THE ENGINE OF
KOREA
HANYANG
UNIVERSITY

리더유형검사
-리더스 프로젝트

3조
ICT융합학부 미디어테크놀로지전공 2017010046 박진규
신문방송학과 2018051703 정혜민
정보사회미디어학과 2020008077 김효림
정보사회미디어학과 2020075105 노영주
정보사회미디어학과 2020007283 최은선

목 차

1. 프로젝트 개요 및 추진 방향

(1) 리더스 SWOT 분석

Strengths(강점)

-유료 온라인 북클럽 : 온라인 독서모임을 위해 필요한 기능 / 유료- 약 2개월 3~5만원의 회비

-기록/공유 SNS : 종이 책에 기술을 더해 추천 알고리즘, 검색, OCR 등 AI와 컴퓨팅기술을 적용 / 읽은 책 기록을 체계적으로 정리 / 피드를 통해 다른 유저들과의 상호작용

-개발자-사용자 간 활발한 소통 : 사용하면서 불편한 점을 접수할 수 있는 메뉴가 따로 존재
→ 함께 만들어가는 독서 어플

Weakness(약점)

-스크랩 기능 : 책을 읽다가 핸드폰을 통해 촬영 > 스크랩 기능이 지원되기 때문에 중간에 집중이 깨질 수 있음 / 기술 사용의 어려움 존재

-업데이트 문제 : 태블릿 모드 개발 필요 / 아쉬운 UI

Opportunities(기회)

-유료 북클럽 / 기록,공유 SNS의 기술 사용 차별화

:외부 어플리케이션과 달리 머신러닝 기술을 통해 사용자가 직접 자신이 읽은 책의 구절을 담아낼 수 있음. → 차별화 분야

:돈을 내고 목표를 달성하는 컨셉의 북클럽 기능은 책장에 쌓여있어 시도를 하지 못하고 있는 책들에 대한 도전심을 불러일으킨다는 컨셉

-개발자-사용자간 소통

:개발자와 사용자 간 어플 내/외부에서의 소통이 활발

→ 이에 대한 어플 사용자의 만족도 높음

ex) 2~3번 어플 내에서 수정 요청을 보냈는데 이에 대한 의견이 받아들여졌다 / 블로그 어플 사용 후기에서의 개발자의 활발한 댓글 소통 등

=> 서비스 개발 시 장점을 극대화 시킬 수 있는 하나의 방향이 될 수 있음

-코로나-19 유행 이후 독서량의 증가 > 독서 앱이 활성화 됨

→독서 어플의 활성화(리더스와 연계의 가능성 존재)

Threat(위협)

-리더스를 나타낼 수 있는 차별화 포인트에 대한 부각이 부족

:북적북적 등과 같은 유사 어플리케이션의 경우 귀여운 캐릭터, 광고 등으로 자신들 만의 색과 컨셉을 찾아가고 있는 상황. 차별화된 기술과 기능들이 있음에도 이를 알릴 있을 만한 포인트 부족

-> 서비스 혹은 캐릭터 강화를 통해 이를 나타낼 수 있는 전략 필요

북적북적- 귀여운 캐릭터. 북적list (책마다 높이를 달리 해 책을 쌓아 올리는 컨셉)
책을 읽고 저장하면 몇 cm마다 캐릭터 획득 가능
읽었던 책들에 대하여 간단히 메모 남길 수 있음

✓ **책의 분야별로 캐릭터 존재 >> 내 캐릭터로 바꿀 수 있다면?**

2. 코드 구현

(1) Active user 분류

Active User 설정하기

전체 유저 데이터를 다루지 않고 최근 3개월동안 활동한(데이터 상 가장 최근 활동기록이 2022년 3월이었으므로 2021년 12월 1일부터 활동(장바구니 기록)한 사용자를 조건으로 설정) 사용자만 추출하기로 했음

```
[ ]  active <- data4 %>%
       select(user_id, modified_at) #유저 아이디와 수정시각 추출
```

```
[ ]  active <- active[active$modified_at >= "2021-12-01",] #최근 3개월 사용자 추출
```

+ 코드 + 텍스트

```
summary(data1)
```

```
    user_id           birth_year      gender           created_at
 Min.   :    10936   Min.   :1930   Length:2634        Length:2634
 1st Qu.: 79645584   1st Qu.:1981   Class :character   Class :character
 Median :195332158   Median :1988   Mode  :character   Mode  :character
 Mean   :218319169   Mean   :1988
 3rd Qu.:350487606   3rd Qu.:1995
 Max.   :555758878   Max.   :2022
                     NA's   :161
    직업               직종
 Length:2634        Length:2634
 Class :character   Class :character
 Mode  :character   Mode  :character
```

```
[ ]  active2 <- unique(active$user_id) #중복 제거
```

```
[ ]  data1 %>% filter(user_id %in% c(active2)) -> data1_ac
     summary(data1_ac)
```

```
    user_id           birth_year      gender           created_at
 Min.   :    73870   Min.   :1930   Length:1944        Length:1944
 1st Qu.: 98821018   1st Qu.:1981   Class :character   Class :character
 Median :229784821   Median :1987   Mode  :character   Mode  :character
 Mean   :240939857   Mean   :1988
 3rd Qu.:374046208   3rd Qu.:1994
 Max.   :555758878   Max.   :2022
                     NA's   :115
    직업               직종
 Length:1944        Length:1944
 Class :character   Class :character
 Mode  :character   Mode  :character
```

(2) 지표별 데이터값 생성

팔로우한 유저 수

유저의 팔로우와 팔로워 관계 정보가 담긴 data3 사용

```
[ ]  data3_ac <- data3 %>% filter(user_id %in% c(active2)) #Active user 필터링

     class1 <- data3_ac %>% group_by(user_id) %>% summarise(count_follow = n()) #유저별 팔로잉 수 합계(user_id를 groupby하여 target_user_id의 갯수 합계)
```

스크랩수

유저의 스크랩 기록이 담긴 data8 사용

```
[ ]  data8_ac <- data8 %>% filter(user_id %in% c(active2)) #Active user 필터링

     class2 <- data8_ac %>% group_by(user_id) %>% summarise(count_scrap = n()) #유저별 스크랩 수 합계(user_id를 groupby하여 스크랩 수 합계)
```

'읽기 전' 비율

책장에 담아만 놓고 읽기 시작하지 않은 책의 비율

유저의 책장에 담기 책과 독서 상태를 보여주는 data4 사용

```
[ ]  data4_ac <- data4 %>% filter(user_id %in% c(active2)) #Active user 필터링

     d41 <- data4_ac %>% group_by(user_id) %>% summarise(count1 = n()) #유저별 책장 기록 횟수 총합 추가(user_id를 groupby하여 책장 기록 합계)

     d42 <- data4_ac %>% group_by(user_id, read_status ) %>% filter(read_status == "READ_STATUS_BEFORE") %>% summarise(count2 = n()) %>% select(user_id, count2)
     #유저별 '읽기 전' 횟수 추가(read_status_before인 상태만 추출하여 기록 합계)

     d43 <- merge(d41, d42, key = 'user_id') #데이터 결합

     class3 <- mutate(d43, ratio_before = count2/count1) %>% select(user_id, ratio_before)
     #count2(읽기 전 책의 수) / count1(전체 책의 수) 연산을 통해 '읽기 전' 책의 비율 추출
```

낮시간 활동 비율

낮시간(07 ~ 18시로 임의 설정)동안 활동(스크랩)하는 비율

유저의 스크랩 기록이 담긴 data8 사용

```
[ ]  scrap <- data8_ac %>% select(user_id, created_at) %>% separate(col= 'created_at', into = c('date', 'time'), sep= ' ')
     #modified_at에서 (2022-03-20 11:08:02.451517)형태로 되어있는 데이터를 공백 기준으로 날짜와 시간 구분

     scrap_time <- scrap %>% select(user_id, time) %>% separate(col= 'time', into = c('hour', 'minute', 'second'), sep= ':')
     scrap_data <- scrap_time %>% select(user_id, hour) #':'을 기준으로 나누어 시간만 추출

     scrap_day <- subset(scrap_data, hour %in% c('07', '08', '09', '10', '11', '12', '13', '14', '15', '16', '17', '18'))
     scrap_day <- scrap_day %>% group_by(user_id) %>% summarize(day=n())
     scrap_all <- scrap_data %>% group_by(user_id) %>% summarize(all=n()) #시간에 7~18이 포함되어있는 데이터를 추출해 낮과 전체시간의 스크랩수 구분

     scrap_time_ratio <- merge(scrap_day, scrap_all, key ='user_id')
     scrap_time_ratio <- mutate(scrap_time_ratio, ratio_daytime = day/all) %>% select(user_id, ratio_daytime)
     class4 <- scrap_time_ratio #day(낮시간 스크랩 횟수)/all(전체 스크랩 횟수) 연산을 통해 낮시간 활동 비율 추출
```

읽은 페이지 수

책장 기록이 담긴 data4와 책에 대한 정보가 담긴 data5 결합 사용

```
[ ]  # 읽은 페이지 수
     ## 책장 기록이 담긴 data4와 책에 대한 정보가 담긴 data5 결합 사용

     book_page <- data5 %>% select(id,page)
     book_page <- rename(book_page,book_id=id) #data5의 책 페이지 변수명을 data4와 일치

     id <- data4_ac %>% group_by(user_id) %>% select(user_id,book_id,read_status)
     test <- merge(book_page,id,by="book_id") %>% select(user_id, page) %>% group_by(user_id) #data4와 data5 결합(book_id를 기준으로 하여 user_id, page, read_status정보가

     test$page <- ifelse(id$read_status == "READ_STATUS_DONE",test$page,0)
     class5 <- test %>% summarise(sum_page = sum(page,na.rm = T)) #read_status_done인 책의 페이지수만 합계
```

(3) K-means clustering

머신러닝을 위한 데이터 정리

```
merge(class1, class2, key = "user_id") -> chunk1
merge(class3, class4, key = "user_id") -> chunk2
merge(chunk1, chunk2, key = "user_id") -> chunk3
merge(chunk3, class5, key = "user_id") -> user0
rownames(user0) = user0$user_id
user0 %>% select(-user_id) -> user1
head(user1) # 데이터를 추출하여 5가지 지표를 user_id 기준으로 하나의 데이터로 정리
```

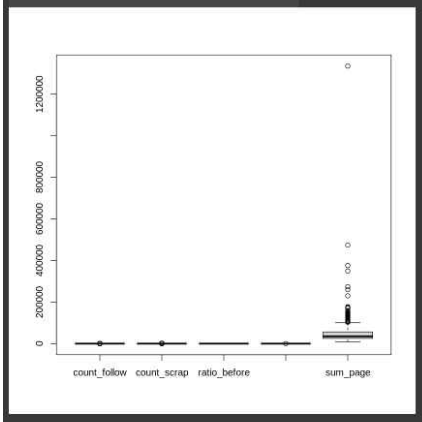
A data.frame: 6 × 5

	count_follow	count_scrap	ratio_before	ratio_daytime	sum_page
	<int>	<int>	<dbl>	<dbl>	<dbl>
73870	17	327	0.04878049	0.6666667	26323
414454	4	11	0.29629630	0.5454545	32301
532918	16	557	0.42583732	0.6104129	35866
629170	1	43	0.69918699	0.8604651	23717
991966	2	449	0.13333333	0.8084633	26544
1117834	63	439	0.35864979	0.1845103	42982

```
boxplot(user1)$stats #박스플롯을 그려 데이터 확인(이상치 확인)
```

A matrix: 5 × 5 of type dbl

1	1	0.003144654	0.1000000	7647
1	17	0.161971831	0.4788419	25279
3	56	0.423963134	0.6285714	35282
8	169	0.676056338	0.7659574	56383
18	391	0.998560775	1.0000000	100555



이상치 변환 및 제거

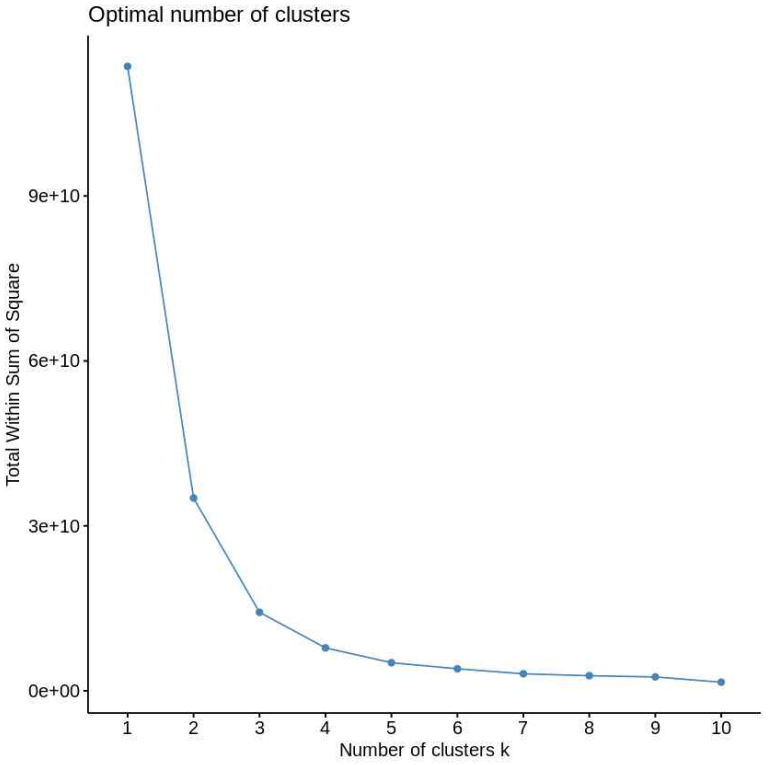
```
ifelse(user1$count_follow < 1  | user1$count_follow > 18, NA, user1$count_follow) -> user1$count_follow
ifelse(user1$count_scrap < 1   | user1$count_scrap > 391, NA, user1$count_scrap) -> user1$count_scrap
ifelse(user1$ratio_before < 0.003144654  | user1$ratio_before > 0.998560775, NA, user1$ratio_before) -> user1$ratio_before
ifelse(user1$ratio_daytime < 0.1000000   | user1$ratio_daytime > 1.0000000, NA, user1$ratio_daytime) -> user1$ratio_daytime
ifelse(user1$sum_page < 7647   | user1$sum_page > 100555, NA, user1$sum_page) -> user1$sum_page

user1 <- na.omit(user1)
#각 지표별로 이상치에 해당되는 데이터는(ifelse사용) NA처리하여 한 번에 제거(na.omit)
```

정규화

```
user1_scaled <- scale(user1, center = T, scale = T) #데이터 정규화
```

최적의 클러스터 갯수 파악하기

```
gl<-fviz_nbclust(user1, kmeans, method = "wss")
gl #결과는 4개
```



K-Means Clustering

```
[ ] set.seed(1000) #샘플 갯수 설정
    km.res <- kmeans(user1_scaled, 4, nstart = 25) #클러스터 갯수, 시행 횟수 설정
    print(km.res)
```

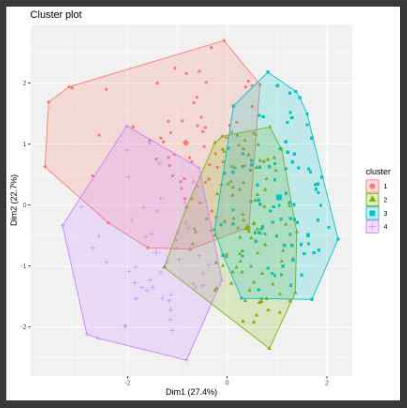
```
K-means clustering with 4 clusters of sizes 66, 91, 94, 56

Cluster means:
  count_follow count_scrap ratio_before ratio_daytime   sum_page
1    0.2612656  -0.0220053    0.4420758    0.13153706  1.4529640
2   -0.3566644  -0.3794722    0.1818904   -0.95476079 -0.4493952
3   -0.4264275  -0.5215042   -0.3264674    0.88528805 -0.4856781
4    0.9872859   1.5179592   -0.2685909   -0.07330876 -0.1669093

Clustering vector:
   73870   414454   629170  1121536  2280262  3487114  3786976  3979480
       4        2        3        1        1        2        3        2
 7103968  7733308  8051680  8473708  9476950 10376536 11168764 12508888
       4        2        3        2        1        4        3        1
```

```
[ ] dd <- data.frame(user1_scaled, cluster = km.res$cluster) #유저 데이터에 클러스터 변수 추가
    head(dd)
```

A data.frame: 6 × 6

	count_follow	count_scrap	ratio_before	ratio_daytime	sum_page	cluster
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
73870	3.2382526	2.4025850	-1.3825523	0.1307131	-0.6321124	4
414454	-0.0789615	-0.7701629	-0.4674353	-0.4225759	-0.3218200	2
629170	-0.8444724	-0.4488719	1.0221347	1.0153322	-0.7673787	3
1121536	0.4313791	2.2419396	1.3707490	0.4340436	3.0422416	1
2280262	0.4313791	0.5250412	1.1720159	0.4132857	1.1405717	1
3487114	-0.3341318	0.1736292	-0.2477902	-0.6952683	-0.6473727	2

```
fviz_cluster(km.res, data = user1_scaled, geom = "point") #K-means clustering 실행
```



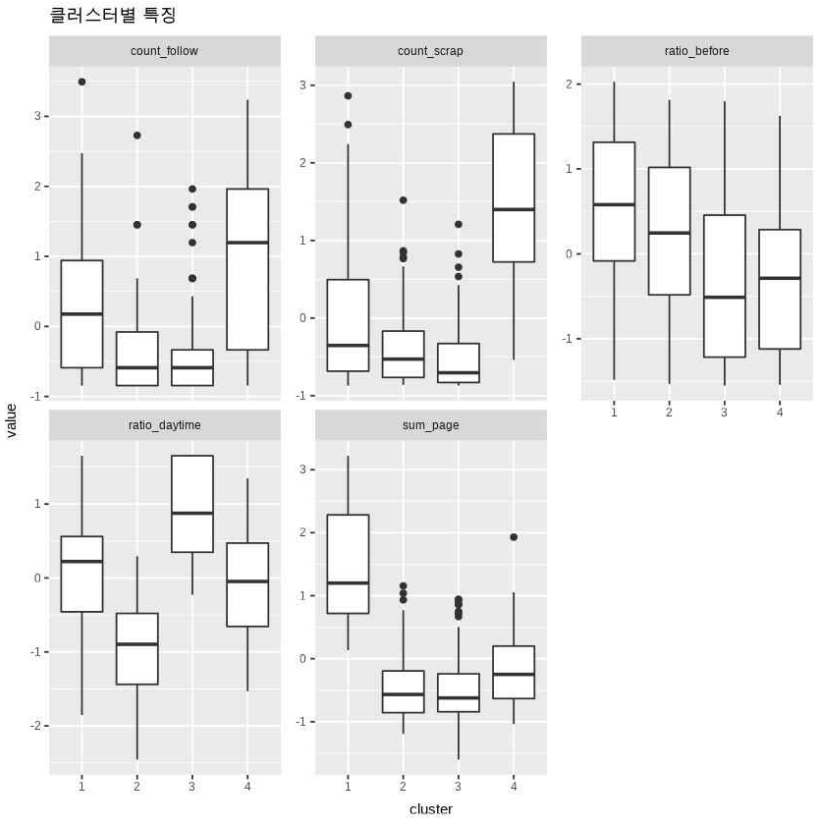

```
[ ] dd %>% group_by(cluster) %>% summarize_all(mean) #클러스터링이 잘 되었는지 확인
```

A tibble: 4 × 6

cluster	count_follow	count_scrap	ratio_before	ratio_daytime	sum_page
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0.2612656	-0.0220053	0.4420758	0.13153706	1.4529640
2	-0.3565644	-0.3794722	0.1818904	-0.96476079	-0.4493952
3	-0.4264275	-0.5215042	-0.3264674	0.88528805	-0.4856781
4	0.9872859	1.5179592	-0.2685909	-0.07330876	-0.1669093

박스플롯 그리기

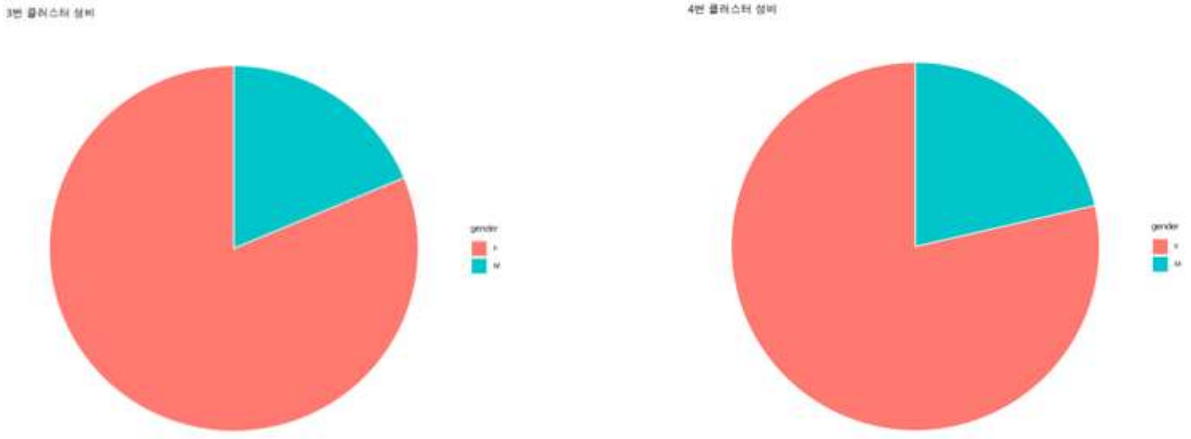
```
[ ] library(reshape2)

    dd %>% melt(id.vars="cluster") %>% mutate(cluster=as.factor(cluster)) %>%
    ggplot(aes(x=cluster, y=value))+geom_boxplot()+facet_wrap(~variable, scale="free_y") + ggtitle("클러스터별 특징") #각 지표별로 경향성 파악
```



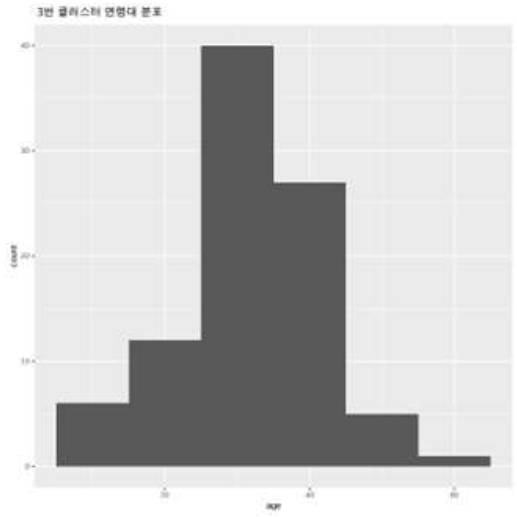
3. 클러스터 유형 별 분석

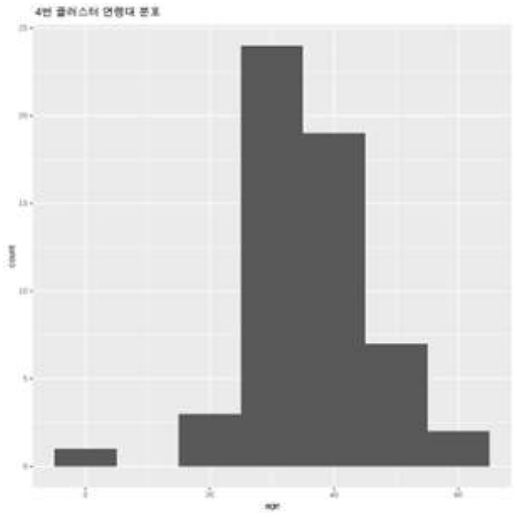
1. 첫번째 유형은 책을 많이 읽는 사람으로 읽은 페이지 수 · before 비율이 많다.
2. 두번째 유형, 스크랩 수 · 팔로우 평균적이고, 비교적 밤에 활동량이 많다.
3. 세번째 유형, 스크랩 수 · 팔로워 · before 비율이 모두 적고, 대체적으로 활동량이 낮지만 주로 낮에 활동한다.
4. 마지막 유형은 성실히 활동하여 팔로워 · 스크랩 수가 많고, before 비율도 평균적으로 낮은 편이다.

(1) 성별

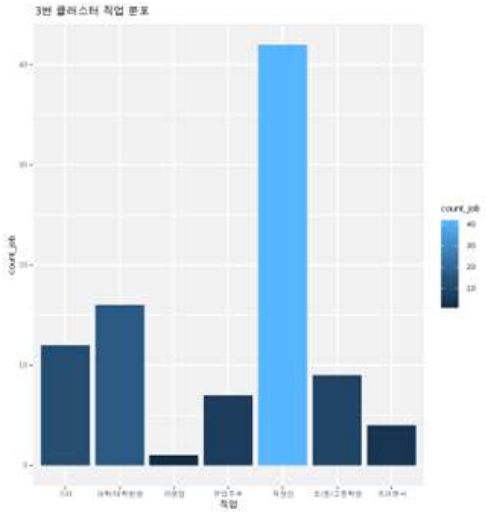


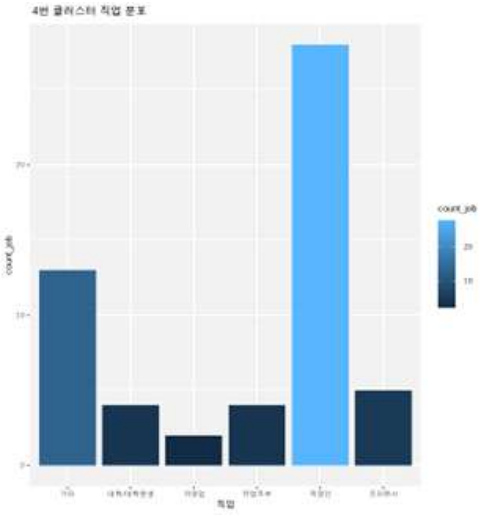
(2) 나이


3번 클러스터 연령대 분포
count
age

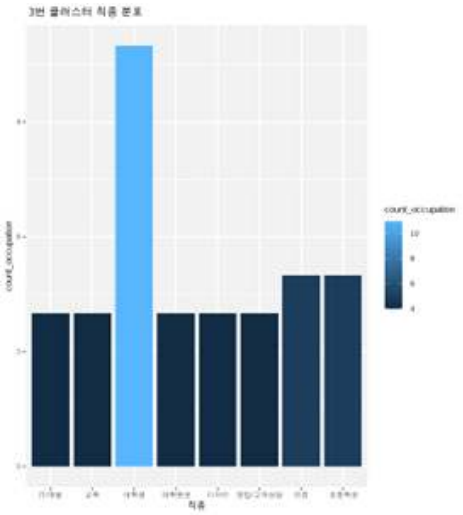

4번 클러스터 연령대 분포
count
age

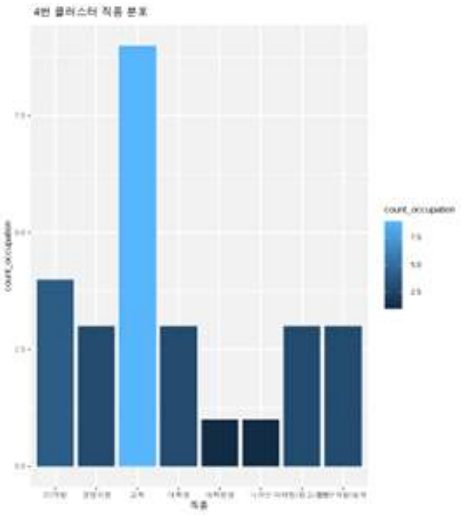
(3) 직업


3번 클러스터 직업 분포
count_job
직업

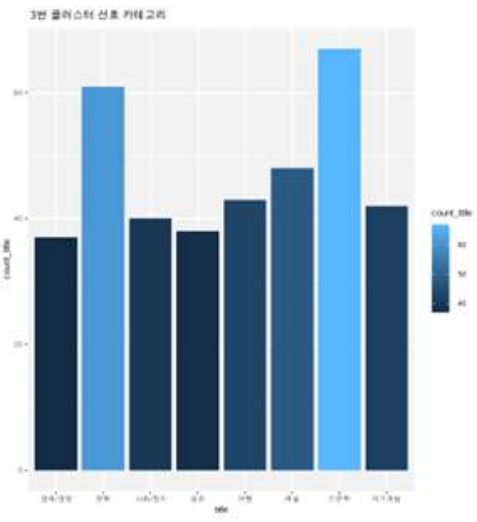

4번 클러스터 직업 분포
count_job
직업

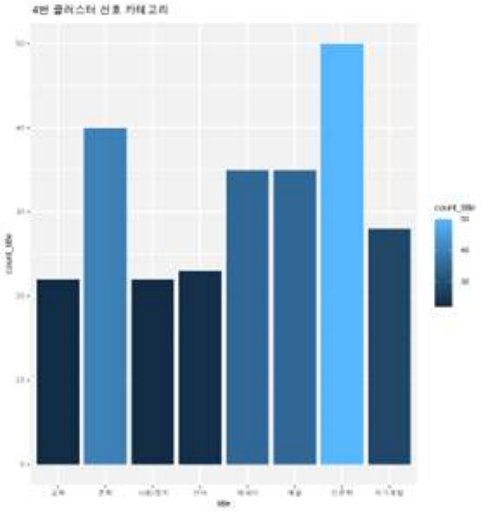
(4) 직종


3번 클러스터 직종 분포
count_occupation
직종


4번 클러스터 직종 분포
count_occupation
직종

(5) 선호 카테고리



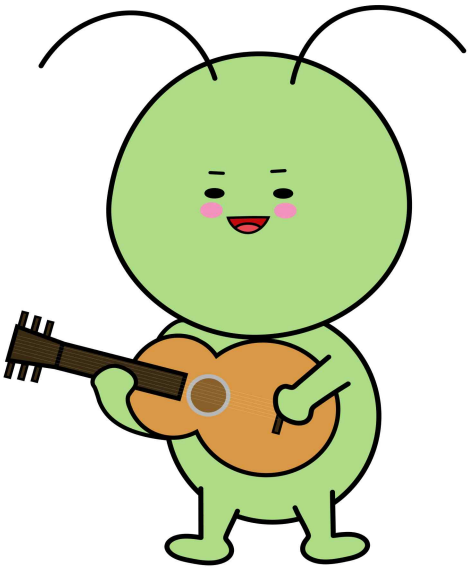


4. 디자인 및 서비스 제안

-전래동화 캐릭터를 모티브로 각 성격에 맞는 리더 유형 캐릭터 생성

(클러스터 유형 순서)

1. 토끼와 거북이-거북이
2. 미운 오리새끼-야행성 아기오리
3. 개미와 베짱이-베짱이
4. 아기돼지 삼형제-막내 아기돼지

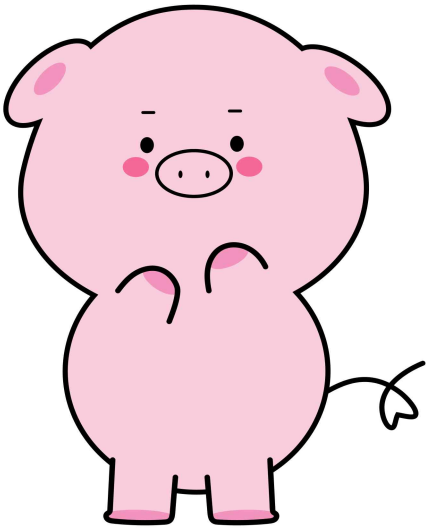


베짱이형 리더(Reader)

당신은
(베짱이형)
리더입니다.

"나는 여유롭게 책을 읽는게 좋아!"

베짱이는 밤보단 낮에 책 읽는 것을 더 좋아해요.
낮에 더 여유롭게 책을 읽을 수 있으니까요!
다른 유저와의 소통보다는
혼자만의 독서 시간을 가진답니다.
게을러보이지만... 꾸준히 책은 읽는다구요.

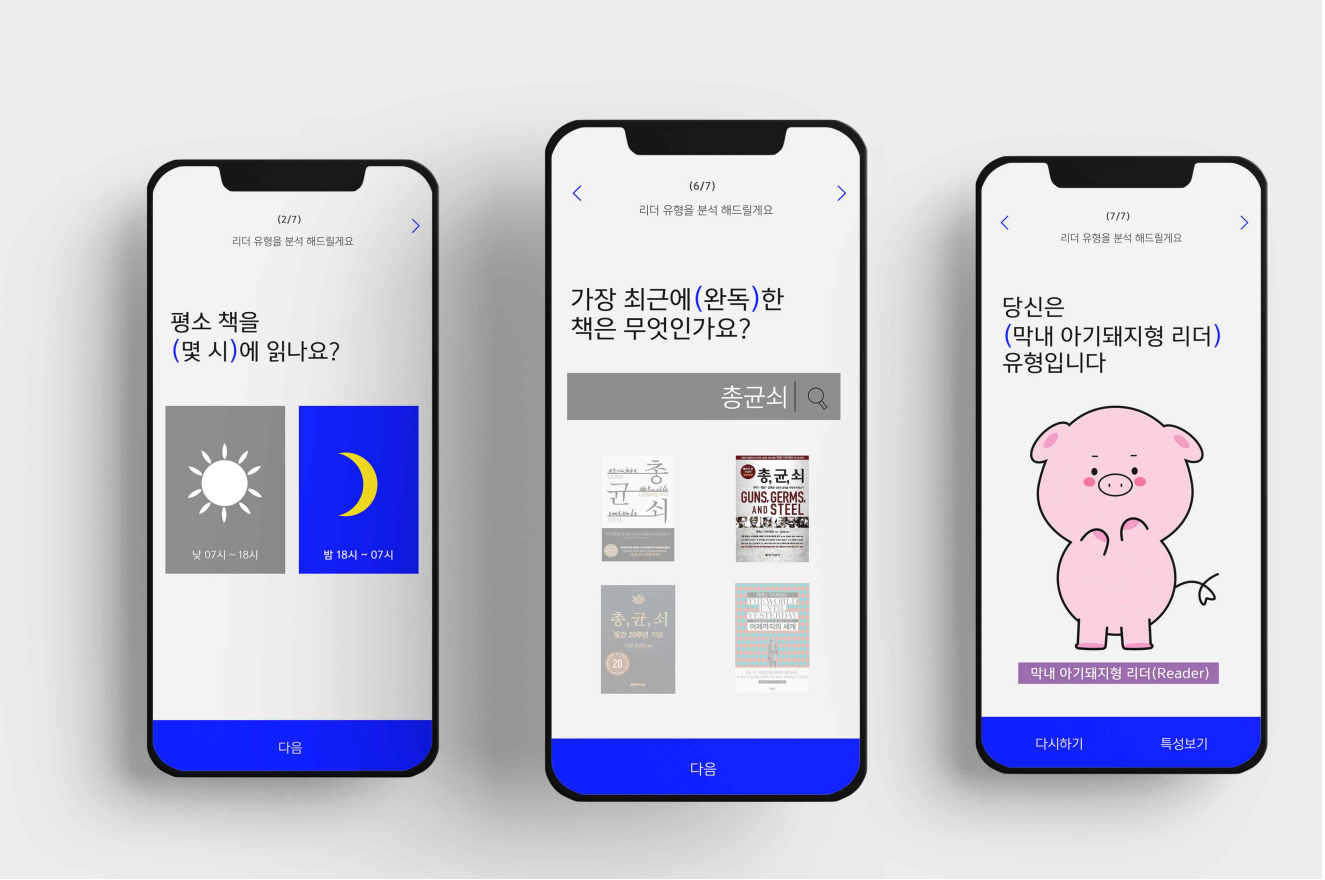


막내 아기돼지형 리더(Reader)

당신은
(막내 아기돼지형)
리더입니다.

"나는 성실한 인싸가 될거야!"

막내 아기돼지는 모든 방면에서 성실해요.
다른 유저와의 활발한 소통과 SNS 활동으로
리더스에서 인싸가 되는게 목표랍니다.
책 읽는 것을 게을리하지 않고
부지런히 리더스 활동을 할거예요!



-리더스 어플에 가입할 때, mbti와 같은 간단한 검사로 분야 확인을 대체한다.

1. 선호 분야가 어떻게 되나요? (분야별 선택)-기존에 존재
2. 평소 책을 몇 시에 읽나요? (오전/오후)
3. 월 평균 독서량이 어떻게 되나요? (0~1 / 2~3 / 4권 이상)
4. 가장 최근에 완독한 책이 무엇인가요? (검색을 통한 책 선택-복수 가능)
5. SNS 활동에 열심히 참여하는 편인가요? (네 / 평균 / 아니요)
6. 책을 구입 후 독서 시기가 어떻게 되나요? (바로 읽는다 / 시간이 지나고 읽는다 / 잘 읽지 않는다)

-이탈률을 최대한 줄이기 위해 최종 수정시각이 오래된(어플을 사용한지 오래 된) 사용자에게 '리더 유형 검사' 이벤트 알람을 보낸다.
  -더불어 SNS 공유 이벤트를 함께 진행하여 참여율을 높이고, 새로운 유저의 유입을 기대한다.
  -기존 active user에게는 데이터를 기반으로 리더 유형 검사 결과를 부여한다.
-유료 북클럽 / follow / 책 추천 관련 서비스에 활용한다.
  -'나와 같은 유형이 시작한 북클럽' 혹은 '이런 북클럽은 어때요?' 형태로 북클럽을 추천한다.
  -나와 같은 유형의 사용자와 나와 같은 유형의 사용자가 읽은 책을 추천한다.
-각 유형 별로 시스템 배치 구도를 다르게 추천해준다. (ex) 베짱이 형 리더 테마 적용하기
(색상변경, 구도변경 추천 등등)
-같은 유형에 매칭된 유저들이 많이 읽은 랭킹 탭을 제공한다.
- 매주 혹은 매일 사용자가 정해둔 시각에 해당 유형이 많이 읽은 책 한~두 권을 추천한다.
- 매주 일요일, "나와 정반대/모든 유형이 많이 읽은 책입니다" 형태로 책을 추천한다.
- 텍스트 분석을 통해 같은 클러스터에 매칭 된 유저들이 많이 스크랩한 책 문구를 배너 알림 서비스로 제공한다.

5. 마무리

-서비스 UI 개선
  : 단조로운 UI, 업데이트가 느림, 북적북적 등과 같은 경쟁사들에 비해 디자인적 차별성이 부족하다.
-ISBN 데이터 수정 필요(총균쇠의 절망..)