

# DB 프로젝트 과제

12191557 김동환

## 1. 구조

- **MVC** (모델-뷰-컨트롤러)라는 소프트웨어 디자인 패턴을 기반으로 제작되어진 코드입니다.
  - 모델 : 앱이 포함해야할 데이터가 무엇인지를 정의
  - 뷰 : 앱의 데이터를 보여주는 방식을 정의
  - 컨트롤러 : 앱의 사용자로부터의 입력에 대한 응답으로 모델 및/또는 뷰를 업데이트하는 로직

## 2. 웹페이지의 구성

- ① userID를 입력받는 form
- ② userID와 맵핑되는 사용자정보
- ③ 같은 직업을 가진 사람들이 좋아하는 영화 Top 10
- ④ 같은 연령대를 가진 사람들이 좋아하는 영화 Top 10
- ⑤ 같은 성별을 가진 사람들이 좋아하는 영화 Top 10
- ⑥ 피어슨 상관계수를 이용한 추천 알고리즘에 의한 영화 Top 10
- ⑦ 코사인 유사도를 이용한 추천 알고리즘에 의한 영화 Top 10

### 3. 코드 파일 설명

- title.jsp : 사용자에게 웹 페이지를 제공하며, 사용자의 ID를 입력 받아 해당 사용자의 정보 및 추천 영화 리스트를 화면에 출력하는 역할을 합니다.
- titleController.java : JSP 페이지로부터 request를 받아 적절한 결과를 리턴해줍니다. 사용자의 요청에 따라 titleService.java의 메소드를 호출하고, 그 결과를 ModelAndView 객체를 통해 JSP에 전달합니다. ModelAndView는 뷰와 모델을 함께 전달할 수 있는 객체로서, 여기서 뷰는 클라이언트에게 보여줄 JSP 페이지를, 모델은 해당 페이지에 전달할 데이터를 의미합니다.
- titleService.java : DAO를 통해 데이터를 가져온 후, 그 결과를 Controller에게 전달합니다. 추천 알고리즘이 적용된 데이터를 얻기 위하여 파이썬 스크립트를 실행하여, 추천 알고리즘을 적용된 결과를 DAO를 사용하여 받아옵니다.
- titleDAO.java : MyBatis를 사용하여 SQL 쿼리를 실행하고, 데이터베이스에 접근하여 데이터를 조회하는 역할을 하는 Data Access Object입니다. 이 때, DTO를 사용하여 필요한 정보를 가져오는데, DTO를 사용하는 이유는 데이터 접근 로직과 비즈니스 로직을 분리하기 위함입니다.
- titlemapper.xml : MyBatis mapper 파일로, 이 파일에 정의된 SQL 쿼리를 DAO에서 호출하여 사용합니다.
- userDTO.java : 사용자 정보를 전달하는 Data Transfer Object입니다.
- titleDTO.java : 영화 정보를 전달하는 Data Transfer Object입니다.
- recommend.py : titleService.java에서 호출되며, 데이터베이스와 연동 후, 데이터베이스에 저장된 데이터를 사용합니다. surprise library에서 제공하는 API를 이용하여 피어슨 상관계수와 코사인 유사도에 따른 사용자별 추천영화목록을 생성하고, 그 결과를 데이터베이스에 저장합니다.

#### 4. 추천 알고리즘에 의한 영화를 구하는 과정

- titleService.java에서, java의 ProcessBuilder를 통해 recommend.py를 실행시키는 과정을 아래 주석을 통하여 표시하였습니다. recommend.py를 통해 피어슨 상관 계수와 코사인 유사도에 따른 사용자별 추천영화 목록을 생성하고, 그 결과를 데이터베이스에 저장합니다.

```
public List<titleDTO> selectRecommendedMovies(String userId) {
    // 파이썬 스크립트 실행하는 코드
    List<titleDTO> recommendedMovies = new ArrayList<>();

    try {
        // 파이썬 스크립트 경로 // 아나콘다 환경에서 파이썬을 실행시켜야함.
        String scriptPath = "C:/Users/wlffj/Miniconda3/python.exe";

        // 파이썬 스크립트 실행 명령어
        String command = scriptPath + "
C:/Users/wlffj/IdeaProjects/DBproject1/src/main/Python/recommend.py " + userId;

        // ProcessBuilder 를 사용하여 파이썬 스크립트 실행
        ProcessBuilder builder = new ProcessBuilder("cmd.exe", "/c", command);
        builder.inheritIO();

        Process process = builder.start();

        // 프로세스가 종료될 때까지 대기. 여기서 자바가 부모프로세스, 파이썬이
        자식프로세스이다.
        int exitCode = process.waitFor();

        if (exitCode != 0) { // 에러 발생시의 예외처리부분들
            System.out.println("error occurred");
        }

    } catch (IOException | InterruptedException e) {
        System.out.println("error occurred");
        e.printStackTrace();
    }

    // DAO 를 통해 DB 에서 쿼리를 적용하여 원하는 데이터를 얻어옵니다.
    // 그리고 그 데이터는 titleController 에 리턴되어 modelAndView 객체에 추가되게됩니다.
    recommendedMovies = titleDAO.selectRecommendedMovies(userId);
    return recommendedMovies;
}
```

- recommend.py에서 피어슨상관계수를 적용하여 추천영화목록을 구하는 과정을 아래 주석을 통하여 표시하였습니다. 구현은 surprise라이브러리의 API를 사용하여 구현하였습니다.

```
def recommend_movies(user_id):  
    # 데이터 로드  
    data = load_data_from_db()  
  
    # KNN 알고리즘 설정  
    # 이웃의 최대개수=40, 메모리 기반 협업 필터링 알고리즘, 유저기반 pearson 상관계수 적용  
    algo = KNNWithMeans(k=40, sim_options={'name': 'pearson', 'user_based': True})  
  
    # 알고리즘 학습  
    trainset = data.build_full_trainset()  
    algo.fit(trainset)  
  
    # 사용자가 평가하지 않은 영화를 찾아서 데이터셋 형성.  
    testset = trainset.build_anti_testset(fill=False)  
    user_testset = [test for test in testset if test[0] == user_id]  
  
    # 예측 수행  
    predictions = algo.test(user_testset)  
  
    # 예측 평점 순으로 정렬 (익명함수 람다를 이용하여 predictions 리스트의 원소의  
    # est 속성값을 기준으로 내림차순정렬) (x를 입력으로 하여 x.est를 리턴)  
    predictions.sort(key=lambda x: x.est, reverse=True)  
  
    # 상위 10개만 선택  
    top10 = predictions[:10]  
  
    # 결과를 DB에 저장  
    result = [{'movieId': pred.iid, 'predictedRating': pred.est} for pred in top10]  
    save_to_db(user_id, result)
```

- 코사인 유사도를 적용하였을 때는 아래 코드와 같습니다.

```
# 이웃의 최대개수=40, 메모리 기반 협업 필터링 알고리즘, 유저기반 cosine 유사도 적용  
algo = KNNWithMeans(k=40, sim_options={'name': 'cosine', 'user_based': True})
```