**2021.11.01.**

프로젝트 진행을 위한 데이터셋 조사를 실시함. Kaggle에서 탐색한 결과

BoardGameGeek Reviews(<https://www.kaggle.com/jvanelteren/boardgamegeek-reviews>)

Board Game Geek Rankings(<https://www.kaggle.com/mseinstein/bgg_top2000>)

등 유용할 수도 있는 데이터셋을 몇가지 찾을 수 있었음. 특히 BoardGameGeek Reviews 데이터셋의 경우 csv 파일의 형태로 특정 유저의 닉네임. 유저가 남긴 보드게임 평점. 보드게임 평론과 같은 데이터를 저장하고 있어서 본 작품을 개발하는데 있어서 큰 도움이 될 것으로 예상됨. 필요한 경우 해당 데이터셋에 더해서 Board Game Geek이라는 웹사이트의 리뷰 데이터를 직접 크롤링하여 데이터를 수집하는 방법을 이용해도 괜찮을 것으로 보임.

**2021.11.11**

BoardGameGeek Reviews 데이터셋에 대한 분석을 진행하였음. 해당 데이터셋을 상세히 살펴본 결과 데이터셋이 협업 필터링에 기반한 추천 시스템 구현에는 적합하지 않을 수도 있을 것으로 생각됨. 한 유저가 다양한 보드게임에 대한 리뷰를 남기기 보다 유저 당 2~3개 정도의 적은 리뷰를 남기는 경우가 많음. 추가적인 데이터셋 탐색이 필요할 것으로 판단됨.

**2021.11.18**

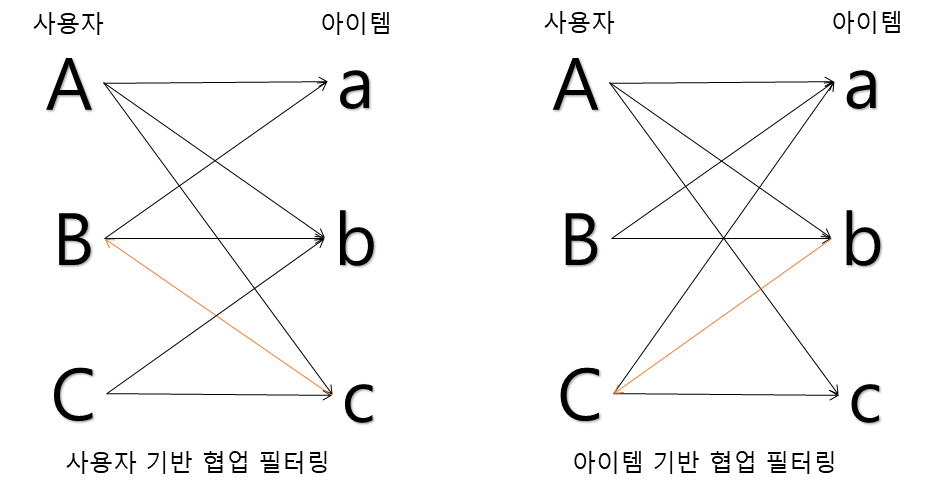
데이터셋의 추가적인 탐색 및 분석을 진행하였음

1. Board Game Data( <https://www.kaggle.com/mrpantherson/board-game-data?select=bgg_db_2018_01.csv> ): 다양한 보드게임의 종류와 각 보드게임 별로 간단한 정보의 요약이 제공됨. 또한 해당 보드게임의 더 상세한 정보를 담고 있는 BoardGameGeek 사이트로의 링크 정보도 제공하고 있기 때문에 크롤링으로 정보를 수집할 경우 유용하게 사용할 여지가 있음.
2. Board Games Dataset( <https://www.kaggle.com/gabrio/board-games-dataset> ): 90000여개의 보드게임에 관한 정보를 담고 있는 데이터셋. 위 Board Game Data에 비해 조금 더 상세한 설명이 포함되어 있는 점을 제외하고는 큰 차이가 없음. 콘텐츠 기반 필터링 구현에 이용할 수 있을 것으로 보임.
3. Board Games( <https://www.kaggle.com/andrewmvd/board-games> ): 위의 데이터셋들과 비교했을 때 Board Game Geeks의 유저 평점 데이터, 전체 랭킹이 추가되었음.

**2021.11.25**

Python을 활용한 협업 필터링 구현 방법에 대한 학습을 진행하였음. 우선 협업 필터링의 개념에 관하여 연구논문작품 제안서에 작성한 내용을 첨부함.

먼저 협업 필터링은 사용자들의 과거 경항이 미래에도 계속될 것이라는 전제를 이용한 알고리즘이다. 협업 필터링은 또다시 사용자 기반 협업 필터링과 아이템 기반 협업 필터링으로 나뉜다. 사용자 기반 협업 필터링은 사용자들의 기호를 분석하여 ‘A 사용자의 기호와 B 사용자의 기호가 서로 비슷하다면 A 사용자가 선호하는 아이템을 B 사용자도 선호할 것이다’라는 추측을 통해 추천하는 방식이다. 그리고 아이템 기반 협업 필터링은 ‘a 아이템을 선호하는 사용자와 b 아이템을 동시에 선호하는 사용자가 많다면 a 아이템과 b 아이템은 서로 유사한 것이다’라는 추측을 통해 추천하는 방식이다.



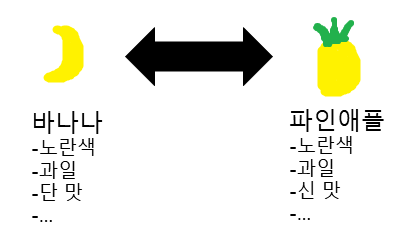
위 그림은 사용자 기반 협업 필터링과 아이템 기반 협업 필터링에 대해 나타내고 있다. 사용자 기반 협업 필터링을 보면 A 사용자와 B 사용자의 기호가 비슷하다고 판단하여 A 사용자만 선호하는 아이템 c를 B 사용자에게도 추천한다. 그리고 아이템 기반 협업 필터링에서는 a 아이템과 b 아이템을 동시에 선호하는 사용자가 많기 때문에 a 아이템을 선호하는 사용자 C에게 b 아이템도 추천하는 모습이다.

Python의 scikit learn을 이용하면 협업 필터링을 쉽게 구현할 수 있음. 사용자가보드게임에 준 평점을 바탕으로 시스템을 구현한다면, 평점은 실수 형태이기 때문에 간단하게 유사도를 구할 수 있기 때문임. Pandas를 활용해 csv 형태로 되어있는 데이터셋을 로드하고, scikit learn의 유사도 계산 함수들을 활용하여 사용자 및 아이템 간의 유사성을 모두 계산하고 이를 이용하여 추천 시스템을 만들면 됨.

**2021.12.02**

Python을 활용한 콘텐츠 기반 필터링 구현 방법에 대한 학습을 진행하였음. 우선 콘텐츠 기반 필터링의 개념에 관하여 연구논문작품 제안서에 작성한 내용을 첨부함.

콘텐츠 기반 필터링은 문자 그대로 각 콘텐츠(아이템)들이 가진 특성들의 유사도를 이용하여 추천하는 방식이다.



위 그림은 콘텐츠 기반 필터링에 대해 나타내고 있다. 바나나라는 아이템과 파인애플이라는 아이템의 특성들이 비슷하므로 바나나를 좋아하는 사용자에게는 파인애플을, 파인애플을 좋아하는 사용자에게는 바나나를 추천하게 될 것이다.

콘텐츠 기반 필터링에서 협업 필터링과는 다르게 주의를 기울여야 할 부분은 콘텐츠 기반 필터링에는 문자열 형태의 데이터가 이용된다는 점임. 따라서 협업 필터링처럼 간단하게 유사도를 계산할 수가 없음. 문자열 간의 유사도를 계산하기 위해서는 먼저 문자를 벡터화하는 작업이 필요함. Python의 scikit learn에서 제공하는 CountVectorizer 및 TfidfVectorizer, gensim의 Word2Vec 등을 고려해 볼 만 함.

**2021.12.09**

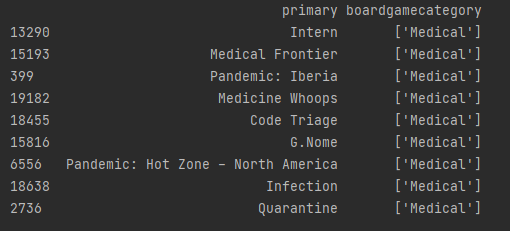
프로젝트의 Front-end 구현을 위해 Django 프레임워크에 관한 학습을 진행하였음. Django는 웹사이트를 편리하게 개발할 수 있도록 고안된 파이썬 웹 프레임워크로 오픈소스로 제공되어 다양한 커뮤니티로부터 정보를 얻을 수 있다는 특징이 있음. Django는 크게 urls, views, models, templates의 4가지 구조로 나누어 볼 수 있음. Urls는 python 코드로 이루어져 사용자의 HTTP 요청을 올바른 veiw들로 연결해 주는 역할을 함. Views 역시 python 코드로 이루어진 부분으로 urls로부터 받은 HTTP 요청에 대한 응답을 처리하는 부분임. 이때 views가 응답을 어떤 서식으로 보낼지 결정하기 위해 사용되는 부분이 templates임. Templates는 주로 HTML 파일들로 구성되어 있음. 마지막으로 models는 python 코드로 이루어진 부분으로 주로 데이터베이스의 관리 작업과 관련된 일을 처리하는 역할을 함. 이러한 Django의 특성을 살펴봤을 때 데이터베이스에 보드게임 추천과 관련된 정보를 저장하고, 해당 정보를 이용하여 사용자에게 추천 시스템을 제공하는 방식으로 본 프로젝트의 Front-end를 구현한다면 괜찮을 것이라 생각함.

**2021.12.16**

기본적인 추천 알고리즘의 틀을 작성함. 현재는 BoardGameGeek Reviews 데이터셋의 보드게임 카테고리를 기준으로 각 보드게임 사의의 유사도를 계산할 수 있는 단계임. 추가적으로 해당 데이터셋의 저작권 관련 라이선스를 조사하여 깃허브에 라이선스 파일을 추가하였음.

**2021.12.23**

보드게임의 카테고리를 기반으로 한 콘텐츠 기반 필터링을 완성하였음. 아래 사진은 구현된 코드를 이용하여 보드게임 ‘Pandemic’과 유사한 10개의 보드게임을 추천 받은 결과임.



데이터셋에서 보드게임의 카테고리 외에도 어떤 column을 콘텐츠 기반 필터링에 이용할 수 있을지 고민해 보고, 해당 알고리즘을 개선할 예정임.

**2021.12.30**

콘텐츠 기반 필터링을 수정하였음. 이제 보드게임의 카테고리 뿐만 아니라 해당 보드게임의 최소 인원, 최대 인원, 보드게임 플레이 방식의 정보도 필터링에 이용됨. 본 프로젝트가 보드게임 카페에서 사람들이 더욱 원활하게 어떤 보드게임을 플레이 할지 고를 수 있도록 돕는다는 목적을 가지고 있다는 점을 고려하였음. 보드게임 카페에서 같이 보드게임을 하는 경우 플레이하는 인원에 대한 제한이 꽤 중요하기 때문임.

**2022.01.06**

잠재 요인 협업 필터링을 기반으로 한 보드게임 추천 기능을 구현중임. 잠재 요인 협업 필터링은 이용자와 아이템 평점 행렬에 잠재되어 있는 요인이 있다고 가정하고, Singular Value Decomposition 기법을 통한 행렬 분해를 하여 잠재 요인들을 찾아내는 방식임. 원래 아이템 기반 협업 필터링을 기반으로 구현하려고 했으나 계획을 변경하였음. 잠재 요인 협업 필터링이 아이템 기반 협업 필터링에 비해 일반적으로 공간을 더 효율적으로 활용할 수 있기 때문임. 그럼에도 불구하고 데이터의 사이즈가 매우 커서 그런지 현재 메모리 관련 문제를 겪고 있음. 따라서 해당 문제를 해결하는 것을 최우선으로 진행중이며, colab과 같은 클라우드 환경을 활용하는 방법도 고려중임.

**2022.01.13**

아직 메모리 부족 문제를 해결하고자 시도중임. 아래는 적용해 본 방법의 목록임

1. 데이터셋의 일부를 무작위로 삭제하여 데이터셋의 크기를 줄임

데이터셋의 크기가 문제가 아니라 보드게임의 다양한 종류와 리뷰를 남긴 유저의 아이디의 개수가 많아서 발생하는 메모리 부족임. 따라서 일부 보드게임과 관련된 데이터 전체를 제거하는 방법을 이용하지 않고 무작위로 삭제하는 방법으로는 해결되지 않았음.

1. Colab 사용

colab에서 제공하는 메모리의 사이즈가 그렇게 크지 않기 때문에 colab 환경에서도 똑같이 메모리 부족 문제가 발생하였음.

1. 기타 코드 최적화

불필요한 메모리의 사용을 최대한 줄이는 작업을 진행하였음. 그러나 python 자체가 개발자에게 프로그램이 사용하는 메모리와 관련된 상세한 작업을 할 수 있도록 제공하고 있지 않고 코드 최적화를 통해 문제를 해결하지 못하였음.

**2022.01.20**

어쩔 수 없이 데이터셋에 포함된 일부 보드게임의 데이터를 아예 제거하는 방법을 이용하여 메모리 부족 문제를 해결하였음. 합리적인 기준을 정해서 데이터의 제거가 이루어진 것이 아니므로 이는 해당 문제의 궁극적인 해결방법이 될 수 없음. 추후에 보드게임의 인기와 같은 기준을 정해서 몇몇 보드게임은 제외하는 방식으로 개선할 필요가 있음. 혹은 메모리 크기를 보강하여 하드웨어적인 방법으로도 해결이 가능함. 이로써 본 작품의 핵심이 되는 아이템 기반 협업 필터링, 그리고 콘텐츠 기반 필터링의 구현이 완료되었음. 지금부터는 이를 활용한 Front-end 개발 작업에 돌입할 예정임.

**2022.01.28**

본 연구작품의 Front-end 부분인 웹 어플리케이션 개발을 위해서는 Django 프레임워크를 이용할 예정임. 현재는 본 작품의 제안서에서 제시되었던 웹 UI 구상도와 비슷하도록 html 코드(Django의 템플릿에 해당함)를 작성하는 중임. 아래는 제안서의 구상도를 첨부한 것임.

