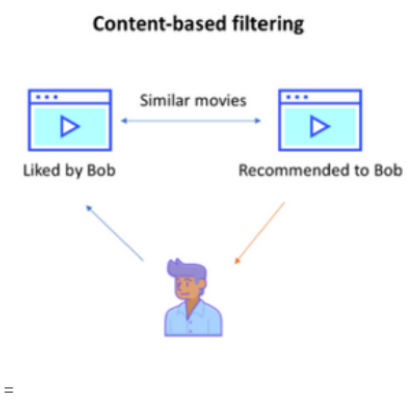
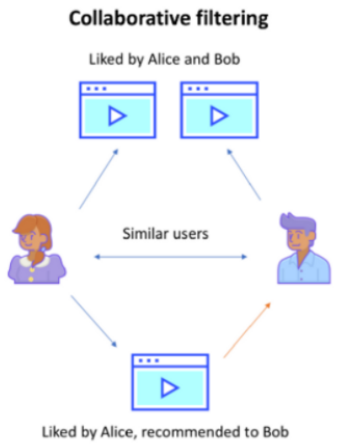
컨텐츠기반모델 이해

Writer	최윤서(학부학생/공과대학 도시공학)
키워드	
Reference	
주차	1주차

컨텐츠기반 필터링 vs 협업필터링

- 컨텐츠 기반 = 해당 사용자가 **좋아하는 컨텐츠와 비슷한 컨텐츠**를 추천
- 협업필터링 = **다른 사용자들로부터** 얻은 정보를 바탕으로 컨텐츠 추천
    - ex) 나와 취향이 비슷한 사용자들이 선호하는 상품을 추천
    - ex) 다른 사람들로부터 특정 상품과 유사한 평가를 받은 상품을 추천





컨텐츠기반 모델

> 해당 사용자가 **좋아하는 콘텐츠와 비슷한 콘텐츠**를 추천

How

1. 아이템을 벡터 형태로 표현
   ex) TF-IDF, Word2Vec
2. 벡터들간의 유사도를 계산
3. 자신과 유사한 벡터를 추출
    a. implicit feedback data라면 유사한 벡터들 순서대로 추출
    b. explicit feedback data라면 유사도 높은 순서대로 그대로 출력하는 것보다 컨텐츠별로 가중 평점을 계산 → 유사도 높은 컨텐츠 중에 평점이 좋은 컨텐츠 순으로 추천하는 방향으로.

Vectorization 방법(1단계)

- **Count Vectorizer**
    - 단순 빈도만으로 중요도를 계산하기에는 조사, 관사처럼 의미는 없지만 많이 등장하는 단어들이 있음
    - 이들이 중요하지 않다는 페널티(IDF)를 주기 위해서 TF-IDF
- **TF-IDF**
    - 다른 문서에서는 등장하지 않지만 특정 문서에서만 자주 등장하는 단어를 찾아 가중치를 계산

    - TF = 특정 문서에 단어 등장 횟수

      DF = 특정 단어가 등장한 문서의 수

      IDF = DF에 반비례하는 수 (소프트웨어마다 조금씩 정의하는 방법이 다름)

      → TF * IDF 매트릭스 만들어주기
    - 장점) 직관적인 해석이 가능함
    - 단점) 메모리 문제. (높은 차원, sparse 데이터)
- **Word2Vec**
    - 비슷한 위치에 등장하는 단어들은 비슷한 의미를 가진다는 가정 하에 단어 의미의 유사도를 반영하는 임베딩 방법.
    - CBOW 알고리즘, skip-gram 알고리즘 (2주차에 이어서 할 예정)
    - TF-IDF의 sparsity 문제(메모리 문제) 해결

유사도 계산 방법(2단계)

- 유클리디안 유사도
    - 1/유클리디안 거리로 문사간의 유사도를 계산 (거리 가까우면 유사도 높음)
    - 계산 쉽고 **각 column별 크기에 민감**할 때 효과볼 수 있음
    - 단어별 분포가 다르거나 범위가 다른 경우 상관성을 놓침
- 코사인유사도
    - 거리가 아니라 **방향이 비슷할때** 유사도 높음
    - 단어들 빈도수(크기)가 비슷하지 않아도 **문서내에서 얼마나 나왔는지 비율**
    - 벡터의 크기가 중요한 경우 잘 작동X
- 피어슨 유사도 = 상관계수
- 자카드 유사도 = 단어 교집합의 크기 / 단어 합집합의 크기
- 그 외

  여러가지 비교해보고 적절하게 사용하자

  여러가지 유사도 매트릭스에서 가중치 조합해서 새로운 유사도 매트릭스를 만들기도함

  or 아예 추천 이후에 결과 결합하기도 함