2021년 데이터 마이닝 실습 중간고사

제출안내: 코드를 따로 제출할 필요는 없이 분석 보고서를 작성하면 됩니다. 코드 설명이 필요한 경우는 코드를 보고서에 넣으면 됩니다. (캡처하는 경우는 해상도가 떨어지므로 보기 좋게 편집해서 넣어주세요)

분석결과에 대한 해석과 설명을 모두 가급적 자세하게 포함해야 합니다.

제출 마감: 10월 22일 금요일 자정 전까지 (23:59)

데이터 설명

take-home exam에 사용할 데이터는 왓챠(watcha.com)에서 사용자들의 영화 평점, 도서 평점을 수집한 데이터이다.

movie_train, movie_test는 영화 평점 데이터이고, book_train, book_test는 도서 평점 데이터이다. 각 데이터는 첨부된 csv파일들을 통해서 읽어올 수 있고, 동일한 파일을 RData 파일을 통해서 읽어올 수 있다. (파일 인코딩 문제가 발생하는 경우는 구글링을 통해 스스로 인코딩 문제를 해결하여야 합니다.)

train 데이터는 추천 모델을 학습하는 용도로 사용하고, 학습된 모델을 이용하여 test 데이터에 있는 평점을 얼마나 잘 예측하는지를 두고 성능을 평가합니다.

분석 및 추천의 편의성을 위해서, 평점 기록이 5개 이하인 영화/도서/사용자는 데이터로부터 제외하였습니다.

변수

user_code: 사용자들의 일련번호(ID) 영화/도서 데이터 공통, 동일 사용자는 동일 user_code를 가짐

movie_code: 영화의 일련번호(ID), 동일 영화는 동일 code를 부여받음.

book_code: 도서의 일련번호(ID), 동일 도서는 동일 code를 부여받음.

score: 영화 혹은 도서에 대한 사용자의 평점, 0.5점에서 5점 사이로 기록됨

movie_title: 영화제목

book_title: 책 제목

Task1 영화분석

영화들의 평균 평점과 평점의 수를 비교해보자. 평점의 수가 높은 영화는 많은 관객들이 감상한 인기있는 영화라고 생각할 수 있다. 인기 있는 영화일수록 평균 평점이 높다고 볼 수 있는가?

평점의 수가 적은 영화이지만 평점이 높은 영화인 경우 숨겨진 명작일 가능성이 있다. 이에 해당하는 영화가 있는지 확인하여보자.

반대로 평점의 수가 높지만 평점이 낮은 영화는 호/불호가 강하거나 피해야할 작품일지도 모른다. 여기에 해당하는 영화가 있는지 확인하여보자.

Task2 도서분석

도서의 경우 인기 도서가 평균 평점도 높다고 생각할 수 있는가?

근거를 들어서 설명해보자.

Task1과 마찬가지로 잘 알려지지 않은 명작 도서가 있는지 확인해보자.

반대로 인기도서지만 호불호가 강하거나 평가가 낮은 도서가 있는지 확인해보자.

Task3 사용자분석

user_code는 도서 데이터와 영화 데이터에서 공통으로 매겨진 것으로 같은 code는 동일 사용자를 의미한다. 도서와 영화 모두에 대해서 평점 기록이 있는 사용자들만을 추려서 아래의 내용을 분석해보자.

영화를 많이 보는(평점 기록이 많은) 사용자는 도서에 대한 평점기록도 많은 경향이 있는가?

비교적 관대하게 평점을 매기는 사용자가 있는가 하면 평점이 박한 사용자가 있을 것이다. 사용자의 이러한 경향은 영화와 도서에 대해 동일하게 나타나는가 아니면 서로 다른가?

i.e. 영화 평점이 generous한 사용자는 도서 평점도 generous한 경향이 있는가? 그 반대는 어떠한가?

Task4 영화 추천

UBCF와 IBCF를 사용해서 영화 추천을 수행해보자. train 데이터를 사용해서 test 데이터의 평점을

예측해보자.

예를 들어 train data에 사용자 A가 영화 a1과 a2에 대해 각 4.5, 3.5 점의 평점을 부여한 기록이 있을 때, test data에 사용자 A가 영화 a3에 대해 4.0이라는 평점 기록이 있다고 가정하자.

우리는 test data에 (A, a3, 4.0)이라는 정보를 모르는 것으로 하고, train 데이터의 정보만으로 A 사용자가 영화 a3를 얼마나 좋아할지 예측 평점을 계산하여야 한다. 가령 추천 시스템이 a3에 대한 평점을 3.5로 예측했다면, 예측 평점(3.5)과 실제 평점(4.0)의 차이(0.5)가 평점 예측의 오차가 된다. 평점 예측의 MAE(Mean Absolute Error)를 계산하여 보고 얼마나 정확하게 평점을 예측하는지 확인하여보자.

UBCF와 IBCF의 추천 성능을 비교하여보자.

Task5 도서 추천

Task4의 작업을 도서 데이터에 대해서 수행해보자.

추천 성능을 계산하여보고 영화 추천에 비해서 추천이 더 잘되는지, 잘 안되는지 설명하고 그 이유를 추론하여 설명하여보라.

Task6 self 추천

자신이 전에 감상했던 영화 20편에 대해서 본인의 평가를 담은 평점을 기록하고, 자신에게 추천을 수행해보라. 어떤 영화가 자신에게 추천되었으며 reasonable 추천 결과인지 설명해보라.

동일하게 도서에 대해서도 20(혹은 읽은 책이 없다면 더 적게 해도 됨)개의 평점을 데이터에 추가한 후 자신에 대한 도서 추천을 수행하고, 결과에 대해서 평가해보라.

Bonus Task1: cross-platform 추천

영화 데이터와 도서 데이터를 통합해서 추천 시스템(UBCF, IBCF)을 수행하고 테스트해보자. 추천 성능이 더 높아지는가, 더 낮아지는가? 이유는 무엇이라고 생각하는가?

도서 평점 데이터는 영화 평점 데이터에 비해서 분량이(volume)이 많이 적다. 따라서 도서 평점이 없는 사용자도 많이 있을 수 있다. 영화 평점기록은 있지만 도서 평점기록이 없는 사용자에게 어떻게 추천을 수행할 수 있을지 고민해보고, 고민한 방법을 사용하여 도서 추천을 수행하여보라.

실제로 영화는 많은 사람들이 관람하지만 이에 비해서 정기적으로 독서를 하는 인구는 많지 않다. 영화 취향을 바탕으로 해서 도서를 추천해줄 수 있다면, 어떤 책을 읽어야할 지 모르는 사용자에게 유용한 추천 시스템이 될 수 있을 것이다.

도서 추천의 평점 예측 성능을 평가하기 위해서는 영화 평점 기록과 도서 평점 기록이 모두 있는 사용자 중 일부를 선정하여, 이 사용자의 도서 평점기록을 없는 것으로 취합한 후 도서 추천을 수행하여, 추천된 도서의 예상 평점과 실제 도서 평점 사이의 오차를 계산할 수 있을 것이다.

Bonus Task2: 영화 추천 성능 개선

영화 추천 시스템에서 UBCF나 IBCF보다 더 추천 성능을 좋게 만들 수 있는 방법이 있는가? UBCF/IBCF 외에 다른 추천 알고리즘을 사용할 수도 있을 것이고, UBCF/IBCF를 개선하는 방법을 수행할 수 있을 것이다.

아마도 영화 혹은 도서의 평균 평점을 추천에 반영할 수 있을 것이고, 평점 수를 반영할 수도 있을 것이다.

또는 영화의 genre, 등급, 상영관 수, 감독, 국가 등의 meta 정보를 추천에 반영할 수 있을 것이다. 영화의 meta 정보는 추가적인 데이터 수집이 필요한데, 한국영화진흥원이나 IMDB와 같은 site에서 얻을 수 있을 것이다.

추천 성능을 개선했다면 예측 평점 오차(MAE)를 사용해 얼마나 추천 성능이 높아졌는지 설명하여보자.