

# Database Project 2

Implementing a Simple Database Application

[영화 예매 Application]

# Overview

- **Python과 MySQL**을 이용한 응용 시스템 구현
  - 영화 예매 시스템
- 주요 기능
  - 영화 정보, 관객 정보 **삽입 / 삭제 / 출력**
  - 관객이 영화를 **예매**, **평점** 부여
  - 관객을 위한 영화 **추천**
- **과제 Due : 6/14(수) 23:59**
  - 하루 Delay 마다 -10%, 3일 Delay 시 0점

# Details (1/3)

- Raw data 제공 (data.csv)
- DB schema는 자유롭게 설계.
  - 주어진 raw data를 기반으로 적합한 DB schema를 설계.
  - csv 파일을 읽어와서 데이터베이스를 초기화하는 것으로 프로그램 시작.



	A	B	C	D	E	F
1	title	director	price	name	age	class
2	Eternal Sunsh	Michel Gondr	10000	Abigail	62	basic
3	Inglourious B	Quentin Tarar	15000	Alexander	66	basic
4	Spider-Man: I	Bob Persichet	25000	Aria	26	basic
5	Django Unch	Quentin Tarar	25000	Ava	63	basic

[data.csv 예시]

## Details (2/3)

- 영화 관리
  - 영화 삽입 / 삭제 / 출력
- 관객 관리
  - 관객 삽입 / 삭제 / 출력
- 영화 예매
  - 관객이 지정한 영화를 예매
- 평점 부여
  - 관객이 **예매한 영화** 중 평점 부여 (1~5점)

# Details (3/3)

- 관객에 적합한 영화 추천
  - Popularity-based
  - Item-based Collaborative Filtering
  - 지정한 관객에게 가장 적합한 영화 N편을 추천.

# Popularity-based Recommendation (1/3)

- 예시 :

- User 1~4, Item 1~4
- 목표 : User4 에게 영화 추천하기
  - 평균 평점 기반 추천
  - 관객 수 기반 영화 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5

# Popularity-based Recommendation (2/3)

- 예시 : 평균 평점 기반 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5
Average	2.67	3	2.5	4.5

- 평균 평점

- User 4가 보지 않은 영화 중 평균 평점이 가장 높은 Item 2 추천

# Popularity-based Recommendation (3/3)

- 예시 : 관객 수 기반 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5
Count	3	3	2	2

- 관객 수

- User 4가 보지 않은 영화 중 관객 수가 가장 많은 Item 2 추천



# Item-based Collaborative Filtering (1/6)

- 예시 :
  - User 1~4, Item 1~4
  - 목표 : User4 에게 가장 적합한 영화 N편을 추천하기
    - 예시에서는  $N=1$ 로 설정

# Item-based Collaborative Filtering (2/6)

- 예시 :

- User 1~4, Item 1~4

- 목표 : User4 에게 가장 적합한 영화 1편을 추천하기.

1. 각 사용자가 부여한 평점 정보를 기반으로 user-item matrix 구축

- Item에 부여된 평점 기반으로 User 4가 아직 평점을 부여하지 않은 item (item2, item3) 중 가장 적합한 item을 추천.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2	?	?	5

[user-item matrix 예시]

# Item-based Collaborative Filtering (3/6)

- 예시 :

- User 1~4, Item 1~4

- 목표 : User4 에게 가장 적합한 영화 1편을 추천하기.

2. 관객이 아직 평점을 부여하지 않은 영화에 대해서는 영화의 평균 평점으로 user-item matrix 완성

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	3	2.5	5

[user-item matrix 예시]

# Item-based Collaborative Filtering (4/6)

- 예시 :

- User 1~4, Item 1~4

- 목표 : User4 에게 가장 적합한 영화 1편을 추천하기.

3. Adjusted Cosine-similarity를 이용해서 User 사이의 유사도를 측정한다.

$$\text{Adjusted cosine similarity} = \frac{\sum_{i=1}^n (A_i - \mu)(B_i - \mu)}{\sqrt{\sum_{i=1}^n (A_i - \mu)^2} \sqrt{\sum_{i=1}^n (B_i - \mu)^2}}$$

$\mu$  = 전체 평균

4. 모든 영화 사이의 similarity를 계산해서 similarity-matrix를 구축한다.

	Item 1	Item 2	Item 3	Item 4
Item 1	1	0.5896	0.6197	-0.3496
Item 2	0.5896	1	0.9462	-0.0972
Item 3	0.6197	0.9462	1	-0.4060
Item 4	-0.3496	-0.0972	-0.4060	1

[similarity matrix 예시]

# Item-based Collaborative Filtering (5/6)

- 예시 :

- User 1~4, Item 1~4

- 목표 : User4 에게 가장 적합한 영화 1편을 추천하기.

5. 아이템 사이의 Similarity를 weight으로 해서 weighted average 값으로 각 영화에 대한 평점을 예측한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	ⓐ	ⓑ	5

$$\textcircled{a} = \frac{0.5896 \times 2 + 0.9462 \times 2.5 + (-0.0972) \times 5}{0.5896 + 0.9462 + (-0.0972)} \approx 2.1261$$

$$\textcircled{b} = \frac{0.6197 \times 2 + 0.9462 \times 3 + (-0.4060) \times 5}{0.6197 + 0.9462 + (-0.4060)} \approx 1.7657$$

[Item1과 Item2에 대한 평점 예측]

# Item-based Collaborative Filtering (6/6)

- 예시 :

- User 1~4, Item 1~4
- 목표 : User4 에게 가장 적합한 영화 1편을 추천하기.

6. 예측된 평점이 가장 높은 Item 1개를 최종 추천한다.

- User 4의 경우 Item2가 Item3보다 높게 예측되었기 때문에 Item2를 추천하게 된다.
- 추천 가능 개수가 추천 영화 개수보다 적을 경우 가능한 개수만큼만 출력한다.
  - 예를 들어, 아래의 경우, N이 3일 경우, 두 편만 출력한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.5
User 2	1	3	2.5	4.5
User 3	2.67	2	1	4
User 4	2	2.13	1.77	5

[User4에 대한 최종 예측]

# MySQL Connector Python Tutorial

# MySQL Connector Python

- Python으로 MySQL 연동을 해주는 패키지
- 설치
  - **`pip install mysql-connector-python`**
- Reference
  - <https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html>
- Also, there are other options
  - PyMySQL
  - ...



# Python – MySQL 연동 예시 코드

```
from mysql.connector import connect

connection = connect(
    host='astronaut.snu.ac.kr',
    port=7000,
    user='TA1',
    password='TA1',
    db='TA1',
    charset='utf8')

with connection.cursor(dictionary=True) as cursor:
    cursor.execute("select * from stage")
    result = cursor.fetchall()
    print(result)

connection.close()
```

# Connect to Server DB

- 접속 정보
  - DB 서버 주소
    - astronaut.snu.ac.kr (포트: 7000)
  - ID = PW = DB name
    - DB[학번]
    - EX) DB2023\_12345