

# 스프린트미션 18

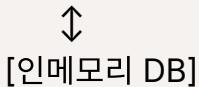
## 서비스 개요

“영화 관리 및 리뷰 감성 분석 시스템”

- 본인이 감상한 영화를 등록하고 리뷰를 남겨보세요!
- 영화에 대한 리뷰를 남겨주시면 AI가 자동으로 리뷰에 대한 별점을 계산해 줍니다!

## 서비스 구조도

[사용자] → [Streamlit UI] → [FastAPI Backend] → [ONNX Model 추론]



## 주요 기능 요약

- 영화 관리: 등록(Create), 조회(Read), 삭제(Delete)
- 리뷰 관리: 등록(Create), 조회(Read), 삭제(Delete)
- 별점/감성 분석: ONNX Runtime 기반 자동 추론

## 모델 설명

- 학습/기반 모델: [nlptown/bert-base-multilingual-uncased-sentiment](#)  
(다국어 1~5 별점 분류 모델)
- 배포 형식: PyTorch → ONNX 변환 → INT8 양자화
- 추론 러너: **ONNX Runtime** (ExecutionProvider = [CPUExecutionProvider](#) 고정)
- 토크나이저: HuggingFace [AutoTokenizer](#)
- 출력 처리: [softmax](#) 확률 → 최빈 클래스 = 별점, 그 확률 = confidence

$$\text{감성점수} = (\text{별점} - 3)/2 \times \text{confidence}, \text{ 범위}[-1, 1]$$

## 데이터 베이스 구조도(ERD)

MOVIE (id PK)

- └ title (TEXT, NOT NULL, UNIQUE on lower(btrim(title)))
- └ release\_date (DATE, NOT NULL)
- └ director (TEXT, NOT NULL)
- └ poster\_url (TEXT, NULL)
- └ genre (genre\_enum[] NOT NULL DEFAULT '{}')

REVIEW (id PK)

- └ movie\_id (BIGINT, FK → MOVIE.id ON DELETE CASCADE)
- └ author (TEXT, NOT NULL)
- └ content (TEXT, NOT NULL)
- └ created\_at (TIMESTAMPTZ NOT NULL DEFAULT now())
- └ sentiment\_label (TEXT, NOT NULL)
- └ sentiment\_score (DOUBLE PRECISION NOT NULL)
- └ star (SMALLINT NULL)
- └ star\_conf (DOUBLE PRECISION NULL)
- └ soft\_star (DOUBLE PRECISION NULL)

인덱스

- uq\_movie\_title\_norm: UNIQUE (lower(btrim(title)))
- idx\_movie\_director\_norm: INDEX on lower(director) [옵션]
- idx\_movie\_release\_date: INDEX on release\_date [옵션]
- idx\_review\_movie\_created\_desc: INDEX on (movie\_id, created\_at DESC)

## 관계 (Relationship)

- MOVIE 1——< N REVIEW

- 영화 삭제 시 해당 리뷰 자동 삭제 (**ON DELETE CASCADE**)

# FastAPI Docs 설명

default	
GET	/ Hello
GET	/health Health
POST	/movie Add Movie
GET	/movie List Movies
DELETE	/movie Clear Movies
GET	/movie/{movie_id} Get Movie
DELETE	/movie/{movie_id} Delete Movie
POST	/review Create Review
GET	/review List Reviews
DELETE	/review/{review_id} Delete Review
GET	/rating/average Rating Average

- `/` → Hello API (서비스 환영 메시지 확인)
- `/health` → 모델/백엔드 상태 점검용 API
- `/movie` 관련 API
  - `POST /movie` : 영화 등록
  - `GET /movie` : 영화 목록 조회
  - `DELETE /movie` : 영화 전체 삭제
  - `GET /movie/{movie_id}` : 특정 영화 상세 조회
  - `DELETE /movie/{movie_id}` : 특정 영화 삭제
- `/review` 관련 API
  - `POST /review` : 리뷰 등록 (리뷰 내용 분석 후 저장)
  - `GET /review` : 리뷰 목록/검색 조회
  - `DELETE /review/{review_id}` : 특정 리뷰 삭제
- `/rating/average` → 특정 영화의 평균 별점 계산 및 반환

Schemas	
BaseResponse	> Expand all object
Genre	> Expand all string
HTTPValidationError	> Expand all object
MovieIn	> Expand all object
MovieOut	> Expand all object
RatingOut	> Expand all object
ReviewIn	> Expand all object
ReviewOut	> Expand all object
ReviewWithTitle	> Expand all object
SuccessMovie	> Expand all object
SuccessMovies	> Expand all object
SuccessRating	> Expand all object
SuccessReview	> Expand all object
ValidationError	> Expand all object

- **BaseResponse**

- 성공 응답 래퍼의 공통 베이스(성공 여부·메시지 포함)

- **Genre**

- 영화 장르를 정의하는 **Enum** (액션, 코미디, 드라마 ...).

- **MovieIn / MovieOut**

- **MovieIn** : 영화 등록 요청(제목, 개봉일, 감독, 장르, 포스터 URL[ **uri** ]).
- **MovieOut** : 응답 모델( **MovieIn** + **id** ).

- **ReviewIn / ReviewOut / ReviewWithTitle**

- **ReviewIn** : 리뷰 등록 요청( **movie\_id** , 작성자, 내용).
- **ReviewOut** : 리뷰 응답
- **ReviewWithTitle** : 리뷰 응답 + **movie\_title**

- **RatingOut**

- 영화 하나에 대한 집계 결과: 리뷰 수(`count`), 평균 별점(`avg_star`), 소프트 평균(`avg_soft_star`), 확신도 가중 평균(`avg_star_conf_weighted`).
- **SuccessMovie / SuccessMovies / SuccessReview / SuccessRating**
  - 성공 응답 래퍼(`BaseResponse` 확장): 각각 단일 영화, 영화 목록, 리뷰, 평점 데이터 포함.
- **ValidationError / HTTPValidationError**
  - 유효성 오류 시 반환되는 표준 에러 구조(어떤 필드에서 어떤 오류인지 포함).

## 디렉토리 구조

```
mission18/
└── backend/
    ├── backend.py          # FastAPI 서버
    └── models/
        └── star-rating-onnx-int8/ # ONNX INT8 모델 + 토크나이저
            ├── model_quantized.onnx # 양자화 모델
            ├── model.onnx          # 양자화 전 원본
            ├── config.json
            ├── tokenizer.json
            ├── tokenizer_config.json
            ├── special_tokens_map.json
            ├── ort_config.json      # ORT 설정
            └── vocab.txt
        └── quantization.ipynb     # ONNX/양자화 모델 생성 코드
    └── frontend/
        └── frontend.py          # Streamlit UI
```

## 서버 실행

1) FastAPI 백엔드 기동 (포트 8000)

```
uvicorn backend:app --reload --port 8000
```

2) Streamlit 프론트엔드 기동

```
streamlit run frontend.py
```

## 서비스 동작 캡쳐 이미지

### 1. 홈/목록 화면

The screenshot shows the main interface of the 'Movie Manager' application. At the top left is a navigation icon followed by the text '컨트롤'. Below it are two buttons: '새로고침' (refresh) and '전체 데이터 삭제' (delete all data), with the latter being red. The main title '무비 매니저' is centered above a horizontal line. To the right of the title is a movie camera icon. Below the title, there's a section titled '전체 영화' with a small movie icon. It displays the message '총 0건' (0 items total). A light blue banner at the bottom states '영화 데이터가 비었습니다.' (There is no movie data). On the left side of the screen, there's a sidebar with three items: '영화 추가' (Add movie), '검색/필터/삭제' (Search/filter/delete), and '리뷰 관리' (Review management), each preceded by a small icon.

### 2. 영화 등록 폼

- 제목, 개봉일, 감독, 장르, 포스터 URL 입력 후 등록
- 중복 제목 방지 및 URL 유효성 검사
- 모든 항목 검증 후 성공/오류 메시지 출력
- 필드 옆 ? 버튼을 통해 입력 규칙 설명 확인 가능

### 3. 영화 검색

- 제목·감독·장르·개봉일 기준 필터링 지원
- 검색 결과에서 영화 개별 삭제 지원

## 4. 리뷰 작성/추론 화면

- Review 버튼을 누르면 해당 영화의 리뷰 작성 화면으로 이동합니다.
- 리뷰를 작성·등록하면 최신 10개 리뷰로 갱신되며, 예측 별점과 감성 분석 결과가 함께 표시됩니다.
- 평균 별점은 영화 목록 카드에서도 확인할 수 있습니다.
- 리뷰 관리를 통해 리뷰 검색 및 삭제가 가능합니다

**컨트롤**

새로고침 전체 데이터 삭제

> 영화 추가

> 검색/필터/삭제

▼ 리뷰 관리

영화 제목 리뷰 내용

괴물

작성자

별점 필터

전체  정확히

범위

표시 개수

20 - +

검색 실행

리뷰 검색 결과

총 12건

2025-09-04T04:03:23.900493Z

작성자: Zoe Alvarez

2025-09-04T04:03:23.900493Z

작성자: Zoe Alvarez

**괴물 (ID: 4)**

3.75/5 · 리뷰 12개

2006-07-27 · 봉준호 · 액션, 드라마, SF, 모험, 가족

가족의 사투가 시작된다  
한강, 가족, 그리고...  
괴물  
2006년 7월 27일 개봉  
한국영화 박태일 페미니스트 미술 운동 부조, 2006  
2006년 7월 27일 개봉, 당시 폭기 했던 사건이 담아낸다

**리뷰 작성**

작성자 \*

리뷰 내용 \*

감상평을 적어주세요.

등록 및 분석(별점/감성)

**최신 리뷰 10개**

2025-09-04T04:03:23.900493Z  
작성자: Zoe Alvarez  
Dark humor lands, and the family bond grounds the chaos.  
4/5 · conf 0.31  
감성: positive · 점수: 0.1528

2025-09-04T04:03:08.419039Z  
작성자: Ravi Menon  
Striking monster, thin arcs and a meandering plot.

**리뷰 검색 결과**

총 12건

작성자: Zoe Alvarez  
Dark humor lands, and the family bond grounds the chaos.  
★ 4/5 - conf 0.31  
감성: positive - 점수: 0.1528

작성자: Ravi Menon  
Striking monster, thin arcs and a meandering plot.  
★ 4/5 - conf 0.53  
감성: positive - 점수: 0.2659

작성자: 윤재호  
화면이 어둡고 톤이 들쭉날쭉해 몰입이 깨졌다.  
★ 2/5 - conf 0.49  
감성: negative - 점수: -0.2438

작성자: 김민아  
사회 풍자에 감정선까지 살아 있어 여운이 길다.  
★ 5/5 - conf 0.38  
감성: positive - 점수: 0.3834

작성자: 장예린  
메시지는 큰데 인물 서사가 알아 공감이 어렵다.  
★ 3/5 - conf 0.43  
감성: neutral - 점수: 0.0

작성자: 이도윤  
괴수 연출과 주걱 시퀀스가 신선했고 리듬감 있다.  
★ 5/5 - conf 0.39  
감성: positive - 점수: 0.3867

**최신 리뷰 10개**

작성자: Zoe Alvarez  
Dark humor lands, and the family bond grounds the chaos.  
★ 4/5 - conf 0.31  
감성: positive - 점수: 0.1528

작성자: Ravi Menon  
Striking monster, thin arcs and a meandering plot.  
★ 4/5 - conf 0.53  
감성: positive - 점수: 0.2659

작성자: 윤재호  
화면이 어둡고 톤이 들쭉날쭉해 몰입이 깨졌다.  
★ 2/5 - conf 0.49  
감성: negative - 점수: -0.2438

작성자: 김민아  
사회 풍자에 감정선까지 살아 있어 여운이 길다.  
★ 5/5 - conf 0.38  
감성: positive - 점수: 0.3834

작성자: 장예린  
메시지는 큰데 인물 서사가 알아 공감이 어렵다.  
★ 3/5 - conf 0.43  
감성: neutral - 점수: 0.0

작성자: 이도윤  
괴수 연출과 주걱 시퀀스가 신선했고 리듬감 있다.  
★ 5/5 - conf 0.39  
감성: positive - 점수: 0.3867

## 결론 및 개선 방향

### 결론

- Streamlit + FastAPI + ONNX Runtime 조합으로 **GPU 없이도** 리뷰 입력 시 **실시간 별점/감성 분석이 가능한 경량 서비스** 프로토타입을 구현했다.
- FastAPI의 [/docs](#) 자동 문서화와 명확한 Pydantic 스키마 덕분에 **API 사용성/확장성이 높다.**

### 한계

- **인메모리 저장소**라 재시작 시 데이터가 소실된다.
- 별점 모델은 다국어 기반이지만 **한국어 표현에 약한 경향**이 있었다.

## 개선 방향

1. **영속 DB 도입:** SQLite/PostgreSQL로 이관
2. **모델 개선:** 한국어 특화 모델 파인튜닝(예: KoBERT) 후 ONNX INT8 재변환.
3. **배포/운영:** Docker 컨테이너화