[추천시스템들어가기전-사전학습: DB의 이해] 데이터베이스란 와 DBMS의 종류

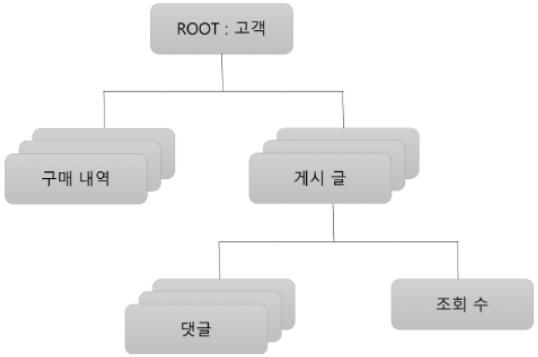
2. DataBase의 이해.ipynb

데이터베이스란 여러 사람들이 공유하고 사용할 목적으로 통합 관리되는 **데이터들의 모임**이다.
등산할 때 기반이 되는 기지를 베이스캠프라 하듯이 데이터베이스라는 용어도 1950년대 미국에서 데이터의 기지라는 뜻에서 데이터베이스라는 용어를 처음 사용했다고 한다.
데이터베이스를 관리하는 시스템을 DBMS라 한다.
.

데이터베이스 관리 시스템의 종류

계층형 (Hierarchical DataBase)

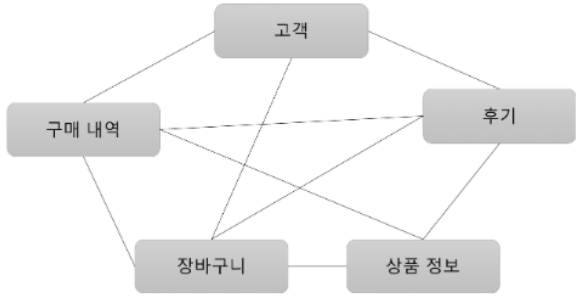
- 데이터 간의 관계가 트리 형태의 구조이다. 트리는 부모-자식 관계로 표현되며 부모와 자식 간에는 1:N (일 대 다)로 구성될 수 있다.
  데이터를 세그먼트 (레코드) 단위로 관리하며 세그먼트 간 계층을 트리구조로 관리한다. 구조가 간단하고 구현, 수정, 검색이 쉽지만 부모 자식 간에 N:N (다 대 다) 관계 처리가 불가능하고, 구조 변경이 어렵다.
- DBMS 예 : IMS (IBM 의 Information Management System)



< 계층형 데이터베이스 모델 구조 >

네트워크형 (Network DataBase)

- 계층형 데이터베이스의 단점을 보완하여 데이터 간 N:N (다 대 다) 구성이 가능한 망 형 모델이다. 계층 구조에 링크를 추가하여 유연성과 접근성을 높였다. 하지만 구조가 복잡해 유지보수가 어렵다.
- DBMS 예 : IDMS (Integrated Data Store)



< 네트워크형 데이터베이스 모델 구조 >

[추천시스템들어가기전-사전학습: DB의 이해] 데이터베이스란 와 DBMS의 종류

2. DataBase의 이해.ipynb

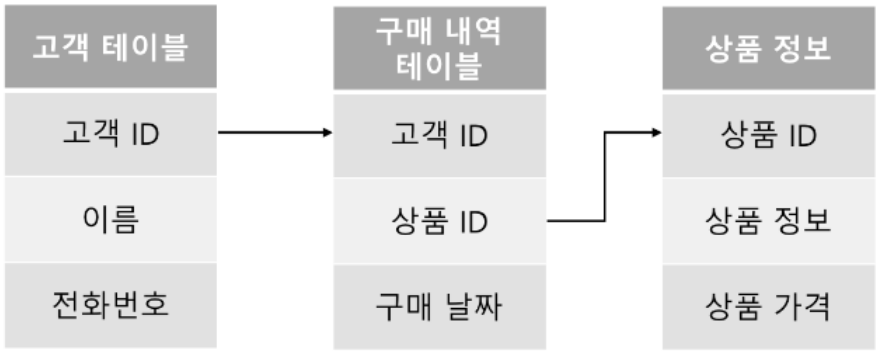
데이터베이스란 여러 사람들이 공유하고 사용할 목적으로 통합 관리되는 **데이터들의 모임**이다.
등산할 때 기반이 되는 기지를 베이스캠프라 하듯이 데이터베이스라는 용어도 1950년대 미국에서 데이터의 기지라는 뜻에서 데이터베이스라는 용어를 처음 사용했다고 한다.
데이터베이스를 관리하는 시스템을 DBMS라 한다.

.

데이터베이스 관리 시스템의 종류

관계형 (Relational DataBase)

- 관계형 데이터베이스 모델은 키(key)와 값 (value)으로 이루어진 데이터들을 행(row)과 열 (Column)로 구성된 테이블 구조로 단순화 시킨 모델이다. SQL (Structured Query Language) 를 사용하여 테이터를 처리한다. 데이터 모델링이 간단하지만 CAD/CAM , GIS 등과 같은 비정형 데이터들을 다루거나 실시간 분석에는 적합하지 않다.
- DBMS 예 : MySQL



< 관계형 데이터베이스 모델 구조 >

데이터베이스 관리 시스템의 특징

데이터의 독립성 / 데이터의 무결성 / 데이터의 보안성 / 데이터의 일관성 / 데이터의 중복 최소화

[추천시스템들어가기전-사전학습: DB의 이해] 데이터베이스의 무결성 규칙

무결성 제약 조건이란 데이터베이스에 들어 있는 데이터의 정확성, 일관성, 유효성, 안정성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말한다.
- 무결성 규정의 대상으로는 **도메인, 키, 종속성, 관계성 등**이 있다

1) 개체 무결성 (Entity integrity)

모든 테이블이 기본 키 (primary key)로 선택된 필드 (column)를 가져야 한다. 기본 키로 선택된 필드는 고유한 값(중복허용하지 않음) 을 가져야 하며, 빈 값은 허용하지 않는다.

2) 참조 무결성 (Referential integrity)

관계형 데이터베이스 모델에서 참조 무결성은 참조 관계에 있는 두 테이블의 데이터가 항상 일관된 값을 갖도록 유지되는 것을 말한다.

3) 도메인 무결성 (Domain integrity)

도메인 무결성은 테이블에 존재하는 필드의 무결성을 보장하기 위한 것으로 필드의 타입, NULL값의 허용 등에 대한 사항을 정의하고, 올바른 데이터의 입력 되었는지를 확인하는 것이다. 예를 들어, 주민등록번호 필드에 알파벳이 입력되는 경우는 도메인 무결성이 깨지는 경우라고 볼 수 있다. DBMS의 기본값 설정, NOT NULL 옵션 등의 제약 사항으로 도메인 무결성을 보장할 수 있다.

4) 무결성 규칙 (Integrity rule)

데이터베이스에서 무결성 규칙은 데이터의 무결성을 지키기 위한 모든 제약 사항들을 말한다. 비즈니스 규칙 (business rule)은 데이터베이스를 이용하는 각각의 유저에 따라 서로 다르게 적용되지만, 무결성 규칙은 데이터베이스 전체에 공통적으로 적용되는 규칙이다.

기본키는 중복되거나 NAN일수 없으며 외래키는 다른 테이블의 기본키로서 두테이블을 연결하는 key

customerDB

id	age	gender	grade
1	25	F	A+
2	45	M	B+
3	32	F	A+

discountDb

grade	discount_rate
A	0.3
A+	0.4
B	0.1
B+	0.3

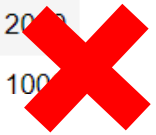
pumMokDB

pumid	pumMokNmae	Danga
10	book	1000
20	ball	240
30	pen	2000
40	brush	1500

	id	pumid	cnt	age	gender	grade	discount_rate	pumMokNmae	Danga
0	1	10	5	25	F	A+	0.4	book	1000
1	2	10	1	45	M	B+	0.3	book	1000
2	1	20	2	25	F	A+	0.4	ball	240
3	1	10	1	25	F	A+	0.4	book	1000
4	1	20	2	25	F	A+	0.4	ball	240
5	2	10	1	45	M	B+	0.3	book	1000
6	1	40	1	25	F	A+	0.4	brush	1500
7	1	20	2	25	F	A+	0.4		40

```
# 아래와 같이 DB를 구축하면 삽입, 삭제, 갱신 이상이 생길수 있음.
import pandas as pd
df=pd.DataFrame({'id':[1,2,1,2,1,1,1,2,3,1],
                 'id_gender':['F','M','F','M','F','F','F','M','M','F'],
                 'id_age':[25,45,25,45,25,25,25,45,32,25],
               'pumMok':['pen','book','pen','pen','pen','book','pen','book
               'count':[5,1,3,1,3,2,1,1,3,1],
                'danga':[2000,10000,2000,2000,2000,1000,2000,1000,1500,15
df
```

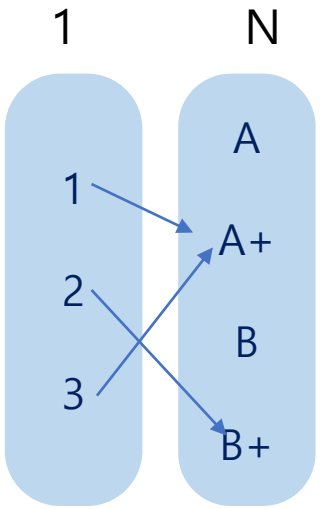
	id	id_gender	id_age	pumMok	count	danga
0	1	F	25	pen	5	2000
1	2	M	45	book	1	10000
2	1	F	25	pen	3	2000
3	2	M	45	pen	1	2000
4	1	F	25	pen	3	2000
5	1	F	25	book	2	1000
6	1	F	25	pen	1	20
7	2	M	45	book	1	100



구글검색

'데이터베이스] 이상(Anomaly)'

기본키는 중복되거나 NAN일수 없으며 외래키는 다른 테이블의 기본키로서 두테이블을 연결하는 key




```
customerDB=pd.DataFrame({'id':[1,2,3],
                         'age':[25,45,32],
                         'gender':['F','M','F'],
                         'grade':['A+','B+','A+']})
customerDB
```

	id	age	gender	grade
0	1	25	F	A+
1	2	45	M	B+
2	3	32	F	A+

```
discountDb=pd.DataFrame({'grade':['A','A+','B','B+'],
                         'discount_rate':[0.3,0.4,0.1,0.3]})
discountDb
```

	grade	discount_rate
0	A	0.3
1	A+	0.4
2	B	0.1
3	B+	0.3

고객정보(기본키)에
할인율정보(외래키)를 넣을때 ⭕

1, 25, F, A+, 0.3
2, 45, M, B+, 0.1
3, 32, F, A+, 0.1

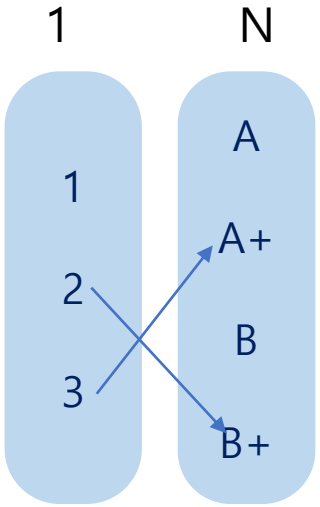
→ 할인율 참조무결성O

할인율정보(기본키)에
고객 정보(외래키)를 넣을때 ❌

A, 0.3 , Nan
A+, 0.4, 1
A+, 0.4, 3
B, 0.1, Nan
B+, 0.3, 2

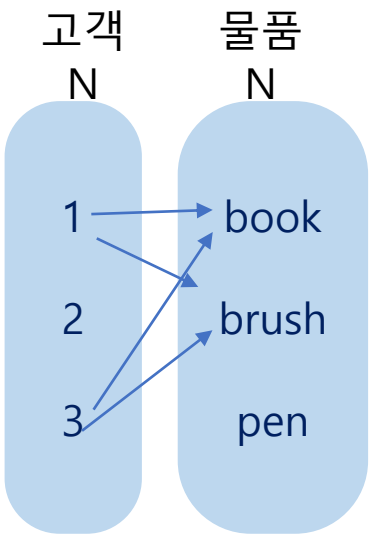
→ 고객id 참조무결성X

기본키는 중복되거나 NAN일수 없으며 외래키는 다른 테이블의 기본키로서 두테이블을 연결하는 key




```
# 고객정보의 grade 값에 따라 할인율 정보를 갖고옴
df=pd.merge(customerDB,discountDb, on='grade', how='left')
df
```

	id	age	gender	grade	discount_rate
0	1	25	F	A+	0.4
1	2	45	M	B+	0.3
2	3	32	F	A+	0.4



- **한명의 고객은 여러물품을 구매할수 있으며, 구매이력이 없는 고객이 존재할수도 있다.**
  - 기본키를 고객id로 하고 물품id를 외래키로 했을때 구매 중복과, Nna 값이 생길수 있음.
- **물품은 여러명에게 팔릴수 있으며, 판매이력이 없는 물품도 존재할수 있다.**
  - 기본키를 물품id로 하고 고객id를 외래키로 했을때 중복과, Nan값이 생길수 있음.

이러한 경우는 고객id와 물품id를 별도의 테이블로 구성하고 중복과 NAN이 없는 새로운 index 기본값을 생성한다. (다대다 관계임)

2. DataBase의 이해.ipynb

기본키는 중복되거나 NAN일수 없으며 외래키는 다른 테이블의 기본키로서 두테이블을 연결하는 key

```
### 생성된 다대다 관계는 merge 하여
## 모든 테이블을 연결함.
## 다대다 관계의 판매데이터를
## 고객데이터(df: 고객데이터+할인율)와 연결
df1=pd.merge(pan,df,on='id', how='left')
df1
```

	id	pumid	cnt	age	gender	grade	discount_rate
0	1	10	5	25	F	A+	0.4
1	2	10	1	45	M	B+	0.3
2	1	20	2	25	F	A+	0.4
3	1	10	1	25	F	A+	0.4
4	1	20	2	25	F	A+	0.4
5	2	10	1	45	M	B+	0.3
6	1	40	1	25	F	A+	0.4
7	1	20	2	25	F	A+	0.4

```
df1=pd.merge(df1,pumMokDB,on='pumid', how='left')
df1
```

	id	pumid	cnt	age	gender	grade	discount_rate	pumMokNmae	Danga
0	1	10	5	25	F	A+	0.4	book	1000
1	2	10	1	45	M	B+	0.3	book	1000
2	1	20	2	25	F	A+	0.4	ball	240
3	1	10	1	25	F	A+	0.4	book	1000
4	1	20	2	25	F	A+	0.4	ball	240
5	2	10	1	45	M	B+	0.3	book	1000
6	1	40	1	25	F	A+	0.4	brush	1500
7	1	20	2	25	F	A+	0.4	ball	240

[추천시스템들어가기전-사전학습: DB의 이해] 무비렌즈 데이터 셋 (https://grouplens.org/da

2. DataBase의 이해.ipynb

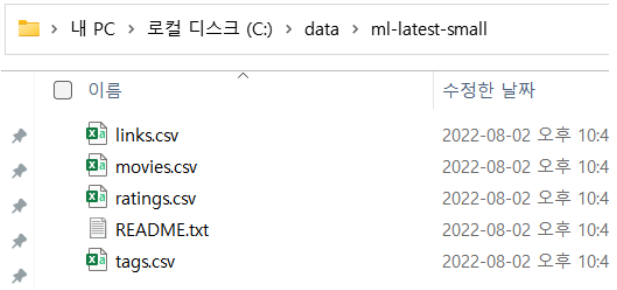
MovieLens Latest Datasets

These datasets will change over time, and are not appropriate for r
links stable for automated downloads. We will not archive or make

Small: 100,000 ratings and 3,600 tag applications applied to 9,000

- README.html
- ml-latest-small.zip (size: 1 MB)

다운로드 압축해제



ratings.csv(영화평점)

userId	movieId	rating	timestamp
2	318	3	1445714835
2	333	4	1445715029
2	1704	4.5	1445715228
2	3578	4	1445714885
2	6874	4	1445714952
2	8798	3.5	1445714960
2	46970	4	1445715013
2	48516	4	1445715064
2	58559	4.5	1445715141
2	60756	5	1445714980
2	68157	4.5	1445715154
2	71535	3	1445714974

tags.csv(영화평)

userId	movieId	tag	timestamp
2	60756	funny	1445714994
2	60756	Highly quo	1445714996
2	60756	will ferrell	1445714992
2	89774	Boxing sto	1445715207
2	89774	MMA	1445715200
2	89774	Tom Hardy	1445715205
2	106782	drugs	1445715054
2	106782	Leonardo	1445715051
2	106782	Martin Scc	1445715056
7	48516	way too lo	1169687325
18	431	Al Pacino	1462138765
18	431	gangster	1462138749

movies.csv(영화정보)

movieId	title	genres
60735	Shotgun Stories (2007)	Drama|Thriller
60737	Watching the Detectives (2007)	Comedy|Romance
60753	Felon (2008)	Crime|Drama
60756	Step Brothers (2008)	Comedy
60760	X-Files: I Want to Believe, The (2008)	Drama|Mystery|Sci-Fi|Thriller
60766	Man on Wire (2008)	Documentary
60803	Little Drummer Boy, The (1968)	Animation|Children|Musical
60818	Hogfather (Terry Pratchett's Hogfathe	Adventure|Fantasy|Thriller
60832	Pathology (2008)	Crime|Horror|Thriller
60857	Tracey Fragments, The (2007)	Drama
60885	Zone, The (La Zona) (2007)	Drama|Thriller
60894	Edge of Love, The (2008)	Drama|Romance|War

https://www.imdb.com/title/tt0838283/

ratings.csv의 평화평점 정보의 userid별 moviedid의 영화에 대한
영화평을 tags.csv의 userid별 moviedid의 tag 정보와
movies.csv의 moviedid의 title 및 genres를 연결하여서
한개의 영화정보 자료를 제작할수 있음.

추천시스템(recommender system) 개요

3. 추천시스템 TF-IDF.ipynb

1) 정의

- 데이터를 통해서 사용자가 아직 소비하지 않은 상품(item)중 선호할 만한것을 예측하는 것임.

- Amazon과 같은 인터넷 쇼핑 사이트나 Netflix등의 온라인 비디오 콘텐츠 제공
- 사이트에서 사용자가 각각의 상품에 대한 평점을 기반으로 추천시스템 활용

전체 추천	그룹 추천	개인 추천	아이템 기반추천
불특정 다수에 대한 무작위 추천 TV광고, 웹사이트 광고배너, 실시간 검색어등	사용자를 특정 segment 로 나누어 특화된 추천을 제공, 지역별 부동산 추천등	사용자의 이력을 바탕으로 관심사를 추측 주로 협업필터링 많이 사용함	최근에 본 유사한 상품, 추천 영화 동영상 (개인추천+하이브리드형태로결합)

2) rating(평가, 평점)

- 아이템에 대한 사용자의 호감도를 매기는 것을 rating이라 표함

- Explicit rating: 사용자가 평점을 매기는 것처럼 사용자의 분명한 피드백을 지칭

✓ 평점 및 후기에 대한 데이터 부족

✓ 점수 편향존재(상품 경험이 좋지 않은 사람은 큰 확률로 피드백을 주지 않을 확률 높음)

✓ 점수에 대한 범위가 다름(2~4, 3~5점등 사용자별 점수범위에 대한 의미가 다름)

- Implicit feedbacK: 조회여부, 구매여부에 따라 추청하며 상황 별 점수부여 가능

✓ -1점: 목록에 노출되었으나 조회하지 않을 때 / 0점: 목록에 노출되지 않을 때

✓ 1점: 조회 / 3점: 구매나 특정 액션을 취할 때

https://tv.naver.com/v/2297146

0-0. 추천시스템알고리즘의 소개.pdf 의 5p~

https://brunch.co.kr/@tobesoft-ai/8

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링)

1) 콘텐츠 기반 필터링(content based filtering)

- 초반에 많이 사용하던 추천시스템 방식임

- 사용자가 특정 아이템을 선호하는 경우 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천해주는 방식

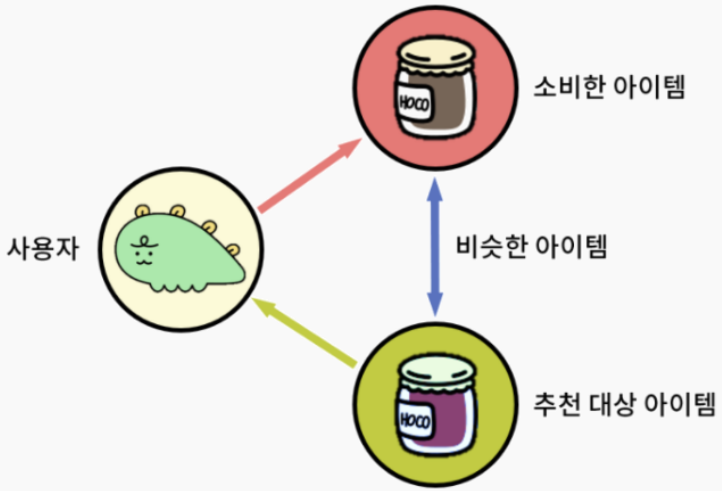
- 각각의 사용자와 아이템에 대하여 프로필을 작성하고, 이를 기반으로 추천

✓ user-based recommendation: 영화추천시 사용자에 대한 성별, 연령대, 지역 등의 프로필 작성 후 프로필이 비슷한 다른 사용자가 선호하는 영화를 추천함

✓ item-based recommendation: 영화추천시 영화에 대한 장르,배우,흥행 여부등의 프로필 작성 후 이를 기반으로 특정 영화를 좋아했던 사용자에게 비슷한 영화를 추천함.

- 프로필 작성에 대한 시간과 주관적 자료가 포함 될수 있음.

유저A가 높은 평점을 주거나 큰 관심을 갖는 아이템 x와 유사한 아이템 y를 추천
웹사이트, 블로그, 뉴스 게시글 추천 / 장르,배우,감독등 비슷한 특징을 갖는 영화 추천



출처: 카카오 기술블로그

카카오 컨텐츠 기반 필터링
https://tech.kakao.com/2021/12/27/content-based-filtering-in-kakao/

콘텐츠 기반 필터링은 우측상단 표현된 것과 같이, 사용자가 소비한 아이템에 대해 아이템의 내용(content)이 비슷하거나 특별한 관계가 있는 다른 아이템을 추천하는 방법을 말합니다. 아이템의 내용은 아이템을 표현할 수 있는 데이터를 지칭하는데, 아이템 카테고리, 아이템 이름과 같은 텍스트 데이터, 이미지 데이터가 주로 사용됩니다. 다른 사용자의 아이템 소비 이력을 활용하는 협업 필터링(Collaborative filtering) 과는 주로 사용하는 데이터가 다르다는 차이점이 있습니다.

콘텐츠 기반 필터링은 아이템 정보만 있으면 추천이 가능하기 때문에 소비 이력이 없는 새로운 아이템에 대한 추천이 바로 가능하다는 장점이 있습니다. 하지만 충분한 소비 이력이 쌓인 아이템에 대해서는 협업 필터링에 비해 추천 성능이 밀린다는 인식이 보편적입니다. 이런 이유로, 콘텐츠 기반 필터링은 추천 대상 아이템이 빠르게 바뀌는 상황이나 소비 이력이 적은 아이템에 대해, 협업 필터링을 보완하는 용도로 많이 활용됩니다.

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링)

3. 추천시스템 TF-IDF.ipynb

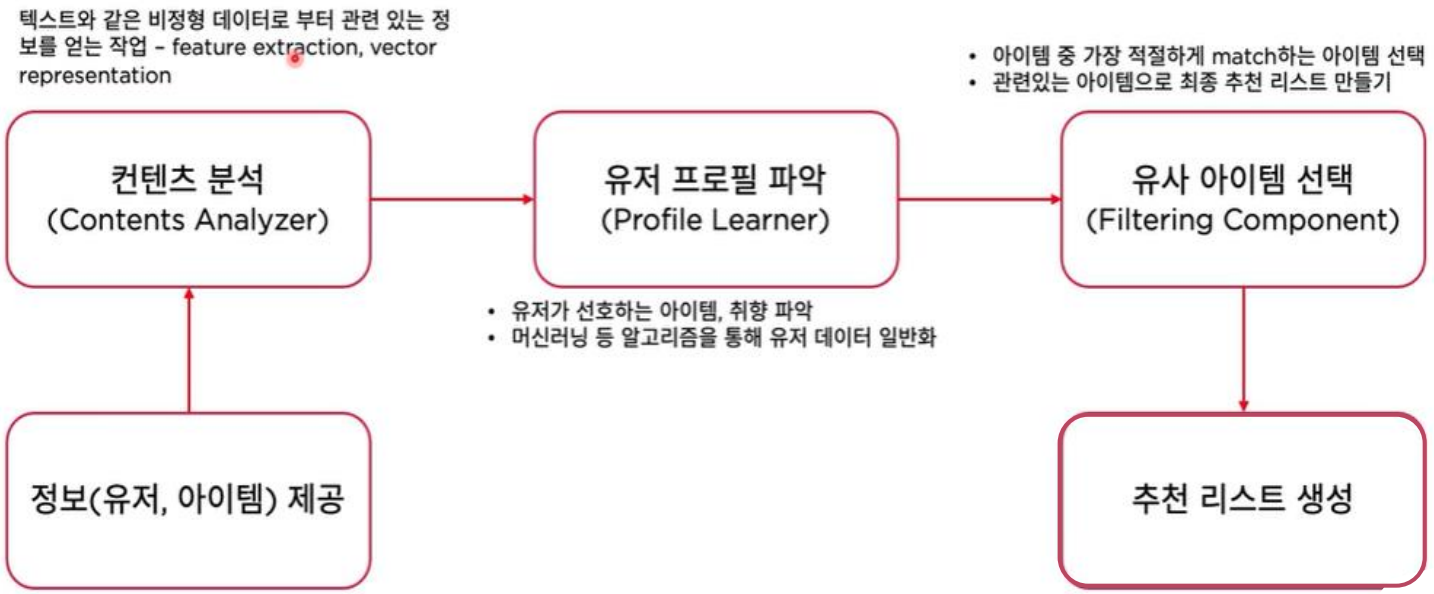
예) 판도라 음악 서비스

[1] 특성추출(features extraction): 판도라는 신곡이 나오면 장르, 비트, 음색 등 400여 항목의 특성을 추출함

[2] profile learner: 유저로부터 'like'를 받은 음악의 특색과 해당 유저의 프로파일 준비

[3] filtering componenet: 음악의 특징과 사용자 프로필을 바탕으로 선호도를 고려해서 유사한 음악을 추천하는 서비스를 제공

https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=datageeks&logNo=221090860432



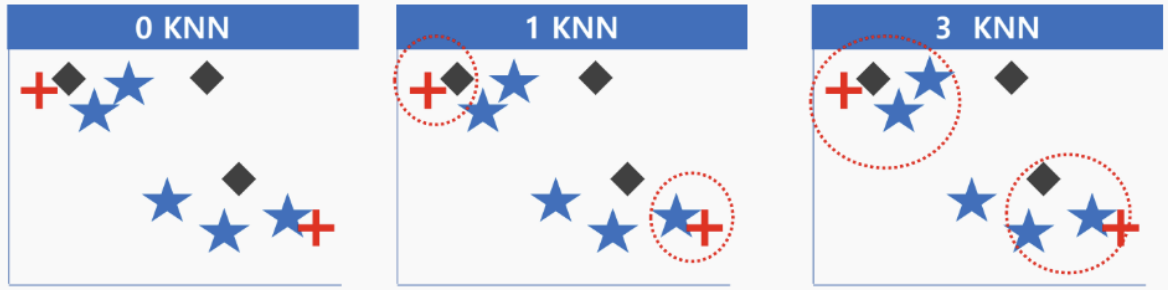
- **컨텐츠 기반 필터링에서는 컨텐츠의 내용을 분석하는 분석 알고리즘을 사용해야한다.**
  (Clustering, Machine Learning, TF-IDF등)

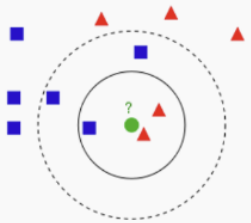
- **추천시스템의 성능을 높일수 있는 적절한 컨텐츠를 사용할것**
  (영화추천을 위해 영화 정보를 활용하고, 뉴스 추천을 위해 뉴스 내용을 활용)

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링) – K근접이웃 알고리즘

1) KNN(K-Nearest Neighbor: 최근접 이웃 알고리즘) 개요

- 데이터를 묶는 모델링-묶은 데이터 그룹을 하나의 데이터로 인식
  - ✓ 매개변수 K값이 커질수록 정확성이 감소하지만 대그룹으로 묶을 수 있으나
  - ✓ 항목간 경계가 불분명해짐





- k=3일때
  - ■ 은 1개 , 확률은 1/3
  - ▲ 은 2개, 확률은 2/3
  - ==> ● 은 ▲ 로
  - 분류됨

- k=5일때
  - ■ 은 3개 , 확률은 3/5
  - ▲ 은 2개, 확률은 2/5
  - ==> ● 은 ■ 로
  - 분류됨

유클리디안 거리를 사용함 – KNN

K값은 홀수로 하는 것이 일반적

. 장점: 수치기반 데이터에서 우수한 성능을 나타내는 알고리즘 / 단순하고 효율적이다.
. 단점: 명목 또는 더미 변수 처리의 어려움. 변수가 많은 데이터 비효율적(정확도 떨어짐)
적절한 k값을 선택하기 어렵다. 레이블과의 관계파악 어려움
변수마다 스케일이 다른경우 동일한 거리 척도 사용이 어렵다

. KNN구현시 반드시 필요 작업
- 정규화 (Z-score, Min-Max Normalization등)
- 데이터간의 거리 측정 효율화(Locality Sensitive Hashing, Network based Indexer등)
- 주어진 공간안에 위치와 거리가 적절하게 분포되어 있어야한다.(아웃라이너 처리)

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링) – 나이브베이즈

1) 개요

- 베이즈 정리에 기반한 통계적 분류기법(확률 분류기)으로 가장 단순한 지도학습
  - 텍스트분류에 많이 사용됨 (예: 리뷰에서 긍부정어 분류)
- Feature 끼리의 독립성 전제조건
  - 스팸메일 분류에서 Label: 스팸메일 여부
  - 스팸메일 분류에서 Feature독립정이어야함: 제목의 일정단어, 비속어, 표현등)

장점	• 간단하고 빠르며 정확한 모델 • 큰 데이터셋에 적합 • 연속형보다 이산형에 적합하며 Multiple class 예측도 가능함
단점	• Feature 독립성이 있어야함.

2) 베이즈 정리

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- B(특징)조건이 주어졌을때 A(레이블)의 확률 구함
- 예: 음식배달 리뷰에서 어떤 단어가 긍정인지 확인하는 베이즈 계산법
  '맛있게 맵다' (맛있다 긍정어확률*맵다 긍정어확률)

$$P(\text{positive} \mid \text{review}) = \frac{P(\text{review} \mid \text{positive}) \cdot P(\text{positive})}{P(\text{review})}$$

3) 베이즈 정리

- 스무딩(smoothing)

  긍정리뷰에 한번도 등장하지 않으면 모두 0이 되어 버림 (오타등)

  - 분자에 1을 더하고 분모에 N을 더해서 부드럽게 만듦

$$P(\text{positive} \mid \text{review}) = \frac{P(\text{review} \mid \text{positive}) \cdot P(\text{positive})}{P(\text{review})}$$

4) 베이즈 분류 실습 (사전, 사후확률)

- 양성판정을 받았을때 진짜 병에 걸렸을 확률 계산
  - 병 걸린 사람이 테스트하면 정확하게 분류할 확률이 99%일때
- 양성이 떴을때 병에 걸렸을 확률 => P(병/양성)
- [사전확률] 병에 걸릴확률(전체 인구중 환자수) => P(병)=0.001
- 병에 거렸을때 양성일 확률 => P(양성 | 병)=0.99

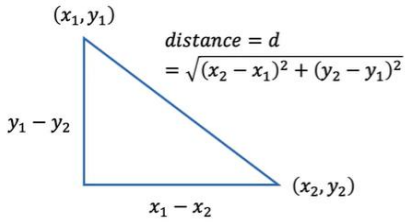
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{P(A)P(B|A) + P(-A)P(B|-A)}$$

$$P(\text{병}|\text{양성}) = \frac{p(\text{양성}|\text{병})P(\text{병})}{P(\text{병})P(\text{양성}|\text{병}) + P(\text{멀쩡})P(\text{양성}|\text{멀쩡})}$$

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링) – 평가 지표

유클리드 거리 – Euclidean Distance

- 두 점 사이의 거리를 계산할 때, 사용하는 evaluation metric
- 여러 차원을 갖는 점과 점 사이의 거리를 계산 할 수 있다
- 거리 기반 유사도 측정 방법



- 두 점 A와 B가 있을 때, $A = (a_1, a_2, \cdots, a_n)$와 B $= (b_1, b_2, \cdots, b_n)$, 두 점 사이의 거리
- $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$

벡터간의 스케일 차이가 크게 나면
코사인유사도,
크지 않다면 유클리드 거리

그외평가지표

맨하탄, 피어슨상관계수, Jaccard Similarity등

코사인 유사도 – Cosine Similarity

- 벡터를 비교할 때, 가장 많이 유용하게 사용되는 평가 지표(evaluation metric)
- 코사인 유사도는 벡터 A와 벡터 B 사이의 각도로 계산된다.
- 코사인 유사도는 -1 과 1사이의 값을 가지며, -1은 완전히 반대, 0은 서로 독립, 1은 완전히 같은 경우
- 각도 기반 유사도 측정 방법; 두 벡터간 유사한 정도를 코사인 값으로 표현

$$a \cdot b = \|a\|\|b\| \cos\theta$$

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

코사인 유사도 계산

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Like/ DisLike
Movie 2	0	1	1	0	0	0	0	Like
Movie 3	1	1	1	1	0	0	0	Dislike
				.....				
New 1	1	1	0	0	0	1	1	?
New 2	0	0	0	0	0	0	1	?

- New 1과 가장 비슷한 영화 찾기(Movie2, Movie3과의 코사인 유사도 계산)
- $cos(Movie_2, New_1) = \frac{0\cdot1+1\cdot1+1\cdot0+0\cdot0+0\cdot0+0\cdot1+0\cdot1}{\sqrt{0^2+1^2+1^2+0^2+0^2+0^2+0^2}\sqrt{1^2+1^2+0^2+0^2+0^2+1^2+1^2}} = \frac{1}{\sqrt{2}\sqrt{4}} \cong 0.354$
- $cos(Movie_3, New_1) = \frac{1\cdot1+1\cdot1+1\cdot0+1\cdot0+0\cdot0+0\cdot1+0\cdot1}{\sqrt{1^2+1^2+1^2+1^2+0^2+0^2+0^2}\sqrt{1^2+1^2+0^2+0^2+0^2+1^2+1^2}} = \frac{2}{\sqrt{4}\sqrt{4}} = 0.5$

추천시스템(recommender system) 종류 – (컨텐츠 기반 필터링)

- TF– IDF
  - TF: 단어(Word) w가 문서(Document) d에 등장한 빈도수
  - DF: 단어(Word) w가 등장한 문서(Document) d의 수
  - N: 전체 문서의 수
- $\mathrm{TF}(w,d) = \frac{\text{문서 내 단어 } w\text{의 수}}{\text{문서 내 모든 단어의 수}}$
- $\mathrm{IDF}(w) = \log\left(\frac{\text{전체 문서 수}_{(N)}}{\text{단어 } w\text{가 포함된 문서 수}_{(DF(w))}}\right)$
  - *If* "단어 w가 포함된 문서 수 = 전체 문서 수", *then TF-IDF(w,d) = 0*
  - *단어 w의 정보력은 없다*

사전학습_데이터분석 모델 제작.pptx

→ 실습: 3. 추천시스템-TF-IDF.ipynb

추천시스템(recommender system) 종류 – 이웃기반(메모리기반) 협업필터링

2) 협업 필터링(collaborative filtering)

- 기본가정은 'Likely-mind'로 나와 비슷한 성향을 가진 다른 사용자들이 좋아하는 것이면 나도 좋아할 가능성이 높다

분류	내용
모델 기반	▪ 나이브 베이지언 이나 뉴럴네트워크등 다양한 머신러닝 기법 사용
메모리 기반	▪ 유저와 아이템에 대한 레이팅 모두 메모리 위에 두고 계산하여 메모리 기반이라 함. ▪ 장점: 간단하며 추천의 성능이 좋음 우연에 관한 추천이 가능 ▪ 단점: 적용범위가 낮은편임

2) 협업 필터링(collaborative filtering)

2-1. 메모리 기반(최근접 이웃기반:nearest neighbor based collaborative filtering))

- 사용자가 아이템에 매긴 평점, 상품 구매 이력 등의 사용자 행동양식(user behavior)를 기반으로 추천해주는 방식임 (프로필 없이 사용자의 과거 행동 데이터만 가지고 추천)
- 사용자-아이템 행렬에서 사용자가 아직 평가하지 않은 아이템을 예측하는 것이 목표임

사용자	상품	평점
U1	A	1
U2	B	3
U3	C	2
U4	A	1

사용자	상품A	상품 B	상품C
U1	1	3	
U2	2		1

▪ 상품추천시스템 행렬

많은 사람의 의견으로 더 나은 추천을 한다
개인보다 단체 또는 그룹의 선택과 취향에 의존한다.
다수의 의견으로 더 나은 선택을 한다.

참고: 모델기반협업필터링 (딥러닝협업필터링)
하이브리드 협업필터링(협업+컨텐츠)

2) 협업 필터링(collaborative filtering)

2-1. 메모리 기반(최근접 이웃기반:nearest neighbor based collaborative filtering))

- 사용자 기반: 비슷한 고객들이 선택한 아이템을 소비했다.
- 아이템 기반: 선택한 아이템을 소비한 고객들은 다음과 같은 상품도 구매했다.

[사용자기반 협업필터링: 사용자간의 유사도를 구한뒤 작업함]

	캐치 미 이프 유 캔	쥬라기 공원	라이언 일병 구하기	마이너리티 리포트	미션 임파서블	미이라
갑	3	3	4	3.5	1	2
을	3.5	2		5	5	
병	4.5		2	?	2.5	3
정	5	1	2.5	4	1.5	
무		2	5	1	3	

- User '병'
  - User '갑' = 0.3, User '을' = 0.6, User '정' = 0.45, User '무' = 0.15 유사하다고 가정하자.
- User "병"의 마이너리티 리포트 평점은?
  - 0.6으로 가장 유사한 User '을'의 평점 = 5로 예측

[item 협업필터링: item간의 유사도를 구한뒤 작업함]

	캐치 미 이프 유 캔	쥬라기 공원	라이언 일병 구하기	마이너리티 리포트	미션 임파서블	미이라
갑	3	3	4	3.5	1	2
을	3.5	2		5	5	
병	4.5		2		2.5	3
정	5	1	2.5	4	1.5	
무		2	5	1	3	

- 마이너리티 리포트
  - 라이언 일병 구하기 = 0.7, 미션 임파서블 = 0.85 유사하다고 가정하자
- User '병'의 마이너리티 리포트 평점은?
  - $\frac{(0.7 \times 2) + (0.85 \times 2.5)}{0.7+0.85} = 2.27$

- **정확도**
  - user based(user < item수) / - item based(item < user수)
- **모델 견고함**
  - 유저와 아이템이 얼마나 자주 변하는지에 따라 선택함
- **새로운 추천가능**
  - item based는 과거 item에 의존하기 때문에 새로운 item을 추천하기 어렵다.
  - user based는 여러 유저의 데이터를 이용하기 때문에 새로운 추천이 가능하다.

2-1. 메모리 기반(최근접 이웃기반:nearest neighbor based collaborative filtering))

문제점	내용
Cold Start	▪ 신규사용자의 경우, 관찰된 행동 데이터가 없거나 적음 ▪ 추천의 정확도가 급격히 떨어지는 Cold start 문제 발생하여 사용자에 대한 추론을 이끌어 낼수 없음
계산효율 저하	▪ 추천의 효율성이 떨어짐. ▪ 사용자가 많아질수록 계산에 걸리는 시간이 매우 많아짐
Long Tail	▪ 파레토의 법칙(전체 결과의 80%가 전체 원인의 20%에서 일어나는 현상)에서 80%부분에 해당하는 것처럼 ▪ 사용자들이 관심을 많이 보이는 소수의 아이템이 전체 추천 아이템으로 보이는 비율이 높은 '비대칭적 쏠림 현상 ' 발생

→ 딥러닝 기반 추천 알고리즘

→ 모델기반 협업필터링

https://cmdlinetips.com/2018/03/sparse-matrices-in-python-with-scipy/

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.unstack.html

https://pandas.pydata.org/pandas-docs/stable/user_guide/reshaping.html

→ 실습: 4. 이웃기반(메모리기반) 협업필터링.ipynb

추천시스템 고려요소와 장애요소

1) 추천시스템 구축 개발시 고려요소

요소	내용
시스템 업데이트	▪ 추천 아이템의 리스트 업데이트 주기
알고리즘 성능	▪ 추천모델링의 계산량과 연산속도 ▪ Collaborative Filtering / Deep Learing /Association Rule ▪ 빅데이터를 처리할수 있는 환경구축

2) 장애요소

요소	내용
Sparsity Problem	▪ 추천할 아이템과 고객은 계속해서 늘어남 ▪ 고객이 실제로 소비하게 되는 콘텐츠나 아이템의 비율은 현저하게 감소하게 됨
Information Utilization Problem	▪ 데이터, 정보들을 올바르게 활용하기 위한 고민에서 나온 문제점임 ▪ Implicit Score(암묵점수, 장바구니에 넣었다가 구매함)가 많음 ▪ 로그데이터 속에 숨어있는 정보를 고민해야함. ▪ 평점과 같은 명시적 점수처럼 데이터를 이용(UtilizatioN)하는 과이 필요함