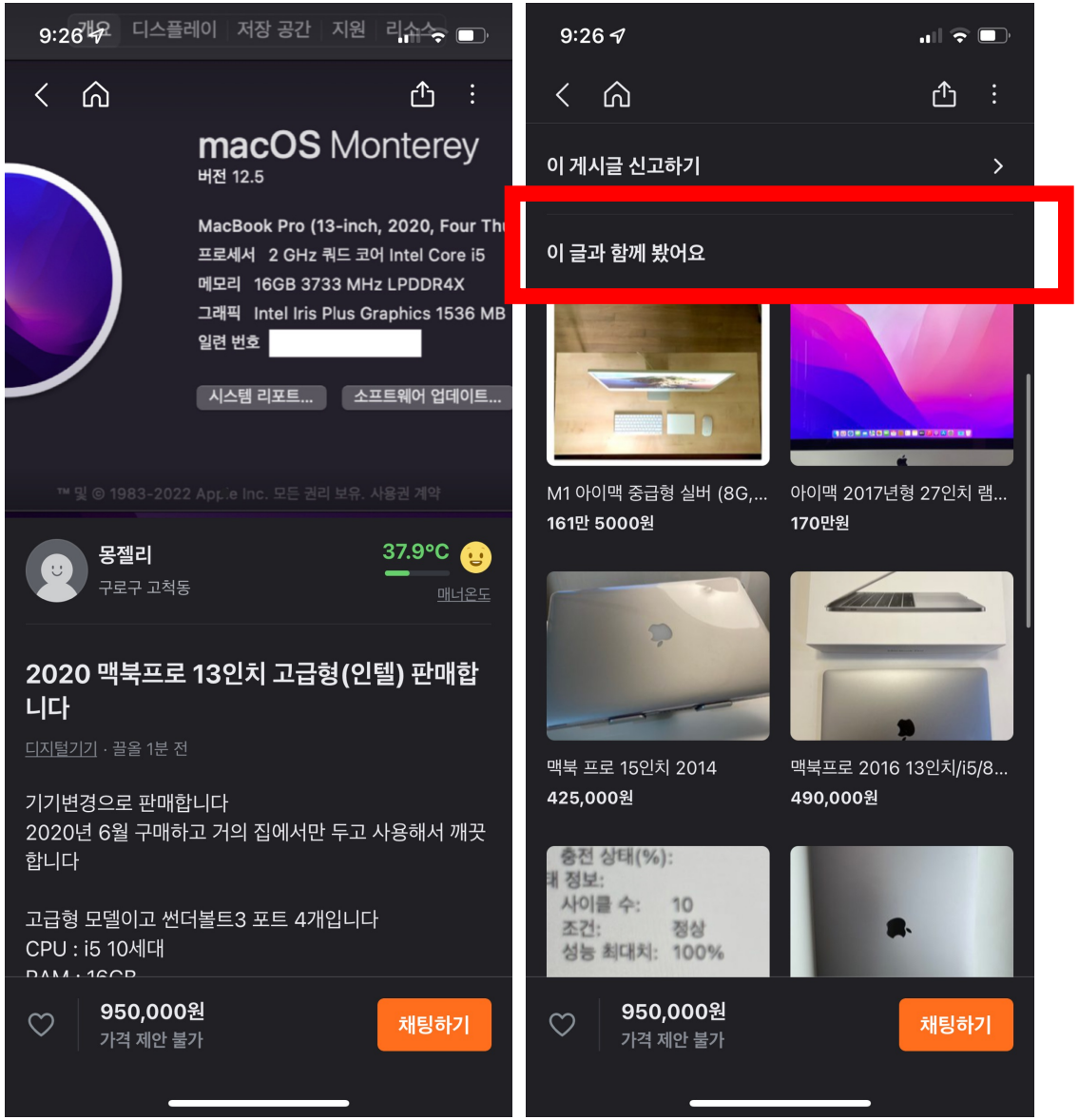
Introduction to Recommender System

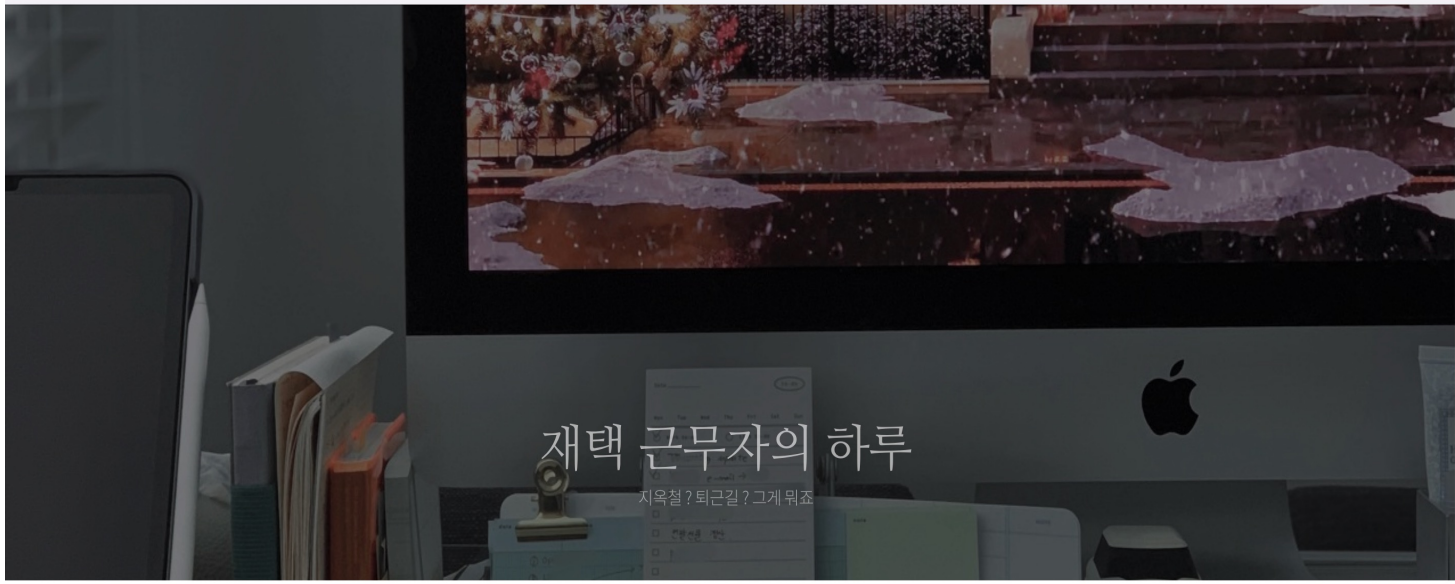
2022.09.08 /6기 이승재

당근마켓


9:26
디스플레이
저장 공간
지원
macOS Monterey
버전 12.5
MacBook Pro (13-inch, 2020, Four Th
프로세서 2 GHz 쿼드 코어 Intel Core i5
메모리 16GB 3733 MHz LPDDR4X
그래픽 Intel Iris Plus Graphics 1536 MB
일련 번호
시스템 리포트...
소프트웨어 업데이트...
™ 및 © 1983-2022 App.e Inc. 모든 권리 보유. 사용권 계약
몽젤리
구로구 고척동
37.9°C
매너온도
2020 맥북프로 13인치 고급형(인텔) 판매합니다
디지털기기 · 끌올 1분 전
기기변경으로 판매합니다
2020년 6월 구매하고 거의 집에서만 두고 사용해서 깨끗합니다
고급형 모델이고 썬더볼트3 포트 4개입니다
CPU : i5 10세대
950,000원
가격 제안 불가
채팅하기
9:26
이 게시글 신고하기
이 글과 함께 봤어요
M1 아이맥 중급형 실버 (8G,...
161만 5000원
아이맥 2017년형 27인치 램...
170만원
맥북 프로 15인치 2014
425,000원
맥북프로 2016 13인치/i5/8...
490,000원
충전 상태(%):
정보:
사이클 수: 10
조건: 정상
성능 최대치: 100%
950,000원
가격 제안 불가
채팅하기

brunch

brunch.co.kr



콘텐츠 생산자가 되야 하는 이유

안녕하세요? 신사동 마케터입니다. 이번 포스팅은 콘텐츠 생산자가 되어야 하는 이유에 대해 이야기해 보려고

by 신사동 마케터

마케팅과 전략 기획, 어떻게 다른 직무일까요?

멘토님 조언 읽고, 마케터로서의 취업에 대한 의사가 더 강해졌습니다. 모자란 지식 더 보충하고 싶은데, 멘토님

by 잇다 itdaa

공무원을 그만뒀어도 하늘은 무너지지 않았다.

의원면직 후 반년을 보내며 | < 사진 임자 = 글임자 > 예전에 어떤 영상을 우연히 보게 됐는데 10년 동안 고등학교 교사 생활을 해 온 사람이 그동안 너무 힘들었다며 …

by 글임자

슬기로운 재택근무 생활

재택근무 환경에서 일잘러 되는 법 | 얼마전 회사에서 한 인턴분과 점심을 먹는 중이었다. "재택근무는 정말 바람직한 제도인것 같습니다. 적극 권장 및 도입해야 한다…

by 아이스라떼부장

11시에 방으로 출근하는 재택근무자

아침 7시 반, 옆에서 리나가 꿈틀거린다. 30개월짜리 우리 딸이 꿈나라에서 돌아온 것이다. 몸을 일으켜 암막 커튼을 걷어 올리고 기지개를 켠다. 솔직히 그렇게 개운…

by 최혁재

스타트업에게 문제란 무엇인가?(ft. 류중희 대표님)

스타트업은 빠르게 성장하기 위해 설계된 회사다. | 퓨처플레이 류중희 대표님의 특강 '문제가 문제다'를 들었다. 스타트업 투자자로, 창업의 선배로서 겪었던 경험과 …

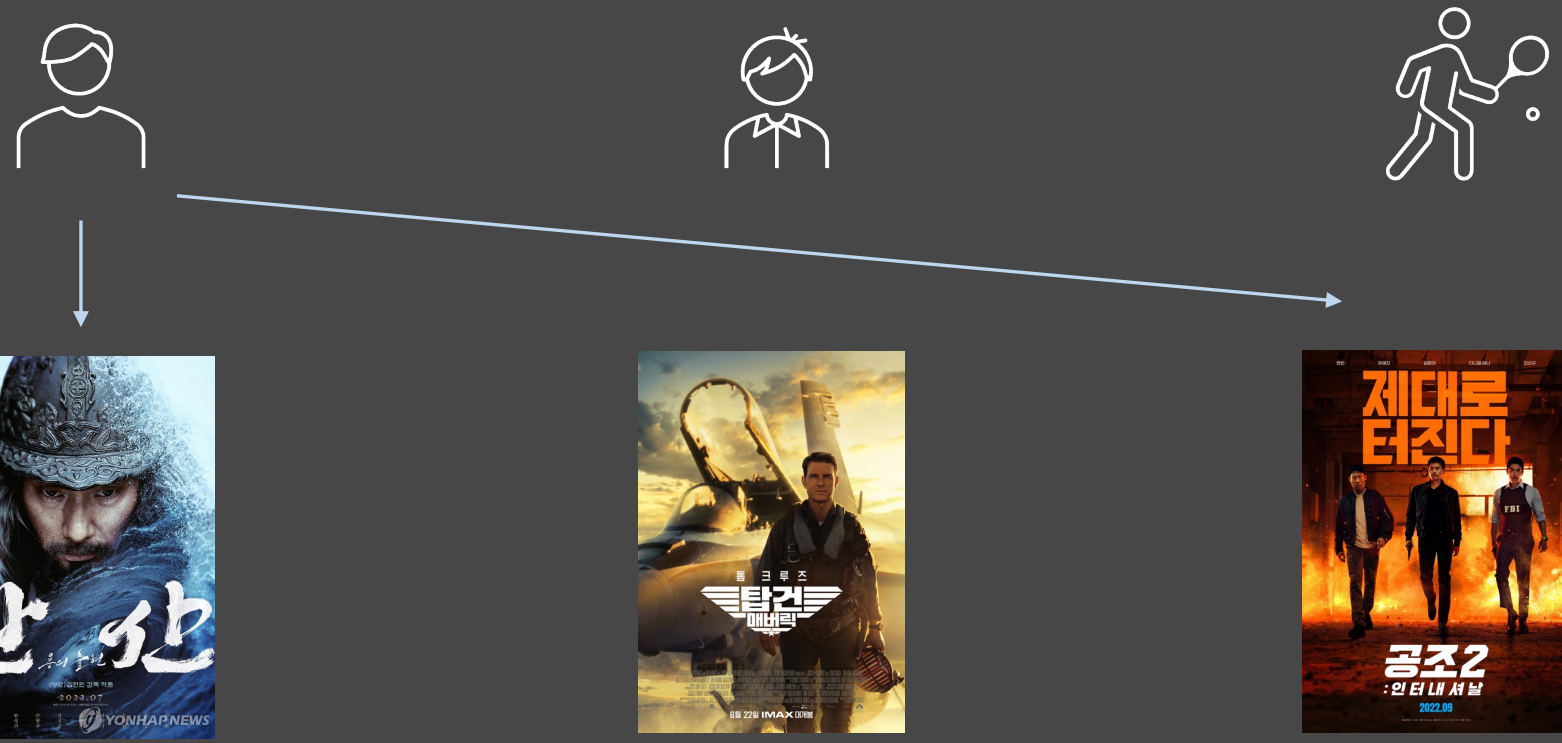
by 규규

목차

1. Basics

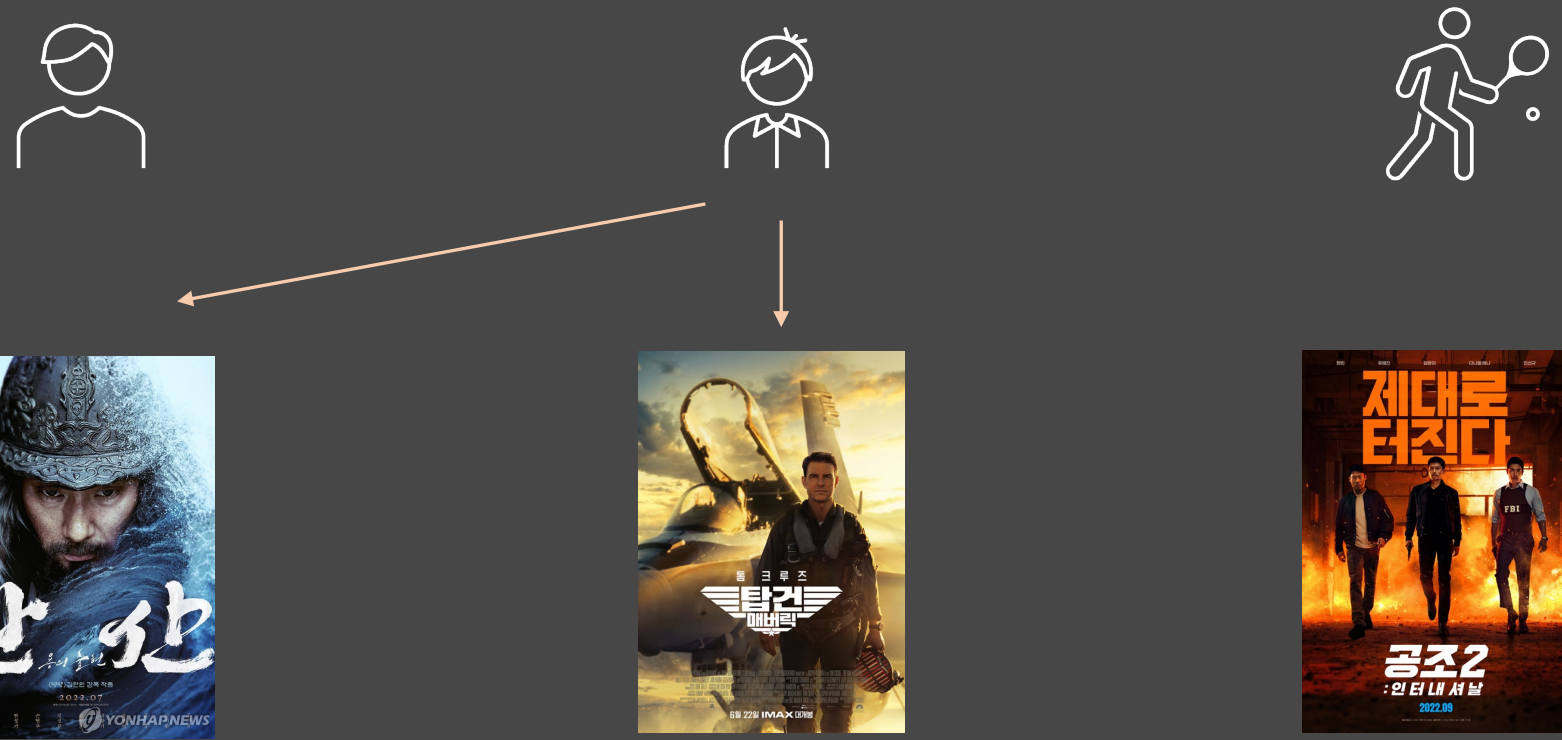
Content-based Filtering

사용자가 특정 아이템을 선호하는 경우, 그 아이템과 비슷한 컨텐츠를 추천하는 방법



Content-based Filtering

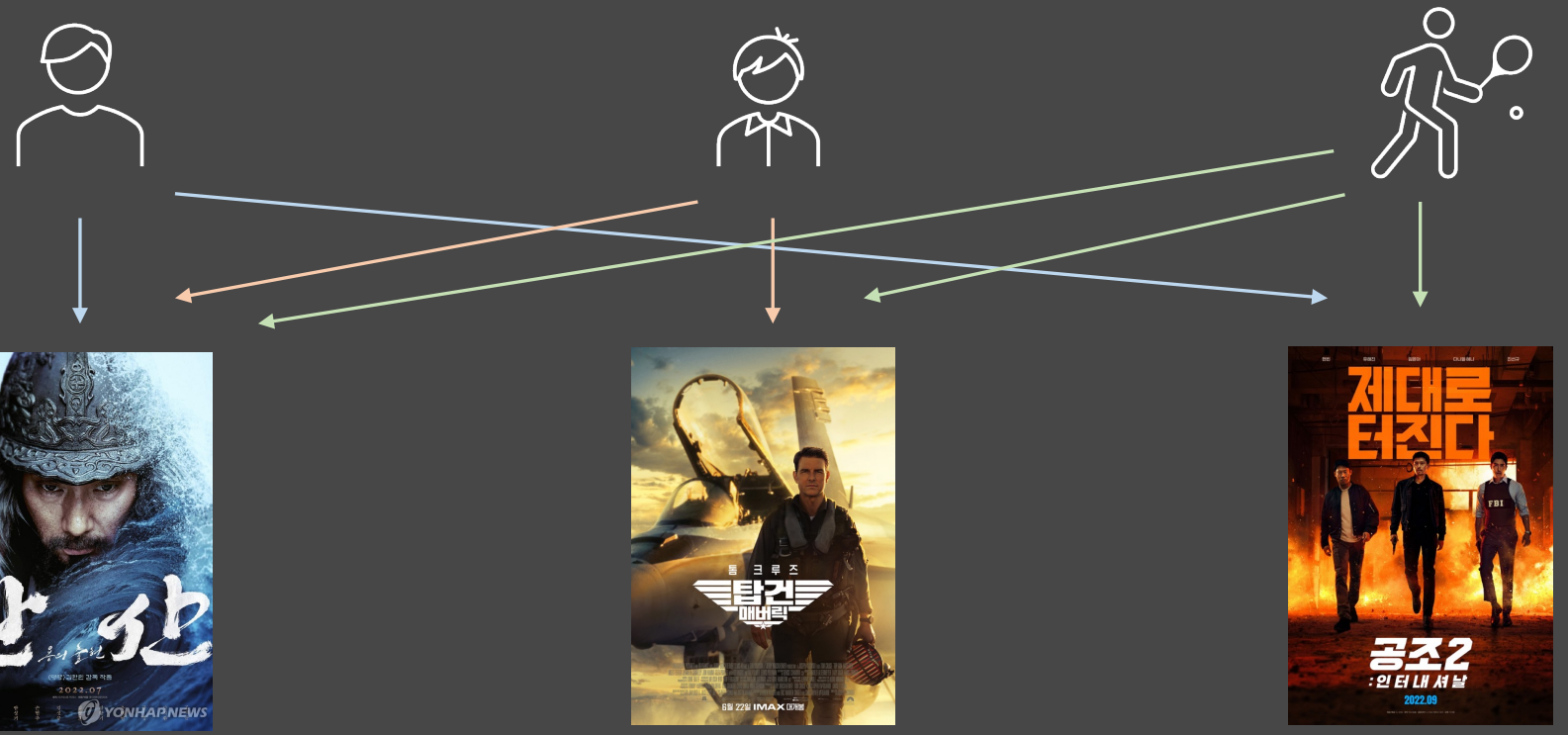
사용자가 특정 아이템을 선호하는 경우, 그 아이템과 비슷한 컨텐츠를 추천하는 방법



1. Basics

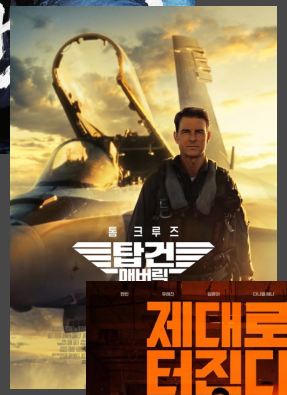
Content-based Filtering

사용자가 특정 아이템을 선호하는 경우, 그 아이템과 비슷한 컨텐츠를 추천하는 방법



1. Basics

Content-based Filtering



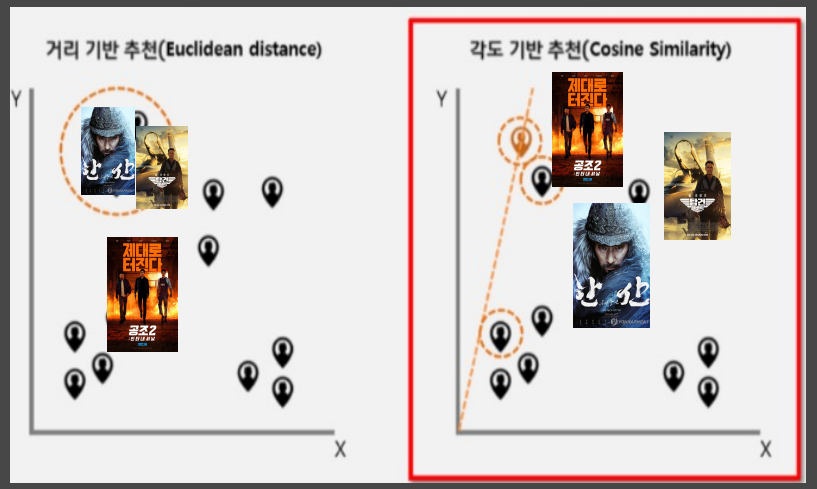
Genre

액션, 역사

액션

액션, 코미디, 공포

Vectorize



Calculation

Recommendation

CountVectorizer

문장에 어떤 단어가 몇 번 등장했는지를 세서(count) 문장을 벡터화

```
Text = [ "나는 테니스를 좋아해."
         "나는 러닝을 좋아해."
         "나는 DSL을 좋아해."]
```

```
[ "나는" : 0,
 "테니스" : 1,
  "러닝" : 2,
 "좋아해" : 3,
  "DSL" : 4,
   "을" : 5,
   "를" : 6 ]
```

```
array( [1, 1, 0, 1, 0, 0, 1],
       [1, 0, 1, 1, 0, 1, 0],
       [1, 0, 0, 1, 1, 1, 0], dtype=int64 )
```

TF-IDF

특정 문장 내에 단어가 얼마나 자주 등장하는지를 이용하되 너무 자주 등장하는 단어에 대해 가중치를 주는 방법

$$TF - IDF = TF * IDF$$

TF : 특정 문서 A에서 특정 단어 t의 등장 횟수

DF : 특정 단어 t가 등장한 문서의 수

IDF : DF에 반비례하는 수

$$idf(A, t) = \log(\frac{n}{1+df(t)})$$

TF-IDF

문서	내용
0	먹고 싶은 사과
1	먹고 싶은 바나나
2	길고 노란 바나나 바나나
3	저는 과일이 좋아요

TF : 특정 문서 A에서 특정 단어 t의 등장 횟수

	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

TF-IDF

문서	내용
0	먹고 싶은 사과
1	먹고 싶은 바나나
2	길고 노란 바나나 바나나
3	저는 과일이 좋아요

DF : 특정 단어 t가 등장한 문서의 수

	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
총합	1	1	1	2	3	1	2	1	1

IDF : DF에 반비례하는 수

$$idf(A,t) = \log(\frac{n}{1+df(t)})$$

단어	IDF
과일이	$\ln(\frac{4}{1+1}) = 0.693147$
길고	$\ln(\frac{4}{1+1}) = 0.693147$
노란	$\ln(\frac{4}{1+1}) = 0.693147$
먹고	$\ln(\frac{4}{1+2}) = 0.287682$
바나나	$\ln\left(\frac{4}{1+3}\right) = 0$
사과	$\ln(\frac{4}{1+1}) = 0.693147$
싶은	$\ln(\frac{4}{1+2}) = 0.287682$
저는	$\ln(\frac{4}{1+1}) = 0.693147$
좋아요	$\ln(\frac{4}{1+1}) = 0.693147$

TF-IDF

$$TF - IDF = TF * IDF$$

	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	0.287682	0	0.693147	0.287682	0	0
문서2	0	0	0	0.287682	0	0	0.287682	0	0
문서3	0	0.693147	0.693147	0	0	0	0	0	0
문서4	0.693147	0	0	0	0	0	0	0.693147	0.693147

TF-IDF

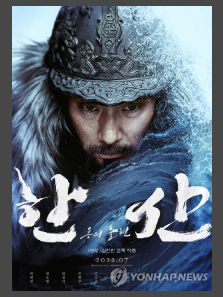
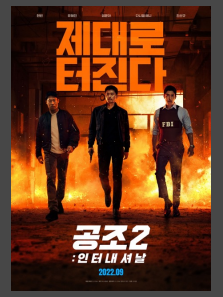
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

문서	내용
0	먹고 싶은 사과
1	먹고 싶은 바나나
2	길고 노란 바나나 바나나
3	저는 과일이 좋아요

	문서1	문서2	문서3	문서4
문서1	1	0.5601	0	0
문서2	0.5601	1	0	0
문서3	0	0	1	0
문서4	0	0	0	1

1. Basics

Latent Factor Collaborative Filtering

				
한빈 (가상인물)	3	?	3	
유리	4.5	5	?	
승재	3	5	4	

1. Collaborative Filtering

SVD (특이값 분해) 를 이용한 latent factor CF

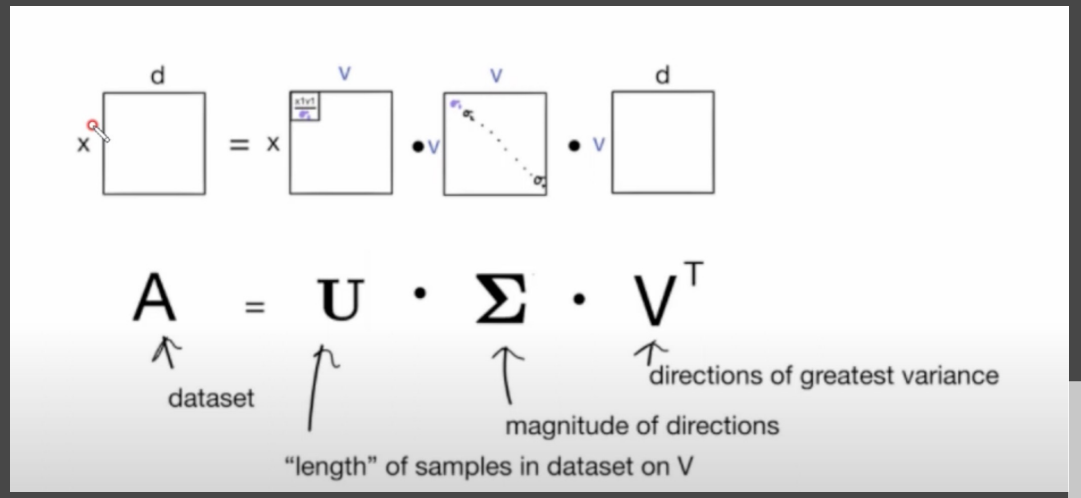
$$A = U \Sigma V^t$$

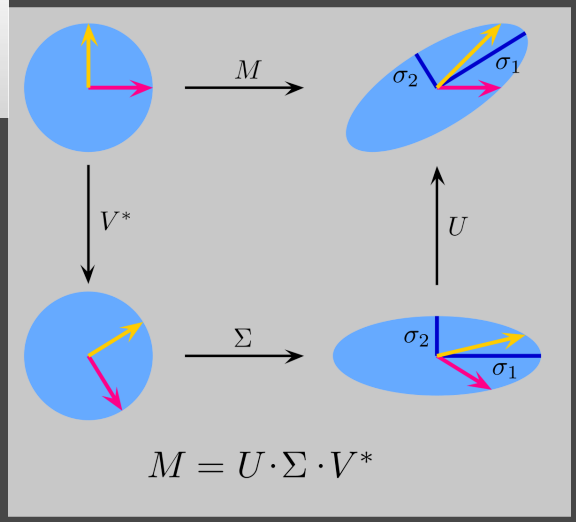
A : m x n rectangular matrix

U : m x m orthogonal matrix

Σ : m x n diagonal matrix

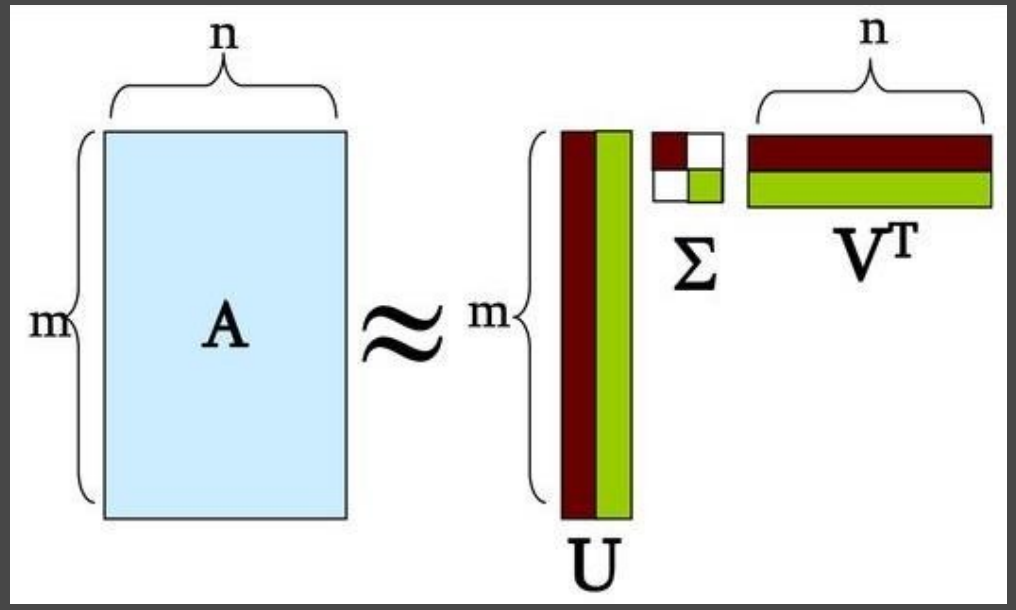
V^t *: n x n orthogonal matrix*





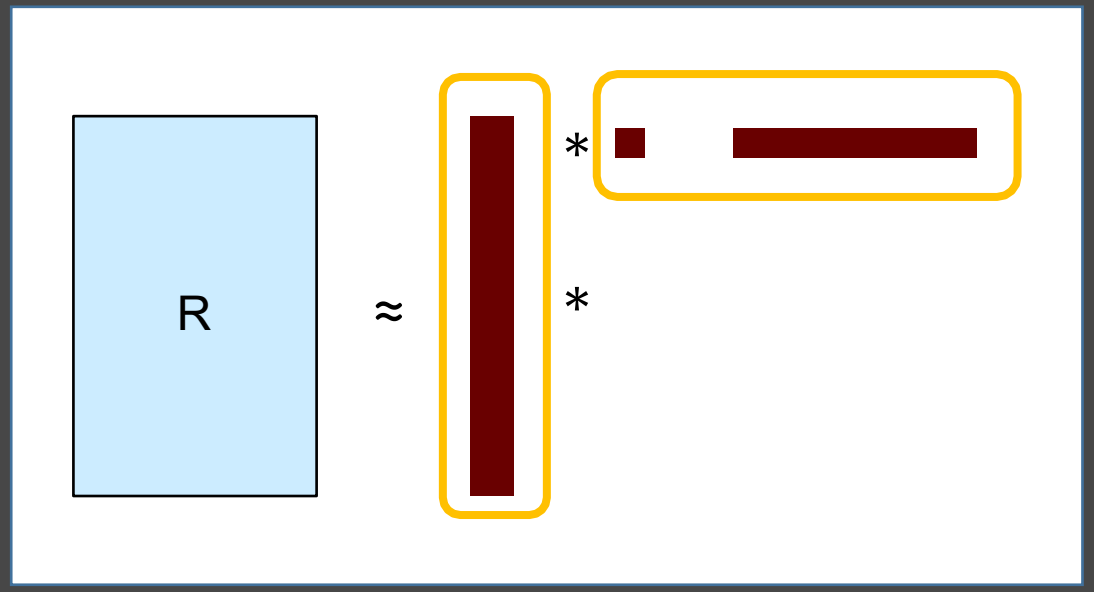
1. Collaborative Filtering

$A = U \Sigma v^T$



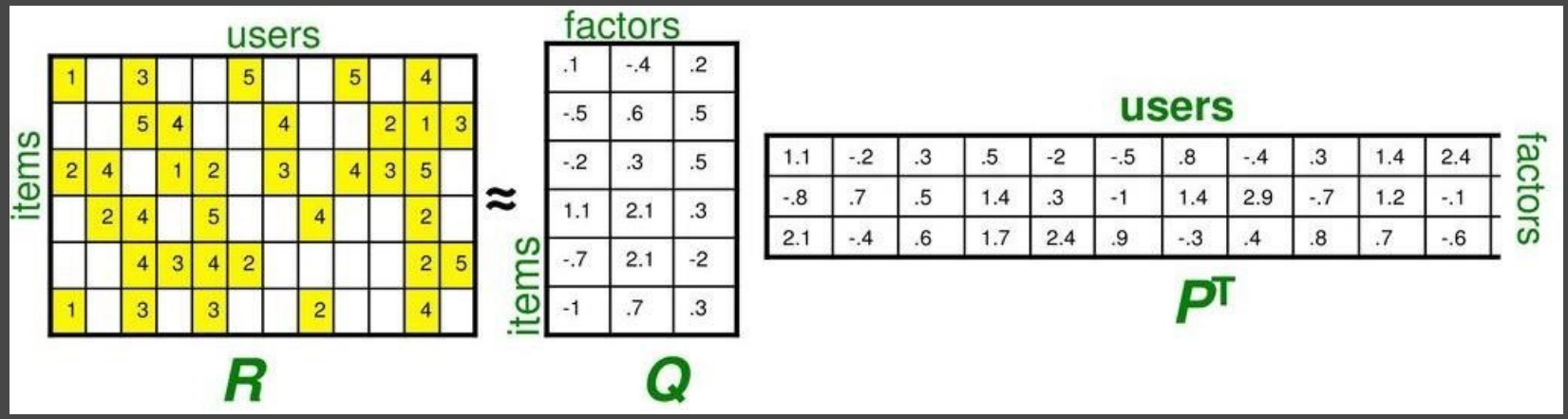
A: input data matrix
U: “user-to-concept” matrix
Σ: strength of each concept
V: “movie to concept” matrix

$R \approx Q P^T$



R: rating matrix
Q: user-latent factor matrix
P^T: item-latent factor matrix

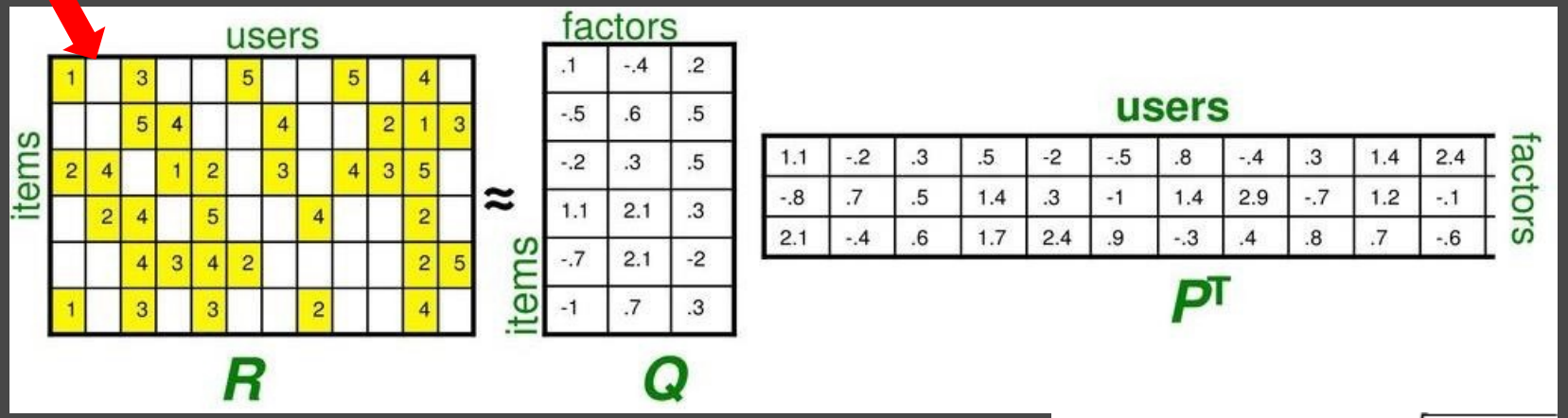
1. Collaborative Filtering



$$\text{RMSE} = \sqrt{MSE} = \sqrt{\frac{\Sigma(\hat{y}-y)^2}{}}$$

1. Collaborative Filtering

SVD를 이용하게 되면 빈 값들은 0으로 체워줘야 한다.



1		3			5			5		4	
		5	4			4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6

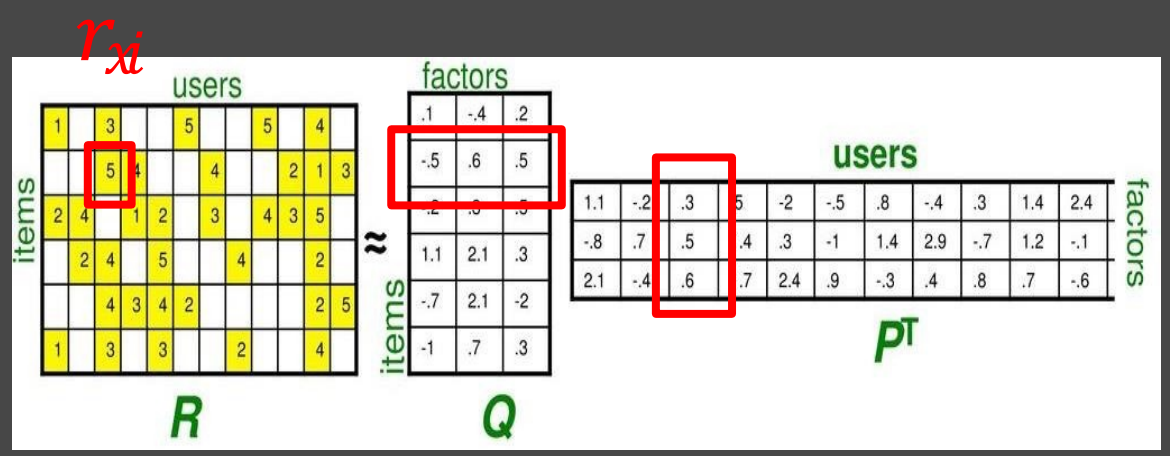
$$\text{RMSE} = \sqrt{MSE} = \sqrt{\frac{\sum(\hat{y}-y)^2}{}}$$

1. Collaborative Filtering

SGD를 이용한 latent factor CF

Cost Function : $\min_{P,Q} \Sigma(Rxi - Qi * Px)^2$

Update : Gradient Descent

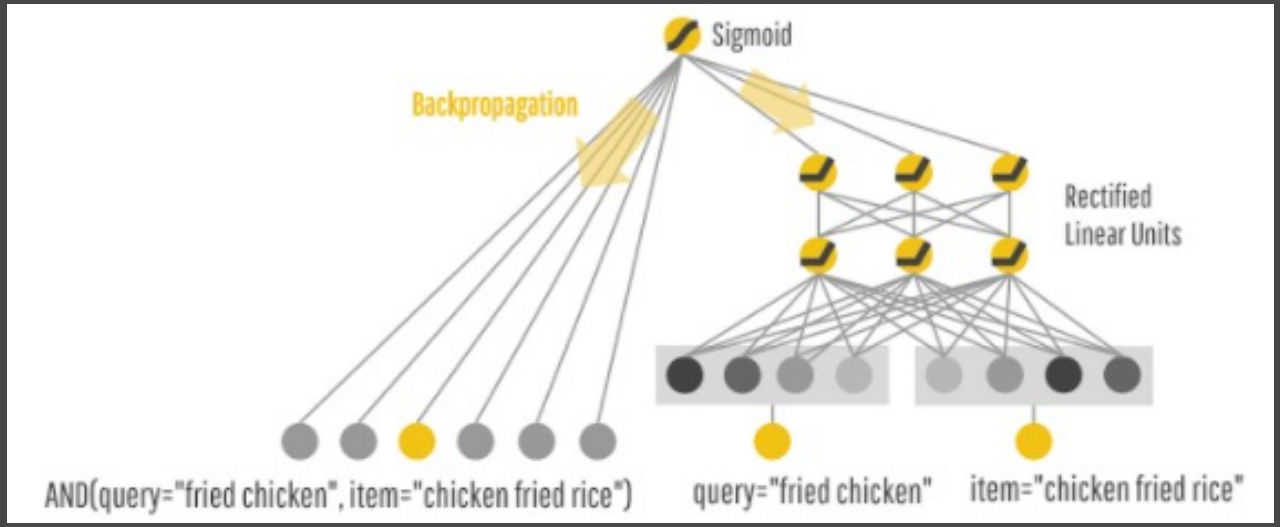


1. Basics - 비교

Content Based Filtering	Collaborative Filtering
- 아이템/유저에 대한 데이터 퀄리티 - 도메인 지식 필요 O - 다양한 항목 추천 어려움 - Cold Start Problem 해결	- 아이템/유저에 대한 historical data - 도메인 지식 필요 X - 편향된 추천 - Cold Start Problem 존재

2. DeepLearning Method

Wide and Deep



Wide

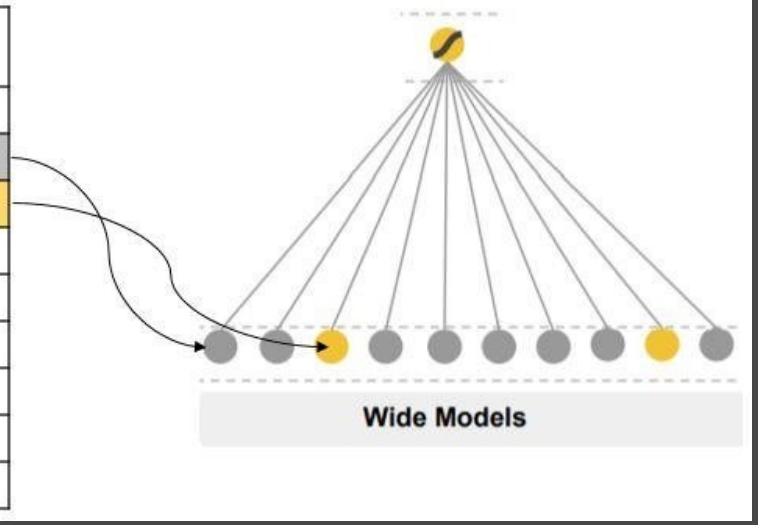
- Memorization에 특화된 선형 모델
- Memorization이란: 함께 자주 등장하는 속성의 correlation을 추출하여 활용하는 것

선형 식

$$y = w^T x + b$$

y = prediction
x = 속성 벡터
w = 모델 파라미터
b = bias

Install	Impression	(Install, Impression)	Install x Impression
A	A	(1,1)	1
A	B	(1,0)	0
A	C	(1,1)	1
B	A	(1,1)	1
B	B	(1,0)	0
B	C	(1,1)	1
C	A	(0,1)	0
C	B	(0,0)	0
C	C	(0,1)	0



Cross-Product Transformation

$$\phi_k(\mathbf{x}) = \prod_{i=1}^{d} x_i^{c_{ki}} \qquad c_{ki} \in \{0, 1\}$$

x = 속성 벡터
c_{ki} = k번째 transformation에서 i번째 속성의 boolean 값
d = Memorization에 사용할 아이템 속성의 개수

예시)

- $\text{gender} = [\text{male}, \text{female}] = [1, 0]$
- $\text{education} = [\text{high}, \text{low}] = [1, 0]$
- $\text{language} = [\text{eng}, \text{kor}] = [1, 0]$

$$\mathbf{x} = [\text{gender}, \text{education}, \text{language}] = [\text{male}, \text{low}, \text{kor}] = [1, 0, 0]$$

$$\phi_2(\mathbf{x}) = x_1^{c_{21}} x_2^{c_{22}} x_3^{c_{23}} = 1^1 0^0 0^1 = 0, \quad 0^0 \equiv 1$$

Deep

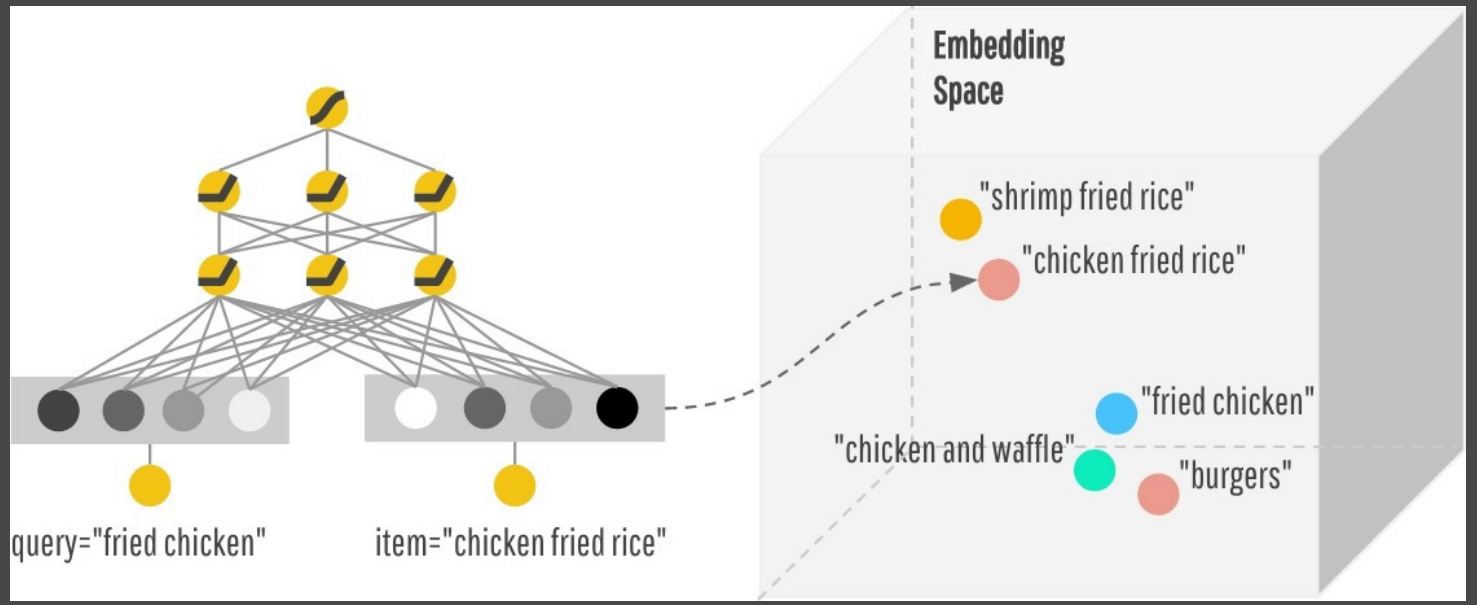
- Generalization에 특화된 순전파 신경망 모델

모델 학습 과정

1. Continuous 속성과 Categorical 속성을 concatenate 한 벡터를 인풋으로 사용함

2. Sparse, 고차원 속성 벡터를 dense, 저차원 벡터로 임베딩함

3. 딥러닝 레이어에 넣어 학습시킴



Deep

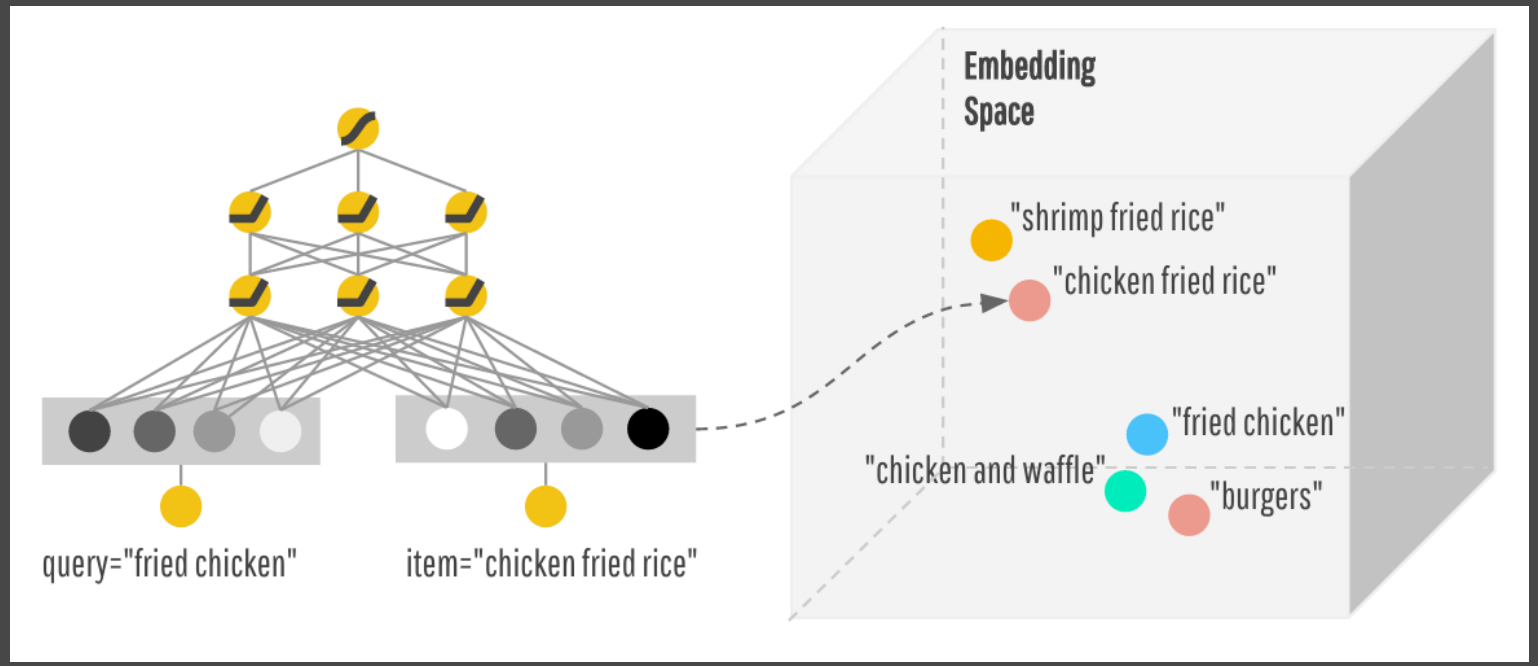
- Generalization에 특화된 순전파 신경망 모델

모델 학습 과정

1. Continuous 속성과 Categorical 속성을 concatenate 한 벡터를 인풋으로 사용함

2. Sparse, 고차원 속성 벡터를 dense, 저차원 벡터로 임베딩함

3. 딥러닝 레이어에 넣어 학습시킴



Deep

$$a^{(l+1)} = f(W^{(l)}a^{(l)} + b^{(l)})$$

l = 레이어 넘버

f = Activation function (논문에서는 ReLU 사용)

$a^{(l)}$ = l번째 레이어의 activation

$b^{(l)}$ = l번째 레이어의 bias

$W^{(l)}$ = l번째 레이어의 모델 weights

Joint Training

- 어러 모델을 결합하는 앙상블과 달리, joint training 기법은 아웃풋의 gradient를 wide와 deep 모델에 동시에 backpropagation하여 학습함

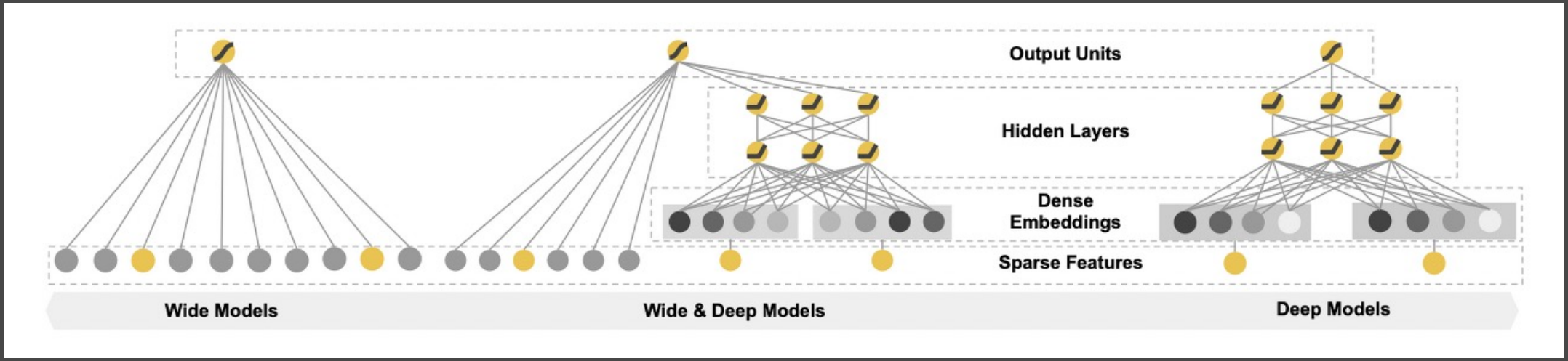
모델 Prediction

$$P(Y=1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(l_f)} + b)$$

Y = 클래스 라벨
$\sigma()$ = 시그모이드 function
$\phi(x)$=cross product transformations
b = bias

w_{wide} = wide 모델의 weights vector
w_{deep} = 마지막 activation에 적용된 deep 모델 weights
Wide 모델 optimizer = FTRL알고리즘 + L1 정형화
Deep 모델 optimizer = Adagrad

Wide and Deep



3. Metrics

MAP (Mean Average Precision)

$$MAP@K = \frac{1}{|U|}\sum_{u=1}^{|U|}(AP@K)_u$$

Rank	Recommendation	Result
1	Forrest Gump	Correct positive
2	Titanic	False positive
3	Seven	False positive
4	The lion king	Correct positive
5	The Truman show	False positive
6	Jaws	Correct positive

3. Metrics

AP@K k=3

$$AP@K = \frac{1}{m}\sum_{i=1}^{K} P(i) \bullet rel(i)$$

P(i) : 해당 index 까지의 Precision

Rel(i) : 해당 index 에서의 user engagement

m : 실제로 사용자가 engage한 횟수

Recommendation	Precision@k (k=3)	AP@k (k=3)
[0, 0, 1]	[0, 0, 1/3]	$\frac{1}{3} * \left(0 + 0 + \frac{1}{3}\right) = \frac{1}{9}$
[0, 1, 1]	[0, 1/2, 2/3]	$\frac{1}{3} * \left(0 + \frac{1}{2} + \frac{1}{3}\right) = \frac{5}{18}$
[1, 1, 1]	[1/1, 2/2, 3/3]	$\frac{1}{3} * (1 + 1 + 1) = 1$

MAP@K k=3

$$MAP@K = \frac{1}{|U|}\sum_{u=1}^{|U|}(AP@K)_u$$

# of users	MAP@K (k=3)
N	$\frac{1}{N} * \left(\frac{1}{9} + \frac{5}{18} + \frac{1}{1}\right)$

4. To wrap up