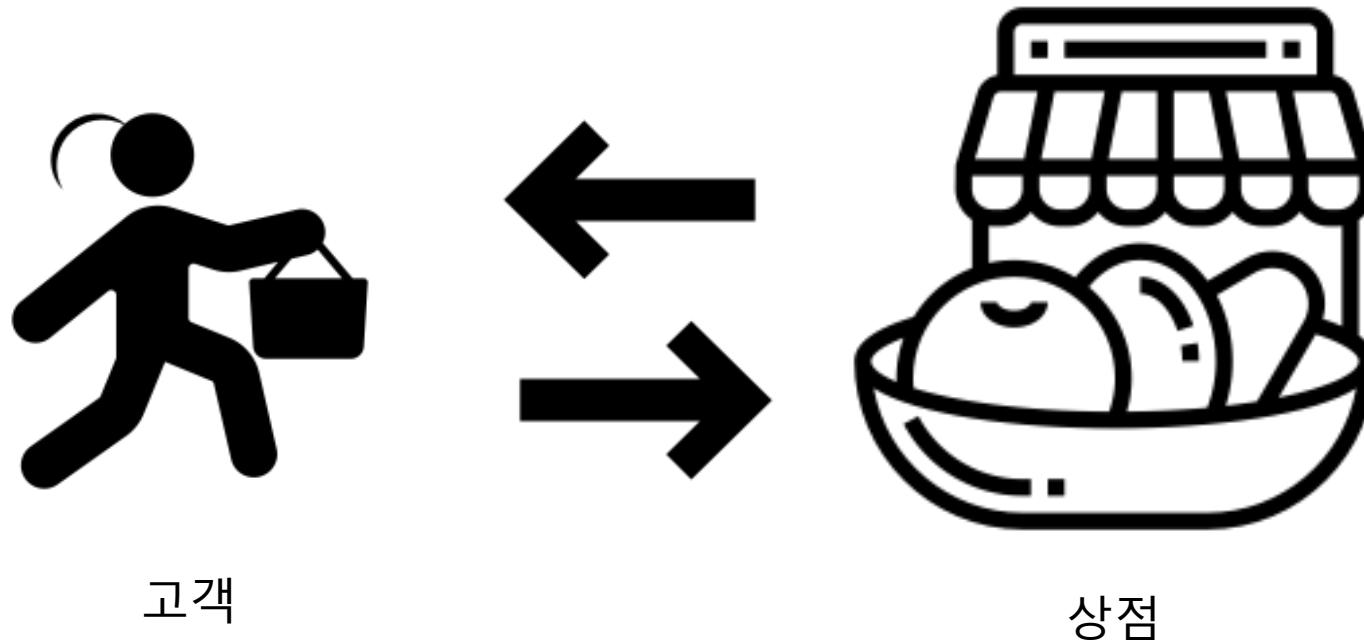


추천 시스템

<Recommender System>

이 수업의 주제 : 추천



핵심 이슈 : 고객에게 어떠한 제품을 권할 것인가?

이 수업의 구성

1. 고객이 첫 방문을 했을 때

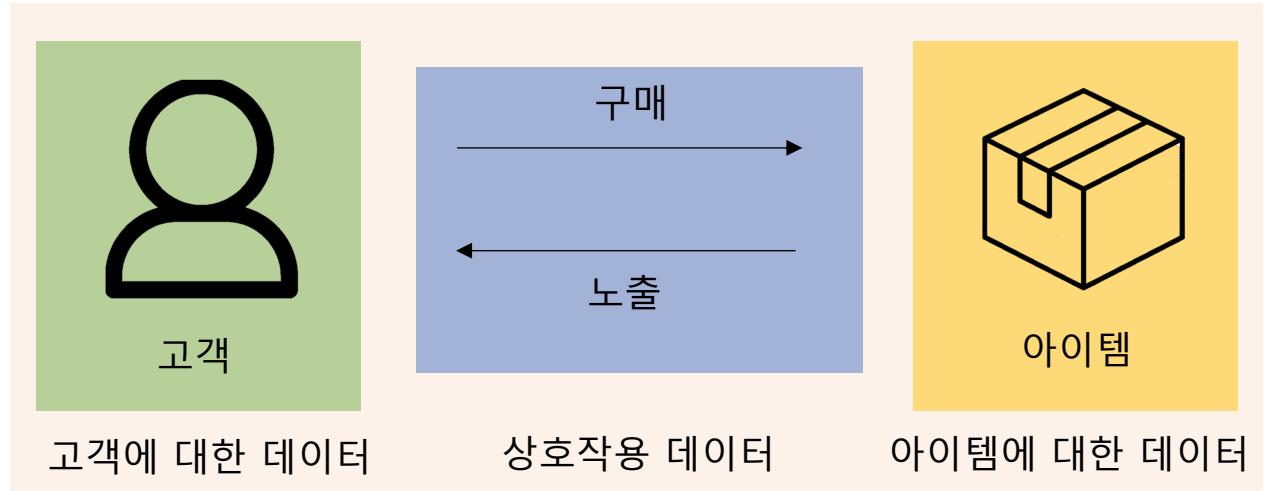
: 고객에 대한 정보가 없을 때

2. 고객이 재차 방문을 했을 때

: 고객의 이전 구매 정보가 있을 때

3. 고객이 단골이 되었을 때

: 고객 신원 정보가 있을 때



이 수업의 주요 토픽

1. 고객이 첫 방문을 했을 때

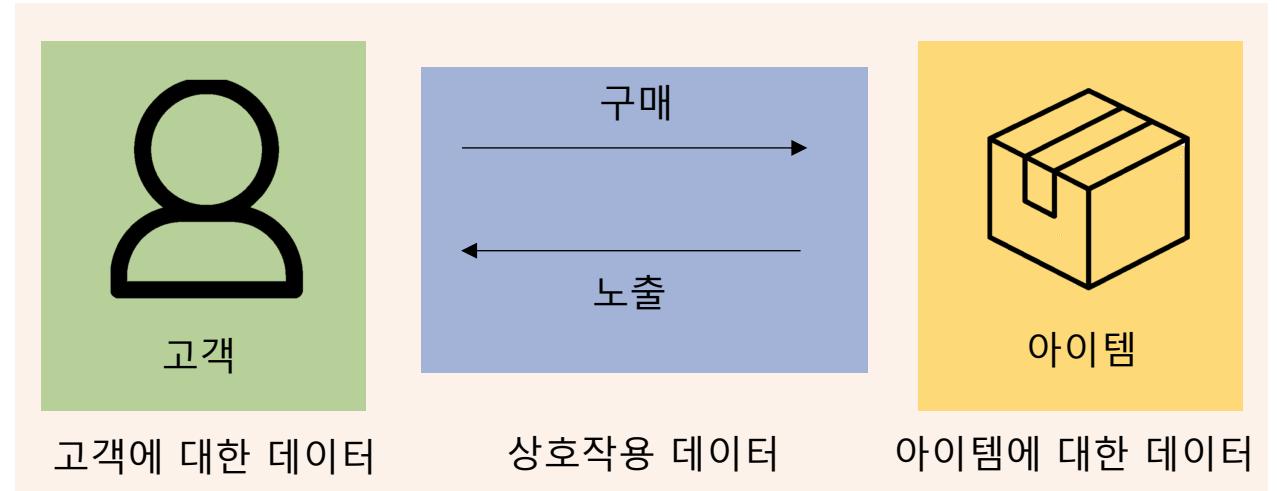
- 비개인화 추천
- 연관 분석

2. 고객이 재차 방문을 했을 때

- Collaborative Filtering

3. 고객이 단골이 되었을 때

- Factorization Machine



수업 목표

현업에서 적용 가능한 수준으로

1. 추천시스템 알고리즘의 이해도 향상
2. 추천시스템 알고리즘의 코드 작성 능력 배양

수업 환경

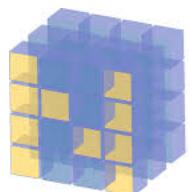
수업 코드 환경 : Jupyter Notebook



다루는 언어 : Python 3.6



다루는 패키지



NumPy

Pandas



PAI
PUBLIC AI

수업 코드 리파짓토리

<https://github.com/public-ai/recommender-ewha20201Q>

The screenshot shows the GitHub repository page for 'public-ai / recommender-ewha20201Q'. The repository has 0 stars, 0 forks, and 0 issues. It contains 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit was made 2 days ago by sangjaekang. The repository includes files like ppt, reference, script, .gitignore, and README.md.

public-ai / recommender-ewha20201Q

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

이화여대에서 진행한 2020년 봄학기 추천 시스템에 대한 교육 자료 Edit

Manage topics

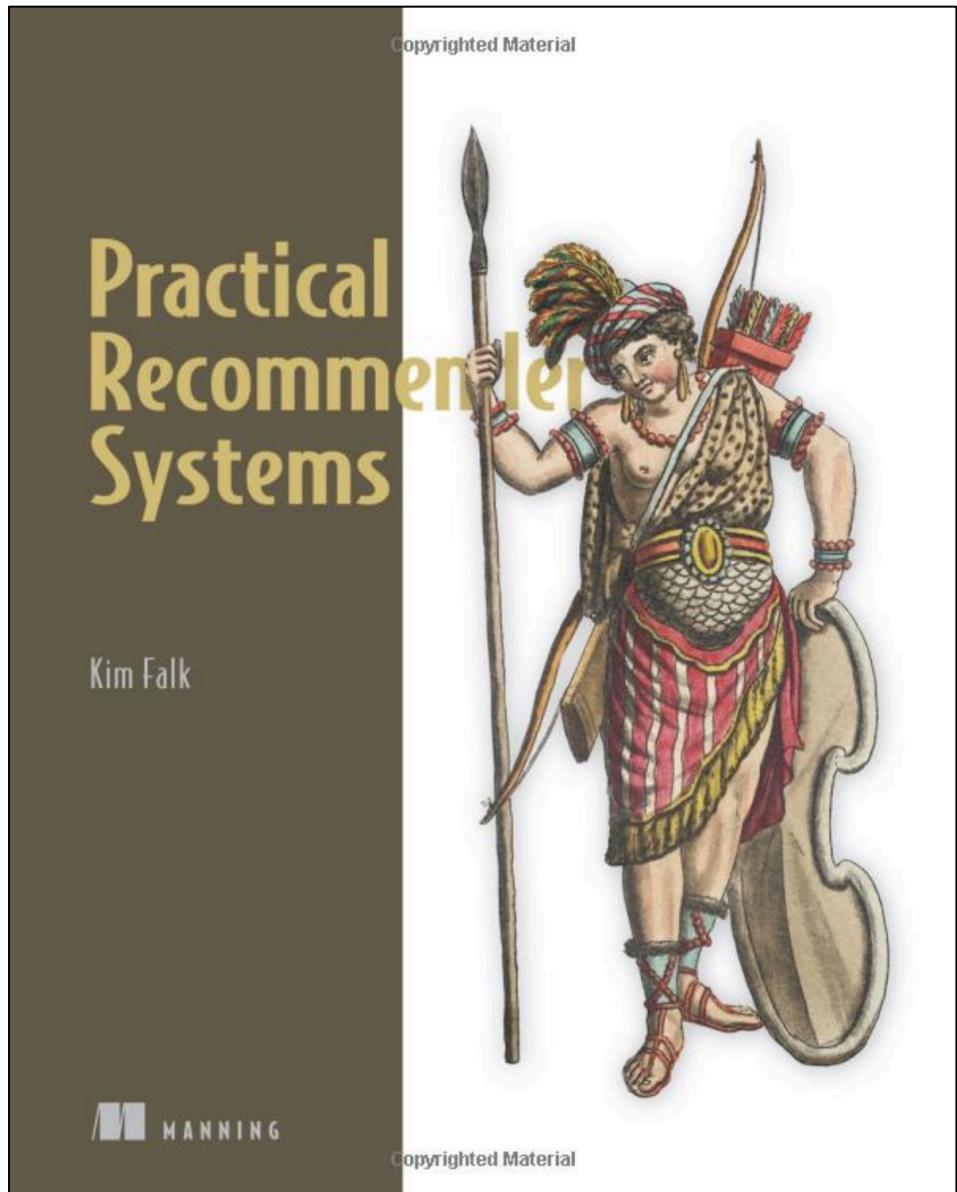
1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

sangjaekang initial commit	Latest commit 4619ae1 2 days ago
ppt	initial commit
reference	initial commit
script	initial commit
.gitignore	initial commit
README.md	initial commit

Help people interested in this repository understand your project by adding a README. Add a README

이 수업을 설계하면서 다루었던 참조자료들



이 책에서 다룬 내용들

- 연관분석
- Collaborative Filtering
- Bayesian Personalized Ranking

이 수업을 설계하면서 다루었던 참조자료들

Deep Learning based Recommender System: A Survey and New Perspectives

SHUAI ZHANG, University of New South Wales

LINA YAO, University of New South Wales

AIXIN SUN, Nanyang Technological University

YI TAY, Nanyang Technological University

Deep Learning 관련된 추천시스템 모델이 무엇이 있는지를 요약한 모델

With the ever-growing volume of online information, recommender systems have been an effective strategy to overcome such information overload. The utility of recommender systems cannot be overstated, given its widespread adoption in many web applications, along with its potential impact to ameliorate many problems related to over-choice. In recent years, deep learning has garnered considerable interest in many research fields such as computer vision and natural language processing, owing not only to stellar performance but also the attractive property of learning feature representations from scratch. The influence of deep learning is also pervasive, recently demonstrating its effectiveness when applied to information retrieval and recommender systems research. Evidently, the field of deep learning in recommender system is flourishing. This article aims to provide a comprehensive review of recent research efforts on deep learning based recommender systems. More concretely, we provide and devise a taxonomy of deep learning based recommendation models, along with providing a comprehensive summary of the state-of-the-art. Finally, we expand on current trends and provide new perspectives pertaining to this new exciting development of the field.

CCS Concepts: •Information systems → Recommender systems;

Additional Key Words and Phrases: Recommender System; Deep Learning; Survey

ACM Reference format:

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 1, 1, Article 1 (July 2018), 35 pages.

DOI: 0000001.0000001

이 수업을 설계하면서 다루었던 참조자료들

Microsoft에서 제공하는 추천시스템에 관련된 Sample Codes Repository

The screenshot shows the GitHub repository page for `microsoft/recommenders`. The page includes the repository name, a star count of 7k, a fork count of 1k, and metrics for used by (1), watched (220), stars, forks, issues (88), pull requests (5), projects (0), wiki, security, and insights. Below the header, there's a section for 'Best Practices on Recommendation Systems' with a link to <https://microsoft-recommenders.readth...>. The repository has tags for machine-learning, recommender, ranking, deep-learning, python, jupyter-notebook, recommendation-algorithm, rating, operationalization, kubernetes, azure, microsoft, recommendation-system, recommendation-engine, recommendation, data-science, tutorial, and artificial-intelligence. At the bottom, it shows 5,431 commits, 13 branches, 0 packages, 5 releases, 43 contributors, and an MIT license.

<https://github.com/microsoft/recommenders>

이 수업을 설계하면서 다루었던 참조자료들

Google에서 제공하는 추천시스템

Recommendations AI^{베타}

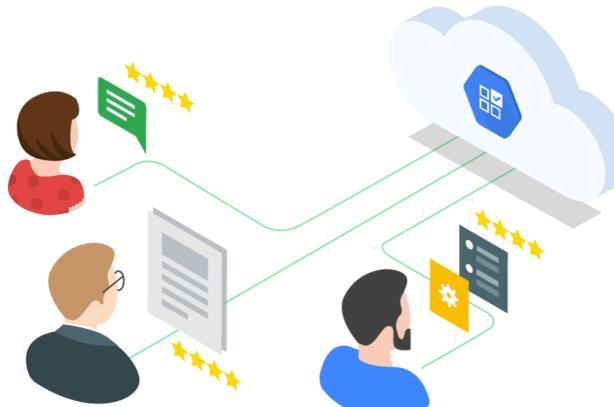
규모에 따라 고도로 맞춤화된 제품 추천

문서 보기

고객의 취향에 따른 추천

고객을 잘 이해하고 있음을 보여주면 고객의 신뢰와 충성도를 높일 수 있습니다. Google에서는 수년 동안 Google Ads, Google Search, YouTube와 같은 대표적인 제품 전반에서 추천 콘텐츠를 제공해 왔습니다.

Recommendations AI는 그러한 경험을 바탕으로 모든 터치포인트 전반에서 각 고객의 취향이나 선호도에 맞는 맞춤 추천을 제공합니다. 제한된 사용자에게만 제공되는 이 베타 서비스에 대해서는 Google 계정 관리자에게 문의하세요.



<https://cloud.google.com/recommendations?hl=ko>

이 수업을 설계하면서 다루었던 참조자료들

AWS에서 제공하는 추천시스템

빌트인 레시피



개인화 & 추천을 위한 **최신 기반 알고리즘(Predefined Recipes)**을 다양하게 제공합니다.

아울러, 높은 성능의 모델 생성을 위한 **파라미터 최적화(Hyperparameter Optimization)**,

알고리즘 자동 선택(AutoML)* 기능도 제공합니다.

DeepFM* (Deep Factorization Machines)	Recommended for fast Training and Inference with good general performance
FFNN* (Feed-Forward Neural Network)	A general purpose feed-forward neural network
HRNN* (Hierarchical Recurrent Neural Network)	Recommended when user behavior is changing with time (the evolving intent problem) - HRNN-coldstart - HRNN-metadata
PersonalizeReranking	Use for personalized reranking of search results or curated lists
Popularity-baseline	Use as a baseline to compare other personalization recipes
SIMS	Item-to-Item similarities. Use for improving item discoverability and in detail pages for the fast performance

비개인화 추천

비개인화 추천시스템

고객에 대한 정보가 **전혀 없을 때**, 어떤 것을 추천하는 것이 가장 좋을까

1. 최신(신규) 아이템 추천하기
2. 최고 평점 아이템 추천하기
3. 최근 인기 아이템 추천하기

비개인화된 요약 통계

1. 최신(신규) 아이템 추천하기

현재상영영화

상영영화 영화제

리스트뷰 이미지뷰

12 첫키스만 50번째

네이즌 ★★★★☆ 5.38 참여 8명

개요 멜로/로맨스, 코미디 | 114분 | 2020.03.26 개봉

감독 후쿠다 유이치

출연 야마다 타카유키, 나가사와 마사미

예매하기 포토보기 예고편

15 온다

네이즌 ★★★★★ 7.26 참여 87명 | 기자·평론가 ★★★★☆ 5.40 참여 5명

개요 공포 | 134분 | 2020.03.26 개봉

감독 나카시마 테츠야

출연 오카다 준이치, 쿠로키 하루, 고마츠 나나, 마츠 다카코, 츠마부키 사토시

예매하기 포토보기 예고편 메이킹

전체 바이올렛 에버가든 -영원과 자동 수기 인형-

네이즌 ★★★★★ 9.33 참여 96명 | 기자·평론가 ★★★★☆ 6.00 참여 1명

개요 애니메이션 | 91분 | 2020.03.26 개봉

감독 후지타 하루카

예매하기 포토보기 예고편

전체 모리의 정원

네이즌 ★★★★★ 7.73 참여 11명 | 기자·평론가 ★★★★☆ 6.33 참여 6명

개요 드라마 | 99분 | 2020.03.26 개봉

감독 오기타 슈이치

출연 암자카 츠토무, 키키 키린

예매하기 포토보기 예고편

비개인화된 요약 통계

2. 최고 평점 아이템 추천하기

랭킹				
영화 - 영화인				
영화 랭킹				
조회순	평점순 (현재상영영화)	평점순 (모든영화) ▾	2020.03.29	◀
순위	영화명	평점	변동폭	
1	그린 북	★★★★★ 9.60	평점주기	- 0
2	가버나움	★★★★★ 9.58	평점주기	- 0
3	베일리 어게인	★★★★★ 9.52	평점주기	- 0
4	주전장	★★★★★ 9.51	평점주기	- 0
5	포드 V 페라리	★★★★★ 9.50	평점주기	- 0
6	아일라	★★★★★ 9.49	평점주기	- 0
7	원더	★★★★★ 9.49	평점주기	- 0
8	당갈	★★★★★ 9.47	평점주기	- 0
9	쇼생크 탈출	★★★★★ 9.44	평점주기	- 0
10	터미네이터 2:오리지널	★★★★★ 9.44	평점주기	- 0

비개인화된 요약 통계

3. 최근 인기 아이템 추천하기

현재상영영화

상영영화 영화제 ▾ 예매순 ▾ 개봉순 ▾ 평점순 ▾ 좋아요순

전체 올지마 톤즈 2 : 슈크란 바바

네이션 ★★★★★ 9.53 참여 154명 | 기자·평론가 ★★★★★ 4.00 참여 1명
개요 다큐멘터리 | 82분 | 2020.01.09 개봉
감독 강성욱
출연 이금희, 이태석

예매하기 다운로드 포토보기 예고편 메이킹

12 이장

네이션 ★★★★★ 9.52 참여 44명 | 기자·평론가 ★★★★★ 6.33 참여 6명
개요 드라마 | 94분 | 2020.03.25 개봉
감독 정승오
출연 장리우, 이선희, 공민정, 윤금선아, 곽민규

예매하기 포토보기 예고편 메이킹

12 포드 V 페라리

네이션 ★★★★★ 9.50 참여 7,949명 | 기자·평론가 ★★★★★ 7.63 참여 8명
개요 액션, 드라마 | 152분 | 2019.12.04 개봉
감독 제임스 맨글드
출연 맷 데이먼, 크리스찬 베일

다운로드 포토보기 예고편 메이킹

전체 산티아고의 흰 지팡이

네이션 ★★★★★ 9.43 참여 21명
개요 다큐멘터리 | 98분 | 2020.03.19 개봉
감독 이종은
출연 박재한, 김다희

예매하기 포토보기 예고편 메이킹

연관 분석
<Association Analysis>

연관 분석

만약 X를 선호(구매)했다면,
Y도 선호(구매)할 것이다

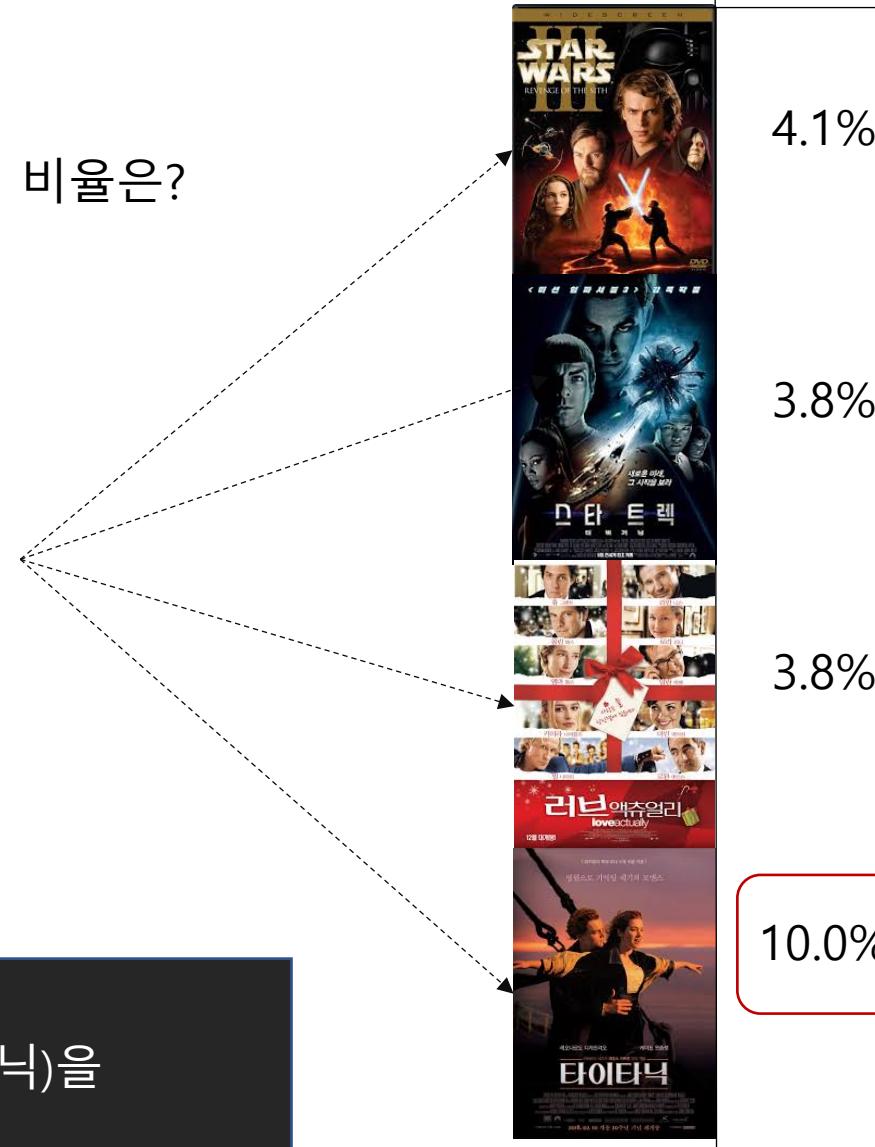
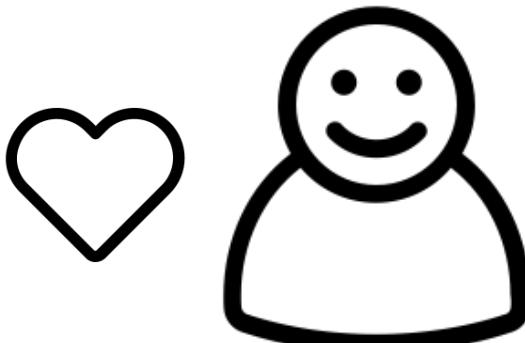
의 연관 관계를 도출하는 것

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

지지도

(1) 지지도 : $Support(X) = \frac{freq(X)}{N}$

: 전체 고객 중에서 영화 X를 선호하는 사람의 비율은?

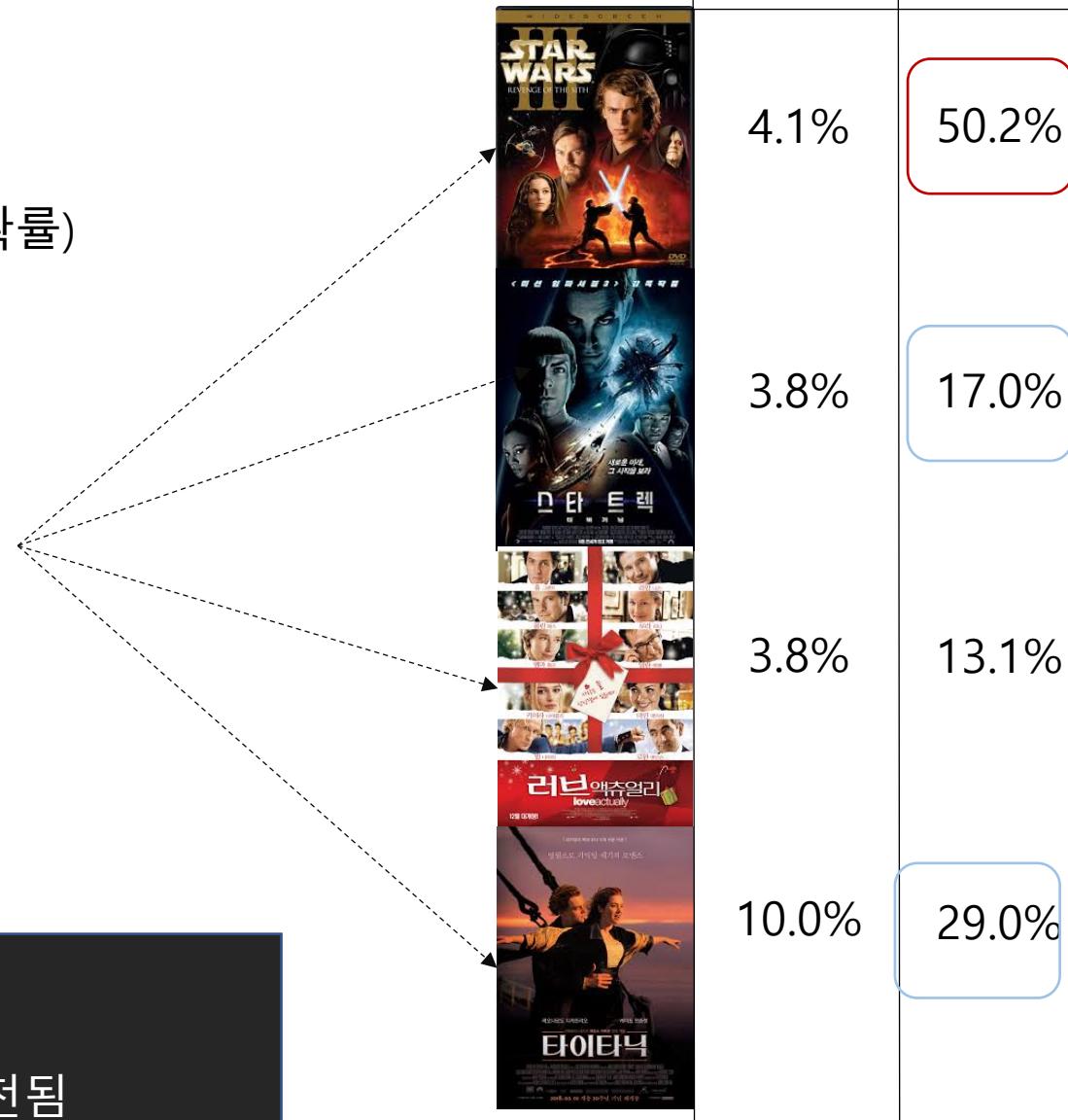
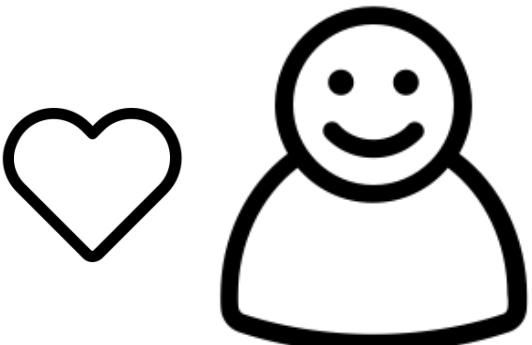


지지도를 기준으로 추천할 경우,
어떤 영화를 선호하던 항상 같은 영화(타이타닉)을
추천하는 문제가 발생

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

(2) 신뢰도 : $confidence(X \rightarrow Y) = \frac{freq(X,Y)}{freq(X)}$

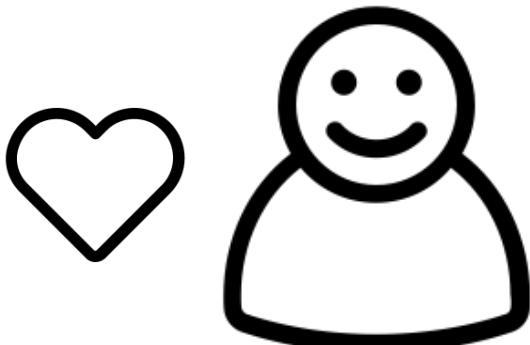
: 영화 X를 선호한 사람 중에서
영화 Y를 선호하는 사람의 비율은?(조건부 확률)



신뢰도를 기준으로 추천할 경우,
대다수 사람들이 선호하는 영화가
소수의 사람들이 선호하는 영화보다 우선 추천됨

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

(3) 리프트 = $\text{lift}(X \rightarrow Y) = \frac{\text{confidence}(X \rightarrow Y)}{\text{support}(Y)}$
: 지지도 대비 신뢰도가 얼마나 올라갔는가?



	지지도	신뢰도	리프트
A movie poster for Star Wars: Episode III - Revenge of the Sith. It features Anakin Skywalker, Obi-Wan Kenobi, and Padmé Amidala.	4.1%	50.2%	12.2
A movie poster for Star Trek. It features Captain Kirk, Mr. Spock, and other crew members.	3.8%	17.0%	4.4
A movie poster for Love Actually. It features a collage of many different actors' faces.	3.8%	13.1%	3.5
A movie poster for Titanic. It features Jack and Rose looking out over the ocean.	10.0%	29.0%	2.9

리프트를 기준으로 추천할 경우,
소수의 사람들이 선호하는 영화이더라도
강한 연관관계가 있으면 영화가 우선 추천됨

연관 분석의 3가지 지표

(1) 지지도 : $Support(X) = \frac{freq(X)}{N}$

(2) 신뢰도 : $confidence(X \rightarrow Y) = \frac{freq(X,Y)}{freq(X)}$

(3) 리프트 : $lift(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{support(Y)} = \frac{N * freq(X,Y)}{freq(X)freq(Y)}$

연관분석을 위해 알아야 하는 값 :

$freq(X), freq(Y), freq(X, Y)$

- $freq(X)$: 아이템(X)을 선호(구매)한 경우의 수
- $freq(Y)$: 아이템(Y)을 선호(구매)한 경우의 수
- $freq(X, Y)$: 아이템(X, Y)을 동시에 선호(구매)한 경우의 수

빈발집합 탐색 알고리즘

연관분석의 단위 : 장바구니

연관분석은 **장바구니**,
즉 고객이 동시에 구매하거나 선호하는 제품군에 대한 정보가 필요

Case) 제품이 a, b, c, d, e 가 있는 경우

	장바구니 (고객이 동시에 구매한 제품 목록)
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}

연관분석의 어려움, 너무 많은 연관 관계

연관분석을 진행하기 위해서는 각 아이템의 동시에 등장한 횟수를 계속 세어야 함
이는 엄청나게 많은 경우의 수가 존재

장바구니 (고객이 함께 선호 혹은 구매한 제품군)		{x} -> {y} : 만약 x를 구매하였으면, y를 구매할 것이다				
0	{b, c}					
1	{b, d, a}	{a} -> {b}	{c} -> {a}	{e} -> {a}	{a,b,c} -> {d}	{b, c} -> {d}
2	{c, e}	{a} -> {c}	{c} -> {b}	{e} -> {b}	{a,b,c} -> {e}	⋮
3	{c, d, e}	{a} -> {d}	{c} -> {d}	{e} -> {c}	{a,b,d} -> {c}	⋮
4	{c, d}	{a} -> {e}	{c} -> {e}	{e} -> {d}	{a,b,d} -> {e}	⋮
5	{b, d}					
6	{b, c, e}	{b} -> {a}	{d} -> {a}	{a,b} -> {c}	{a,c} -> {b}	
7	{d, e}	{b} -> {c}	{d} -> {b}	{a,b} -> {d}	{a,c} -> {d}	
8	{b, c, d}	{b} -> {d}	{d} -> {c}	{a,b} -> {e}	{a,c} -> {e}	
9	{c, e}	{b} -> {e}	{d} -> {e}	{b,c} -> {a}	{b,c} -> {e}	
10	{b, e}					
11	{b, c, d, e}					

: 너무 많은 연관 관계가 존재

빈발집합 찾기 알고리즘 : Apriori

모든 연관 관계를 파악할 수 없기 때문에, “**자주 등장한**” 연관 관계만을 파악하자 -> Apriori의 아이디어

장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}

The diagram illustrates the Apriori algorithm's process of generating candidate itemsets. It shows items a, b, c, d, and e as individual circles. Itemsets are represented as overlapping circles. For example, itemset {b,c} is shown as two overlapping circles, and itemset {a,b,c,d,e} is shown as all five items overlapping. This visual representation helps in understanding how the algorithm checks if a candidate itemset is a subset of any existing itemset in the database.

빈발집합 찾기 알고리즘 : Apriori

1. 제품별로 각각 장바구니에 몇번씩 등장했는지를 셈

장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}

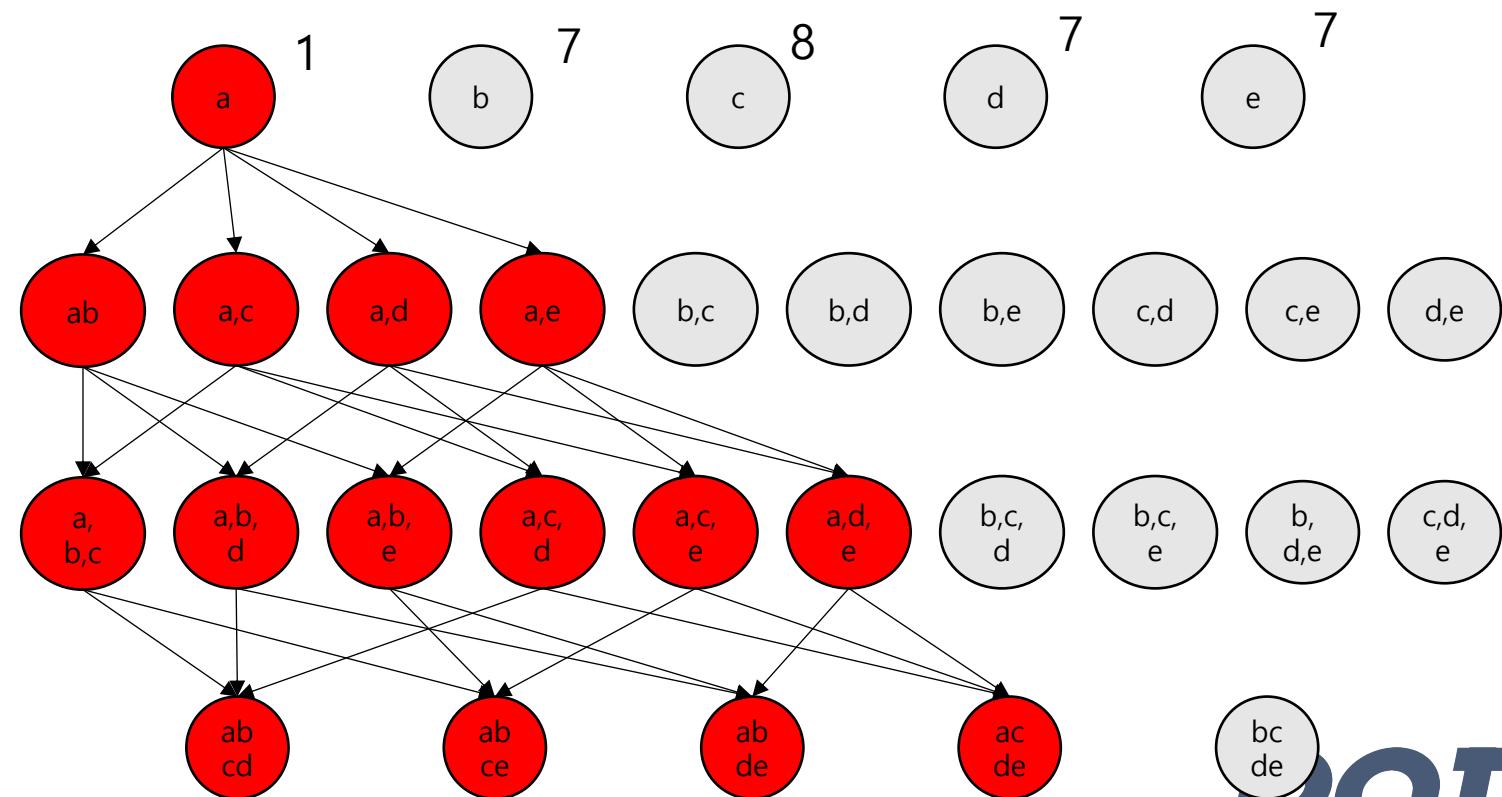
The diagram illustrates the frequency of each item (a, b, c, d, e) across 12 customer baskets. The counts are: a (1), b (7), c (8), d (7), and e (7). The counts are displayed above circles containing the item names.

Item	Count
a	1
b	7
c	8
d	7
e	7

빈발집합 찾기 알고리즘 : Apriori

2. 너무 적게 등장한 연관관계의 경우, 해당 아이템 집합을 포함한 모든 집합은 카운트하지 않음

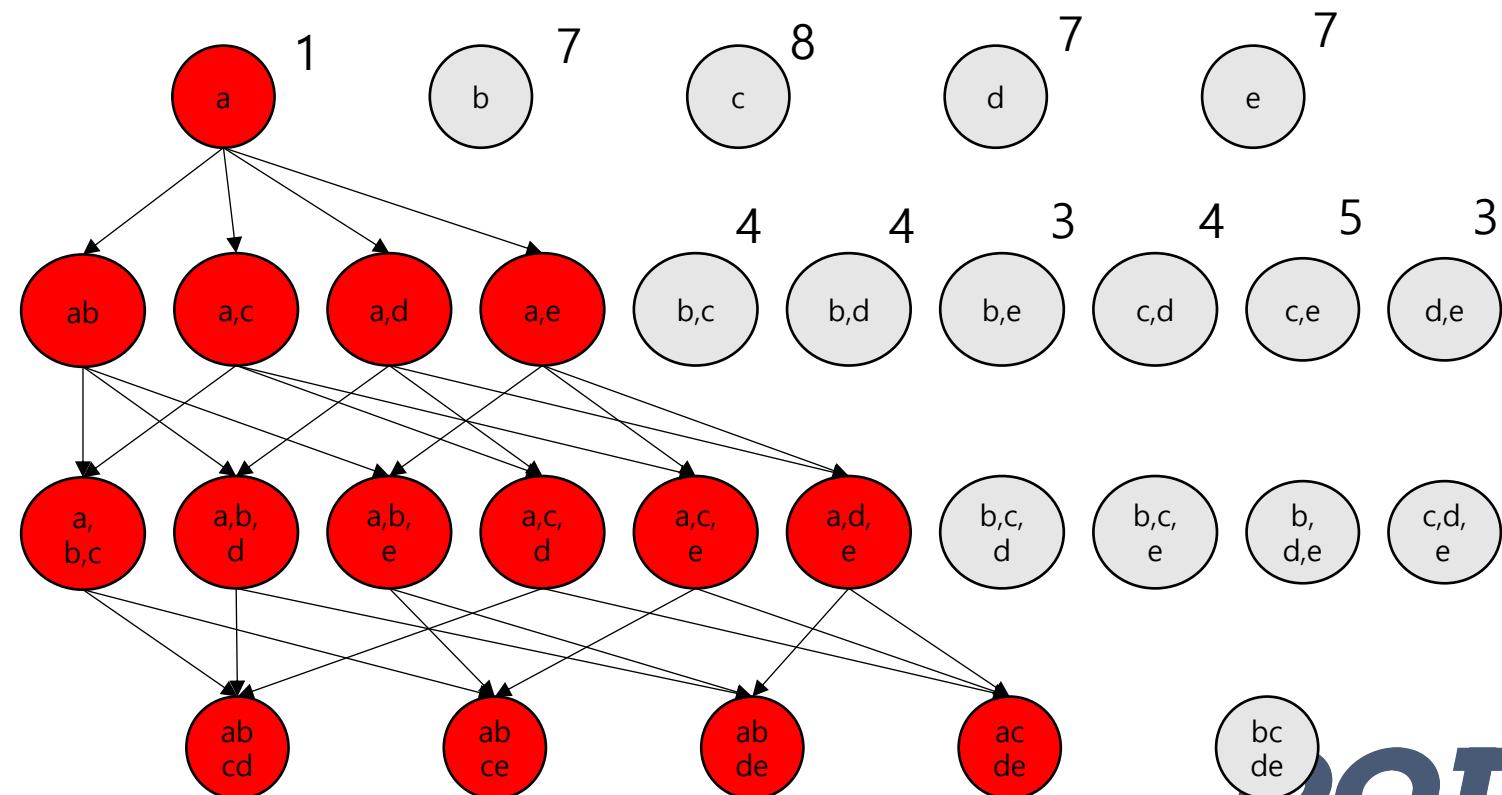
장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}



빈발집합 찾기 알고리즘 : Apriori

3. 생략되고 남은 것들에 한에 카운트를 함

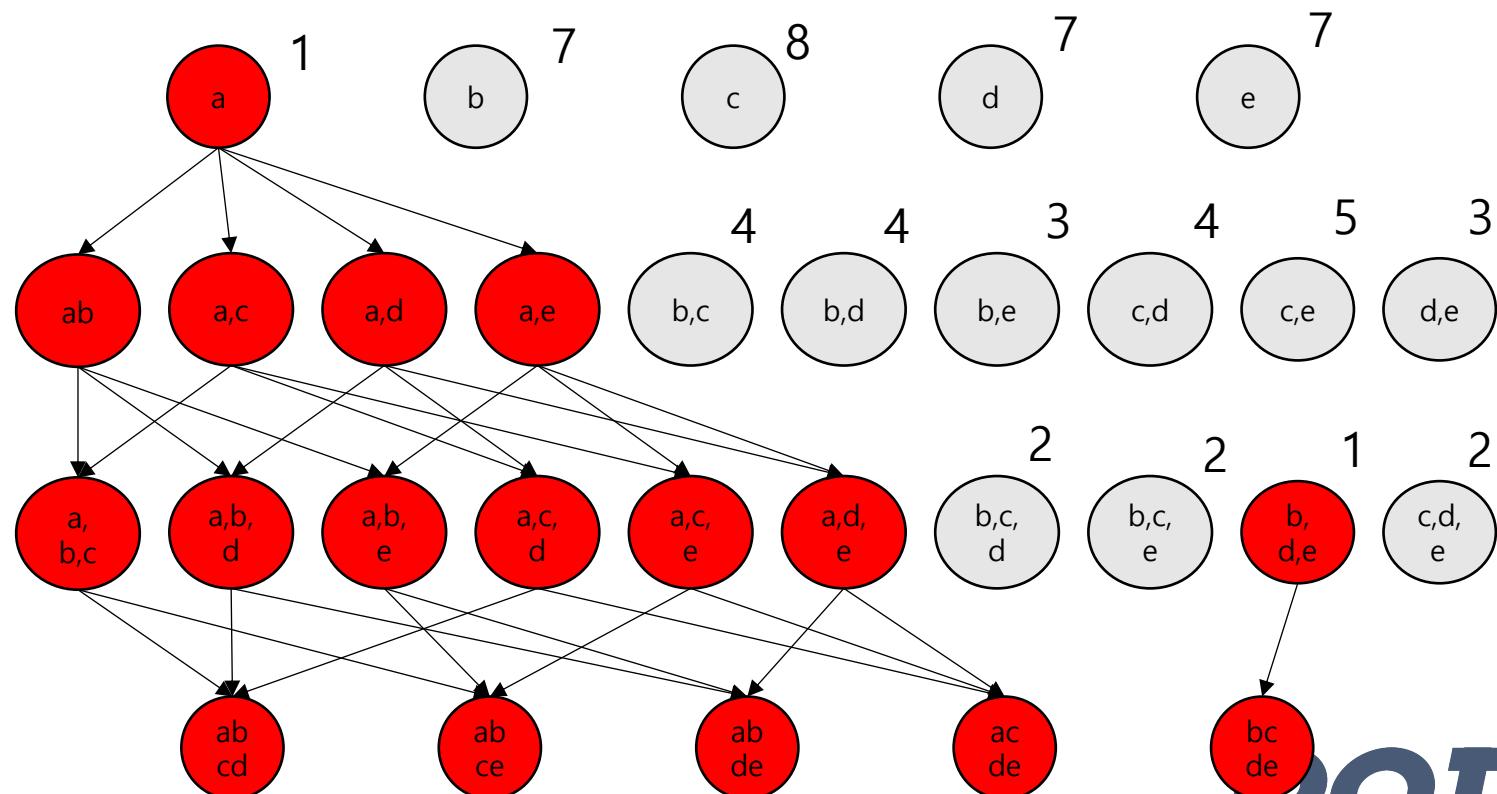
장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}



빈발집합 찾기 알고리즘 : Apriori

3. 생략되고 남은 것들에 한에 카운트를 함

장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}

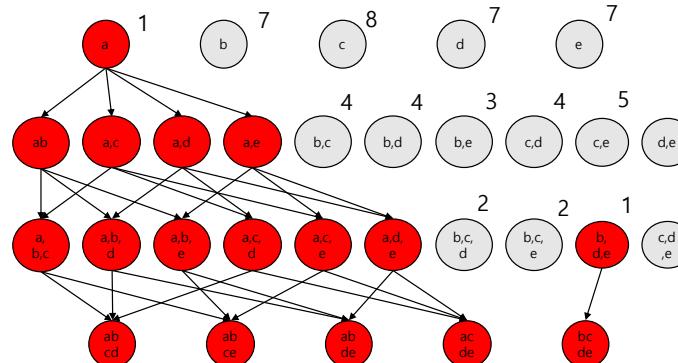


빈발집합 찾기 알고리즘 : Apriori

탐색할 아이템 집합이 없어지면, 종료 후 빈발집합 도출

장바구니 (고객이 함께 선호 혹은 구매한 제품군)	
0	{b, c}
1	{b, d, a}
2	{c, e}
3	{c, d, e}
4	{c, d}
5	{b, d}
6	{b, c, e}
7	{d, e}
8	{b, c, d}
9	{c, e}
10	{b, e}
11	{b, c, d, e}

아프리오리 알고리즘



빈발집합 도출

빈발집합	Freq
{b}	7
{c}	8
{d}	7
{e}	7
{b,c}	4
{b,d}	4
{b,e}	3
{c,d}	4
{c,e}	5
{d,e}	3
{b,c,d}	2
{b,c,e}	2
{c,d,e}	2

빈발집합을 통한 연관 분석

빈발 집합이 도출되고 나면 그 후 바로 연관 관계를 수식으로 계산 가능

빈발집합	Freq
{b}	7
{c}	8
{d}	7
{e}	7
{b,c}	4
{b,d}	4
{b,e}	3
{c,d}	4
{c,e}	5
{d,e}	3
{b,c,d}	2
{b,c,e}	2
{c,d,e}	2

연관 관계 ($b \rightarrow c$)

: 만약 b를 구매하였을 때, c도 동시에 구매할까요?

$$(1) \text{ 지지도} : \text{Support}(c) = \frac{8}{12}$$

$$(2) \text{ 신뢰도} : \text{confidence}(b \rightarrow c) = \frac{\text{freq}(b,c)}{\text{freq}(b)} = \frac{4}{7}$$

$$(3) \text{ 리프트} : \text{lift}(b \rightarrow c) = \frac{\text{confidence}(b \rightarrow c)}{\text{support}(c)} = \frac{14}{12}$$

빈발집합을 통한 연관 분석

빈발 집합이 도출되고 나면 그 후 바로 연관 관계를 수식으로 계산 가능

빈발집합	Freq
{b}	7
{c}	8
{d}	7
{e}	7
{b,c}	4
{b,d}	4
{b,e}	3
{c,d}	4
{c,e}	5
{d,e}	3
{b,c,d}	2
{b,c,e}	2
{c,d,e}	2

연관 관계 ($c \rightarrow e$)

: 만약 c 를 구매하였을 때, e 도 동시에 구매할까요?

$$(1) \text{ 지지도} : \text{Support}(e) = \frac{7}{12}$$

$$(2) \text{ 신뢰도} : \text{confidence}(c \rightarrow e) = \frac{\text{freq}(c,e)}{\text{freq}(c)} = \frac{5}{8}$$

$$(3) \text{ 리프트} : \text{lift}(c \rightarrow e) = \frac{\text{confidence}(c \rightarrow e)}{\text{support}(c)} = \frac{60}{56}$$