빅데이터 연구회 : KU-BIG

MyAnimeList Dataset

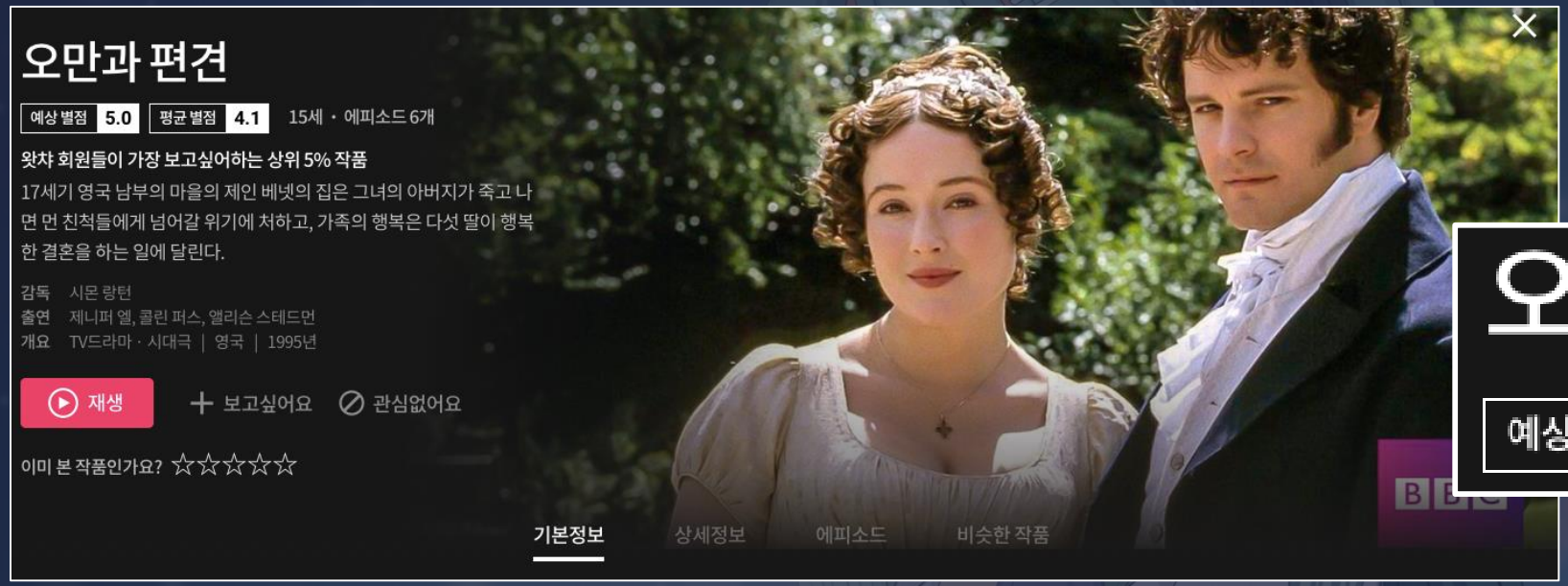
Recommendation

송 민 우경민 이강현
윤재경 정재원 이소담

INDEX

프로젝트

추천시스템



WATCHA PLAY

오만과 편견

예상 별점 5.0 평균 별점 4.1

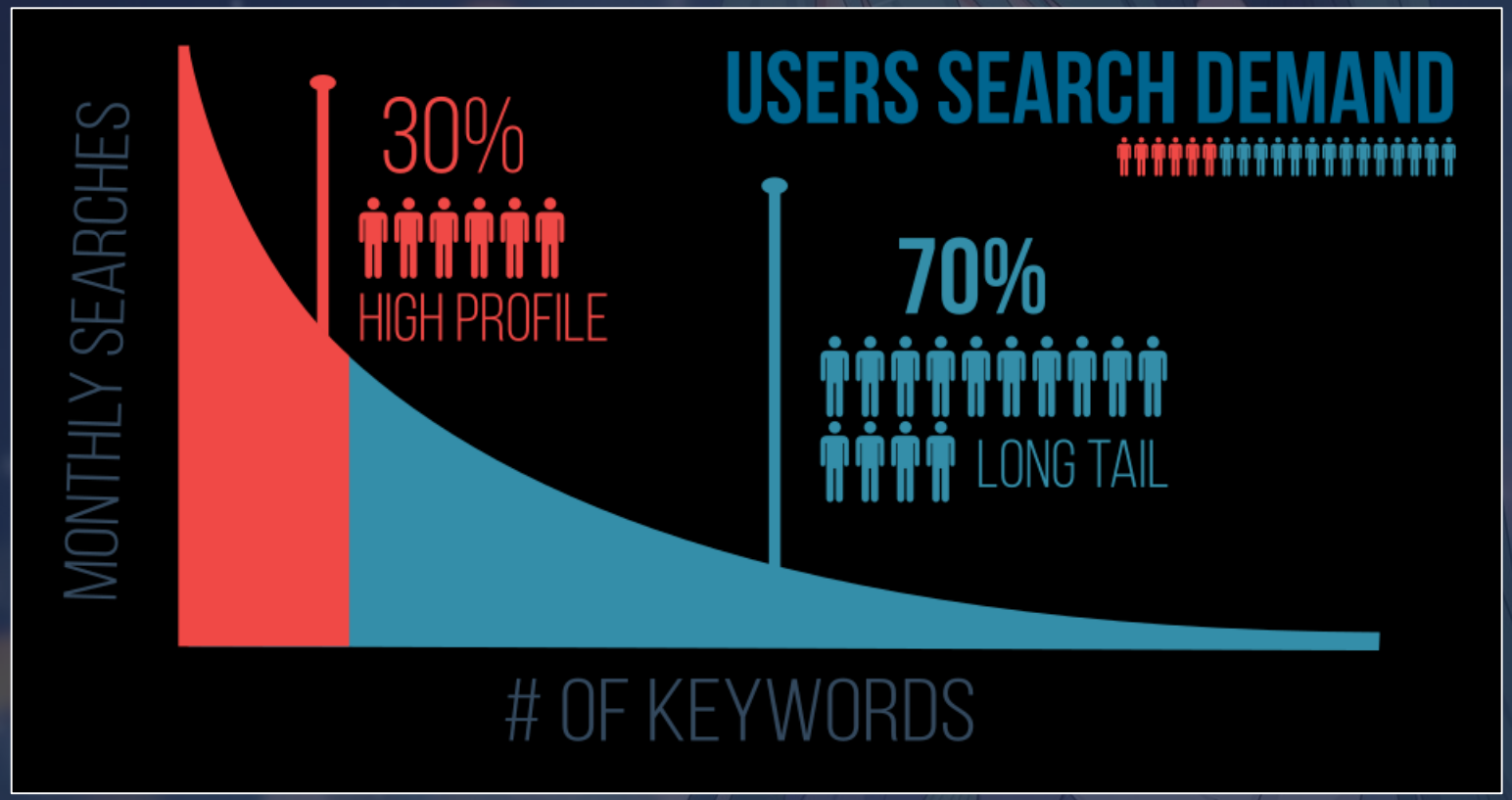
NETFLIX

NETFLIX 영화

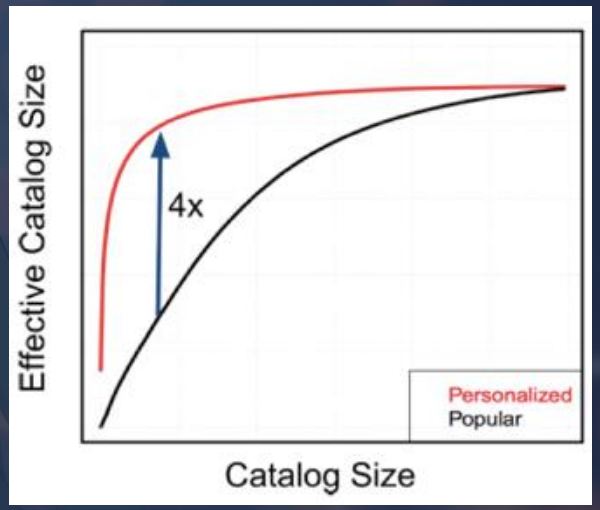
우리의 계절은

80% 일치 2018 12 1시간 14분

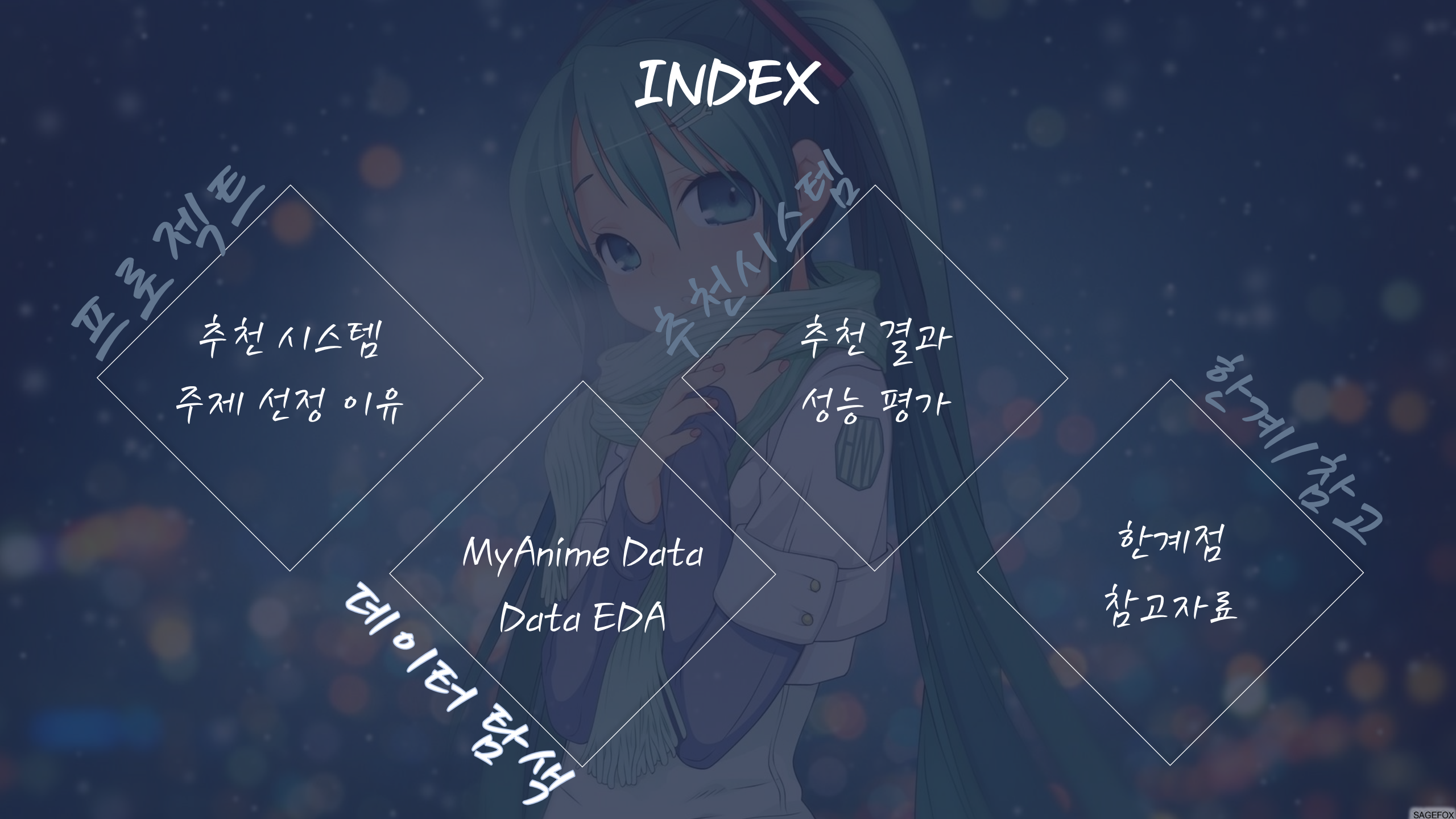
프로젝트 주제 선정 이유



추천 시스템은 1년에 1.2조원의 가치



INDEX



프로젝트
추천 시스템
주제 선정 이유

추천시스템
추천 결과
성능 평가

데이터 탐색
MyAnime Data
Data EDA

한계/참고
한계점
참고자료

데이터 탐색

MyAnime data

MyAnimeList

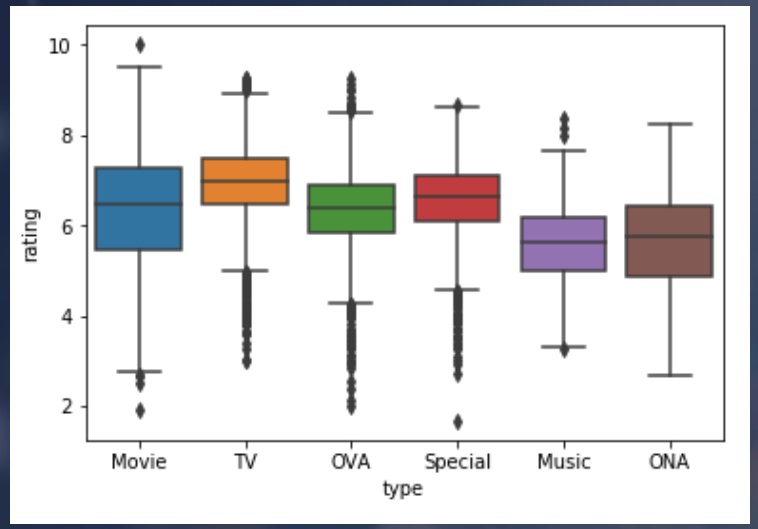
: an anime andmanga social networking and social cataloging application website. The site provides its users with a list-like system to organize and score anime and manga.

Rating.csv

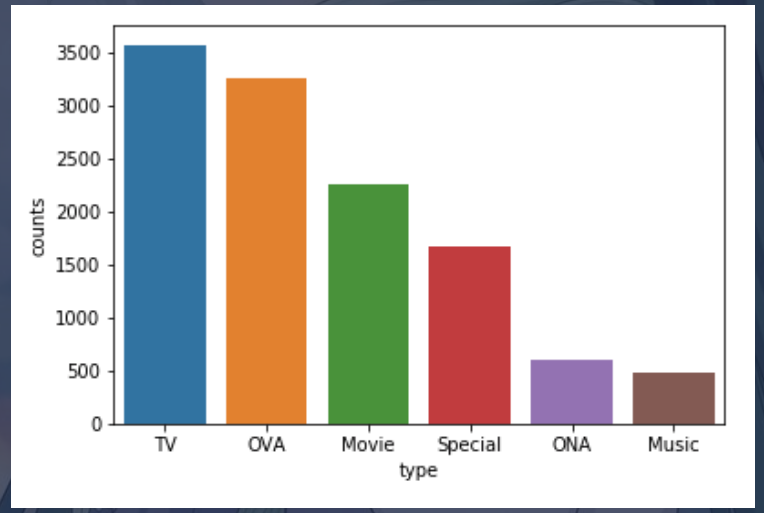
- user_id
- anime_id
- rating

데이터 탐색
Data EDA

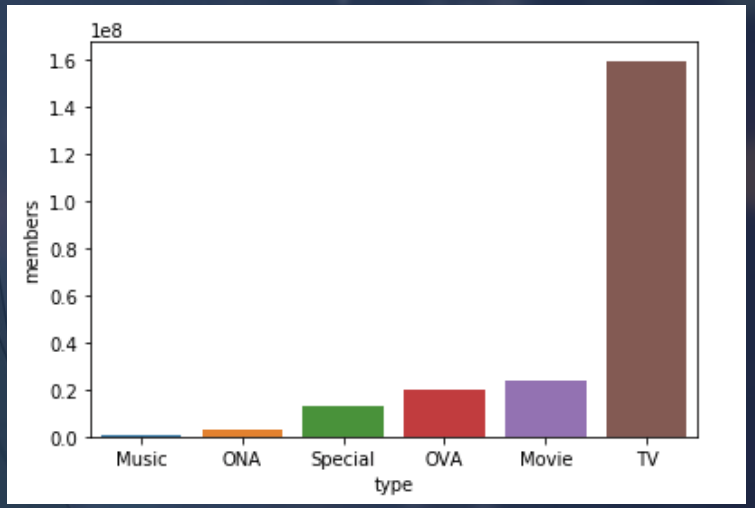
Rating - Type



Count - Type

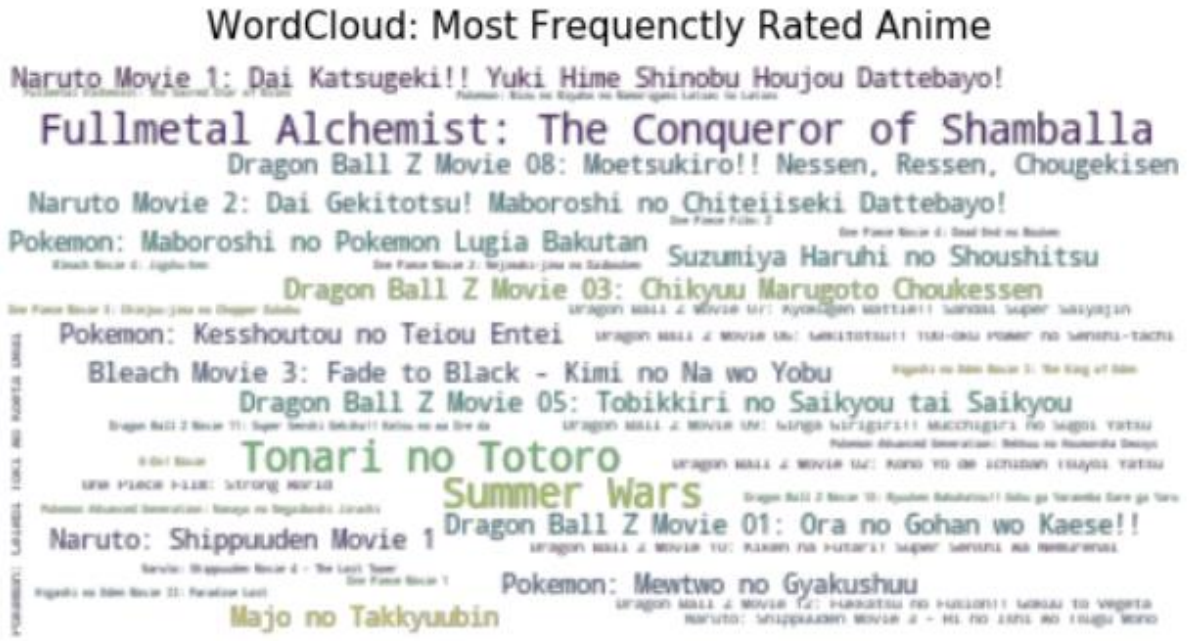


Members - Type



데이터 탐색

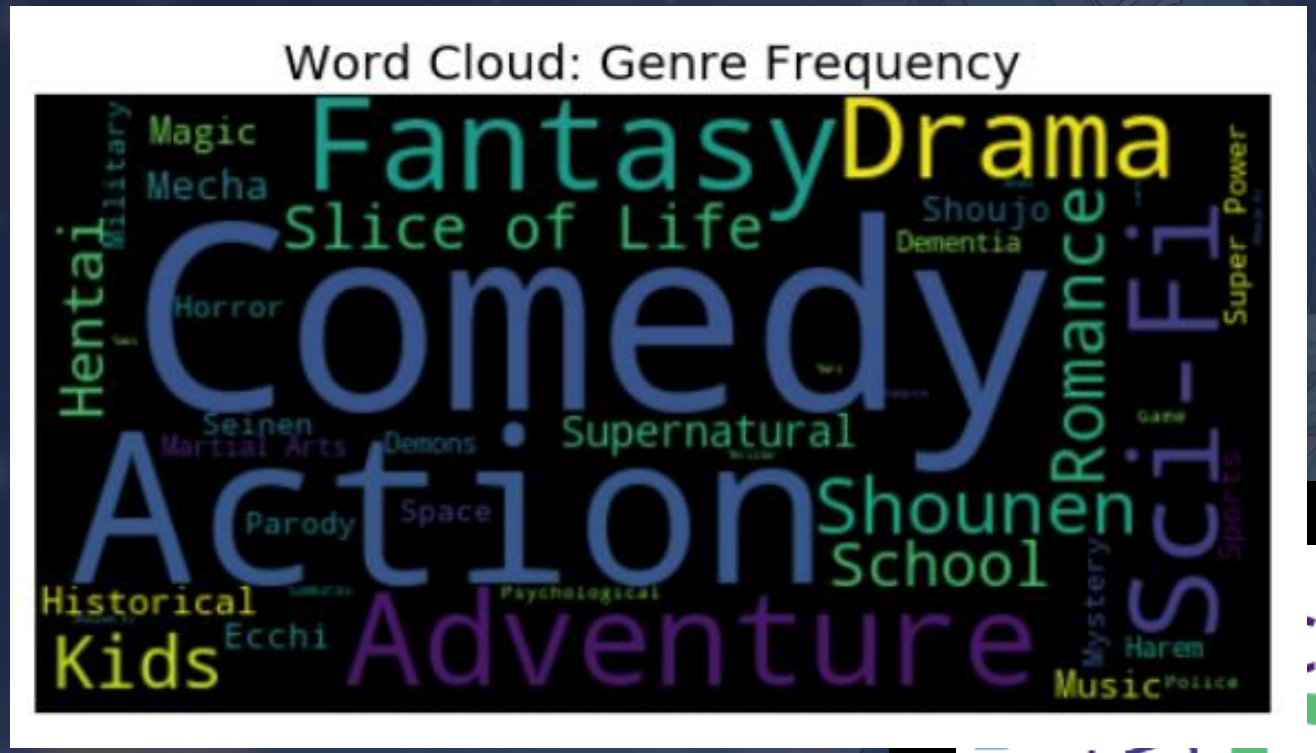
Data EDA



	name	count
493	Fullmetal Alchemist: The Conqueror of Shamballa	3816
523	Tonari no Totoro	3572
5651	Summer Wars	3566
402	Naruto Movie 1: Dai Katsugeki!! Yuki Hime Shinobu Houjou Dattebayo!	3542
528	Pokemon: Mewtwo no Gyakushuu	3503
7511	Suzumiya Haruhi no Shoushitsu	2879
2472	Naruto: Shippuuden Movie 1	2865
1117	Pokemon: Maboroshi no Pokemon Lugia Bakutan	2842
936	Naruto Movie 2: Dai Gekitotsu! Maboroshi no Chiteiiseki Dattebayo!	2817
4835	Bleach Movie 3: Fade to Black - Kimi no Na wo Yobu	2758

데이터 탐색

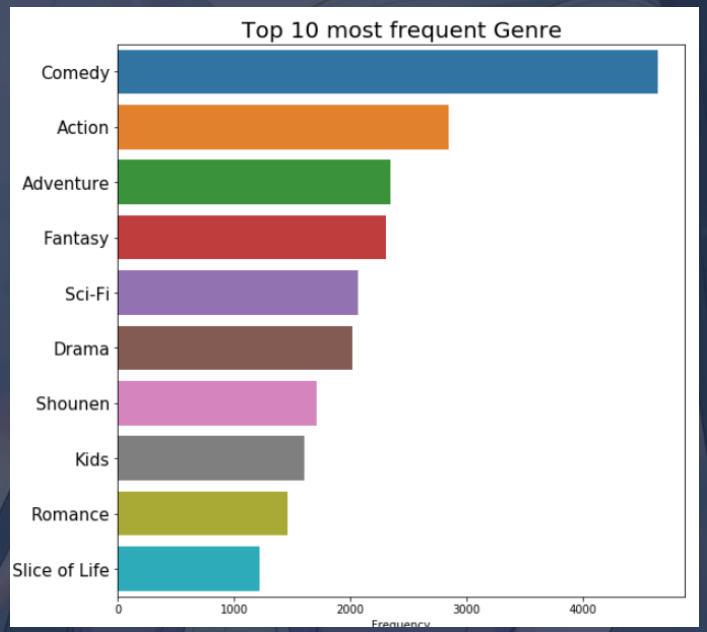
Data EDA



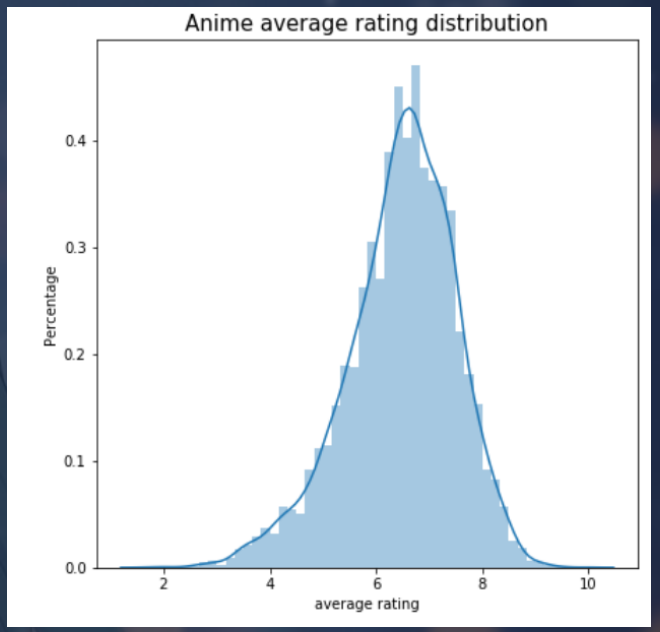
clustering

데이터 탐색
Data EDA

Frequent Genre



Anime average rating



INDEX

추천시스템

사용 알고리즘

- 협업 필터링
  - Collaborative filtering
    : Matrix Factorization

추천시스템

Python Coding

GitHub

enfow Update README.md — Latest commit f722

.gitignore	initial commint- implementing basic recommendation system for movie.
MF.py	initial commint- implementing basic recommendation system for movie.
README.md	Update README.md
data.py	initial commint- implementing basic recommendation system for movie.
sim.py	initial commint- implementing basic recommendation system for movie.

→ https://github.com/KU-BIG/recommendation_system

```python
class MatrixFactorization():

    def __init__(self, matrix, lr, reg_param, epochs):

        self._matrix = matrix
        self._row, self._column = matrix.shape
        self._k = 5
        self._lr = lr
        self._reg_param = reg_param
        self._epochs = epochs


    def fit(self):

        # initialize latent features
        self._P = np.random.normal(size=(self._row, self._k))
        self._Q = np.random.normal(size=(self._column, self._k))

        # initialize biases
        self._b_P = np.zeros(self._row)
        self._b_Q = np.zeros(self._column)
        self._b = np.mean(self._matrix[np.where(self._matrix != 0)])

        #training
        for epoch in range(self._epochs):
            for i in range(self._row):
                for j in range(self._column):
                    if self._matrix[i, j] > 0:
                        self.gradient_descent(i, j, self._matrix[i, j])


    def gradient_descent(self, i, j, rating):

        # get error
        prediction = self.get_prediction(i, j)
        error = rating - prediction
```

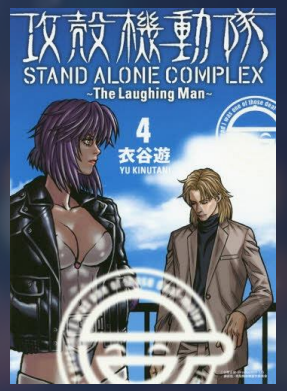
추천시스템 추천 결과

```
#get anime name
get_name(top10_rec_exmp)
```

```
## Recommendation for User 3 ##

 ['Gintama Movie: Kanketsu-hen - Yorozuya yo Eien Nare', 'Gintama: Yorinuki Gintama-san on Theater 2D', 'Tonari no Totoro', 'Gintama Movie: Shinyaku Benizaku
ra-hen', 'Tengen Toppa Gurren Lagann Movie: Gurren-hen', 'Suzumiya Haruhi no Shoushitsu', 'Trigun: Badlands Rumble', 'One Piece Movie 4: Dead End no Bouken',
'One Piece Film: Strong World', 'Majo no Takkyuubin']
```

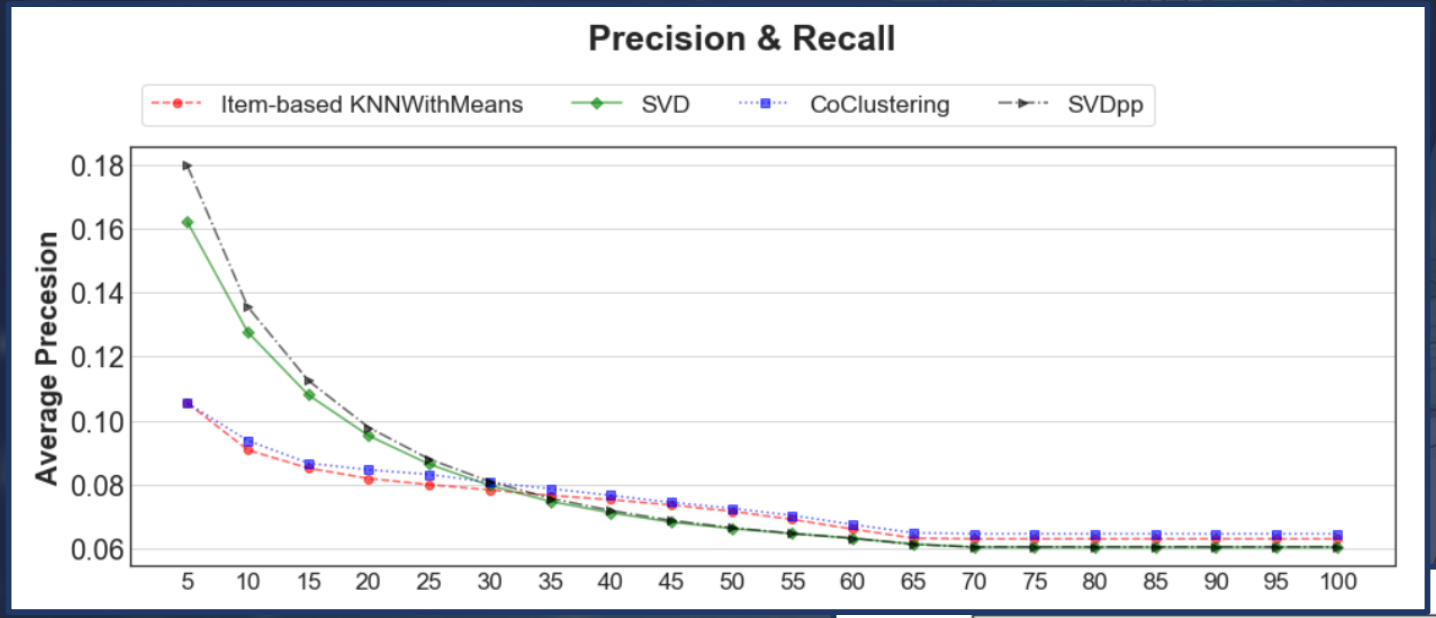
```
## Recommendation for User 19418 ##

 ['Gintama Movie: Kanketsu-hen - Yorozuya yo Eien Nare', 'Detective Conan Movie 08: Magician of the Silver Sky', 'Bo
ruto: Naruto the Movie', 'Detective Conan Movie 07: Crossroad in the Ancient Capital', 'Gintama Movie: Shinyaku Beni
zakura-hen', 'Detective Conan Movie 03: The Last Wizard of the Century', 'Lupin III: Cagliostro no Shiro', 'Crayon S
hin-chan Movie 09: Arashi wo Yobu Mouretsu! Otona Teikoku no Gyakushuu', 'Mind Game', 'Detective Conan Movie 15: Qua
rter of Silence']
```

10개의 Anime 추천 (Top 10)



추천시스템

성능 평가

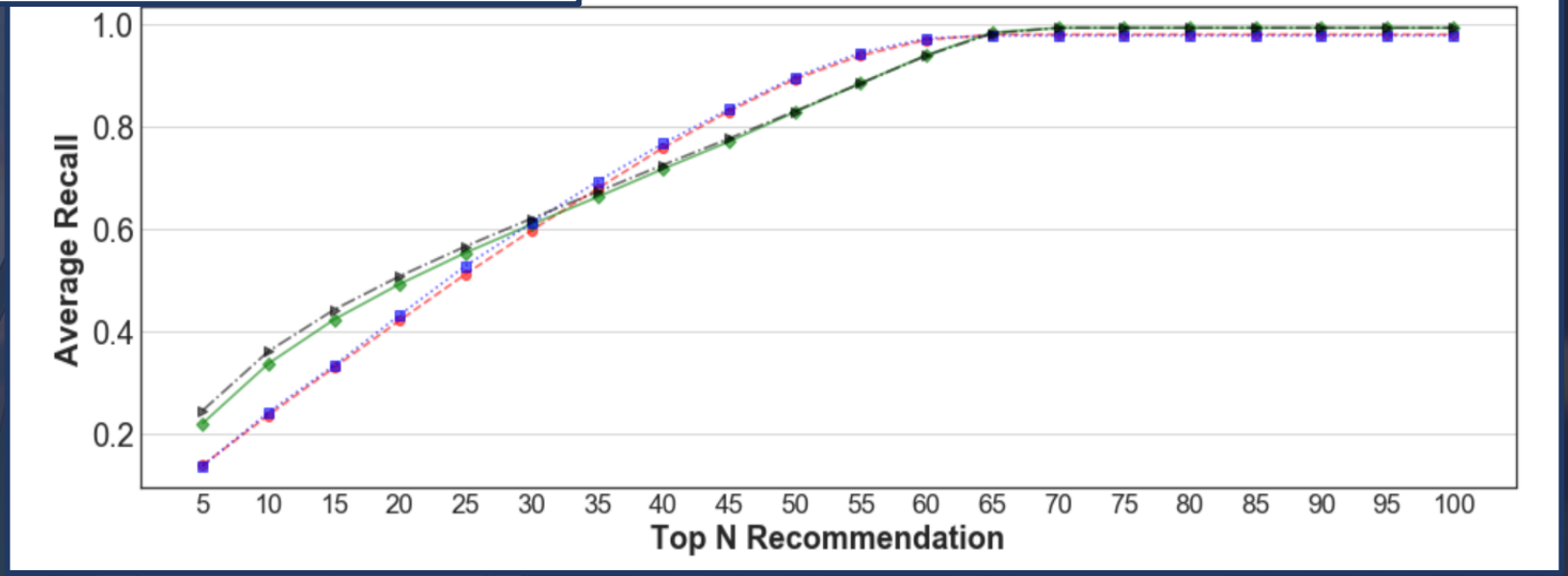


Precision :

a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved

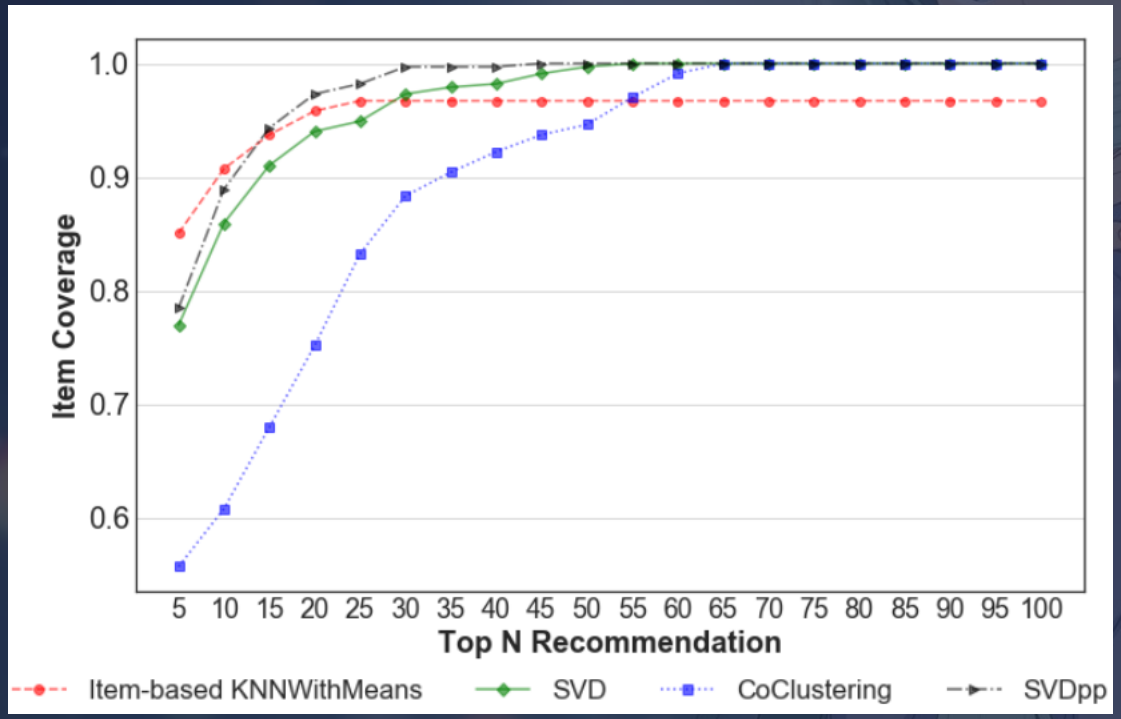
Recall :

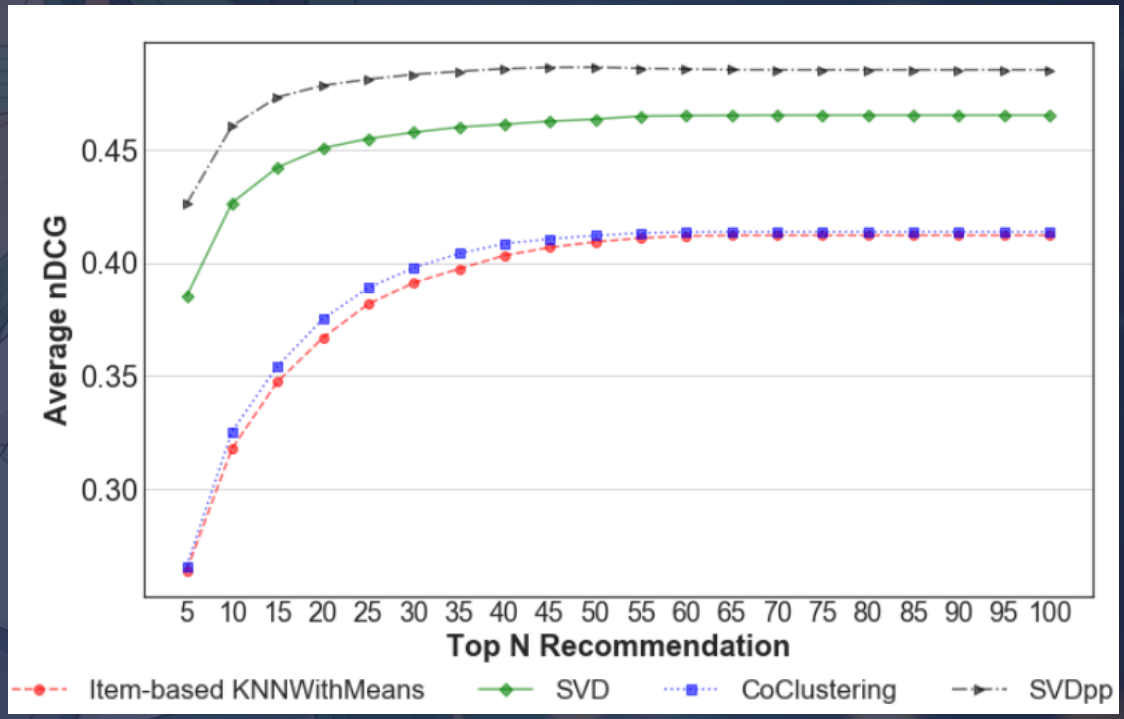
a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items



추천시스템 성능 평가

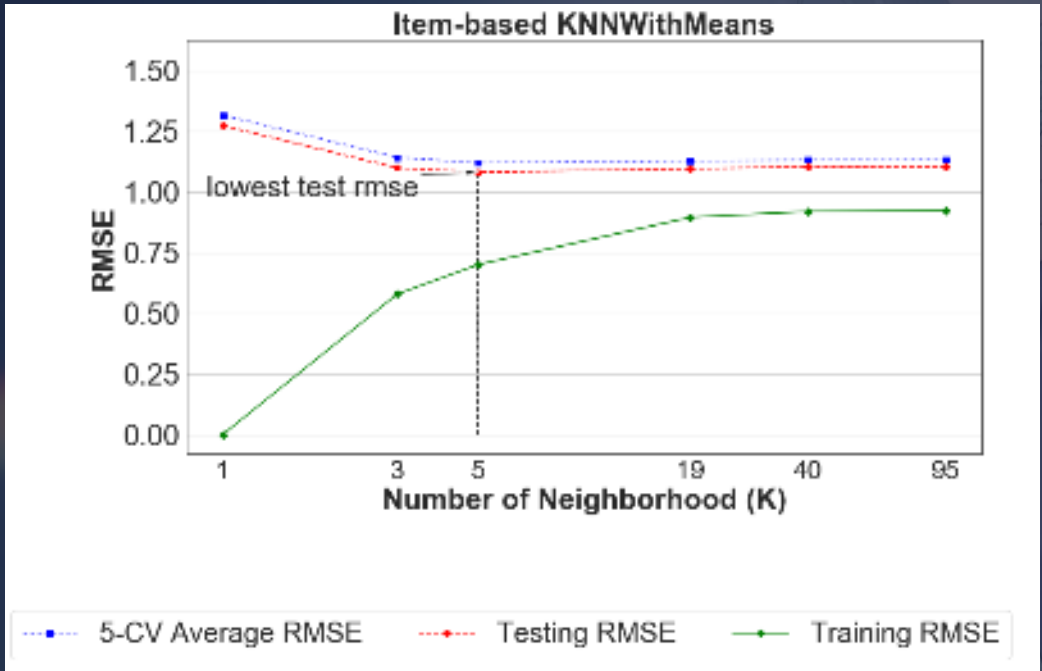
Recommendation Quality Evaluation



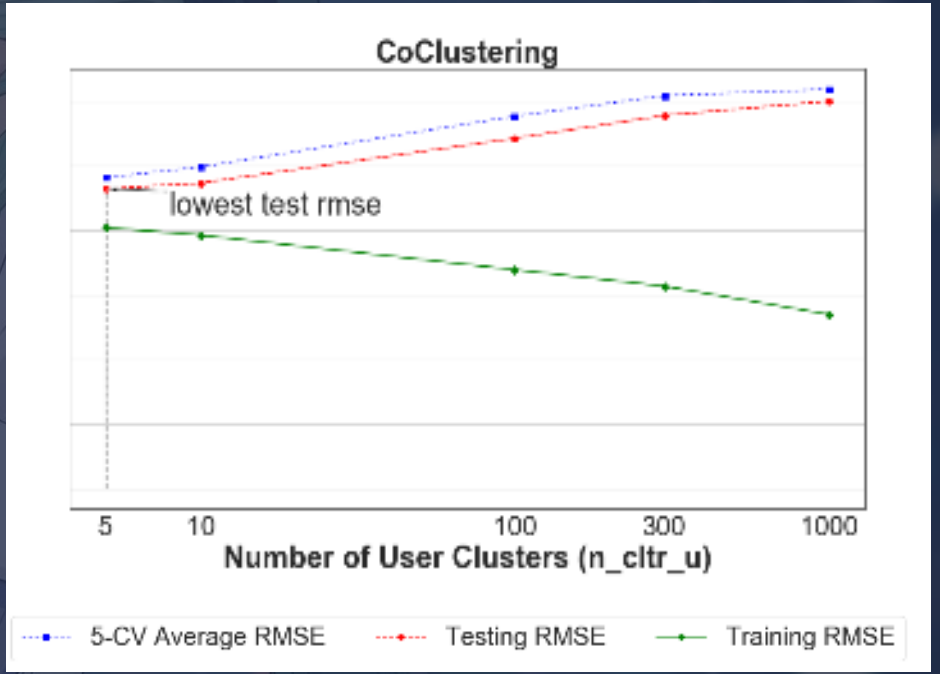


추천시스템 성능 평가

Knn



Co-clustering



추천시스템

Using R

Code

```r
install.packages("recommenderlab")
library(tidyverse)
library(recommenderlab)
library(stringr)
library(devtools)
data <- read.csv("rating.csv", header=T)
data_rating <- data[data$rating != -1,]
index <- sample(1:nrow(data_rating), 10000, replace=F)
data_rating <- data_rating[index,]

rating_df <- data_rating
rating_df <- rating_df %>%
  mutate(user_id = str_c("u", user_id),
         anime_id = str_c("m", anime_id))

rating_mat <- spread(rating_df, anime_id, rating) %>%
  remove_rownames() %>%
  column_to_rownames(var = "user_id")

rating_rrm <- as(as(rating_mat, "matrix"), "realRatingMatrix")

rating_rrm <- rating_rrm[rowCounts(rating_rrm) > 1,
                         colCounts(rating_rrm) > 0]

> print(object.size(rating_mat), units = "auto")
84.9 Mb
> print(object.size(rating_rrm), units = "auto")
302.4 Kb

rating_eval <- evaluationScheme(rating_rrm, method="split", train=0.7, given=1)

ubcf_rmse <- Recommender(getData(rating_eval, "train"), method = "UBCF",
                         param=list(normalize = "center", method="Cosine", nn=5))

ubcf_pred <- predict(ubcf_rmse, getData(rating_eval, "known"), type="ratings")
calcPredictionAccuracy(ubcf_pred, getData(rating_eval, "unknown"))

ubcf_pred <- predict(object = ubcf_rmse, newdata = rating_rrm,  n = 5)

recc_matrix <- sapply(ubcf_pred@items, function(x){
  colnames(rating_rrm)[x]
})
```

Output

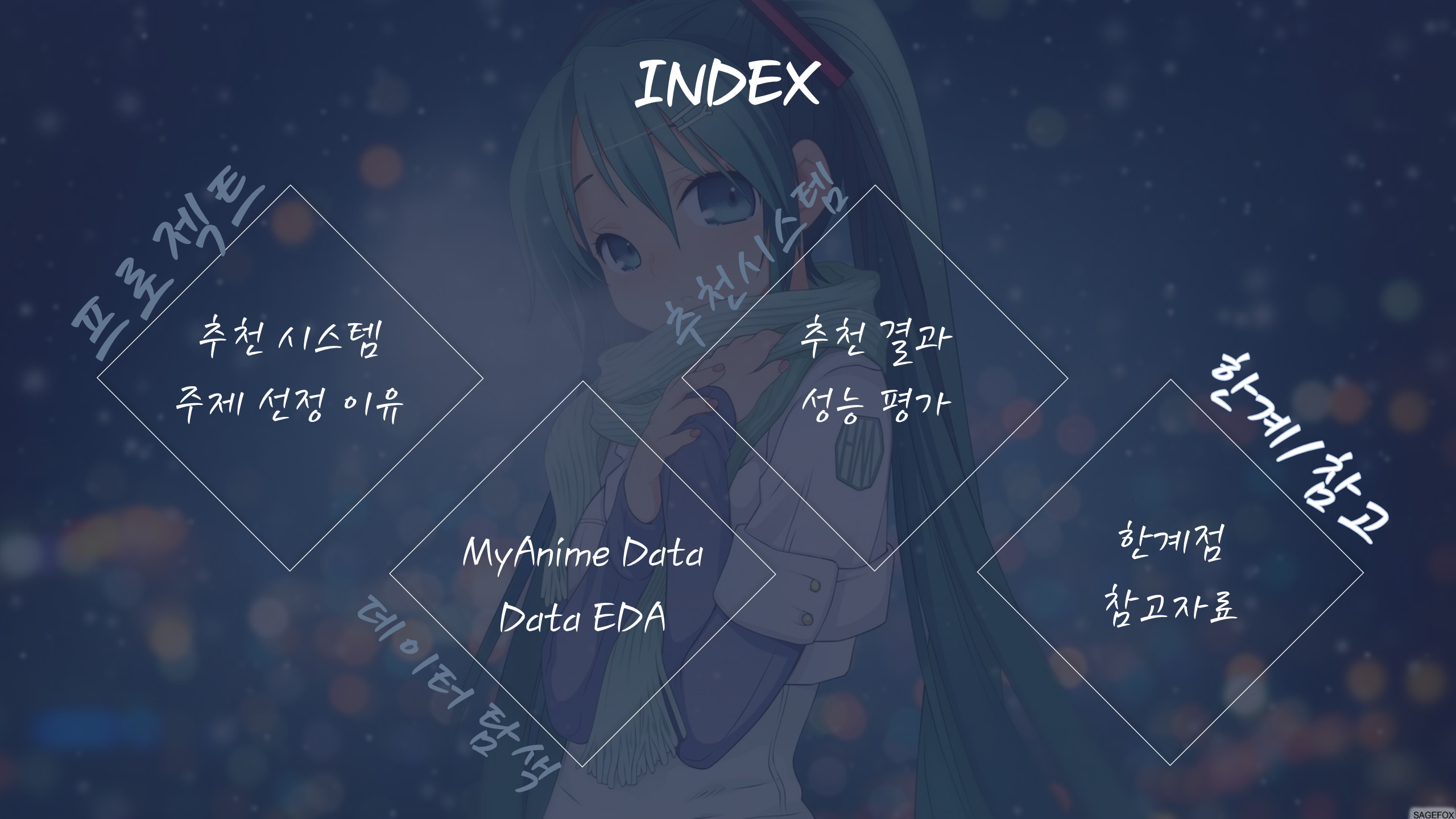
```
> recc_matrix[c(3,13,333)]
$`u10419`
[1] "m9253"  "m29803" "m1"     "m100"   "m10012"

$u10903
[1] "m1943"  "m26055" "m969"   "m2449"  "m1"

$u26192
[1] "m4985"  "m10119" "m9260"  "m1"     "m100"
```

- Steins;Gate /Carnival Phantasm/ Overload/ Cowboy Bebob/ Shin Shirayuki-hime Densetsu Prtear

- Paprika / JoJo no Kimyou na Bouken: Stardust Crusaders 2nd Season/ Tsubasa Chronicle 2nd Season/Ghost in the Shell: Stand Alone Complex - The Laughing Man/Overload

- Mahou Shoujo Lyrical Nanoha: The Movie 1st/Seitokai Yakuindomo OVA/Kizumonogatari I: Tekketsu-hen/ Overload/ Cowboy Bebob

INDEX



프로젝트
추천 시스템
주제 선정 이유

추천시스템
추천 결과
성능 평가

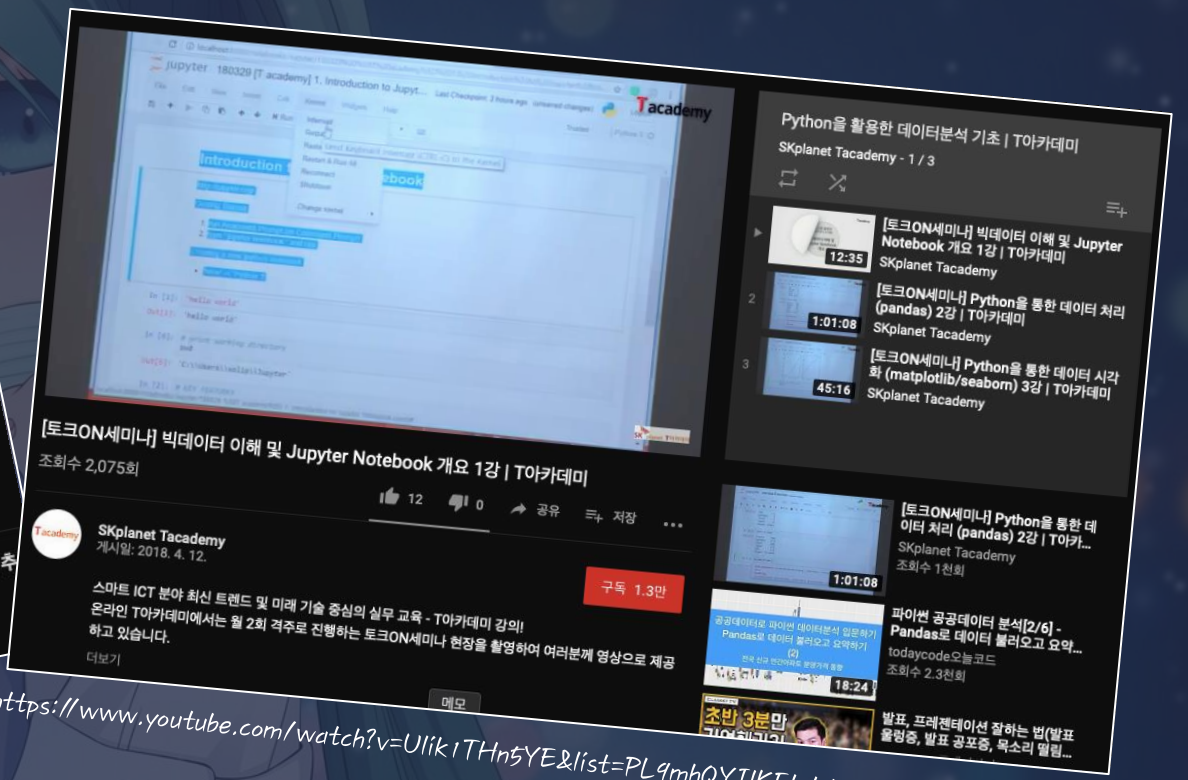
데이터 탐색
MyAnime Data
Data EDA

한계/참고
한계점
참고자료

참고

https://www.youtube.com/playlist?list=PLwQdCtnouPk6rOKwGFs-KOzS8P4qIdJzY

https://www.youtube.com/watch?v=Ulik1THn5YE&list=PL9mhQYIlKEhdnNrVs9MlVLramrRQ8jqis



도움을 주신 많은 논문과,,
블로거/엔지니어 분들 감사합니다^^,,

https://github.com/KU-BIG/recommendation_system

한계점

통컴실,, 열일,,

정재원

아직 * 떠있네 오후 1:24

강현

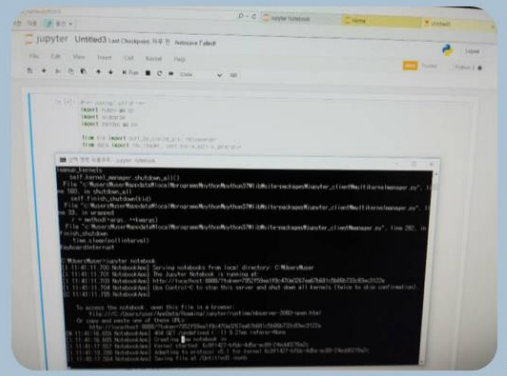
6. 04. 화

한평생 돌려야되나보다 ㅎ 오후 1:26

시간이 너무 오래 걸려요.ㅠ

2019년 6월 5일 수요일

정재원



오전 10:16

송민

아아 오전 10:16

송민

! 오전 10:17

윤 재경

아직도 돌아가고 있어?ㅋㅋㅋ

컴퓨터 열일하네.. 오전 11:44

안녕하살법!

첫 모임

=

마지막 모임

KU-BIG

한학기 동안 모두 수고 많으셨습니다!