Pusan National University
Computer Science and Engineering
Technical Report 2025-09

웹 크롤링 기반 블루레이 추천 서비스

202055539 박덕형
202055549 박태준
202055564 안형찬

조준수

목 차

1. 서론

1.1 연구 배경

블루레이는 영화와 같은 HD 비디오를 저장하기 위한 디지털 광 기록 방식의 저장매체로, 전 세계적으로 CD나 DVD를 대체할 고화질 매체로 주목받고 있다.

-블루레이 플레이어와 블루레이-

블루레이는 그 특성과 상징성 덕분에, 영화를 애호가들 사이에서는 블루레이를 수집하는 것이 하나의 취미로 자리 잡았다.

하지만 국내에서의 인지도는 현저히 낮아, 블루레이 관련 콘텐츠를 제공하는 서비스가 부족하고 정보를 얻기 어렵다. 때문에 관련 취미를 가진 사람은 해외 사이트를 이용하거나 여러 서적 사이트를 일일이 찾아다니며 정보를 수집하는 수고를 들여야 한다.

이렇듯 블루레이에 대한 낮은 접근성을 개선하는데 기여할수 있도록, 웹 크롤링과 LLM 기술을 활용해 블루레이 관련 데이터를 수집,정제하여 보다 풍부하고 정확한 정보를 제공하며, 더 나아가 개인화된 추천 리스트를 제공하는 서비스를 구축하고자 한다.

1.2 기존 문제점

기존의 블루레이 판매 사이트들은 대부분 단순히 상품 정보를 나열하거나 판매량 기반의 단편적인 추천만을 제공할 뿐이다. 이 때문에 이용자의 세부적인 취향이나 구매 패턴을 반영한 맞춤형 추천은 사실상 이루어지지 않는다.

게다가 블루레이 상품과 영화 작품 간 매핑 기준이 통일되어 있지 않아 동일 타이틀이라도 4K, 리마스터, 스틸북, 한정판 등 다양한 에디션이 분산적으로 소개된다. 결국 사용자는 자신에게 맞는 상품을 찾기 위해 여러 판매처와 커뮤니티를 직접 탐색해야 하며, 원하는 정보를 얻기까지 과도한 시간과 노력이 소요된다.

이러한 문제들은 국내 시장에서 블루레이의 인지도가 낮고 관련 정보 제공이 부족하다는 현실과 맞물려, 결과적으로 국내 블루레이 시장을 더욱 제한적이고 폐쇄적인 구조로 만들고 있다.

1.3 연구 목표

본 프로젝트의 목표는 사용자 취향을 반영한 블루레이 추천 서비스를 제공하고, 다양한 블루레이 구매 사이트의 정보를 체계적으로 정리하여 제시함으로써, 국내 사용자들이 블루레이 상품을 탐색하고 선택하는 과정을 보다 편리하게 만드는 데 있다.

2. 연구 배경

2.1 물리 미디어와 수집 시장

블루레이는 화질·음질·부가영상·패키징(스틸북 등) 측면에서 높은 수집 가치를 지니며 전 세계적으로 DVD를 대체했으나, 국내에서는 정보 단절과 파편화로 탐색 비용이 높다. 이러한 환경은 작품–에디션 간 차별화된 스펙과 희소성 정보를 함께 고려하는 큐레이션과 개인화 추천의 필요성을 증대시킨다.

-한정판 블루레이-

2.2 AI 기술의 발전

정제된 블루레이 데이터를 얻기 위해서는 제휴사의 지원이나 대규모 인력이 필요한 분류 작업이 필요하지만, 최근 LLM 기술의 발달로 인해서 다양하고 큰 규모의 데이터를 처리하고정제하는데 필요한 노력이 줄어들었다. 이에 LLM모델을 통하여서 데이터를 정제하는데 다양하게 사용되고 있다.

3. 연구 내용

3.1 블루레이 추천 시스템

3.1.1 데이터 축적블루레이 추천 서비스 구현

본 프로젝트에서는 블루레이 상품 정보를 다양한 온라인 서점(예: Yes24, 알라딘, 교보문고 등)으로부터 크롤링하여 수집한 뒤, OpenAI API를 활용해 정형화된 JSON 데이터로 정제하였다. 크롤링 대상이 되는 영화 타이틀은 무비렌즈(MovieLens) 데이터셋을 활용하였다. 전체 과정은 크게 데이터 크롤링, 텍스트 전처리, 프롬프트 설계, OpenAI API 기반 정제, 배치 처리 최적화의 단계로 나눌 수 있다.

3.1.1.1 웹크롤링

requests와 BeautifulSoup 라이브러리를 사용하여 각 사이트의 HTML 구조를 분석한 후, 상품명, 가격, 저자/출연진, 발매일 등의 기본적인 정보를 추출하였다. 사이트마다 HTML 구조가 상이하기 때문에, Yes24, 알라딘등 다양한 사이트에 대해 별도의 CSS Selector를 지정하여 크롤링 로직을 구현하였다.

특히 일부 사이트의 경우 상세 페이지에서만 제공되는 이미지 URL을 가져오기 위해 한번 더 접근이 필요했고, ThreadPoolExecutor를 활용한 멀티스레드 방식으로 병렬 요청을 수행하였다. 이를 통해 이미지 로딩 시간을 단축하고 전체 크롤링 속도를 개선하였다.

3.1.1.2 프롬프트 설계

OpenAI API 호출 시, 모델이 정확하게 JSON 형식으로 응답하도록 하기 위해 프롬프트를 세심하게 설계하였다. 구체적으로 다음과 같은 요구사항을 명시하였다:

- 출력은 반드시 JSON 배열 형식일 것
- 각 객체는 movie_name, region_code, resolution, limited_edition 네 개의 필드를 포함할 것
- 특정 정보가 없을 경우 "N/A" 또는 false로 처리할 것

또한, Few-Shot Prompting 기법을 활용하여 올바른 JSON 예시를 함께 제공함으로써 모델이 안정적으로 필드를 매핑하도록 유도하였다. 이 과정에서 모델이 불필요한 설명 문장을 출력하지 않도록 “Return JSON only”를 반복적으로 강조하였다.

key	Value
원본 데이터 (크롤링 결과)	Toy Story (토이 스토리) (1995) (한글무자막)(4K Ultra HD + Blu-ray + Digital Code)
movie_name	Toy Story (토이 스토리) (1995)
region_code	N/A
resolution	4K Ultra HD
limited_edition	false

3.1.1.3 모델 선택 및 호출 전략

본 프로젝트에서는 gpt-3.5-turbo 모델을 사용하였다. 그 이유는 다음과 같다:

비용 효율성: GPT-4 계열보다 낮은 비용으로 대량의 데이터 처리 가능

토큰 처리량: 블루레이 타이틀 데이터는 길이가 다양하고 부가 설명이 많아 토큰 수가 빠르게 증가하는데, GPT-3.5-turbo는 이러한 대규모 입력 처리에 적합

JSON 구조화 성능: 단순 텍스트 분류보다는 구조화된 데이터 변환이 목적이므로, 고성능 GPT-4 대신 GPT-3.5를 선택하여 속도와 비용을 절충

단일 요청에 많은 타이틀을 입력하면 응답 시간이 길어지고 JSON 구조가 깨질 위험이 있기 때문에 배치 처리(batch processing) 방식을 적용하였다. 즉, 일정 개수(예: 20~30개)의 타이틀을 묶어 한 번의 API 호출로 처리하고, 이를 반복하여 전체 데이터셋을 소화하였다.

3.1.1.4 최종 데이터 구성

API 호출 결과로 반환된 JSON 객체는 원본 크롤링 데이터와 결합되어 최종 데이터셋을 형성한다. 즉, OpenAI가 정제해준 movie_name, region_code, resolution, limited_edition 필드에 크롤링 단계에서 추출한 price, link, img, auth, date 등의 부가 정보를 병합하였다.

최종적으로 수집된 데이터는 다음과 같은 형식을 갖는다:

Key	Value
movie_name	Toy Story (1995)
region_code	지역코드1
resolution	N/A
limited_edition	false
price	97400
link	https://www.yes24.com/product/goods/13692336
img	https://image.yes24.com/momo/TopCate0010/hani/L_624626.jpg
auth	Tom Hanks,Tim Allen
date	2021년 05월
id	1

3.1.2 추천 알고리즘 구현

3.1.2.1 MF

- 데이터 전처리
  User-Movie 평점 행렬은 희소성이 98.31%로 대부분의 평점이 결측값이다. 단순히 평균값으로 채우는 방식 대신, SVD를 반복적으로 적용하여 결측값을 예측하고 업데이트하는 방식을 사용한다. 초기에는 전체 평균으로 채우고, SVD 분해와 재구성을 반복하며 예측값의 변화가 특정 임계값 이하로 줄어들면 멈춘다.

- Matrix Factorization
  결측값이 채워진 사용자-영화 행렬에 TruncatedSVD를 적용하여 행렬 분해를 수행한다. 결과적으로 두 개의 작은 행렬, 즉 사용자 잠재 요인 행렬과 영화 잠재 요인 행렬이 생성된다. 이 잠재 요인들은 사용자의 취향이나 영화의 특성을 나타내는 추상적인 벡터이다. 이 때 잠재 요인의 개수를 50개로 설정하였다.

- 개인화 추천 로직
  장르 보너스: 사용자가 선호하는 장르의 영화에 가산점을 부여한다.
  인기도 페널티: 너무 많은 사람이 평가한 인기 영화(예: 300회 이상 평가)는 주류 영화일 가능성이 높으므로 취향을 반영하기 위해 감점을 준다.

3.1.2.2 콘텐츠 기반 필터링

사용자 리뷰 이력과 블루레이 판매 메타데이터의 유사도를 활용해, 사용자 취향에 맞는 블루레이를 추천한다. 앞서 MF로 추출한 영화 후보에 대해 "해당 영화의 최적 블루레이"를 순위화하는 단계로도 동작한다.

추천을 하기 위해서 블루레이 속성을 정교하게 벡터화해야 한다. 각 판매 항목에 대해 화질/형식(예: 4K, 3D, Blu-ray, DVD), 리전 코드, 판매 사이트(yes24, aladin, kyobo), 한정판 여부, 가격대와 같은 메타데이터를 토큰으로 추출한 뒤, 이를 TF-IDF로 표현해 블루레이 벡터를 만든다. 여기서 TF는 해당 블루레이의 빈도를 의미하는데 너무 큰 값이 튀지 않도록 로그 스케일(1+log tf)을 사용한다. IDF는 해당 블루레이의 희소성을 의미하는데, 모든 항목을 통틀어 드물고 구분력이 있는 특징일수록 가중치를 크게 줌으로써 흔하지 않은 특징의 중요성을 높여야 하므로 필요하다. 이 둘을 곱하여 빈도가 높고 구분력이 좋은 특징을 찾아내는 것이다. 이렇게 추출된 블루레이 벡터들을 행 단위 L2 정규화로 항목 간 스케일을 맞춘다.

이렇게 계산해낸 행렬들에서 벡터들 간 코사인 유사도를 계산하여 비슷한 벡터들을 찾아낸다. 행 정규화된 항목 벡터와 사용자 프로필의 내적이 곧 유사도 점수가 되며, 임계값 이상 후보를 모아 점수 내림차순으로 정렬해 반환한다.

3.1.2.3 매핑 알고리즘

본 프로젝트에서는 블루레이에 대해서만 리뷰를 남길 수 있지만 MF에서는 영화 평점이 필요하다. 그렇기 때문에 블루레이 평점과 영화 평점을 매핑해 주는 알고리즘이 필요하다.

매핑 과정은 블루레이 리뷰 평점에 포함된 영화 특성(장르, 러닝타임 등) 및 블루레이 특성(화질, 가격, 한정판, 지역, 사이트 등)으로 인한 보너스/패널티를 제거해 영화 자체에 대한 평점을 도출해 낸다. 결과는 MF 학습·평가에 바로 활용되며, 핵심은 블루레이 속성 편향을 걷어내 공정한 영화 선호도를 얻는 것이다.

공식은 다음과 같다. $y=\alpha+\beta x+i\sum i\gamma \cdot featurei$. x는 블루레이 평점이고 α는 절편이다. 또한 $\beta\gamma i$는 가중치이다. 최종 y는 영화 평점이다. α, β, γi는 파인튜닝으로 조절한다.

3.2 블루레이 추천 플랫폼

3.2.1 서비스 설명

서비스 목적: 본 서비스는 온라인에 분산된 블루레이 판매 정보와 영화 메타데이터를 통합하여 신뢰 가능한 사용자 리뷰와 개인화 추천을 제공하는 웹 서비스이다. 사용자는 작품의 기본 정보부터 에디션·스펙·가격까지 한 화면에서 비교·탐색할 수 있으며, 축적된 선호 데이터를 활용한 추천을 통해 탐색 시간을 단축하고 선택의 만족도를 높일 수 있는 사용 경험을 제공받는다. 또한 웹 크롤링과 정제 파이프라인을 통해 최신 판매 정보를 지속적으로 반영함으로써 정보의 정확성과 접근성을 동시에 강화하는 것을 지향한다

핵심 기능:

- 회원가입과 로그인은 JWT 기반 인증 체계로 구성되어 세션 의존도를 낮추고 모듈 간 일관된 권한 검증을 보장한다.
- 영화 목록·검색·상세 보기 기능은 장르, 출시연도 등 메타데이터를 활용한 정교한 필터링과 풍부한 작품 정보를 제공한다
- 블루레이 판매 정보 검색은 판매처별 상품명, 에디션(4K, 리마스터, 스틸북 등), 가격과 같은 핵심 스펙을 통합 노출하여 분산된 정보를 한 곳에서 비교 가능하게 한다
- 리뷰 작성·조회 기능은 사용자 의견을 구조화된 형태로 축적하여 작품·에디션 선택 시 참고 가능한 정성·정량 정보를 함께 제공한다
- 개인화 추천은 사용자–영화 선호 추정과 블루레이 메타 특성을 결합한 하이브리드 알고리즘을 통해 취향 적합도가 높은 상품을 우선적으로 제시한다.

3.2.2 개발 도구

FrontEnd	Vite, React
BackEnd	Spring, MySQL, Nginx
Infra	Docker, AWS EC2
LLM	GPT-3.5-turbo

3.2.3 프론트 엔드

블루레이 추천 시스템의 프론트엔드는 Vite와 React JS를 활용하여 구현되었다.

Vite는 개발 과정에서 반복적인 UI 수정과 기능 테스트를 빠르게 수행할 수 있도록 지원하여, 초기 개발 단계에서 효율성을 높였다. React는 컴포넌트 기반 설계를 통해 온보딩, 추천 리스트, 대시보드, 상세 페이지 등 복잡한 화면을 모듈화하고 재사용할 수 있게 하여, 유지보수와 확장성을 강화하였다. 또한, 상태 관리와 라우팅 기능을 적용하여 사용자 데이터와 화면 전환을 효율적으로 처리할 수 있었다.

사용자는 온보딩 페이지에서 블루레이 선호도를 선택하고, 추천 리스트와 대시보드를 통해 개인화된 블루레이 정보를 확인할 수 있으며, 상세 페이지에서 추가 정보를 조회할 수 있다. 로그인과 회원 관리 기능을 통해 추천 경험을 개인화하고, 선택한 블루레이를 관리할 수 있도록 하여, 사용자가 편리하게 탐색하고 구매할 수 있는 웹 환경을 제공한다.

3.2.4 백엔드

3.2.4.1 서버 측 애플리케이션

본 프로젝트는 AWS EC2 상에서 다수의 Docker 컨테이너로 웹 프론트엔드, 백엔드 API, 데이터 수정·정제 모듈, 추천 엔진, NGINX 프록시 등을 분리 배포하여 모듈별 독립 운영과 장애 격리를 달성한다. 컨테이너 기반 구성은 기능 단위 확장과 롤링 재배포를 단순화하며, 서비스 부하에 따라 특정 모듈만 선별적으로 수평 확장할 수 있어 서버 간 확장성이 한층 용이하다. 아울러 내부 네트워크를 통해 모듈 간 통신을 표준화하고 로그·메트릭 수집을 일원화함으로써 운영 관측성과 유지보수성을 동시에 확보한다.

본 프로젝트의 Nginx는 Reverse Proxy와 정적 리소스 서버의 이중 역할을 수행하여, 단일 진입점에서 프론트엔드와 백엔드 트래픽을 일관되게 제어하는 게이트웨이로 동작한다. 이를 통해 정적 자원 서빙과 API 프록시를 분리된 모듈 없이 처리하며, 네트워크 경로를 단순화하고 운영 관리 포인트를 축소한다.

API 요청은 /api/ 접두어를 기준으로 분기되어 내부 서비스 spring-api:8080/api로 Reverse Proxy되며, 프론트엔드와 백엔드가 동일 도메인 하에서 통신하도록 구성된다. 이 구성은 백엔드 실제 호스트와 포트를 외부로 노출하지 않으면서도 경로 기반 라우팅으로 서비스 모듈을 유연하게 교체·확장할 수 있게 한다.

Nginx는 요청/응답에 대한 액세스 로그와 오류 로그를 관리하여 트래픽 패턴과 장애 상황을 가시화하고, 문제 원인 분석 및 용량 계획 수립에 활용한다.

Spring Boot API로는 RESTful 규약을 따르는 JSON 기반 통신을 통해 프론트엔드와 내부 추천·정제 모듈을 연결하는 핵심 게이트웨이로 동작한다. 컨테이너화와 Reverse Proxy 구성(Nginx) 하에서 일관된 엔드포인트 체계를 유지하며, 서비스 전반의 트래픽을 안정적으로 수용하도록 설계한다.

도메인은 영화, 판매(블루레이 상품), 사용자, 리뷰, 추천의 다섯 축으로 구성되며, 각 엔티티는 서비스 시나리오(목록·검색·상세·리뷰·추천)와 직결되는 불변식과 제약을 명시한다. 영화–블루레이 간 매핑은 에디션·포맷·출시연도 등의 메타 특성을 활용해 연결되며, 추천 결과는 영화 단위 후보를 상품 단위로 해석하는 절차를 포함한다.

인증은 JWT 기반으로 구현되어 역할·권한 정보를 토큰 페이로드로 안전하게 전달한다. 따라서 API 접근을 선별하고, 민감 리소스(리뷰 작성, 개인화 추천)에는 사용자 검증을 안전하게 수행한다.

Method	URL	Description	주요 변수
POST	api/login	유저 로그인	usernameOrEmail, passowrd
POST	api/signup	회원가입	username, email, password
GET	api/me	유저 정보 불러오기	Bearer <token>
GET	api/movies	모든 영화 불러오기	
GET	api/movies/{id}	특정 영화 불러오기 (영화 id로 검색)	movieId
GET	api/movies/search	영화 제목으로	영화 제목

	?query=...	영화 검색	
GET	api/sales	모든 블루레이 정보 불러오기	
GET	api/sales/{id}	특정 블루레이 불러오기 (salesId로 검색)	블루레이 id (salesId)
GET	api/sales/search?query=...	블루레이 타이틀로 검색	블루레이 타이틀
POST	api/recommendations/run	추천 알고리즘 실행	userId, topN
POST	api/reviews	리뷰 등록	salesId, rating, reviewComment
GET	api/reviews?salesId=..	특정 블루레이에 대한 리뷰들 불러오기	salesId
GET	api/reviews?userId=..	특정 사용자의 리뷰를 불러오기	userId

3.2.4.2 데이터베이스

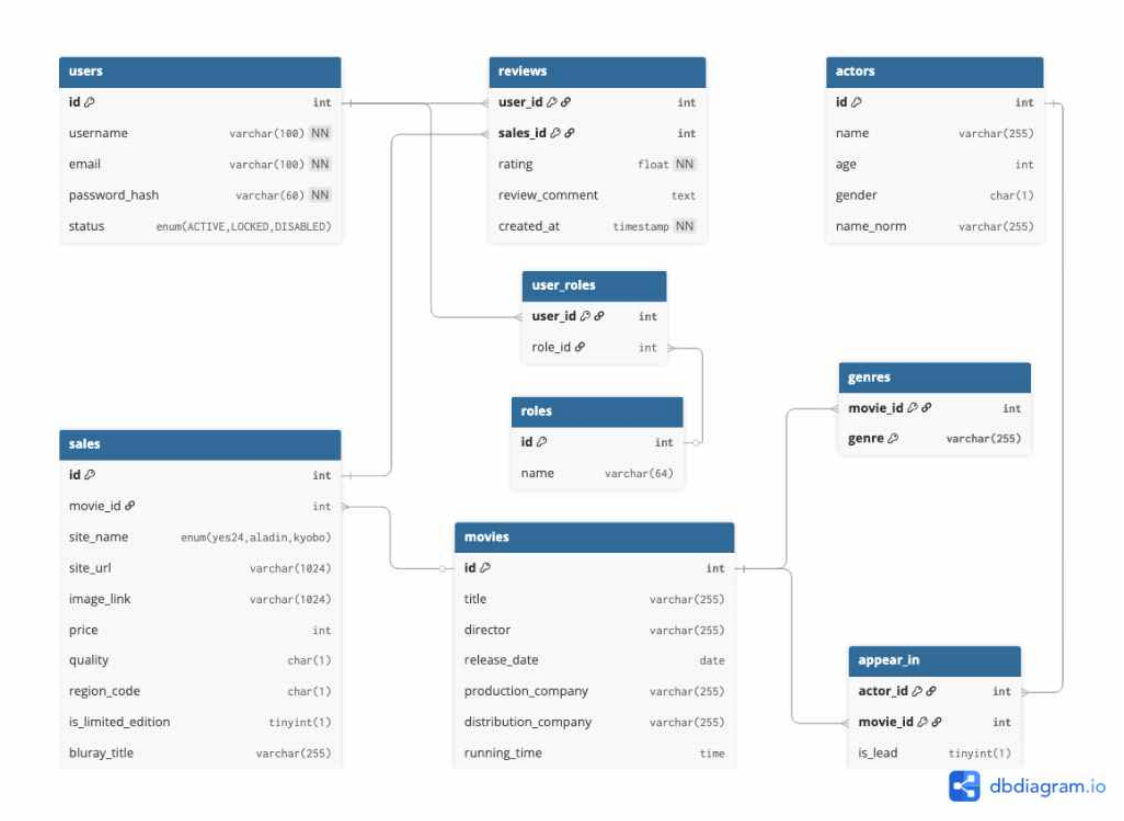

users
id int
username varchar(100) NN
email varchar(100) NN
password_hash varchar(60) NN
status enum(ACTIVE,LOCKED,DISABLED)
reviews
user_id int
sales_id int
rating float NN
review_comment text
created_at timestamp NN
actors
id int
name varchar(255)
age int
gender char(1)
name_norm varchar(255)
user_roles
user_id int
role_id int
roles
id int
name varchar(64)
genres
movie_id int
genre varchar(255)
sales
id int
movie_id int
site_name enum(yes24,aladin,kyobo)
site_url varchar(1024)
image_link varchar(1024)
price int
quality char(1)
region_code char(1)
is_limited_edition tinyint(1)
bluray_title varchar(255)
movies
id int
title varchar(255)
director varchar(255)
release_date date
production_company varchar(255)
distribution_company varchar(255)
running_time time
appear_in
actor_id int
movie_id int
is_lead tinyint(1)
dbdiagram.io

Table	Attribute	Type	Contraint	Explanation
users	id	int	PK	유저를 구분하는 고유 ID
	username	varchar(100)	not null, unique	회원가입 시 유저가 사용하는 아이디로, 고유한 값을 가짐
	email	varchar(100)	not null, unique	회원가입 시 기입하는 고유한 이메일 주소
	password_hash	varchar(60)	not null	유저 비밀번호 정보의 해시값
	status	enum('ACTIVE','LOCKED','DISABLED')		유저의 현재 상태(ACTIVE: 정상, LOCKED: 잠금, DISABLED: 정지)
user_roles	user_id	int	PK,FK	유저를 식별하는 ID 값
	role_id	int	FK, not null	유저의 역할을 위한 ID 값
roles	id	int	PK	역할 구분을 위한 고유 ID
	name	varchar(64)		관리자, 사용자 계정 등 역할을 구분하는 이름

sales	id	int	PK	블루레이 판매 정보를 위한 고유 ID
	movie_id	int	FK	해당 블루레이가 어떤 영화에 대한 것인지 알 수 있는 ID
	site_name	enum('yes24','aladin','kyobo')		블루레이 판매 사이트의 이름 (yes24, aladin, kyobo)
	site_url	varchar(1024)		블루레이 판매 페이지의 URL
	image_link	varchar(1024)		블루레이 이미지를 로드하기 위한 링크 주소
	price	int		블루레이의 가격
	quality	char(1)		화질: 4K Bluray - '4', 3d Bluray - '3', Bluray - 'B', DVD - 'D', N/A - 'U'
	region_code	char(1)		지역코드: DVD는 1~6, Bluray는 'A', 'B','C', 4K Bluray는 Null (지역프리)
	is_limited_edition	tinyint(1)		한정판 여부를 나타내며, 1은 한정판을 의미
	blueray_title	varchar(255)		블루레이 판매 타이틀

reviews	user_id	int	PK,FK	평가한 사용자의 ID
	sales_id	int	PK,FK	평가된 블루레이의 ID
	rating	float	not null	평점
	review_comment	text		평점에 대한 코멘트
	created	timestamp	not null	평점이 생성된 시간
movies	id	int	PK	영화를 구분하기 위한 고유 ID
	title	varchar(255)		영화 제목
	director	varchar(255)		영화 감독
	release_date	date		영화 개봉 일시
	production_company	varchar(255)		영화의 제작사
	distribution_company	varchar(255)		영화의 배포사
	running_time	time		영화의 러닝타임
genres	movie_id	int	PK,FK	영화 ID
	genre	varchar(255)	PK	영화에 해당하는 장르

actors	id	int	PK	영화배우를 구분하기 위한 고유 ID
	name	varchar(255)		영화배우의 이름
	age	int		영화배우의 나이
	gender	char(1)		영화배우의 성별
	name_norm	varchar(255)		중복되는배우를 제거하기 위해 이름을 정상화(대문자-> 소문자로 통일)한 값
appear_in	actor_id	int	PK,FK	출연한 영화배우의 ID
	movie_id	int	PK,FK	출연하는 영화의 ID
	is_lead	tinyint(1)		주연 여부를 나타내며, 1은 주연을 의미

데이터 접근은 JPA 리포지토리를 기반으로 하며, 명세적 쿼리와 동적 조건 조합을 병행하여 목록·검색·상세 조회의 다양한 필터 요구를 수용한다. 스키마는 영화·상품(에디션 포함)·사용자·리뷰·추천 로그의 핵심 관계를 반영한다.

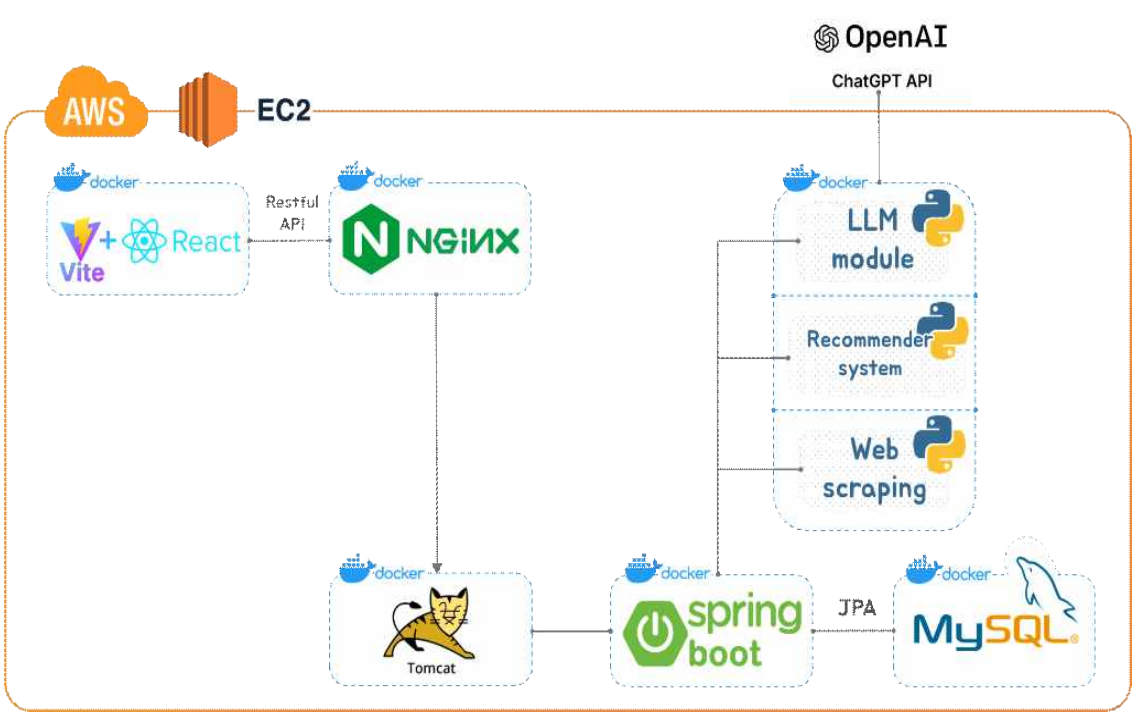
3.2.4.3 파이선 추천 서비스 (py-reco)

추천 모듈은 Capstone/dataset의 스크립트 자산을 서비스 환경에 맞게 래핑하여, 외부에서는 API 호출만으로 추천 결과를 획득하는 방식으로 동작한다. 내부적으로는 콘텐츠 기반, 협업 필터링, 행렬분해의 출력물을 통합·재랭킹하는 파이프라인을 구성하여 요청 시점의 최신 데이터와 사용자 맥락을 반영한다

포트번호 8000번에서 GET /run?userId&topN 형태의 엔드포인트는 필수 파라미터인 userId와 반환 개수인 topN을 입력으로 받아, 요청 유효성 검사를 거친 뒤 JSON 포맷으로 추천 결과를 반환한다. 파라미터 기본값과 상한을 정의하여 과도한 응답 크기를 방지하고, 잘못된 userId에 대해서는 표준화된

오류 응답을 제공한다

3.2.5 서비스 아키텍쳐



1. 회원은 가입을 완료한 뒤 로그인을 수행하며, 서버는 검증 절차를 거쳐 발급한 JWT 토큰을 클라이언트 저장소에 안전하게 보관하도록 안내한다.

2. 사용자는 영화 목록에서 키워드·필터로 검색을 수행하고 상세 화면으로 진입하여 작품 정보와 함께 통합된 블루레이 판매 정보를 열람한다.

3. 구매한 블루레이에 대해 사용자는 리뷰를 작성하고, 다른 사용자들이 남긴 리뷰를 함께 확인하여 품질·에디션 선택에 대한 정성적 근거를 확보한다.

4. 개인화 추천을 실행하면 시스템은 사용자 선호와 메타 특성을 반영한 후보 블루레이/영화를 제시하며, 후보 선정 사유를 함께 노출하여 결과의 설명가능성을 제공한다.

3.2.6 논리 아키텍처

본 프로젝트의 논리 아키텍처는 Client → Nginx → Spring API → (MySQL, py-reco) → dataset의 단일 흐름으로 구성되며, 클라이언트는 모든 요청을

Nginx로 전송하여 정적 리소스 서빙과 API 역프록시를 일관된 진입점에서 처리한다. Spring API는 비즈니스 로직과 데이터 접근을 담당하고, MySQL과 추천 모듈(py-reco) 및 원천 dataset에 연계하여 조회·저장·추론을 조정한다.

- 요청 시퀀스(추천)

추천 기능의 표준 시퀀스는 Client → POST /api/recommendations/run → Spring → GET py-reco /run → Spring 변환 → Client 흐름으로 정의되며, 클라이언트의 실행 요청은 서버에서 인증·검증을 거친 후 내부 추천 모듈로 전달된다. py-reco는 사용자 컨텍스트와 최신 데이터에 기반한 추천 결과를 생성하고, Spring은 이를 클라이언트 규격의 DTO로 변환하여 응답한다.

- 데이터 흐름

데이터는 MovieLens와 판매 원천에서 수집·정규화 과정을 거친 후 DB에 저장되며, API는 해당 데이터를 목적에 맞는 DTO로 구성하여 UI에 반환한다. 이 과정에서 작품–에디션 매핑, 메타 특성(장르·출시연도·4K·리마스터 등) 정제, 중복·동형 처리 규칙이 적용되어 검색·상세·추천 품질을 높인다

4. 연구 결과 분석 및 평가

4.1 기능목록

4.1.1 웹서비스

- 회원가입/로그인(JWT)
- 메인 랜딩페이지
- 주요 화면: 로그인, 회원가입, 홈, 검색, 상세, 프로필
- 사용자 분류

  - 일반 사용자, 관리자

  - 사용자별 차등 권한 부여(리뷰 등록·추천 실행은 로그인 필요, 영화/판매 열람은 공개)

- 사용자 활동

  - 리뷰활동

- 영화(콘텐츠) 구성
  - 영화 목록/상세 조회
  - 영화 검색(제목 기준)
- 판매(Sales) 구성
  - 판매 목록/단건 조회
  - 판매 검색(영화 제목 기준, 부분 일치)
- 리뷰(응시·채점에 대응)
  - 리뷰 등록/업서트(0.5~5.0), 조회 API 제공

4.1.2 개인화 추천(추천 실행/채점에 대응)

- 실시간 추천 실행 API
  - Spring → Python 추천 서비스(py-reco) 연동
  - 매개변수: 사용자 ID, Top-N
  - 응답: 영화/판매 후보, 유사도 점수, 추천 사유 문자열
- 인증 필요 엔드포인트(토큰 기반)

4.1.3 PBT에 대응되는 출력/정적 페이지

- Nginx 정적 페이지 서빙(HTML/CSS/JS)
- SPA 라우팅 지원(임의 경로 → 홈 페이지)
- 추후 PDF 출력 연동 확장 용이(현 시점 기본 정적 자산 제공)

4.1.4 생성 결과 검수/관리(운영 기능)

- 추천 실행 트리거/모니터링용 API 제공
- 데이터 임포트(추천 도메인) API 준비(코드 레벨 지원)

4.1.5 회원별 학습 통계에 대응되는 사용자 통계

- 사용자 리뷰 기반 활동 이력 조회
- 향후 평균 평점/활동량 등 지표 확장에 필요한 엔드포인트 구조 마련

4.1.6 • 보안/권한

- JWT 인증(만료 기본 15분)

- 역할 구조(User/ADMIN) 기반 접근 제어
- 민감 엔드포인트 보호(Authorization 헤더)

4.1.7 • 인프라/운영

- Docker Compose 기반 3컨테이너
  - nginx: 정적 서빙 + Reverse Proxy(`/api → spring-api`)
  - spring-api: 핵심 REST API
    - py-reco: 추천 시스템
- 로그/성능
  - `/api` 전용 접근/에러 로그 분리
  - 프록시 타임아웃·버퍼링 설정
- DB
  - MySQL 연동(JPA 리포지토리)
- 데이터/모델
  - MovieLens + 블루레이 데이터 변환 스크립트
  - 표준 DTO 응답(Movie/Sales/Review/Recommendation)

4.2 성능 평가

612번 사용자의 리뷰 목록 (총 20개):

영화 ID	평점	작성일	영화 제목	장르
163134	3.0	2000-07-30 19:08:20	Your Name. (2016)	Animation|Drama|Fantasy|Rom...
5618	5.0	2000-07-30 18:45:03	Spirited Away (Sen to Chihiro no kamikakushi) (...	Adventure|Animation|Fantasy
176101	2.0	2000-07-30 18:45:03	Kingsman: The Golden Circle (2017)	Action|Adventure|Comedy
119145	3.0	2000-07-30 18:45:03	Kingsman: The Secret Service (2015)	Action|Adventure|Comedy|Crime
1345	4.5	2000-07-30 18:45:03	Carrie (1976)	Drama|Fantasy|Horror|Thriller
42723	1.5	2000-07-30 18:45:03	Hostel (2005)	Horror
32456	5.0	2000-07-30 18:45:03	Pom Poko (a.k.a. Raccoon War, The) (Heisei tanu...	Animation|Comedy|Drama|Fantasy
56728	5.0	2000-07-30 18:45:03	N/A	N/A
53996	3.0	2000-07-30 18:45:03	Transformers (2007)	Action|Sci-Fi|Thriller|IMAX
128968	5.0	2000-07-30 18:45:03	Stitch! The Movie (2003)	Animation|Children|Comedy
60069	5.0	2000-07-30 18:45:03	WALL·E (2008)	Adventure|Animation|Childre...
26662	5.0	2000-07-30 18:45:03	Kiki's Delivery Service (Majo no takkyûbin) (1989)	Adventure|Animation|Childre...
5971	5.0	2000-07-30 18:45:03	My Neighbor Totoro (Tonari no Totoro) (1988)	Animation|Children|Drama|Fa...
101962	5.0	2000-07-30 18:45:03	Wolf Children (Okami kodomo no ame to yuki) (2012)	Animation|Fantasy
3000	5.0	2000-07-30 18:45:03	Princess Mononoke (Mononoke-hime) (1997)	Action|Adventure|Animation|...
109487	4.0	2000-07-30 18:45:03	Interstellar (2014)	Sci-Fi|IMAX
80586	4.5	2000-07-30 18:45:03	Flipped (2010)	Comedy|Drama|Romance
31658	5.0	2000-07-30 18:40:00	Howl's Moving Castle (Hauru no ugoku shiro) (2004)	Adventure|Animation|Fantasy...
53000	2.0	2000-07-30 18:37:04	28 Weeks Later (2007)	Horror|Sci-Fi|Thriller
106696	4.0	2000-07-30 18:37:04	Frozen (2013)	Adventure|Animation|Comedy|...

예시 사용자가 실제로 남긴 평점/리뷰 기록(사용자 프로파일)이다. 추천 적중 여부 판단을 위한 기준 데이터로 활용된다.

```
🎬 영화 추천 결과 (상위 20개):
순위    영화 ID    예상 평점    개인화 점수    평균 평점    영화 제목
------------------------------------------------------------------------------------------
1     4306     3.64     4.78     3.87     Shrek (2001)
2     4886     3.64     4.77     3.87     Monsters, Inc. (2001)
3     2138     3.35     4.76     4.05     Watership Down (1978)
4     76093    3.65     4.70     3.94     How to Train Your Dragon (2010)
5     6350     3.81     4.69     4.06     Laputa: Castle in the Sky (Tenkû...
6     81847    3.59     4.64     3.92     Tangled (2010)
7     661      3.44     4.60     3.45     James and the Giant Peach (1996)
8     4016     3.45     4.58     3.72     Emperor's New Groove, The (2000)
9     27731    3.41     4.57     3.75     Cat Returns, The (Neko no ongaes...
10    162578   3.40     4.56     4.00     Kubo and the Two Strings (2016)
11    78499    3.54     4.56     4.11     Toy Story 3 (2010)
12    166461   3.42     4.55     3.45     Moana (2016)
13    72737    3.36     4.55     3.75     Princess and the Frog, The (2009)
14    2987     3.40     4.53     3.57     Who Framed Roger Rabbit? (1988)
15    2092     3.24     4.51     2.42     Return of Jafar, The (1994)
16    4366     3.34     4.51     3.34     Atlantis: The Lost Empire (2001)
17    2123     3.21     4.50     2.70     All Dogs Go to Heaven (1989)
18    4519     3.34     4.50     3.33     Land Before Time, The (1988)
19    2116     3.34     4.50     3.13     Lord of the Rings, The (1978)
20    53121    3.37     4.50     3.02     Shrek the Third (2007)

... 총 100개 중 상위 20개만 표시됨
```

예시 사용자에 대해 행렬분해 기반 모델이 산출한 상위 N개 추천 목록이다. 사용자–아이템 상호작용을 잠재요인으로 분해해 개인화 성향을 반영하며, 유사한 취향을 보이는 사용자 군집의 패턴이 추천에 반영된다. 사용자가 과거에 높게 평가한 작품들과 주제·스타일이 유사한 항목이 상위권에 분포한다.

```
=== Content-based Filtering 블루레이 추천 결과 (순수 콘텐츠 기반) ===
순위    영화 제목                                  유사도      Sale ID   가격          화질        선택 이유
------------------------------------------------------------------------------------------------------------------------
1     Moana (2016)                           0.637     15663     52,300원     U         유사도 0.637, 미확정, 52,300원,
2     Shrek the Third (2007)                 0.602     10158     56,700원     4         유사도 0.602, 4K, 56,700원, 수
3     Shrek (2001)                           0.479     5863      59,400원     4         유사도 0.479, 4K, 59,400원, 한
4     Tangled (2010)                         0.097     11988     65,500원     4         유사도 0.097, 4K, 65,500원, 수
5     Toy Story 3 (2010)                     0.097     11743     65,500원     4         유사도 0.097, 4K, 65,500원, 수
6     Who Framed Roger Rabbit? (1988)        0.097     4358      64,300원     4         유사도 0.097, 4K, 64,300원, 수
7     Jumanji (1995)                         0.097     9         60,800원     4         유사도 0.097, 4K, 60,800원, 수
8     James and the Giant Peach (1996)       0.074     1046      23,500원     U         유사도 0.074, 미확정, 23,500원,
9     Kubo and the Two Strings (2016)        0.074     15575     28,160원     U         유사도 0.074, 미확정, 28,160원,
10    How to Train Your Dragon (2010)        0.074     11628     122,100원    4         유사도 0.074, 4K, 122,100원,
11    Atlantis: The Lost Empire (2001)       0.060     5917      25,300원     U         유사도 0.060, 미확정, 25,300원,
12    Land Before Time, The (1988)           0.060     6062      21,900원     U         유사도 0.060, 미확정, 21,900원,
13    Emperor's New Groove, The (2000)       0.023     5582      23,500원     U         유사도 0.023, 미확정, 23,500원,
14    Lord of the Rings, The (1978)          0.023     3184      25,900원     U         유사도 0.023, 미확정, 25,900원,
15    Monsters, Inc. (2001)                  0.016     6422      71,300원     U         유사도 0.016, 미확정, 71,300원,
16    Princess and the Frog, The (2009)      0.016     11472     71,300원     U         유사도 0.016, 미확정, 71,300원,
17    Pocahontas (1995)                      -0.094    98        19,800원     U         유사도 -0.094, 미확정, 19,800원,
18    All Dogs Go to Heaven (1989)           -0.407    3210      33,100원     U         유사도 -0.407, 미확정, 33,100원,
19    Watership Down (1978)                  -1.000    N/A       N/A          N/A       블루레이 없음
20    Laputa: Castle in the Sky (Tenkû...    -1.000    N/A       N/A          N/A       블루레이 없음
```

사용자가 과거에 선호한 영화의 메타데이터(장르, 화질, 가격 등등)와의 특징 유사도(코사인 유사도)를 기준으로 산출된 추천 목록이다. 아이템 자체의 속성에 기반한다. 상위권 항목은 장르·주제·톤이 기존 선호 항목과 일관된다.

5. 결론 및 향후 연구 방향

5.1 결론

본 연구는 다원 소스(교보문고·예스24·알라딘 등)에서의 웹 크롤링과 LLM 기반 정제를 결합해 블루레이 상품 데이터를 표준화하고, 영화와의 매핑을 일관되게 수행하는 데이터 파이프라인을 구축하였다. 무비렌즈 기반 행렬분해로 사용자 선호 가능성이 높은 영화 후보군을 도출하고, 4K·리마스터·장르·출시연도 등 블루레이 메타 특성을 반영한 콘텐츠 기반 필터링으로 최종 추천을 제공하는 하이브리드 알고리즘을 구현하였다. React–Spring Boot–MySQL–NGINX–Docker–AWS EC2로 서비스 전 과정을 통합하여 메인·검색·상세 화면을 통해 추천과 상품 정보 제공을 실용적으로 제시하였다.

5.2 향후 연구방향

5.2.1 매핑 정밀도

블루레이에서 관측되는 평점과 영화 고유의 선호를 분리하기 위해, 영화 전반에 공통으로 작용하는 편향과 블루레이에 기인한 편향을 각각 Bias 분해로 추정하고 영화 Bias만을 추출해 매핑의 기준으로 사용한다. 이를 통해 동일 작품의 다양한 블루레이에서 발생하는 표본 편향을 줄이고, 영화 선호 추정의 일관성과 전이 가능성을 높인다

5.2.2 특성 확정

영화 측면에서는 감독, 배우, 수상 경력과 같은 데이터를 추가로 수집·정규화하고, 블루레이 측면에서는 오디오 포맷, 부가영상, 패키징(스틸북·리패키지), 리전 및 한정판 여부 등 구매 결정에 직결되는 피처를 스키마에 편입한다. 확장된 특성은 가중치 학습 대상에 포함되어 추천 점수 산정 시 품질·희소성·수집 가치가 균형 있게 반영되도록 한다.

5.2.3 모델 고도화

콘텐츠 기반 필터링을 다수의 특성과 상호작용 항을 직접 학습하는 Learn to Rank 계열의 LambdaMART 모델로 대체한다.

5.2.4 실시간성 강화

최신 입고·가격 변동을 신속히 반영하기 위해 증분 수집 중심의 실시간 크롤링을 도입하고, 캐시 시스템을 사용한다. 또한 인덱싱 최적화로 조회 지연을 낮추고 요청 시점의 실시간 응답성을 확보한다.

5.2.5 운영 안정성

컨테이너 이미지 무결성 검사를 실행하는 시스템을 도입하여 모니터링·알림을 주기적으로 수행한다. 또한 트래픽 증감에 대비한 수평 확장 정책으로 가용성을 보장한다

5.2.6 UX/UI

사용자에게 동일 작품의 에디션을 한눈에 비교할 수 있는 비교 보기를 제공하고, 필터·정렬 옵션을 확장하여 선호 스펙 중심의 탐색을 가능하게 한다. 더불어 추천 카드에 특성 매칭, 유사 사용자·유사 영화 근거 등 간결한 설명을 노출해 결과 신뢰도를 높이고 상호작용을 유도한다

6. 참고 문헌