94674778	30773
16	가치	94674778	11675
17	가치	103359670	37950
18	가치	103359670	198580
19	가치	103359670	925840
20	가치	103359670	1822
21	가치	293997862	930761
22	가치	293997862	1886673
23	가치	293997862	1922490

▲ 코드 실행 후 결과 화면

③ 책 추천을 위한 요소

[3-1] 팔로우 관계를 파악하는 열 생성

```
follow_df %>% filter(user_id == 290503174) %>% select(target_user_id) -> follower_current_user
follower_current_user$following <- "following user"
scrap6 <- left_join(scrap6, follower_current_user, by = c("user_id" = "target_user_id"))
scrap6$following <- ifelse(is.na(scrap6$following), 0, 1)
```

▲ follow_df 데이터와 scrap 데이터를 결합하여 팔로우 여부를 확인하는 코드 생성

(팔로우한 유저면 1점, 아니면 0점 부여)

	pos_done	user_id	book_id	following
1	가치	3668512	522	1
2	가치	13745356	908	1
3	가치	13745356	621	1
4	가치	255608122	2986	1
5	가치	154410250	5581	1
6	가치	337277944	8025	1
7	가치	26731972	780191	1
8	가치	26731972	1522	1
9	가치	26731972	177192	1
10	가치	26731972	117798	1
11	가치	26731972	3282	1
12	가치	26731972	1386336	1
13	가치	26731972	1186	1
14	가치	26731972	48674	1
15	가치	61393798	304188	1
16	가치	61393798	1285858	1
17	가치	61393798	22577	1
18	가치	61393798	199368	1
19	가치	61393798	925693	1
20	가치	61393798	915731	1
21	가치	61393798	402902	1
22	가치	61393798	402481	1
23	가치	61393798	3533	1

▲ 코드 실행 후 결과 화면

[3-2] 페이지 선호도를 반영한 page_diff 열 생성

*현재 이용자가 읽었던 책의 평균 페이지는 325.5742 페이지 입니다.

```
user_pagelength %>% filter(user_id == 290503174)
```

▲ 현재 이용자의 읽었던 책의 평균 페이지 정보

```
book_df %>% select(id,page) -> book_page
scrap6 <- left_join(scrap6,book_page,by=c("book_id" = "id"))
scrap6$page_diff <- abs(scrap6$page - 325.5742)
scrap6$page <- NULL
```

▲ scrap과 book_df의 page열을 결합한 후, 현재 이용자의 읽은 도서들의 평균 page를 뺀 값을 구한다.

```
scrap6$page_diff <- ifelse(scrap6$page_diff <= 50 ,1,ifelse(scrap6$page_diff<=150,0.5,0))
```

▲ page_diff를 범주화하여(50페이지 이내로 차이나면 1점, 150페이지 이내로 차이나면 0.5점,그이상은 0점을 부여한다.)

	pos_done	user_id	book_id	following	page_diff
1	가치	69249442	303741	0	0.5
2	가치	69249442	904339	0	0.5
3	가치	69249442	174150	0	1.0
4	가치	69249442	90876	0	0.5
5	가치	69249442	552825	0	1.0
6	가치	69249442	966054	0	1.0
7	가치	69249442	306336	0	0.0
8	가치	69249442	7444	0	0.0
9	가치	227983798	60884	0	1.0
10	가치	112233364	4167	0	0.5
11	가치	112233364	629832	0	1.0
12	가치	115487422	1198350	0	0.5
13	가치	115487422	17143	0	1.0
14	가치	132668404	131070	0	1.0
15	가치	94674778	30773	0	0.0
16	가치	94674778	11675	0	0.5
17	가치	103359670	37950	0	0.5
18	가치	103359670	198580	0	0.5
19	가치	103359670	925840	0	1.0
20	가치	103359670	1822	0	0.5
21	가치	293997862	930761	0	0.5
22	가치	293997862	1886673	0	0.5
23	가치	293997862	1922490	0	1.0

▲ 코드 실행 후 결과 화면

[3-3] 같은 직종인지의 여부를 파악하는 열 생성

*현재 이용자의 직종은 교육 직종의 직장인이므로 equal 조건을 교육으로 설정

```
user_df %>% select(user_id, 직종) -> user_job
scrap6 <- join(scrap6, user_job)
user_job %>% filter(user_id == 290503174)
scrap6$equal_job <- ifelse(is.na(scrap6$직종), 0, ifelse(scrap6$직종 == "교육", 1, 0))
scrap6$직종 <- NULL
```

▲ 스크랩 데이터에 현재 이용자와 같은 직업을 가진 사람인지 확인(같은 직종이라면 1점, 아니면 0점 부여)

	pos_done	user_id	book_id	following	page_diff	equal_job
1	가치	69249442	303741	0	0.5	0
2	가치	69249442	904339	0	0.5	0
3	가치	69249442	174150	0	1.0	0
4	가치	69249442	90876	0	0.5	0
5	가치	69249442	552825	0	1.0	0
6	가치	69249442	966054	0	1.0	0
7	가치	69249442	306336	0	0.0	0
8	가치	69249442	7444	0	0.0	0
9	가치	227983798	60884	0	1.0	0
10	가치	112233364	4167	0	0.5	0
11	가치	112233364	629832	0	1.0	0
12	가치	115487422	1198350	0	0.5	0
13	가치	115487422	17143	0	1.0	0
14	가치	132668404	131070	0	1.0	0
15	가치	94674778	30773	0	0.0	1
16	가치	94674778	11675	0	0.5	1
17	가치	103359670	37950	0	0.5	0
18	가치	103359670	198580	0	0.5	0
19	가치	103359670	925840	0	1.0	0
20	가치	103359670	1822	0	0.5	0
21	가치	293997862	930761	0	0.5	1
22	가치	293997862	1886673	0	0.5	1
23	가치	293997862	1922490	0	1.0	1

▲ 코드 실행 후 결과 화면

[3-4] 카테고리 유사도를 파악하는 열 생성

```
head(user_cat_df, 10)
user_cat_df <- read_xlsx("02_user_cat.xlsx")
user_cat_df %>% dcast(user_id ~ title) -> usercat
usercat %>% select(-user_id) -> usercat2
usercat2 <- ifelse(is.na(usercat2), 0, 1)
usercat %>% select(user_id) %>% cbind(usercat2) -> usercat
str(usercat)
```

▲ User_cat 성분 행렬

```
scrap6 %>% select(user_id) -> target_user
target_user[!duplicated(target_user),] -> target_user
target_usercat <- inner_join(usercat, target_user)
str(target_usercat)
```

▲ 이 성분 행렬에 현재 해시태그를 스크랩한 사람들의 user_id를 조인하여 해당 유저들의 관심 카테고리 성분 행렬만 남김

	user_id	건강	결혼	경제/경영	교육	기타	다이어트	데이터분석	디자인	마케팅	목표달성	문학	반려동물	부동산	사회/정치	스타트업	습관	시	언어	에세이	여행	영어	예술	요리	운동/레저	육아	음악	인공지능	
1	29446	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	
2	73870	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	
3	592150	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	
4	1117834	1	0	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	
5	2761522	1	1	1	1	0	1	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	
6	3612982	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	
7	3668512	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1	
8	6611602	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
9	7103968	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	1	
10	8170144	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	
11	8410774	0	0	1	0	0	0	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	1	1	
12	8673616	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
13	8866120	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	
14	8925352	0	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	0	1	0	0	
15	9476950	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	0	
16	9591712	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
17	13501024	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	
18	13515832	1	1	1	0	0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0
19	13745356	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
20	14522776	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

▲ 코드 실행 후 결과 화면

```
target_usercat$cat_total <- 0
target_usercat$not_choose <- 0
```

▲ Sum_row값을 초기화 시키기 위한 임시 도구

```
usercat %>% filter(user_id == 290503174) -> current_user_cat
target_usercat$cat_total <- target_usercat %>% select(-user_id) %>% rowSums()
target_usercat$not_choose <- target_usercat %>% select(-c(건강, 교육, 목표달성, 문학, 습관, 언어, 여행, 영어, 육아, 음악, 인문학, 자기계발, 커뮤니케이션, user_id, cat_total)) %>% rowSums()
target_usercat$choose <- target_usercat$cat_total - target_usercat$not_choose
```

▲ 현재 이용자는 관심 카테고리로 건강, 교육, 목표달성, 문학, 습관, 언어, 여행, 영어, 육아, 음악, 인문학, 자기계발, 커뮤니케이션의 13개 카테고리를 선택했음.

```
target_usercat$cat_similarity <- (target_usercat$choose / target_usercat$cat_total) * 100
target_usercat %>% select(user_id, cat_similarity) -> usercat_similarity
usercat_similarity
```

▲ 선택 카테고리 유사도 = (이용자의 관심 카테고리와 일치하는 선택 카테고리 개수) / (전체 카테고리 선택 개수) * 100 로 정의

	user_id	cat_similarity
1	29446	60.00000
2	73670	42.85714
3	592150	33.33333
4	1117834	47.61905
5	2761522	50.00000
6	3612982	57.14286
7	3668512	38.46154
8	6611602	50.00000
9	7103968	57.14286
10	8170144	57.14286
11	8410774	46.66667
12	8673616	66.66667
13	8866120	60.00000
14	8925352	57.14286
15	9476950	38.46154
16	9591712	33.33333
17	13501024	40.00000
18	13515832	33.33333
19	13745356	55.55556
20	14522776	44.82759
21	17913808	50.00000
22	18206266	75.00000
23	18532042	40.00000

▲ 코드 실행 후 결과 화면

```
scrap6 <- left_join(scrap6, usercat_similarity)
scrap6$cat_similarity <- ifelse(scrap6$cat_similarity < 50, 0, ifelse(scrap6$cat_similarity < 75, 1, 2))
```

- ▲ 얻은 카테고리 유사도 데이터를 scrap 데이터와 결합한 후, 유사도가 50% 미만이면 0점, 75% 미만이면 1점, 75% 이상이면 2점을 부여

	pos_done	user_id	book_id	following	page_diff	equal_job	cat_similarity
1	가치	69249442	303741	0	0.5	0	2
2	가치	69249442	904339	0	0.5	0	2
3	가치	69249442	174150	0	1.0	0	2
4	가치	69249442	90876	0	0.5	0	2
5	가치	69249442	552825	0	1.0	0	2
6	가치	69249442	966054	0	1.0	0	2
7	가치	69249442	306336	0	0.0	0	2
8	가치	69249442	7444	0	0.0	0	2
9	가치	227983798	60864	0	1.0	0	1
10	가치	112233364	4167	0	0.5	0	1
11	가치	112233364	629832	0	1.0	0	1
12	가치	115487422	1198350	0	0.5	0	0
13	가치	115487422	17143	0	1.0	0	0
14	가치	132666404	131070	0	1.0	0	0
15	가치	94674778	30773	0	0.0	1	1
16	가치	94674778	11675	0	0.5	1	1
17	가치	103359670	37950	0	0.5	0	0
18	가치	103359670	198580	0	0.5	0	0
19	가치	103359670	925840	0	1.0	0	0
20	가치	103359670	1822	0	0.5	0	0
21	가치	293997862	930761	0	0.5	1	1
22	가치	293997862	1886673	0	0.5	1	1
23	가치	293997862	1922490	0	1.0	1	1

▲ 코드 실행 후 결과 화면

[3-5] 책 카테고리 통합

```
book_cat_df
book_cat_list <- ddply(book_cat_df, c("book_id"), function(df1) paste(df1$book_category_id, collapse=","))
str(book_cat_list)
write.csv(book_cat_list, "bookcat_transaction.csv", fileEncoding = "EUC-KR", row.names = FALSE, quote = FALSE)
tr2 <- read.transactions("bookcat_transaction.csv", format = "basket", sep = ",", skip = 1)
cat_apriori <- apriori(tr2, support = 0.01, confidence = 0.8, minlen = 5)
summary(cat_apriori)
inspect(cat_apriori)
```

▲ #book_cat를 장바구니 분석하여 책의 카테고리를 하나로 통일하는 코드

*카테고리의 depth와 세분화된 분류 이슈로 인하여 구현 실패

*카테고리 축소 필요

*depth_2를 기준으로 장바구니 분석을 하여 책의 카테고리를 하나로 통일하는 방법 고안

```
> inspect(cat_apriori)
```

	lhs		rhs	support	confidence	coverage	lift	count
[1]	{2811, 2812, 2840, 2844}	=>	{2839}	0.01049295	1.00000000	0.01049295	21.036214	1152
[2]	{2812, 2839, 2840, 2844}	=>	{2811}	0.01049295	1.00000000	0.01049295	21.804965	1152
[3]	{2811, 2812, 2839, 2844}	=>	{2840}	0.01049295	1.00000000	0.01049295	22.282931	1152
[4]	{2811, 2839, 2840, 2844}	=>	{2812}	0.01049295	0.9991327	0.01050206	22.767285	1152
[5]	{2811, 2812, 2840, 2844}	=>	{2767}	0.01049295	1.00000000	0.01049295	20.181618	1152
[6]	{2767, 2812, 2840, 2844}	=>	{2811}	0.01049295	1.00000000	0.01049295	21.804965	1152
[7]	{2767, 2811, 2812, 2844}	=>	{2840}	0.01049295	1.00000000	0.01049295	22.282931	1152
[8]	{2767, 2811, 2840, 2844}	=>	{2812}	0.01049295	0.9991327	0.01050206	22.767285	1152
[9]	{2812, 2839, 2840, 2844}	=>	{2767}	0.01049295	1.00000000	0.01049295	20.181618	1152
[10]	{2767, 2812, 2840, 2844}	=>	{2839}	0.01049295	1.00000000	0.01049295	21.036214	1152
[11]	{2767, 2812, 2839, 2844}	=>	{2840}	0.01049295	1.00000000	0.01049295	22.282931	1152
[12]	{2767, 2839, 2840, 2844}	=>	{2812}	0.01049295	0.9335494	0.01123984	21.272836	1152
[13]	{2811, 2812, 2839, 2844}	=>	{2767}	0.01049295	1.00000000	0.01049295	20.181618	1152
[14]	{2767, 2811, 2812, 2844}	=>	{2839}	0.01049295	1.00000000	0.01049295	21.036214	1152
[15]	{2767, 2811, 2840, 2844}	=>	{2839}	0.01049295	1.00000000	0.01049295	21.036214	1152
[16]	{2767, 2811, 2839, 2844}	=>	{2812}	0.01049295	0.9991327	0.01050206	22.767285	1152
[17]	{2811, 2839, 2840, 2844}	=>	{2767}	0.01050206	1.00000000	0.01050206	20.181618	1153
[18]	{2767, 2811, 2840, 2844}	=>	{2839}	0.01050206	1.00000000	0.01050206	21.036214	1153
[19]	{2767, 2839, 2840, 2844}	=>	{2811}	0.01050206	0.9343598	0.01123984	20.373683	1153
[20]	{2767, 2811, 2839, 2844}	=>	{2840}	0.01050206	1.00000000	0.01050206	22.282931	1153
[21]	{1624, 748, 763, 765}	=>	{1522}	0.01021970	1.00000000	0.01021970	5.717827	1122
[22]	{1522, 748, 763, 765}	=>	{1624}	0.01021970	0.9824869	0.01040187	10.361697	1122
[23]	{1522, 1624, 763, 765}	=>	{748}	0.01021970	1.00000000	0.01021970	31.022323	1122
[24]	{1522, 1624, 748, 765}	=>	{763}	0.01021970	1.00000000	0.01021970	50.898470	1122
[25]	{1624, 1638, 1668, 1670}	=>	{1522}	0.01829890	1.00000000	0.01829890	5.717827	2009
[26]	{1522, 1638, 1668, 1670}	=>	{1624}	0.01829890	1.00000000	0.01829890	10.546398	2009
[27]	{1522, 1624, 1668, 1670}	=>	{1638}	0.01829890	1.00000000	0.01829890	47.692441	2009
[28]	{1522, 1624, 1638, 1670}	=>	{1668}	0.01829890	1.00000000	0.01829890	48.068301	2009
[29]	{1522, 1624, 1638, 1668}	=>	{1670}	0.01829890	0.9152620	0.01999308	48.032878	2009
[30]	{2205, 2207, 2287, 2291}	=>	{2101}	0.01137647	1.00000000	0.01137647	7.162578	1249
[31]	{2101, 2207, 2287, 2291}	=>	{2205}	0.01137647	1.00000000	0.01137647	12.606269	1249
[32]	{2101, 2205, 2207, 2291}	=>	{2287}	0.01137647	1.00000000	0.01137647	26.365994	1249
[33]	{2101, 2205, 2207, 2287}	=>	{2291}	0.01137647	0.9622496	0.01182279	45.379493	1249
[34]	{2205, 2207, 2298, 2300}	=>	{2101}	0.01312530	1.00000000	0.01312530	7.162578	1441
[35]	{2101, 2207, 2298, 2300}	=>	{2205}	0.01312530	1.00000000	0.01312530	12.606269	1441
[36]	{2101, 2205, 2207, 2300}	=>	{2298}	0.01312530	0.9993065	0.01313440	21.609585	1441
[37]	{2101, 2205, 2207, 2298}	=>	{2300}	0.01312530	0.9890185	0.01327103	43.277149	1441
[38]	{2768, 2812, 2840, 2886}	=>	{2811}	0.01159507	1.00000000	0.01159507	21.804965	1273
[39]	{2768, 2811, 2812, 2886}	=>	{2840}	0.01159507	0.9914330	0.01169527	22.092033	1273
[40]	{2768, 2811, 2840, 2886}	=>	{2812}	0.01159507	1.00000000	0.01159507	22.787049	1273

▲ Apriori 코드 실행 후 결과 화면

④ 결과

```
scrap6$recommand_score <- 0
scrap6$recommand_score <- scrap6 %>% select(-c(book_id,pos_done)) %>% rowSums()
scrap6$recommand_score <- scrap6$recommand_score - scrap6$user_id
```

▲ 지금까지 생성한 점수들의 합을 구하는 열 생성

```
scrap6[order(-scrap6$recommand_score,-scrap6$following),] -> result
book_title <- book_df %>% select(id,title)
result <- left_join(result,book_title,by = c("book_id" = "id"))
```

▲ 종합 점수를 내림차순 정렬 (점수가 같다면 팔로우한 유저를 우선적으로 보여줌)

	pos_done	user_id	book_id	following	page_diff	equal_job	cat_similarity	recommand_score	title
1	가치	61393798	199368	1	1.0	1	1	4.0	상식 밖의 경제학 - 이제 상식에 기초한 경제학은 버려라!
2	가치	61393798	925693	1	1.0	1	1	4.0	책 한번 써봅시다 - 예비작가를 위한 책 쓰기의 모든 것
3	가치	61393798	2080	1	1.0	1	1	4.0	과학혁명의 구조 - 출간기념50주년 제4판
4	가치	61393798	106	1	1.0	1	1	4.0	불안 - 개정판
5	가치	61393798	1670	1	1.0	1	1	4.0	책은 도끼다
6	가치	33865726	2111	0	1.0	1	2	4.0	독서 천재가 된 홍 팀장 - 실행력을 높이는 기적의 독서 솔루션
7	가치	33865726	131738	0	1.0	1	2	4.0	광수상각: 오늘, 나에게 감사해
8	가치	33865726	21597	0	1.0	1	2	4.0	심리학을 만나 행복해졌다 - 복잡한 세상과 사람의 마음을...
9	가치	48288718	10871	0	1.0	1	2	4.0	데일 카네기 인간관계론
10	가치	61393798	1285858	1	0.5	1	1	3.5	활동경제학 콘서트 - 소비와 투자의 심리학 이야기
11	가치	61393798	915731	1	0.5	1	1	3.5	위기의 시대, 돈의 미래 - 세계 3대 투자자 짐 로저스가 말하...
12	가치	61393798	402481	1	0.5	1	1	3.5	현명한 투자자 - 벤저민 그레이엄 직접 쓴 마지막 개정판, 개...
13	가치	61393798	3533	1	0.5	1	1	3.5	문구의 모험 - 당신이 사랑한 문구의 파란만장한 연대기
14	가치	61393798	144184	1	0.5	1	1	3.5	어느 도서관의 생산적 책읽기 50 (랜디북) - 미래를 위한 자...
15	가치	33865726	13383	0	0.5	1	2	3.5	발강머리 앤 - 발강머리 앤 100주년 공식 기념판
16	가치	33865726	125449	0	0.5	1	2	3.5	그저 감사했을 뿐인데 - 관계, 사랑, 운명을 바꾸는 감사의 힘
17	가치	61393798	304188	1	0.0	1	1	3.0	파운틴 헤드 1 - 오직 나만이 나의 근원이다
18	가치	61393798	22577	1	0.0	1	1	3.0	누가 나를 쓸모없게 만드는가 - 시장 상품 인간을 거부하고 ...
19	가치	61393798	402902	1	0.0	1	1	3.0	철도원 상담
20	가치	69249442	174150	0	1.0	0	2	3.0	해리 포터와 죽음의 성물 3 (무선)
21	가치	69249442	552825	0	1.0	0	2	3.0	쓰레기에 관한 모든 것
22	가치	69249442	966054	0	1.0	0	2	3.0	커피를 좋아하면 생기는 일
23	가치	293997862	1922490	0	1.0	1	1	3.0	방금 떠난 세계
24	가치	303683663	1846384	0	1.0	1	1	3.0	보통 자리에 데이트 한다는 건 특별한 이야기 3 - 손가락 펴...

▲ 최종 결과 화면