

데이터베이스 Project 2

2024.06.11

기계항공공학부

2020-12696 이민석

※ DB2020_126960에 진행하였습니다 (DB2020_12696이 아닙니다!)

1. 본인이 설계한 데이터베이스 스키마 설명

books(b_id, b_title, b_author, b_quantity)

b_id -> PK / (b_title, b_author) -> UNIQUE / b_quantity -> 책의 잔여 권 수(대출 가능한 권 수)

users(u_id, u_name, u_borrowing)

u_id -> PK / u_borrowing -> 유저가 빌리고 있는 책의 권 수

borrowings(b_id, u_id, rating, now)

(b_id, u_id) -> PK / b_id, u_id -> FK / now -> 현재 대출 중인지 여부

(b_id, u_id)를 PK로 하여, 여러 번 대출이 발생한 경우 최신의 rating으로 덮어씀

2. 프로젝트 결과물 설명

A. 컴파일과 실행방법 : \$ python3 run.py

B. 핵심 모듈과 알고리즘에 대한 간략한 설명

Initialize_database() : 초기 DB안에 테이블이 없다고 가정하여, 테이블을 생성하고, data.csv로 초기 데이터를 설정한다.

remove_book,user() : 삭제가 가능할 경우 borrowings 테이블에 연관된 정보를 모두 지워줬다.

search_books() : 쿼리와 DB안에있는 books title을 모두 lower을 한뒤 like %query%로 비교하였다.

checkout_book() : 이미 b_id, u_id가 동일한 record 가 있으면, now만 true로 바꿔주었다. book의 대출 가능 권수와 user의 대출 가능 권수를 체크한 뒤에, borrowings에 insert를 하고, book의 대

출 가능 권수를 감소시키고, user의 대출 가능 권 수를 감소시켰다.

return_book() : now를 false로 바꾼 후, book의 대출 가능 권 수를 증가 시키고, user의 대출 가능 권수를 증가시켰다.

Recommend_popularity() : 유저가 평점을 남기지 않은 도서 중에 가장 평점 평균이 높은 도서와 가장 평점이 많은 도서를 추천하였다.

recommend_item_based() :

유저가 책을 하나도 안 읽었으면, 모든 유저와의 코사인 유사도가 0이 나오기에, weight sum이 의미 없이 모든 책에 대해 0이 나오게 된다. 그래서, 가장 첫번째 책을 추천한다.

반대로, 유저가 모든 책을 읽었다면, 추천할 책이 출력되지 않는다(칼럼명만 출력)

나머지의 경우에는 정상적으로 user-book-rating matrix를 구축하고, input으로 들어온 user_id와 코사인 유사도를 구한 뒤, weighted sum 방식으로 user가 읽지 않은 책에 대해 예상 평점을 구했다. 그리고 예상평점이 가장 높으면서, b_id가 가장 작은 책을 추천하였다.

대부분의 평점 평균 계산이 필요한 구현에서, books와 borrowings를 left outer join 하여, 평점 평균이 하나도 없는 책이라도 계산 결과에 나오도록 하였다. 그리고 평점평균이 0이이면 'None'을 출력하도록 하였다.

느낀점 : 코사인 기반 추천에서, 예외케이스를 처리하는게 까다로웠다.

Avg_rating이 None인 경우를 처리하는게 번거로웠다.