

# 그래프를 이용한 기계 학습

## #6 그래프를 추천시스템에 어떻게 활용할까? (기본)

---

신기정

(KAIST AI대학원)

1. 우리 주변의 추천 시스템

2. 내용 기반 추천시스템

3. 협업 필터링

4. 추천 시스템의 평가

5. 실습: 협업 필터링 구현

# 1. 우리 주변의 추천 시스템

1.1 아마존에서의 상품 추천

1.2 넷플릭스에서의 영화 추천

1.3 유튜브에서의 영상 추천

1.4 페이스북에서의 친구 추천

1.5 추천 시스템과 그래프

## 1.1 아마존에서의 상품 추천

Amazon.com 메인 페이지에는 고객 맞춤형 **상품** 목록을 보여줍니다

Gifts in Video Games under \$30 [Shop now](#)

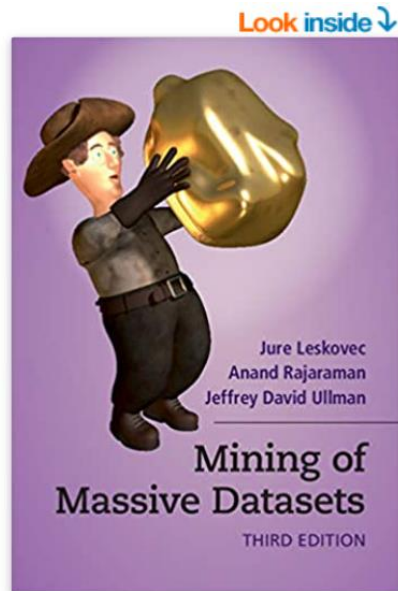


Stuffed Animals & Toys under \$10 [Shop now](#)

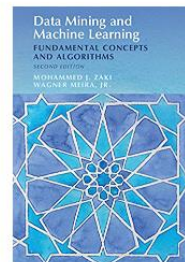


## 1.1 아마존에서의 상품 추천

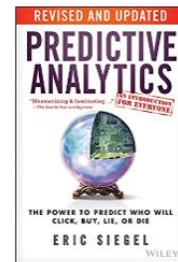
Amazon.com 에서 특정 상품을 살펴볼 때, 함께 혹은 대신 구매할 **상품** 목록을 보여줍니다



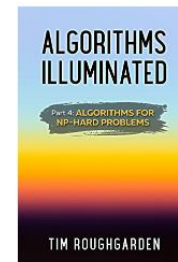
### Customers who bought this item also bought



Data Mining and Machine Learning: Fundamental Concepts...  
Mohammed J. Zaki  
★★★★☆ 7  
Kindle Edition  
\$48.00



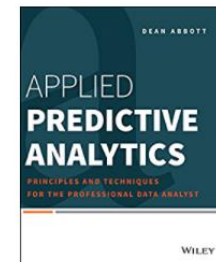
Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die  
> Eric Siegel  
★★★★☆ 292  
Kindle Edition  
\$14.00



Algorithms Illuminated (Part 4): Algorithms for NP-Hard Problems  
> Tim Roughgarden  
★★★★★ 77  
Kindle Edition  
\$9.99



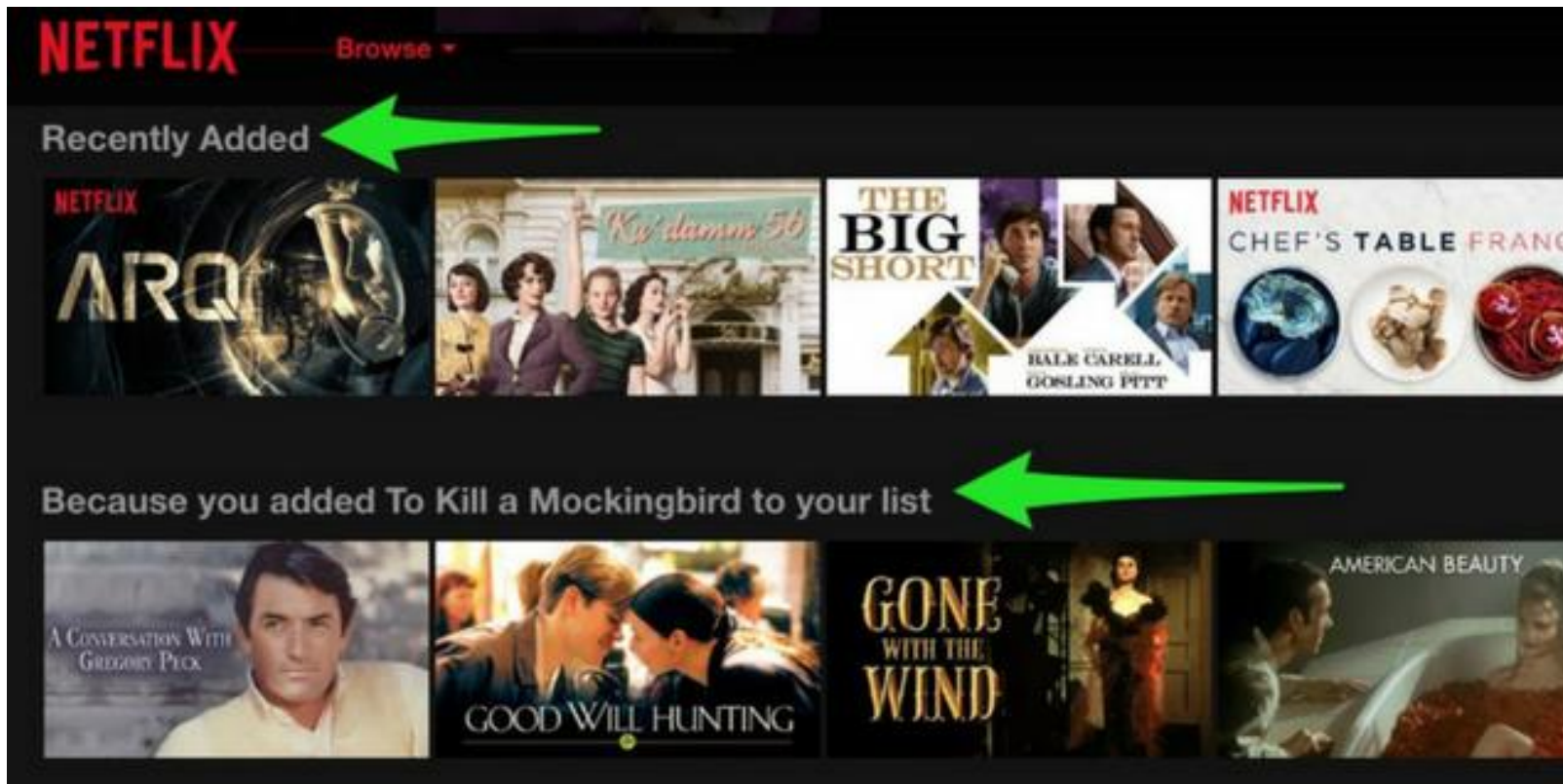
Machine Learning Engineering  
> Andriy Burkov  
★★★★★ 86  
Kindle Edition  
\$34.95



Applied Predictive Analytics: Principles and Techniques for the...  
> Dean Abbott  
★★★★☆ 60  
Kindle Edition  
\$40.00

## 1.2 넷플릭스에서의 영화 추천


넷플릭스 메인 페이지에는 고객 맞춤형 **영화** 목록을 보여줍니다






# 1.3 유튜브에서의 영상 추천


유튜브 메인 페이지에는 고객 맞춤형 **영상** 목록을 보여줍니다




**[전곡가사] 올해 내가 가장 많이 들은 아이유 노래 모음 Most I listened IU song...**  
아이유라는장르-GOIU PLAYLIST  
237K views • 1 month ago




**미리 메리 크리스마스 (원곡:아이유) - 세정 (SEJEONG) [뮤직뱅크/Music Bank]...**  
KBS Kpop  
410K views • 1 year ago




**eight(에잇) - IU 아이유 | Best Songs Of IU 아이유 최고의 노래모음 - IU 최고의 ...**  
K-POP CHARTS  
972K views • 8 months ago




**[IU] '내 손을 잡아(Hold My Hand)' Live Clip (2019 IU Tour Concert 'Love, poem')**  
이지금 [IU Official] ♪  
6.1M views • 1 month ago




**IU 아이유 | Best Songs Of IU 아이유 최고의 노래모음 - IU 최고의 노래 컬렉션 - I...**  
K-POP CHARTS  
4.3M views • 1 year ago




**[Playlist] 지은이가 추천해준 다른 가수들의 노래 모음**  
아이유라는장르-GOIU PLAYLIST  
401K views • 2 months ago




**[ENG SUB] IU prays in the forest for her wish to come true, what might it be...**  
KBS 대구  
158K views • 3 months ago



**아이유·유희열·지코·FT아일랜드·에픽하이,故 종현 조문 (현장)**  
STATV  
5.6M views • 3 years ago



**[Uncle] Lee Sang Soon "Inna, I'm a fan ..."Lee Hyori's Bed & Breakfast 2**  
JTBC Entertainment  
550K views • 3 years ago



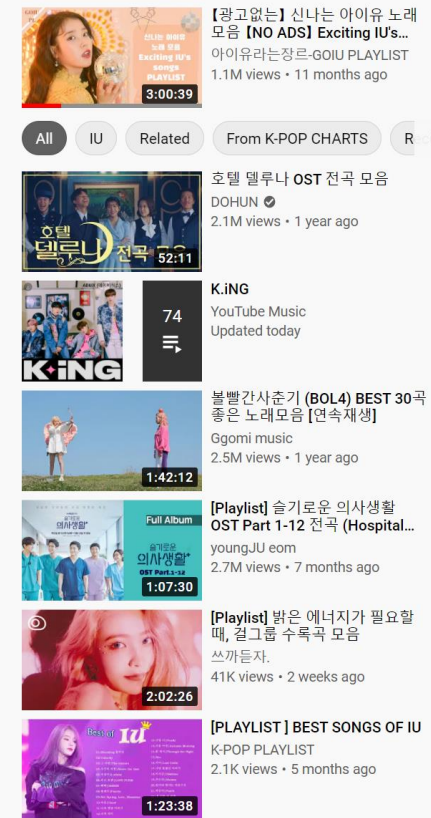
**추석 연휴에 아이유 다이어트 또 했습니 다.**  
일주어터  
614K views • 3 months ago

## 1.3 유튜브에서의 영상 추천

유튜브는 현재 재생 중인 영상과 관련된 영상 목록을 보여줍니다



eight(에잇) - IU 아이유 | Best Songs Of IU 아이유 최고의 노래모음 - IU 최고의 노래 컬렉션 - IU Spring Playlist 🌸

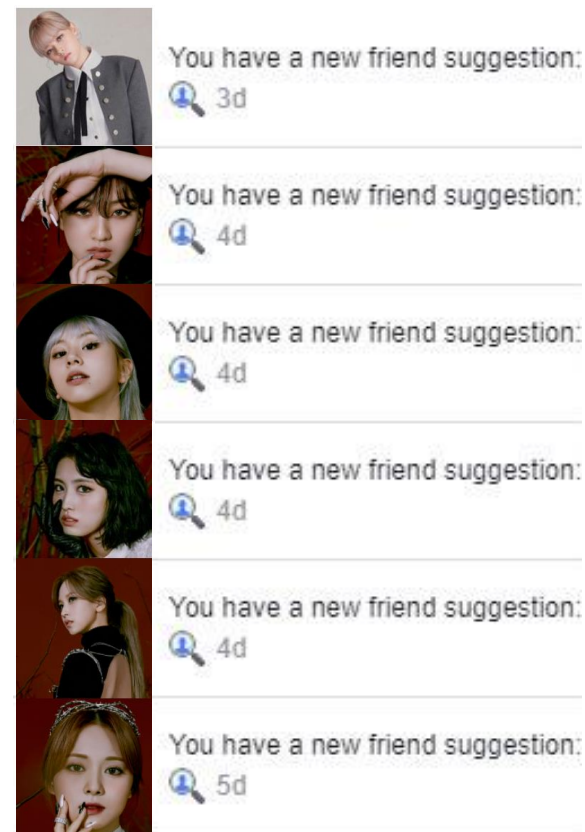




## 1.4 페이스북에서의 친구 추천

페이스북에서는 추천하는 친구의 목록을 보여줍니다

이처럼 추천의 대상은 다양하지만 본 강의에서는 편의상 **상품**을 추천하다고 가정하겠습니다



## 1.5 추천 시스템과 그래프

추천 시스템은 **사용자** 각각이 **구매**할 만한 혹은 **선호**할 만한 **상품**을 추천합니다

사용자별 구매 기록은 아래 예시처럼 **그래프**로 표현 가능합니다  
구매 기록이라는 암시적(Implicit)인 선호만 있는 경우도 있고,  
평점이라는 명시적(Explicit)인 선호가 있는 경우도 있습니다



## 1.5 추천 시스템과 그래프

추천 시스템은 **사용자** 각각이 **구매**할 만한 혹은 **선호**할 만한 **상품/영화/영상**을 추천합니다

추천 시스템의 핵심은 **사용자별 구매**를 예측하거나 **선호**를 추정하는 것입니다

그래프 관점에서 추천 시스템은

“미래의 간선을 예측하는 문제” 혹은

“누락된 간선의 가중치를 추정하는 문제”로

해석할 수 있습니다



## 2. 내용 기반 추천시스템

2.1 내용 기반 추천시스템의 원리

2.2 내용 기반 추천시스템의 장단점



## 2.1 내용 기반 추천시스템의 원리

---

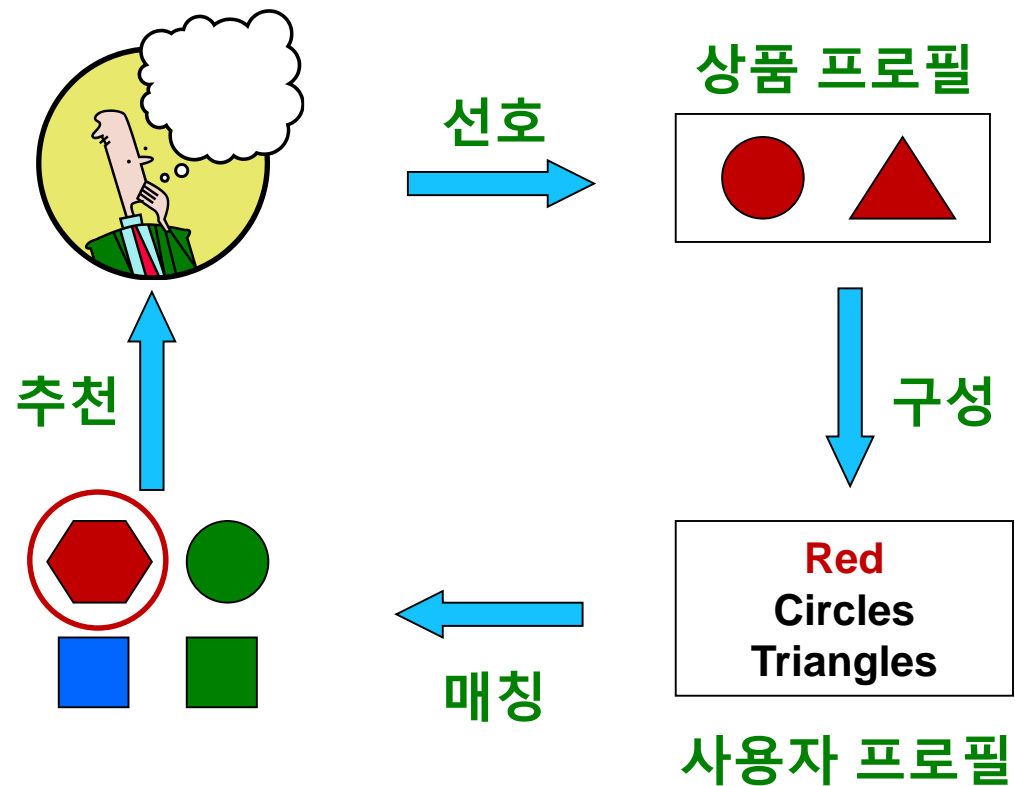
내용 기반(Content-based) 추천은 각 사용자가 구매/만족했던 상품과 유사한 것을 추천하는 방법입니다

예시는 다음과 같습니다

- 동일한 장르의 영화를 추천하는 것
- 동일한 감독의 영화 혹은 동일 배우가 출연한 영화를 추천하는 것
- 동일한 카테고리의 상품을 추천하는 것
- 동갑의 같은 학교를 졸업한 사람을 친구로 추천하는 것

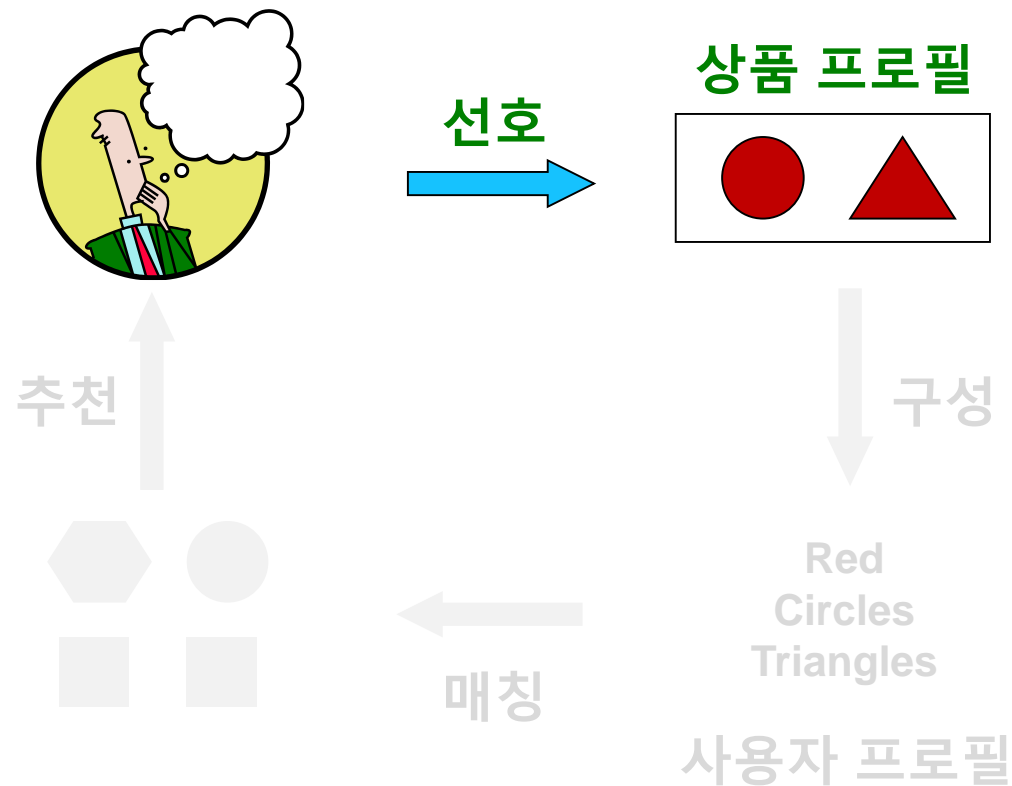
## 2.1 내용 기반 추천시스템의 원리

내용 기반 추천은 다음 네 가지 단계로 이루어집니다



## 2.1 내용 기반 추천시스템의 원리

첫 단계는 사용자가 선호했던 상품들의 **상품 프로필(Item Profile)**을 수집하는 단계입니다



어떤 상품의 **상품 프로필**이란  
해당 상품의 특성을 나열한 **벡터**입니다

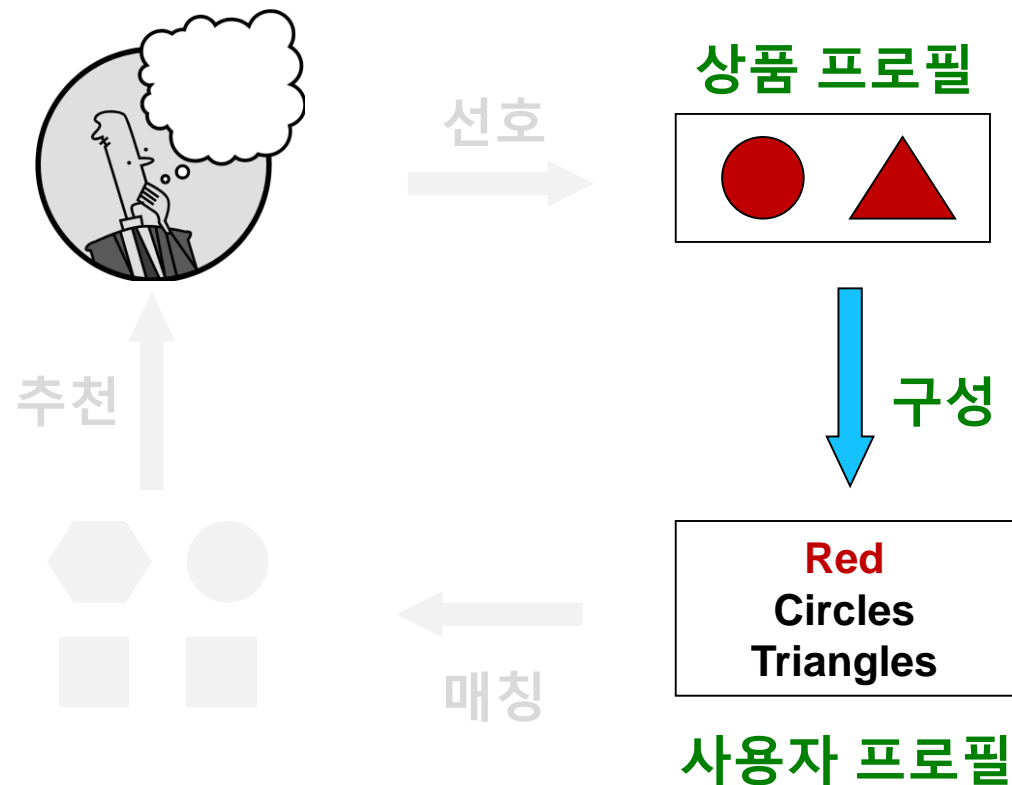
영화의 경우 감독, 장르, 배우 등의  
**원-핫 인코딩**이 상품 프로필이 될 수 있습니다

0	1	0	0				
---	---	---	---	--	--	--	--

로맨스  
코미디  
액션  
공포  
...

## 2.1 내용 기반 추천시스템의 원리

다음 단계는 **사용자 프로필(User Profile)**을 구성하는 단계입니다



**사용자 프로필**은 선호한 상품의 상품 프로필을 선호도를 사용하여 가중 평균하여 계산합니다 즉 사용자 프로필 역시 벡터입니다

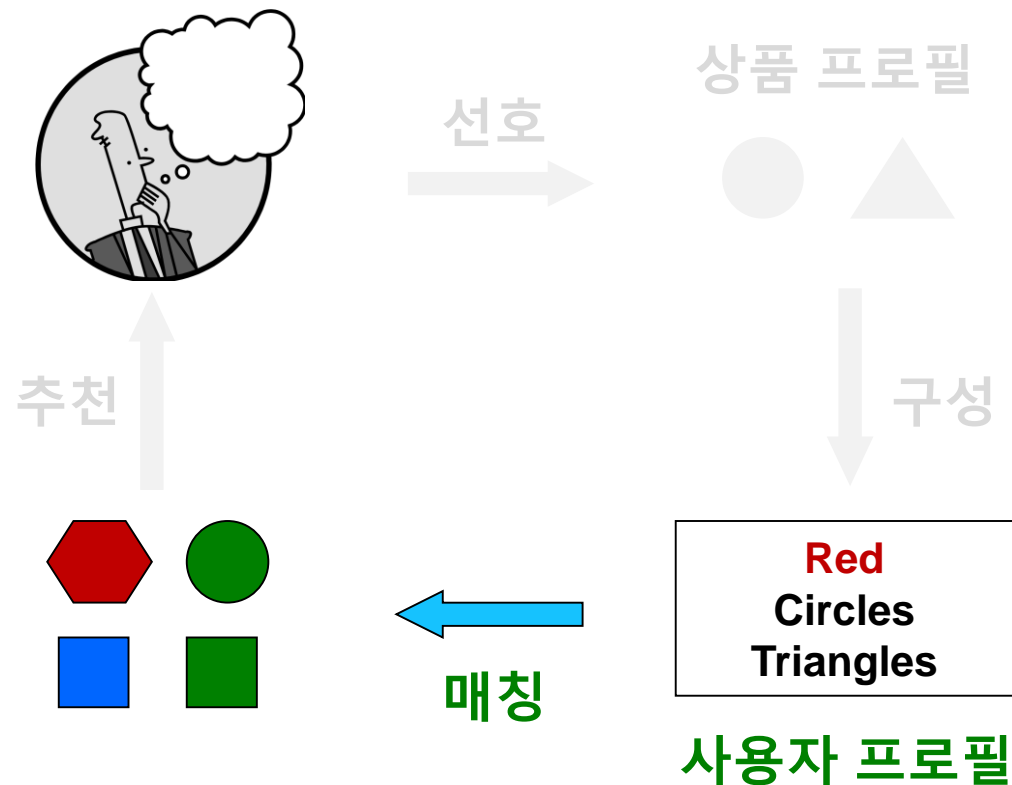
앞선 영화 프로필 예시에서는 다음과 같은 형태의 사용자 프로필을 얻을 수 있습니다

0.1	0.9	0.2	0.3				
로맨스	코미디	액션	공포	...			



## 2.1 내용 기반 추천시스템의 원리

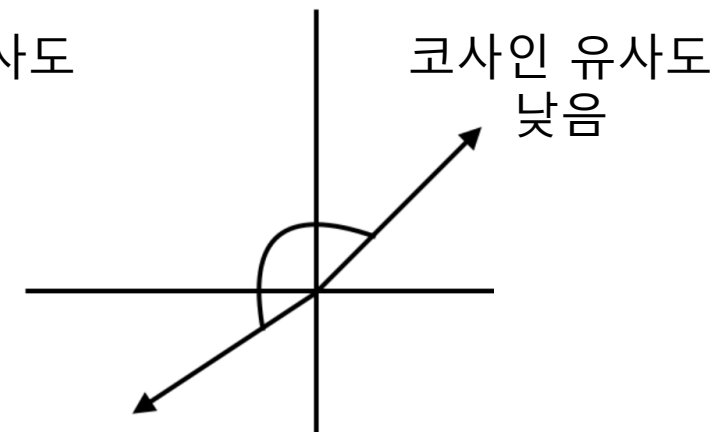
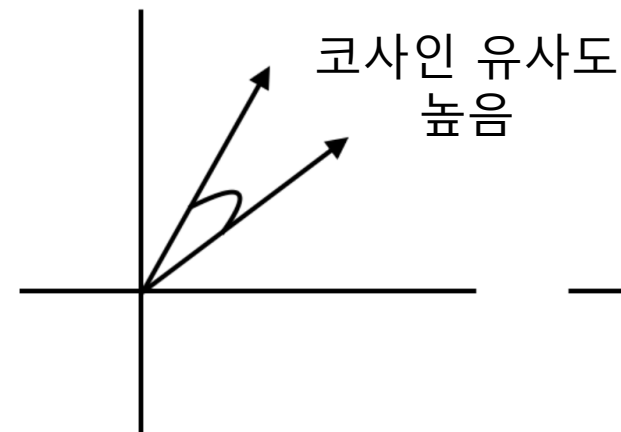
다음 단계는 **사용자 프로필과 다른 상품들의 상품 프로필을 매칭**하는 단계입니다



사용자 프로필 벡터  $\vec{u}$ 와 상품 프로필 벡터  $\vec{v}$

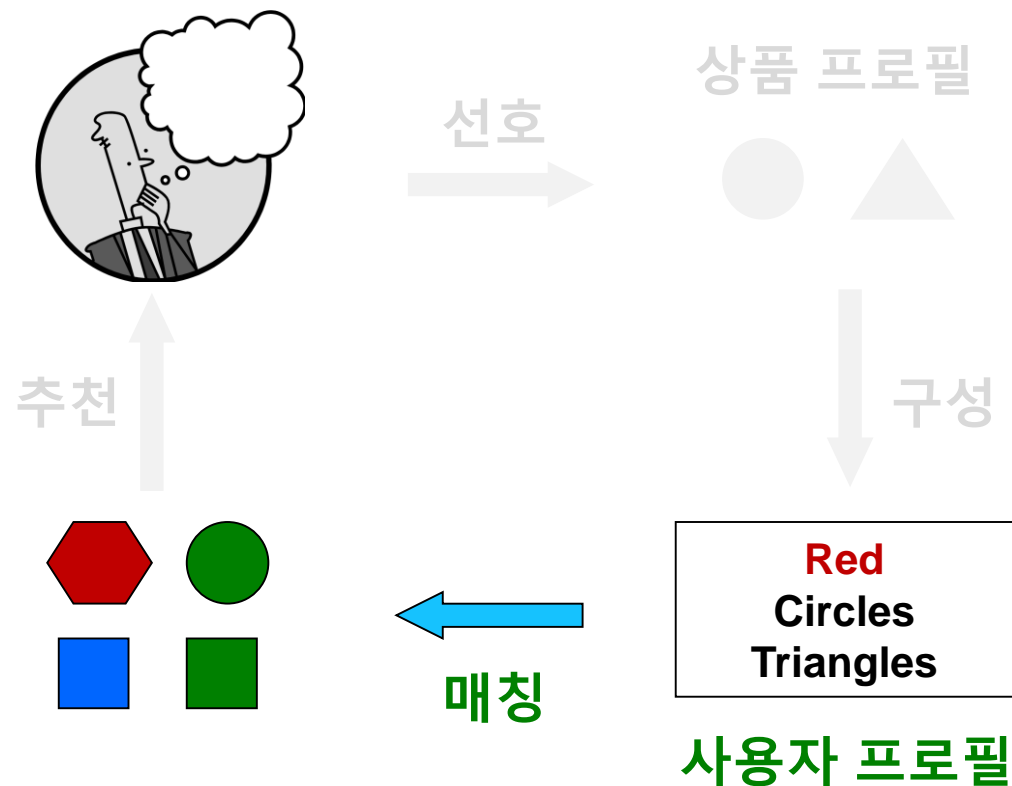
코사인 유사도  $\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|}$  를 계산합니다

즉, 두 벡터의 사이각의 코사인 값을 계산합니다



## 2.1 내용 기반 추천시스템의 원리

다음 단계는 사용자 프로필과 다른 상품들의 상품 프로필을 매칭하는 단계입니다



사용자 프로필 벡터  $\vec{u}$ 와 상품 프로필 벡터  $\vec{v}$

코사인 유사도  $\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|}$  를 계산합니다

즉, 두 벡터의 사이각의 코사인 값을 계산합니다

코사인 유사도가 높을 수록,  
해당 사용자가 과거 선호했던 상품들과  
해당 상품이 유사함을 의미합니다

## 2.1 내용 기반 추천시스템의 원리

마지막 단계는 **사용자에게 상품을 추천**하는 단계입니다



## 2.2 내용 기반 추천시스템의 장단점

---

내용 기반 추천시스템은 다음 **장점**을 갖습니다

- (1) 다른 사용자의 구매 기록이 필요하지 않습니다
  - (2) **독특한 취향의 사용자**에게도 추천이 가능합니다
  - (3) **새 상품**에 대해서도 추천이 가능합니다
  - (4) 추천의 **이유**를 제공할 수 있습니다
- 예시: 당신은 로맨스 영화를 선호했기 때문에, 새로운 로맨스 영화를 추천합니다



## 2.2 내용 기반 추천시스템의 장단점

---

내용 기반 추천시스템은 다음 단점을 갖습니다

- (1) 상품에 대한 **부가 정보가 없는 경우**에는 사용할 수 없습니다
- (2) **구매 기록이 없는 사용자**에게는 사용할 수 없습니다
- (3) 과적합(Overfitting)으로 지나치게 **협소한 추천**을 할 위험이 있습니다

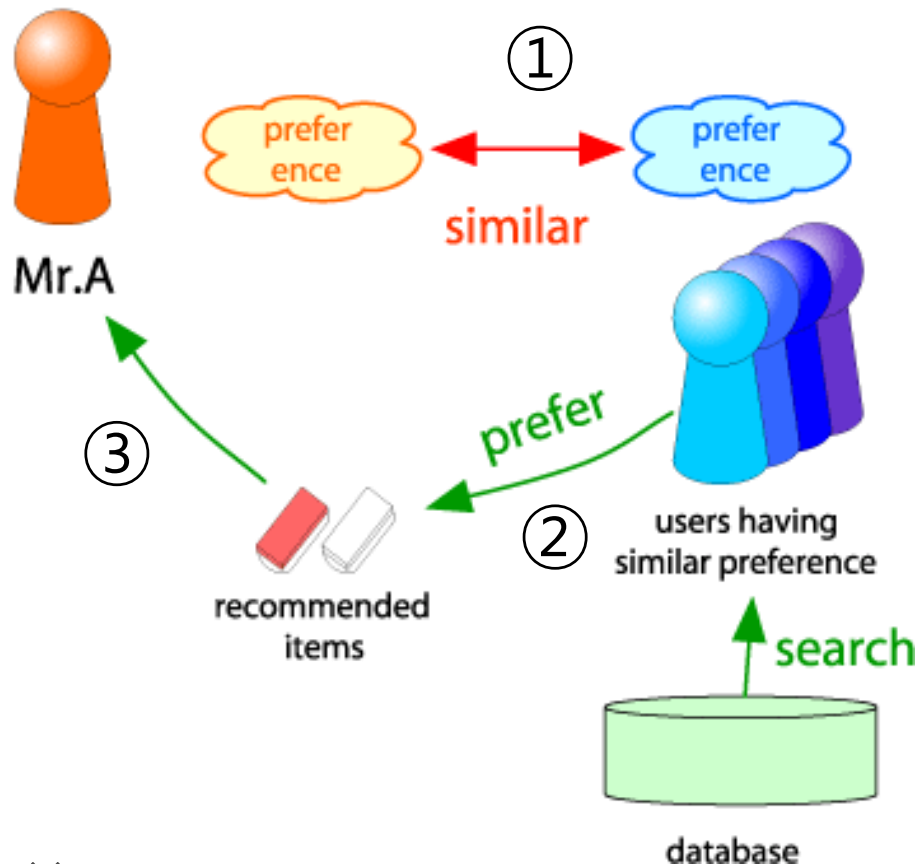
# 3. 협업 필터링 추천시스템

3.1 협업 필터링의 원리

3.3 협업 필터링의 장단점

## 3.1 협업 필터링의 원리

사용자-사용자 협업 필터링은 다음 세 단계로 이루어집니다



추천의 대상 사용자를  $x$ 라고 합시다

우선  $x$ 와 유사한 취향의 사용자들을 찾습니다

다음 단계로 유사한 취향의 사용자들이  
선호한 상품을 찾습니다

마지막으로 이 상품들을  $x$ 에게 추천합니다

## 3.1 협업 필터링의 원리

사용자-사용자 협업 필터링의 핵심은 유사한 취향의 사용자를 찾는 것입니다  
그런데 취향의 유사도는 어떻게 계산할까요?

	반지의제왕	겨울왕국	라푼젤	해리포터	미녀와 야수
지수	4	1	2	5	?
제니	5	1	?	4	2
로제	2	5	5	?	4

위 예시는 사용자 별 영화 평점입니다

'?'는 평점이 입력되지 않은 경우를 의미합니다

지수와 제니의 취향이 유사하고, 로제는 둘과 다른 취향을 가진 것을 알 수 있습니다



## 3.1 협업 필터링의 원리

취향의 유사성은 **상관 계수(Correlation Coefficient)**를 통해 측정합니다

사용자  $x$ 의 상품  $s$ 에 대한 평점을  $r_{xs}$ 라고 합니다

사용자  $x$ 가 매긴 평균 평점을  $\bar{r}_x$ 라고 합니다

사용자  $x$ 와  $y$ 가 공동 구매한 상품들을  $S_{xy}$ 라고 합니다

사용자  $x$ 와  $y$ 의 **취향의 유사도**는 아래 수식으로 계산합니다

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

즉, 통계에서의 상관 계수(Correlation Coefficient)를 사용해 취향의 유사도를 계산합니다

## 3.1 협업 필터링의 원리

예시에서 **취향의 유사도**를 계산해봅시다

	반지의제왕	겨울왕국	라푼젤	해리포터	미녀와 야수
지수	4	1	2	5	?
제니	5	1	?	4	2
로제	2	5	5	?	4

지수와 로제의 취향의 유사도는 0.88 입니다

$$\frac{(4-3)(5-3) + (1-3)(1-3) + (5-3)(4-3)}{\sqrt{(4-3)^2 + (1-3)^2 + (5-3)^2} \sqrt{(5-3)^2 + (1-3)^2 + (4-3)^2}} = 0.88$$

즉 둘의 취향은 매우 유사합니다

## 3.1 협업 필터링의 원리

예시에서 **취향의 유사도**를 계산해봅시다

	반지의제왕	겨울왕국	라푼젤	해리포터	미녀와 야수
지수	4	1	2	5	?
제니	5	1	?	4	2
로제	2	5	5	?	4

지수와 로제의 취향의 유사도는 **-0.94** 입니다

$$\frac{(4-3)(2-4) + (1-3)(5-4) + (2-3)(5-4)}{\sqrt{(4-3)^2 + (1-3)^2 + (2-3)^2} \sqrt{(2-4)^2 + (5-4)^2 + (5-4)^2}} = -0.94$$

즉 둘의 취향은 매우 **상**이합니다

## 3.1 협업 필터링의 원리

따라서, 지수의 취향을 추정할 때는 제니의 취향을 참고하게 됩니다

	반지의제왕	겨울왕국	라푼젤	해리포터	미녀와 야수
지수	4	1	2	5	?
제니	5	1	?	4	2
로제	2	5	5	?	4

예를 들면, 지수는 미녀와 야수를 좋아할 확률이 낮습니다

지수와 제니의 취향은 유사하고, 제니는 미녀와 야수를 좋아하지 않았기 때문입니다

## 3.1 협업 필터링의 원리

---

구체적으로 **취향의 유사도를 가중치로 사용한 평점의 가중 평균**을 통해 **평점을 추정**합니다

사용자  $x$ 의 상품  $s$ 에 대한 평점을  $r_{xs}$ 를 추정하는 경우를 생각합시다

앞서 설명한 상관 계수를 이용하여 상품  $s$ 를 구매한 사용자 중에  $x$ 와 **취향이 가장 유사한  $k$ 명의 사용자  $N(x; s)$** 를 뽑습니다

평점  $r_{xs}$ 는 아래의 수식을 이용해 추정합니다

$$\hat{r}_{xs} = \frac{\sum_{y \in N(x; s)} \text{sim}(x, y) \cdot r_{ys}}{\sum_{y \in N(x; s)} \text{sim}(x, y)}$$

즉, 취향의 유사도를 가중치로 사용한 평점의 가중 평균을 계산합니다

## 3.1 협업 필터링의 원리

---

마지막 단계는 **추정한 평점이 가장 높은 상품을 추천**하는 단계입니다

추천의 대상 사용자를  $x$ 라고 합시다

앞서 설명한 방법을 통해,  $x$ 가 아직 구매하지 않은 상품 각각에 대해 평점을 추정합니다

**추정한 평점이 가장 높은 상품**들을  $x$ 에게 추천합니다

## 3.2 협업 필터링의 장단점

---

협업 필터링은 다음 **장점**과 **단점** 있습니다

- (+) 상품에 대한 **부가 정보가 없는 경우**에도 사용할 수 있습니다
- (-) **충분한 수의 평점 데이터가 누적**되어야 효과적입니다
- (-) **새 상품, 새로운 사용자**에 대한 추천이 불가능합니다
- (-) **독특한 취향의 사용자**에게 추천이 어렵습니다



# 4. 추천 시스템의 평가

4.1 데이터 분리

4.2 평가 지표

## 4.1 데이터 분리

추천 시스템의 정확도는 어떻게 평가할까요?

Diagram illustrating a recommendation system evaluation matrix. The matrix is a 10x6 grid of blue cells. The horizontal axis is labeled '상품' (Product) and the vertical axis is labeled '사용자' (User). The grid contains numerical ratings (1, 2, 3, 4, 5) in some cells, while others are empty.

	상품					
사용자	1	3	4			
		3	5			5
			4	5		5
			3			
			3			
	2			2		2
					5	
		2	1			1
		3			3	
	1					

## 4.1 데이터 분리

먼저 데이터를 **훈련(Training) 데이터**와 **평가(Test) 데이터**로 분리합니다

Diagram illustrating data separation for a recommendation system. The data is organized into a grid where rows represent users (사용자) and columns represent products (상품). The grid is divided into two sections: Training Data (훈련 데이터, blue) and Evaluation Data (평가 데이터, grey).

	상품					
사용자	1	3	4			
		3	5			5
			4	5		5
			3			
			3			
	2			2		2
					5	
		2	1			1
		3			3	
	1					

Annotations:

- 훈련 데이터 (Training Data): Points to the blue-shaded cells (top 5 rows).
- 평가 데이터 (Evaluation Data): Points to the grey-shaded cells (bottom 5 rows).

## 4.1 데이터 분리

평가 데이터는 주어지지 않았다고 가정합니다

Diagram illustrating data separation for a recommendation system. The matrix is divided into training data (blue) and evaluation data (red).

**상품** (Items) and **사용자** (Users) are labeled on the axes.

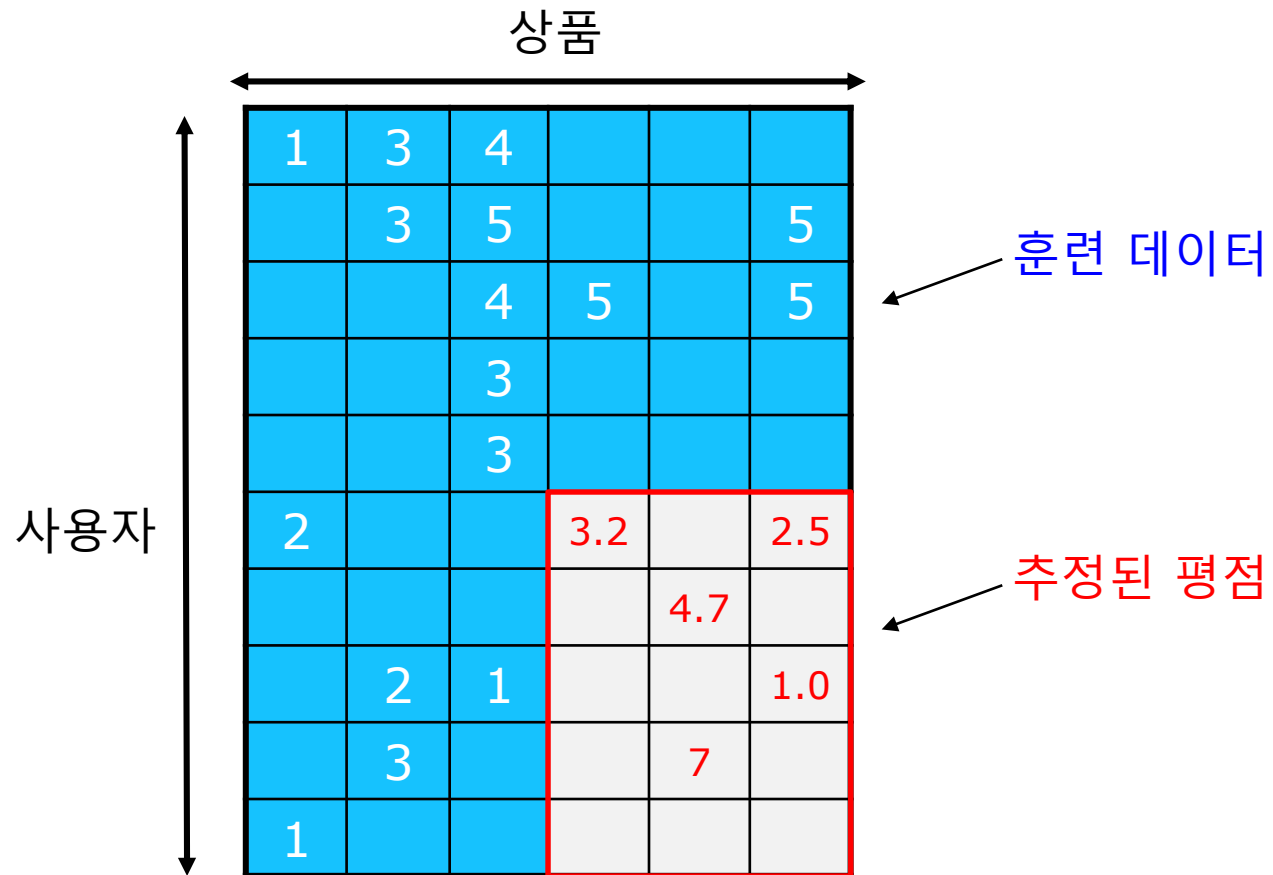
**훈련 데이터** (Training Data) is the blue-shaded area.

**평가 데이터** (Evaluation Data) is the red-shaded area.

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

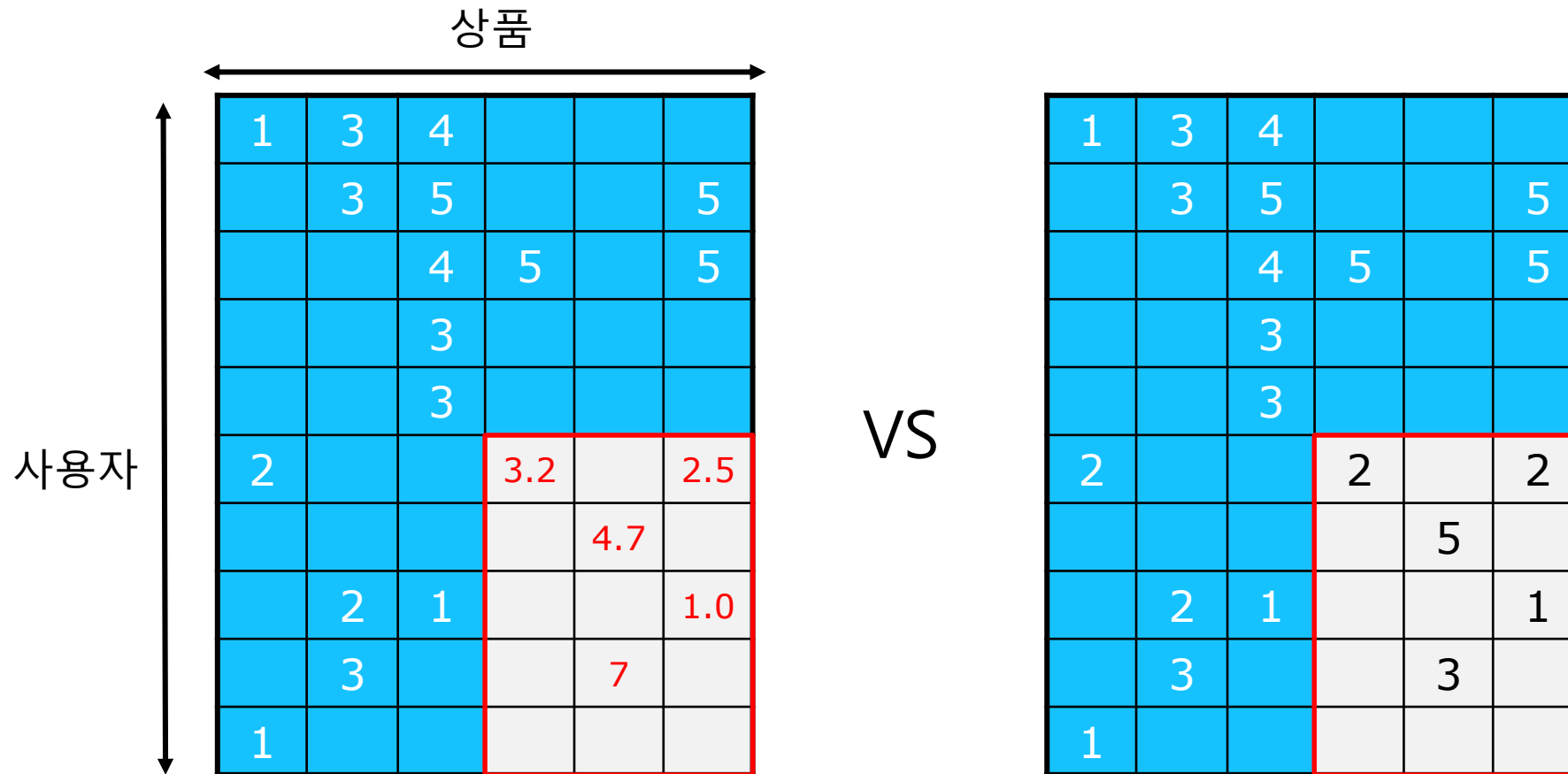
## 4.1 데이터 분리

훈련 데이터를 이용해서 가리워진 **평가 데이터의 평점을 추정**합니다



## 4.1 데이터 분리

추정한 평점과 실제 평가 데이터를 비교하여 오차를 측정합니다



## 4.2 평가 지표

추정한 평점과 실제 평가 데이터를 비교하여 오차를 측정합니다

오차를 측정하는 지표로는 평균 제곱 오차(Mean Squared Error, MSE)가 많이 사용됩니다

평가 데이터 내의 평점들을 집합을  $T$ 라고 합니다

평균 제곱 오차는 아래 수식으로 계산합니다

$$\frac{1}{|T|} \sum_{r_{xi} \in T} (r_{xi} - \hat{r}_{xi})^2$$

평균 제곱근 오차(Root Mean Squared Error, RMSE)도 많이 사용됩니다

$$\sqrt{\frac{1}{|T|} \sum_{r_{xi} \in T} (r_{xi} - \hat{r}_{xi})^2}$$

## 4.2 평가 지표

---

이 밖에도 다양한 지표가 사용됩니다

추정한 평점으로 순위를 매긴 후, 실제 평점으로 매긴 순위와의 상관 계수를 계산하기도 합니다

추천한 상품 중 실제 구매로 이루어진 것의 비율을 측정하기도 합니다

추천의 순서 혹은 다양성까지 고려하는 지표들도 사용됩니다



# 5. 실습: 협업 필터링 구현

5.1 데이터 불러오기 및 전처리

5.2 취향의 유사도 계산

5.3 점수 추정

5.4 정확도 평가

## 5.1 데이터 불러오기 및 전처리

---

먼저 실습에서 사용하는 파이썬 라이브러리를 불러옵니다

```
[ ] import numpy as np
import pandas as pd
from sklearn.metrics import mean_squared_error
```

## 5.1 데이터 불러오기 및 전처리

본 실습에서는 100,000개의 평점으로 구성된 **MovieLens 데이터셋**을 사용합니다

```
### Rating Dataset Format ###
```

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

```
### Movie Dataset Format ###
```

```
Columns of Movie Dataset : Index(['movieId', 'title', 'genres'], dtype='object')
```

	movieId	...	genres
0	1	...	Adventure Animation Children Comedy Fantasy
1	2	...	Adventure Children Fantasy
2	3	...	Comedy Romance
3	4	...	Comedy Drama Romance
4	5	...	Comedy

```
[5 rows x 3 columns]
```

## 5.1 데이터 불러오기 및 전처리

---

데이터를 파일에서 읽어옵니다

```
df_ratings = pd.read_csv('./ratings.csv')  
df_movies = pd.read_csv('./movies.csv')
```

## 5.1 데이터 불러오기 및 전처리

---

데이터에 포함된 사용자 수와 영화 수를 확인합니다

```
n_users = df_ratings.userId.unique().shape[0]
n_items = df_ratings.movieId.unique().shape[0]
print("num users: {}, num items:{}".format(n_users, n_items))
```

```
num users: 611, num items:9724
```

## 5.1 데이터 불러오기 및 전처리

사용자(혹은 영화)의 식별자 범위를 0 ~ 사용자 수(혹은 영화 수) - 1로 변경합니다

```
user_dict = dict()
movie_dict = dict()
user_idx = 0
movie_idx = 0
ratings = np.zeros((n_users, n_items))
for row in df_ratings.itertuples(index=False):
    user_id = row[0]
    movie_id = row[1]
    if user_id not in user_dict:
        user_dict[user_id] = user_idx
        user_idx += 1
    if movie_id not in movie_dict:
        movie_dict[movie_id] = movie_idx
        movie_idx += 1
    ratings[user_dict[user_id], movie_dict[movie_id]] = row[2]
```

×

## 5.1 데이터 불러오기 및 전처리

---

### 학습 데이터와 평가 데이터를 분리합니다

```
test = np.zeros_like(ratings)
train = ratings.copy()
for x in range(ratings.shape[0]):
    nonzero_idx = ratings[x, :].nonzero()[0]
    test_ratings = np.random.choice(nonzero_idx,
                                    size=int(len(nonzero_idx)/5),
                                    replace=False)

    train[x, test_ratings] = 0.
    test[x, test_ratings] = ratings[x, test_ratings]
```

## 5.2 취향의 유사도 계산

사용자간 취향의 유사도를 계산합니다

```
normalized_ratings = np.zeros_like(train_ratings)
for i in range(train_ratings.shape[0]):
    nonzero_idx = train_ratings[i].nonzero()[0]
    sum_ratings = np.sum(train_ratings[i])
    num_nonzero = len(nonzero_idx)
    avg_rating = sum_ratings / num_nonzero
    if num_nonzero == 0:
        avg_rating = 0
    normalized_ratings[i, nonzero_idx] = train_ratings[i, nonzero_idx] - avg_rating
```

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$



## 5.2 취향의 유사도 계산

사용자간 취향의 유사도를 계산합니다

```
n_users = ratings.shape[0]
n_items = ratings.shape[1]
similarity = np.zeros((n_users, n_users))
for i in range(n_users):
    for j in range(i+1, n_users):
        sum_i = sum_j = prod = 0
        for k in range(n_items):
            if normalized_ratings[i][k] != 0 and normalized_ratings[j][k] != 0:
                sum_i += pow(normalized_ratings[i][k], 2)
                sum_j += pow(normalized_ratings[j][k], 2)
                prod += normalized_ratings[i][k] * normalized_ratings[j][k]
        if prod != 0:
            similarity[i][j] = prod / sqrt(sum_i) / sqrt(sum_j)
            similarity[j][i] = similarity[i][j]
```

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

## 5.3 점수 추정

유사도를 사용한 가중 평균을 통해, 사용자 - 영화 쌍 각각에 대해 점수를 추정합니다

```
pred = np.zeros(ratings.shape)
for u in range(ratings.shape[0]):
    for i in range(ratings.shape[1]):
        watched_i = ratings[:,i].nonzero()[0]
        watched_i = np.setdiff1d(watched_i, u)
        similarity_u = similarity[u, watched_i]
        similar_idx = np.argsort(similarity_u)[::-1]
        if len(similar_idx) > k:
            similar_idx = similar_idx[:k]
        sum_similarity = np.sum(similarity[u, similar_idx])
        if sum_similarity == 0:
            sum_similarity = 1
        pred[u, i] = np.sum(np.dot(similarity[u, similar_idx],
                                   ratings[similar_idx, i])) / sum_similarity
```

$$\hat{r}_{xs} = \frac{\sum_{y \in N(x;s)} sim(x, y) \cdot r_{ys}}{\sum_{y \in N(x;s)} sim(x, y)}$$

## 5.4 정확도 평가

---

평가 데이터와 추정한 점수를 비교, 평균 제곱 오차(MSE)를 계산합니다

```
pred = pred[test_ratings.nonzero()].flatten()  
actual = test_ratings[test_ratings.nonzero()].flatten()  
mean_squared_error(pred, actual)
```

# 6강 정리

---

## 1. 우리 주변의 추천 시스템

- Amazon.com, 넷플릭스, 페이스북, 유튜브 등

## 2. 내용 기반 추천 시스템

- 장점: 새로운 상품에 대한 추천이 가능
- 단점: 상품에 대한 부가 정보가 있는 경우에만 사용 가능

## 3. 협업 필터링

- 장점: 부가 정보가 없는 경우에도 사용 가능
- 단점: 새로운 상품에 대한 추천이 불가능

## 4. 추천 시스템의 평가

- 학습/평가 데이터 분리, 평균 제곱(근) 오차

## 5. 실습: 협업 필터링 구현