

2024년 전기 졸업과제

중간보고서

대형 언어모델을 활용한 추천 시스템의 개선 방안



지도교수 : 조준수 교수님

팀명 : LLcommend

201824482 박지환

201924604 하현진

201924562 장재혁

목차

- 1. 과제 목표 및 수정 사항**
 - 1.1. 기존 과제 목표
 - 1.2. 수정된 과제 목표
- 2. 요구사항 및 제약사항 대응 전략**
 - 2.1. LLM에서의 문제점
 - 2.2. 추천 시스템에서의 문제점
 - 2.3. 데이터에서의 문제점
- 3. 설계 상세화 및 변경 내역**
 - 3.1. 시스템 구조
 - 3.2. 모델 관련 사항
 - 3.2.1. 추천 시스템
 - 3.2.2. 대형 언어 모델(LLM)
- 4. 구성원별 진척도**
- 5. 과제 수행 내용 및 중간 결과**
 - 5.1. 데이터 전처리
 - 5.2. 추천 모델 구축
 - 5.3. 대형 언어 모델
 - 5.4. UI
 - 5.5. 프롬프트 엔지니어링
 - 5.5.1. 목적
 - 5.5.2. 프롬프트 실험

1. 과제 목표 및 수정 사항

1.1. 기존 과제 목표

대형 언어 모델을 활용하여 기존 추천 시스템의 한계를 극복하고 추가적인 성능을 개선할 수 있는 방안을 모색한다.

기존 목표였던 '대형 언어 모델을 이용한 추천 시스템의 개선'은 아래와 같은 맹점들이 존재하였고, 추가적인 논의를 통하여 주제를 변경하게 되었습니다. 변경된 주제는 **1.2** 부분에 기재하였습니다.

1. 주제의 모호함

추천 시스템의 어떤 부분에 대형 언어 모델을 적용하여 개선할 지에 대한 고민이 부족했습니다. 따라서, 대형 언어 모델의 어떤 이점을 취해서 사용자에게 개선된 부분을 체감할 수 있도록 할 지 고민하여 새로운 목표를 선정하였습니다. 이에 맞게 시스템 구조를 다시 작성했습니다.

2. 구체적인 구조적 개념의 부재

주제의 모호함 때문에 구체적인 시스템의 구조 또한 명확하지 않고 모호한 형태를 띠고 있었습니다. 새롭게 선정한 목표에 맞게 시스템의 전체 흐름을 다시 정리하고 재작성하였습니다.

1.2. 수정된 과제 목표

본 프로젝트의 주요 목표는 '개인화된 영화 추천 플랫폼의 개발'입니다.

이를 위해 기존의 추천 시스템에 대형 언어 모델(LLM)을 결합하여 개선된 추천 시스템을 구현하고, 사용자에게 더욱 정확하고 맞춤형 영화 추천 서비스를 제공하는 것을 목표로 합니다.

대형 언어 모델을 통해 아래와 같은 이점을 얻고자 합니다.

1. 개인화된 추천 강화

대형 언어 모델은 사용자의 개인 정보와 선호도를 심층적으로 분석하여 더욱 정확한 추천을 제공할 수 있습니다. 사용자의 현재 상황, 기분, 또는 특정 요구사항을 더 잘 이해하고 이를 추천에 반영할 수 있습니다.

2. 다양한 데이터 통합

대형 언어 모델은 영화 정보뿐만 아니라 리뷰, 트렌드, 문화적 맥락 등 다양한 정보를 종합하여 추천의 질을 높일 수 있습니다.

3. Explainability

대형 언어 모델을 통해 사용자의 페르소나를 추측하고 추측 이유에 대해 설명합니다.

4. Novelty

대형 언어 모델에 학습된, 방대한 지식을 활용하여 사용자가 미처 생각하지 못한 새로운 영화나 장르를 소개할 수 있습니다.

단순한 알고리즘 기반 추천을 넘어서 더욱 지능적이고 개인화된 영화 추천 서비스를 제공할 수 있을 것으로 기대됩니다.

2. 요구사항 및 제약사항 대응 전략

2.1 LLM에서의 문제점

(1) 과도한 계산 자원 요구량

- 사전 조사 결과, 대형 언어 모델의 활용에 있어 모델 자체의 거대한 크기로 인해 요구되는 계산 자원의 양이 일반적인 환경에서 구동하기에는 과도하다는 점을 확인하였습니다.
- 이에 대한 첫 번째 대응 전략으로 모델 선정 과정에서 일차적으로 선정된 Llama3 모델군 중 파라미터 수가 제일 적은 LLaMA3.1-8b 모델을 선정하였습니다.
- 또한, 두 번째 대응 전략으로 전체적인 시스템 설계 과정에서 언어 모델의 구동 및 응답 수신을 전담하는 별개의 서버 사이드를 두어 시스템 서비스의 대상이 되는 오퍼레이터의 입장에서는 계산 자원 상의 부담이 없도록 설계하였습니다.
- 마지막으로, 실질적 모델 구동 환경(Google Colab)에서 GPU(NVIDIA Tesla T4) 자원을 사용하도록 구축하여 자원 요구량을 충족할 수 있도록 설정하였습니다. 그러나, 입력값을 모델에 전송한 후 응답 수신까지에 소요되는 시간이 20초대로 적지 않게 측정되어 모델 구동 코드 수정 및 파라미터 조정(토큰 수 등) 등의 응답 대기 시간을 최소화시킬 수 있는 개선 작업을 수행할 계획입니다.

(2) 응답의 신뢰성(Hallucination)

- Llama3.1-8b 모델의 테스트 결과, 일회성의 문답에서는 의도한 것과 크게 벗어나지 않은 응답을 수신할 수 있었지만, 대화가 지속될 수록 그 길이와 비례하여 맥락과 벗어나는 응답을 하는 현상을 확인할 수 있었습니다.
- 이를 개선하기 위한 대책으로 구동 코드 상에서 이전 대화의 내용을 별도로 기록하여 언어 모델의 프롬프트로 계속 삽입될 수 있도록 코드를 구성하였으나, 만족스러운 결과가 도출되지는 않아 추가적인 해결책을 모색중입니다.
- 또한, 목적에 맞는 데이터셋 및 파인 튜닝 라이브러리(TRL)을 사용하여 모델의 파인 튜닝을 진행할 예정입니다.
- 한국어 입출력의 용이성을 위하여 추가적으로 xionic-ko-LLaMA-3-70b 모델의 api를 확보한 뒤 테스트까지 진행은 완료하였으나, 우선 선정된 모델(LLaMA3.1-8b)이 다국어를 지원하기 때문에 큰 이상이 없을 경우 변동 없이 계획에 따라 현재 모델을 기반으로 파인 튜닝을 진행하여 작업을 재개할 예정입니다.

2.2 추천 시스템에서의 문제점

(1) Cold Start 문제

- 신규 사용자가 시스템에 진입하였을 경우, 부족한 정보량으로 인하여 추천의 정확도가 떨어지는 문제 현상입니다.
- 시스템 실행 흐름 내에서 신규 사용자 진입 시 대형 언어 모델의 입출력값과 연계하여 최종 추천의 근거가 될 수 있는 정도의 데이터를 수집할 계획입니다.

(2) 맥락적 요소의 반영 부족

- 기존의 추천 시스템이 사용자의 실시간 상황적 요소를 고려하지 못하고, 일반적인 추천만을 출력하는 현상입니다.
- 이는 사용자의 페르소나 및 상호작용 내역이 포함된 프롬프트를 지속적으로 대형 언어 모델의 입력으로 순환함으로써 개선할 예정입니다.

(3) 과적합 문제

- 훈련 데이터, 또는 기존에 사용하였던 데이터에 과도하게 적합하도록 모델이 학습되어 특정 영화, 사용자 유형에만 적절한 추천이 제공되는, 과적합의 가능성이 존재합니다.

-
- 대형 언어 모델을 사용하여 정확한 사용자의 현재 상황 정보 및 대화 맥락을 추천 시스템에 반영, 특정 그룹이나 아이템에 편향되지 않고 현재 상황이 최우선적으로 고려되도록 시스템을 설계할 예정입니다.

2.3 데이터에서의 문제점

(1) 최신 데이터 부족

- 기존에 선택했던 데이터는 2020년까지의 데이터만 존재합니다. 따라서 최신 영화에 대한 추천은 대형 언어 모델에 온전히 의존해서 기대해야 합니다.
- 대형 언어 모델에 의존할 경우 추천된 영화의 설명이 정확하지 않거나 존재하지 않는 영화를 제공할 수 있습니다.
- 따라서 실제 데이터를 확보하기 위해 kaggle에 제공되는 최신 영화 데이터를 추가하고, 필요한 경우 최신 영화에 대한 정보를 크롤링하여 사용할 계획입니다.

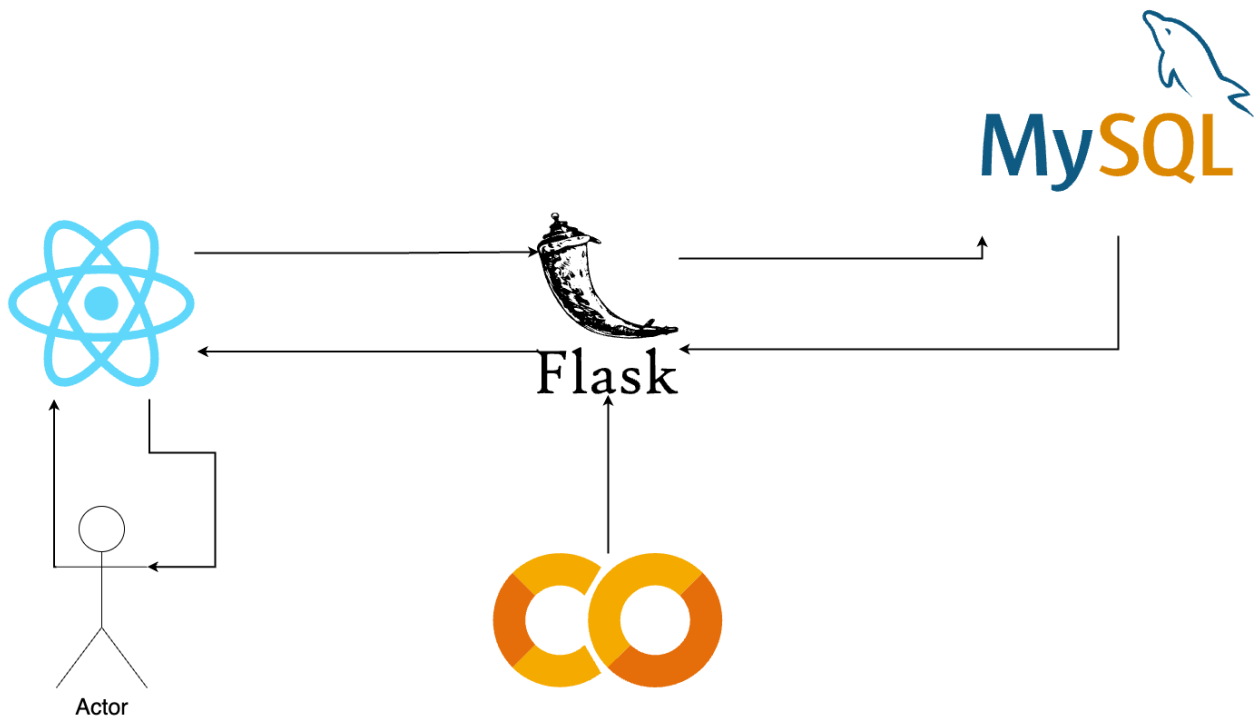
(2) 사용자 정보 확보 어려움

- Netflix등과 같은 영상 플랫폼과 달리 어떤 유저의 선호도, 시청 기록등을 확보하기 어렵습니다.
- 따라서 웹에 사용자가 직접 입력할 수 있는 부분을 제작해 사용자에게 의존해 정보를 받아야 합니다.

-
- 첫 사용에 사용자의 시청기록과 개인 정보 기입을 web ui를 통해 유도하여 해결하고자합니다.
 - 사용자가 오, 미기입 하는 경우를 최대한 억제하고 추가적인 정보를 얻을 수 있는 방법론을 더 찾고 적용하여 해결할 계획입니다.

3. 설계 상세화 및 변경 내역

3.1. 시스템 구조



(1) 메인 페이지 로드 및 사용자 정보 입력

- 사용자가 사이트에 접속하면, 메인 페이지가 로드됩니다. 이 페이지에서 사용자는 개인정보를 입력할 수 있는 폼을 보게 됩니다.
- 사용자 입력: 사용자는 본인의 User ID, 나이, 성별 등을 입력하고 "Next" 버튼을 클릭합니다.
- 서버로 POST 요청 전송: 입력된 정보는 POST 요청으로 서버로 전송됩니다.

(2) 사용자 정보 저장 및 영화 선택 페이지로 이동

- 사용자 정보 저장: 서버는 전달받은 사용자 정보를 데이터베이스에 새 레코드로 저장합니다.
- 영화 선택 페이지로 리디렉션: 사용자 정보를 저장한 후, 서버는 사용자를 영화 선택 페이지로 리디렉션합니다.

(3) 영화 선택 및 페르소나 생성

- 사용자 영화 선택: 사용자는 시청한 영화를 체크박스를 통해 선택합니다.
- 서버로 POST 요청 전송: 사용자가 선택한 영화는 POST 요청을 통해 서버로 전송됩니다.
- 영화 정보 저장: 서버는 사용자가 선택한 영화 목록을 해당 사용자 정보에 저장합니다.
- 페르소나 생성: 사용자의 나이, 성별, 시청한 영화 목록을 바탕으로 서버는 페르소나를 생성하고, 이를 데이터베이스에 저장합니다.
- 추천 영화 페이지로 리디렉션: 페르소나 생성 후, 서버는 사용자를 추천 영화 페이지로 리디렉션합니다.

(4) 추천 영화 페이지 로드 및 추천 영화 표시

- 추천 영화 생성: 서버는 사용자의 시청 영화 목록과 페르소나를 바탕으로 추천 영화를 생성합니다. (코드에서는 임의의 추천 영화 목록이 사용됨)
- 추천 영화 저장: 생성된 추천 영화 목록은 데이터베이스에 저장됩니다.
- 추천 영화 페이지 렌더링: 서버는 추천 영화 페이지를 렌더링하여 사용자에게 추천된 영화를 보여줍니다.

3.2. 모델 관련 사항

3.2.1. 추천 시스템

(1) 직접 구현

- 첫 번째로 Keras 라이브러리를 활용, 다중 레이어 신경망 기반 추천 모델을 구현하였습니다. 해당 모델의 경우 어느 정도 사용자가 시청한 영화의 기록 등의 데이터가 전제되어야 할 것인데, 사용자 초기 진입시에 진행하는 문답의 길이를 고려해 보았을 때 충분한 양의 데이터가 확보되지 않을 것이 맹점으로 예상되었고, 실제 수행 결과 만족스러운 성능을 보이지 못했습니다.
- 두 번째로 추천 시스템에 사용되는 low-level의 요소에 대한 학습을 위하여 행렬 분해, 특이값 분해를 NumPy 라이브러리를 통해 구현 후 이를 활용하여 사용자가 시청한 영화 데이터에 기반한 영화 추천 시스템을 구현해 보았는데, 테스트 결과 상에서 사용자의 주관을 정확하게 반영되지 않는 현상을 높은 빈도로 확인하였고, 사용자의 특성과 주관적 체감이 중요하게 여겨지는 본 과제의 주제에 적용하기에는 다소 아쉬움이 있다고 판단되었습니다.
- 상대적으로 낮은 성능과 시간적 제약 문제로 우선 추천 시스템 라이브러리를 사용하는 방향으로 논의가 진행되었습니다. 추후 진행 과정에서 신경망을 통한 협업 필터링 방식의 추천 모델을 다시 구현해 볼 예정으로, 현재 관련 논문 탐독 및 배경 지식 조사가 진행되고 있는 상태입니다.

(2) LightFM

- 라이브러리 우선적 사용을 결정한 후 Surprise, Spotlight, LightFM 등 여러 추천 시스템 라이브러리 후보군 중 어떤 라이브러리를 사용할지에 대한 논의를 진행하였습니다.
- 후보군 조사 및 비교 결과, explicit 및 implicit 데이터 모두에 대응이 가능하고 패키지 버전 충돌의 제약이 비교적 덜한 LightFM 라이브러리를 사용하는 것으로 결정되었습니다.
- 협업 필터링 방식과 콘텐츠 기반 필터링 방식이 결합된 자체적 하이브리드 모델인 LightFM 모델을 가지고 있고, 연산 식은 아래와 같습니다.

$$q_u = \sum_{j \in f_u} e_j^U \quad p_i = \sum_{j \in f_i} e_j^I$$

$$b_u = \sum_{j \in f_u} b_j^U \quad b_i = \sum_{j \in f_i} b_j^I$$

$$\hat{r}_{ui} = \text{sigmoid}(q_u \odot p_i + b_u + b_i)$$

아이템 특성의 latent vector의 합과 사용자 특성의 latent vector의 합을 내적인 후 각각의 bias를 적용하여 sigmoid 함수로 처리, 최종적으로 추천 점수를 출력합니다. 이러한 과정에서 사용자의 행동뿐만 아닌 사용자 및 아이템의 특징까지 고려한 추천을 진행함으로써 cold start 문제를 완화할 수 있다는 장점이 긍정적으로 고려되었습니다.

- 위의 과정을 통해 출력된 추천 점수를 기반으로 상위 아이템들을 추려내고, 이를 언어 모델로 전송하여 최종적인 추천 응답을 출력할 수 있도록 구조를 설계하였습니다.

(3) BERT4Rec (변경 예정)

- 현재는 LightFM을 활용하여 추천 시스템을 우선 구성해 두었으나, 이전에 논의되었던 BERT4Rec 모델 사용에 대한 재논의가 진행되었습니다.
- 큰 시스템 설계에서 사용자가 시청하였던 영화의 정보를 입력받는 것으로 초기 동작 단계를 변경함에 따라, 순차적 데이터의 처리 및 문맥의 양방향적 이해에 용이한 BERT4Rec 모델에 대한 긍정적인 평가가 있었고, 이에 따라 추가적인 개선을 위하여 해당 모델에 대한 테스트가 진행 중입니다. 테스트 결과에서 치명적인 오류가 없고 어느 정도 만족스러운 결과가 도출된다면 현재 사용하고 있는 LightFM 모델을 대체하여 적용할 계획입니다.

3.2.2. 대형 언어 모델(LLM)

- 이전 작업 과정에서 GPT-3 모델과 LLaMA 3.1 모델 중 어떤 모델을 사용할 것인지에 대한 논의를 진행하였고, 최종적으로 비용과 최신성을 고려하여 LLaMA3.1 모델을 사용하기로 결정한 바가 있습니다.
- 이에 따라 LLaMA 3.1 모델 라인업 중 가장 경량화된 모델인 LLaMA 3.1-8b를 선정하여 작업을 진행하였고, 별도의 변동 사항은 없습니다.

4. 구성원별 진척도

이름	내용
박지환	<ul style="list-style-type: none">● 추천 시스템 직접 구현 시도 및 라이브러리 테스트 (LightFM, Surprise, Spotlight)● 대형 언어 모델 테스트 및 운용 코드 작성 (GPT-3, LLaMA 3.1-8b, xionic-ko-LLaMA-3-70b)● 데이터 전처리 진행
하현진	<ul style="list-style-type: none">● 웹 ui 구성 및 프론트엔드 개발(react)● 언어 모델 관련 프롬프트 엔지니어링● 데이터 전처리 진행
장재혁	<ul style="list-style-type: none">● 데이터베이스 설계 및 구현 (MySQL)● 백엔드 API 개발 및 통합 (Flask)● 추천 시스템과 프론트엔드 간 데이터 통신 설계 및 구현

5. 과제 수행 내용 및 중간 결과

5.1. 데이터

(1) 데이터 선정 및 전처리

kaggle에서 제공하는 영화 데이터셋([kaggle](#))을 사용하였습니다. 총 7개의 .csv파일로 이루어졌으며 그 중 특정 파일과 features를 위 기준에 맞춰 하나의 파일로 만드는 작업을 진행하였습니다. 파일과 features는 kaggle에서 제공된 데이터 설명과 대형 언어 모델에 전달되어야 할 영화 정보를 고려해 선정하였습니다. 원본 데이터에 저장된 data를 전부 text 형식으로 변환해 저장하기로 결정하였습니다.

(2) 파일 및 **feature** 설정

1. Credits.csv

- Cast : 'name'(출연 배우의 이름)값들만 별도로 추출하여 리스트화
- Crew : 'job'의 값이 'Director'인 항목의 'name'만 추출

2. Keywords.csv

- Id : 마찬가지로 별도 수정 없음
- Keywords : 'name' 값들만 추출하여 리스트화

3. Movies_metadata.csv

- Genres : 'name'값들만 추출하여 리스트화
- Id : 수정 사항 없음
- imdb_id : 각 값 앞의 'tt'만 제하고(숫자만) 남도록 수정
- Overview : 별도 수정 없음 → 이후 description 생성에서 활용 가능성
- Production_companies : 'name'값만 추출
- Production_countries : 'name'값만 추출
- Release_date : 연도값만 추출하여 release_year column으로 새로 저장
- Runtime : int로 형변환 외 수정사항 없음
- Spoken_language : 'name'값들만 추출하여 리스트화
- Title : 수정 사항 없음
- Vote_average : 수정 사항 없음, 현재 10점 만점 기준
- Vote_count : int로 형변환 외 수정사항 없음
- Original_language : spoken_language 활용 예정
- Original_title : title 활용 예정

(3) 전처리 결과

```
# 리스트로 저장되어 있는 컬럼을 전부 데이터를 스트링으로 바꾸기
list_columns = ['genres', 'actors', 'directors', 'keywords', 'production_companies', 'production_countries', 'spoken_languages']

for col in list_columns:
    merged_movie_data[col] = merged_movie_data[col].apply(lambda x: ', '.join(x) if isinstance(x, list) else '')

merged_movie_data
```

<ipython-input-7-be85837c04f3>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
merged_movie_data[col] = merged_movie_data[col].apply(lambda x: ', '.join(x) if isinstance(x, list) else '')

	id	title	genres	overview	actors	directors	keywords	imdb_id	production_companies	production_countries	release_year	runtime	spoken_languages	vote_average	vote_count
0	710	GoldenEye	Adventure, Action, Thriller	James Bond must unmask the mysterious head of ...	Pierce Brosnan, Sean Bean, Izabella Scorupco, ...	Martin Campbell	cuba, falsely accused, secret identity, comput...	0113189	United Artists, Eon Productions	United Kingdom, United States of America	1995	130	English, Русский, Español	6.6	1194
1	10634	Friday	Comedy	Craig and Smokey are two guys in Los Angeles h...	Ice Cube, Chris Tucker, Nia Long, Tom Lister J...	F. Gary Gray	rap music, parent child relationship, rapper, job	0113118	New Line Cinema	United States of America	1995	91	English	7.0	513
2	755	From Dusk Till Dawn	Horror, Action, Thriller, Crime	Seth Gecko and his younger brother Richard are...	George Clooney, Quentin Tarantino, Harvey Keit...	Robert Rodriguez	dancing, brother relationship, sexual ...	0116367	A Band Apart, Dimension Films, Los Hooligans P...	United States of America	1996	108	English, Español	6.9	1644
3	5894	Blue in the Face	Comedy	Auggie runs a small tobacco shop in Brooklyn, ...	Harvey Keitel, Lou Reed, Michael J. Fox, Rosea...	Paul Auster, Wayne Wang	smoking, corner shop, cigarette, tobacco, ciga...	0112541	Miramax Films, InterAL	United States of America	1995	83	English	6.8	28
4	9070	Mighty Morphin Power Rangers: The Movie	Action, Adventure, Science Fiction, Family, Fa...	Power up with six incredible teens who out-ma...	Amy Jo Johnson, Jason David Frank, David Yost...	Bryan Spicer	based on tv series, tokusatsu, superhero team...	0113820	Twentieth Century Fox Film Corporation, Saban ...	Japan, United States of America	1995	92	English	5.2	153
...

merged_movie_data.csv 파일의 최종 결과입니다. 모든 feature는 모두 string으로 변환해 저장하였습니다. 이는 영화에 대한 설명 정보를 파이썬으로 쉽게 처리하고, 생성된 설명 정보를 대형 언어 모델의 입력으로 삼입하기 위함입니다.

(4) 영화 설명 데이터 생성

```
# 각 영화에 대한 설명 생성
merged_movie_data['description'] = merged_movie_data.apply(create_description, axis=1)

# 결과 출력
print(merged_movie_data['description'][0])
```

```
##GoldenEye## (1995)
##Overview:## James Bond must unmask the mysterious head of the Janus Syndicate and prevent the leader from utilizing the GoldenEye weapons system to inflict devastating revenge on Britain.
##Genres:## Adventure, Action, Thriller
##Starring:## Pierce Brosnan, Sean Bean, Izabella Scorupco, Famke Janssen, Joe Don Baker, Judi Dench, Gottfried John, Robbie Coltrane, Alan Cumming, Tch ky Karyo, Desmond Llewelyn, Samantha Bond, Michael Kitchen,
##Directed by:## Martin Campbell
##Keywords:## cuba, falsely accused, secret identity, computer virus, secret base, secret intelligence service, kgb, satellite, special car, cossack, electromagnetic pulse, time bomb, st. petersburg russia, ejecti
##Production Companies:## United Artists, Eon Productions
##Production Countries:## United Kingdom, United States of America
##Runtime:## 130 minutes
##Spoken Languages:## English, Русский, Espa ol
##Vote Average:## 6.6 (based on 1194 votes)
```

전처리한 데이터를 이용한 영화 설명 생성 함수의 결과입니다. 이는 후에 사용자의 설명, 생성된 영화 후보군과 함께 대형 언어 모델의 입력으로 전달되어 최종 추천 영화 선정에 사용됩니다.

5.2. 추천 시스템

```
from lightfm.datasets import fetch_movielens
data = fetch_movielens(min_rating=4.0)
```

- LightFM 라이브러리, 그리고 테스트용 데이터셋으로 내장된 movielens 데이터셋을 사용하기 위해 import하였습니다. 또한, 최소 평점이 4.0 이상인 데이터만 추려내었습니다.

```
import pandas as pd
import numpy as np
# 영화 제목을 별도로 로드
url = "https://files.grouplens.org/datasets/movielens/ml-100k/u.item"
movie_titles = pd.read_csv(url, sep='|', encoding='latin-1', header=None, usecols=[0, 1], names=['movie_id', 'title'])
movie_titles['movie_id'] = movie_titles['movie_id'] - 1 # ID가 1부터 시작하므로 0부터 시작하도록 조정

train_data = data['train']
test_data = data['test']
```

- 작업 진행 결과, 데이터셋 중에서 'title' 열에서 문제가 있는 것으로 확인되어, 추가적으로 영화의 제목을 가지는 데이터셋을 로드, 'movie_id' 값과 함께 준비하였습니다.
- 그 후, 위에서 로드된 데이터를 테스트 세트와 훈련 세트로 분할하였습니다.

```

from lightfm import LightFM
from lightfm.evaluation import precision_at_k
from lightfm.evaluation import auc_score
model = LightFM(loss='warp', learning_rate=0.05, no_components=30)
%time model.fit(train_data, epochs=200, num_threads=4)

train_precision = precision_at_k(model, train_data, k=10).mean()
test_precision = precision_at_k(model, test_data, k=10).mean()

train_auc = auc_score(model, train_data).mean()
test_auc = auc_score(model, test_data).mean()

print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))

```

CPU times: user 19.9 s, sys: 52.8 ms, total: 20 s

Wall time: 11.9 s

Precision: train 0.71, test 0.05.

AUC: train 0.99, test 0.90.

- 라이브러리를 활용, 자체 LightFM 클래스를 호출합니다.
 - 손실 함수는 WARP(Weighted Approximate Rank Pairwise)를 사용합니다. 샘플링을 통하여 사용자가 선호하는 항목과 비선호 항목의 순위 차 정보를 최대화하고, 기존 순위가 낮고 잘못 배정된 항목에 대해서는 더 큰 페널티를 부여합니다.
 - 학습률은 0.05, 컴포넌트의 수(잠재 벡터 길이)는 30으로 설정하였습니다.
- 이후 훈련 데이터에 대한 모델의 학습을 진행합니다.
 - epoch 수는 200, 사용할 쓰레드의 수는 4개로 설정하였습니다.
- 그리고, precision@10 및 auc 점수 측정 결과를 출력하였습니다.

- precision의 경우 대부분의 학습에서 테스트 세트 측정시 매우 낮은 수치가 기록되었습니다. 이에 따라 , 추가적인 개선(또는 추천 시스템 모델 변경)이 계획되어 있습니다.
- AUC 점수의 경우 테스트, 훈련 모두에서 0.9 이상의 수치를 기록하였습니다.

```
import numpy as np

def recommender(user_id, model, data, items=10):
    user_x = user_id
    scores = model.predict(user_x, np.arange(data['train'].shape[1]))

    item_scores = [(item_id, score) for item_id, score in zip(np.arange(data['train'].shape[1]), scores)]
    top_items = sorted(item_scores, key=lambda x: x[1], reverse=True)[:items]

    return top_items
```

```
recommendations = recommender(1, model, data)
print("추천 영화 : ")
for item_id, score in recommendations:
    print(f"영화 ID {item_id} - 점수 {score:.2f}")
```

```
추천 영화 :
영화 ID 285 - 점수 1.40
영화 ID 301 - 점수 1.37
영화 ID 274 - 점수 1.23
영화 ID 299 - 점수 1.15
영화 ID 126 - 점수 0.96
영화 ID 99 - 점수 0.89
영화 ID 312 - 점수 0.80
영화 ID 284 - 점수 0.77
영화 ID 302 - 점수 0.68
영화 ID 254 - 점수 0.58
```

- 영화 id값과 점수 값을 출력하여 확인하기 위한 테스트용 함수입니다.

id 배열을 생성한 뒤 predict 메소드를 사용, 해당 사용자의 id값과 영화의 id값 및 데이터셋 내 사용자 특성, 평점 기록, 아이템 특성을 활용하여 예측을

수행합니다. 최종적으로, 추천하는 영화의 id값과 상대적인 점수를 출력합니다.

```
# 추천 함수 정의
def recommend_movies_with_titles(user_id, model, data, movie_titles, num_items=10):
    user_x = user_id
    scores = model.predict(user_x, np.arange(data.shape[1]))
    top_items = np.argsort(-scores)[:num_items]
    recommended_movies = [(movie_titles.iloc[i]['title'], scores[i]) for i in top_items]
    return recommended_movies

# 사용자 1에게 추천 생성 및 제목으로 출력
recommendations = recommend_movies_with_titles(123, model, train_data, movie_titles)
print("추천 영화:")
for title, score in recommendations:
    print(f"영화 제목: {title} - 점수 {score:.2f}")
```

```
추천 영화:
영화 제목: Blade Runner (1982) - 점수 3.21
영화 제목: Raiders of the Lost Ark (1981) - 점수 3.04
영화 제목: Pulp Fiction (1994) - 점수 3.01
영화 제목: Twelve Monkeys (1995) - 점수 2.92
영화 제목: Monty Python and the Holy Grail (1974) - 점수 2.82
영화 제목: Alien (1979) - 점수 2.77
영화 제목: Terminator, The (1984) - 점수 2.62
영화 제목: Die Hard (1988) - 점수 2.59
영화 제목: Monty Python's Life of Brian (1979) - 점수 2.58
영화 제목: Star Wars (1977) - 점수 2.54
```

- 영화 제목 데이터와의 연동 및 추천 결과 출력을 위한 함수를

정의하였습니다. 추천 점수 기준으로 정렬 후, num_items개 만큼의 상위
아이템을 남긴 후에 이들의 id 값을 기준으로 영화 제목을 찾아 함께
출력합니다.

- 위 결과는 주어진 데이터셋 내의 123번 사용자의 평가 기록 및 사용자 특성,
영화 특성을 고려한 결과로 추가적인 데이터셋 증강 및 평가 기록 갱신을
통하여, 보다 최신화된 영화들이 추천되도록 조정이 가능할 것으로 생각되어
개선 계획입니다.

5.3. 대형 언어 모델

```
!pip install transformers accelerate
!pip install --upgrade torch torchvision
!pip install bitsandbytes
!pip install openai
```

- Hugging Face를 활용한 언어 모델 사용을 위한 transformers, 모델 학습과 추론을 위한 accelerate, torch, 모델의 양자화 및 최적화를 위한 bitsandbytes, 그리고 추가적인 GPT 관련 테스트를 위한 openai 라이브러리를 설치하였습니다.

```
import torch
from transformers import BitsAndBytesConfig

bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)
```

- 모델의 메모리 사용량 조정을 위한 양자화 설정(BitsAndBytesConfig)을 진행했습니다.
- 모델을 4비트로 로드, 이중 양자화를 사용하고 양자화 방식은 NF4, 연산에 사용할 데이터 타입은 bfloat16으로 설정했습니다.

```
[3] from huggingface_hub import login
```

- 개인 토큰을 사용하여 언어 모델 로드를 위해 HuggingFace의 모델 허브에 로그인하였습니다(login 호출부는 개인 토큰을 포함하여 이미지를 절삭함).

```
from transformers import AutoTokenizer, AutoModelForCausalLM
model_id = "meta-llama/Meta-Llama-3-8B"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForCausalLM.from_pretrained(model_id, quantization_config=bnb_config, device_map={"":0})
```

- 모델 허브에서 Meta-Llama-3-8b 모델과 상응하는 토크나이저를 로드하였습니다(AutoTokenizer, AutoModelForCausalLM).
- 로드 옵션으로 상기한 양자화 설정을 적용하고(quantization_config), 모델을 GPU에 로드하였습니다(device_map).

```
# 이전 대화 턴의 내용을 저장할 변수
conversation_history = ""
max_length = 1000 # 문맥 길이 제한
```

- 테스트 결과, 이전에 진행하였던 대화의 내용을 반영하지 못하고 매번 새로운 대화의 시작과 유사한 의미의 발화를 하는 빈도가 높았던 것을 확인하였고, 대화 기록을 어느 정도 형식화하여 모델에 전송하기 위한 별도의 history 변수를 선언하였습니다.

```

# 대화 루프
while True:
    # 사용자 입력 받기
    user_input = input("user: ")

    # 이전 대화 내용과 사용자 입력을 합쳐 모델 입력 생성
    model_input = conversation_history + "user: " + user_input + "\nassistant:"

    # 문맥 길이 제한 적용
    if len(model_input) > max_length:
        model_input = model_input[-max_length:]

    # 토큰화
    input_ids = tokenizer.encode(model_input, return_tensors="pt").to(model.device)

    attention_mask = torch.ones_like(input_ids)

    # 모델 실행
    output = model.generate(
        input_ids,
        attention_mask=attention_mask,
        max_new_tokens=200,
        pad_token_id=tokenizer.eos_token_id,
        no_repeat_ngram_size=2, # 반복 방지
        num_beams=5, # 빔 서치 적용으로 더 논리적인 응답 생성
    )

    # 생성된 텍스트 디코딩
    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)

    # 챗봇 답변 추출 (응답 부분만 잘라내기)
    assistant_response = generated_text[len(model_input):].strip()

    # 답변 출력
    print("assistant: " + assistant_response)

    # 대화 내용 업데이트 (정확한 포맷으로 대화 내용 추가)
    conversation_history += "user: " + user_input + "\nassistant: " + assistant_response + "\n"

```

- 반복 대화 테스트를 위하여 루프를 설정, 지속적으로 대화 기록을 모델의 입력값으로 할당하도록 설정하였습니다.
- 이후, 모델 입력값을 토큰화한 후 input_ids와 동일한 모양을 갖는 tensor를 활용, attention_mask를 생성합니다. 그리고 모델의 출력값을 생성합니다

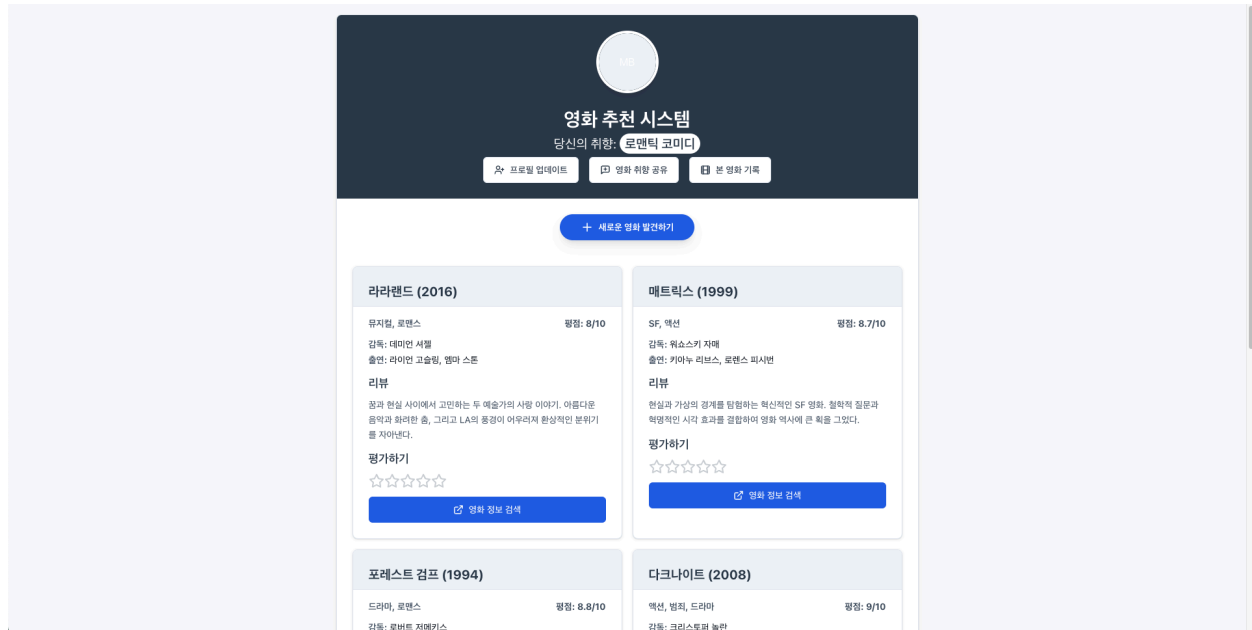
- 입력으로 토큰화한 `input_ids`를 사용합니다.
- `attention_mask`를 배정하여 유효한 토큰을 표시합니다.
- 문장의 끝을 나타내는 `eos_token`의 id로 패딩 토큰 id를 설정합니다.
- 두 단어가 동일하게 조합되지 않도록(`2_gram`) 방지하여 다양성을 높입니다.
- 마지막으로, 응답 생성에서 사용되는 빔 서치 방식의 빔 수를 5개 경로로 설정합니다. (연산 시간을 고려하여 우선 5로 설정하였음)
- 최종적으로 응답 출력 후 대화 기록에 추가합니다.

user: 한국의 로맨스 영화 두 가지만 추천해줘.
 한국의 로맨스 영화 중 추천하는 두 작품은 다음과 같습니다. 첫 번째는 '이창동 감독'의 '버닝'입니다. 이 영화는 2018년에 개봉했으며, 하루의 삶에 대한 묵상을 통해 사랑과 외로움을 그리.
 user: 박하사탕 영화에 대해 더 자세히 알려줄래?
 박하사탕은 2000년에 개봉된 대한민국의 영화로, 김지운 감독이 연출하였다. 이 영화는 1980년대 초반에 서울에서 일어난 실제 살인 사건을 기반으로 만들어졌다. 영화의 제목인 '박하사탕'은
 user: 미국의 액션 영화 세 가지만 추천해줘
 다음은 미국의 액션 영화 세 가지를 추천합니다. 첫 번째는 1988년에 개봉한 'Die Hard'입니다. 이 영화는 브루스 윌리스가 주연을 맡았으며, 크리스마스 파티 중에 테러 사건이 발생하면서 경

- 대화 기록 상에서 한국어 인식 및 한국어 답변 생성 부분에서는 큰 문제가 없음을 확인하였습니다.
- 다만, 특정 글자가 깨지는 현상 및 영화 설명 상에서 `hallucination` 문제가 다소 있음을 확인하였고, 이는 추가 데이터셋을 통한 파인 튜닝으로 개선할 계획입니다.

5.4. UI

(1) 요소 설명



react를 이용해 기능을 테스트할 수 있도록 제작한 웹페이지입니다. 아래는 각 요소에 대한 설명입니다.

1. 프로필 업데이트, 본 영화 기록 : 사용자의 시청 기록, 나이, 성별등 기초적인 정보를 입력합니다. 사용자가 입력한 내용을 바탕으로 대형 언어 모델을 통해 사용자 페르소나를 생성합니다.
2. 영화 취향 공유 : 특정 배우, 감독과 같이 영화 정보에 관한 내용, 특정 테마에 맞는 영화 추천등 영화 추천을 위한 추가적인 정보를 제공합니다. 해당 정보는 대형 언어 모델의 입력으로 전달되어 추천에 추가적인 정보로 사용됩니다.

3. 영화 카드 : 영화에 대한 간략한 정보를 나타냅니다.

4. 새로운 영화 추천 : 추가적인 영화 추천을 받을 수 있습니다.

5. 영화 정보 검색 : 클릭시 즉시 해당 영화에 대한 정보를 검색하는 창으로
이동해 본인이 이용하는 ott 및 개봉관을 찾아 시청에 용이하도록 합니다.

모든 요소는 사용자 경험 개선, 시스템 구조상의 이유로 개발 중 추가, 수정할 수
있습니다.

5.5. 프롬프트 엔지니어링

5.5.1. 목적

(1) 정확도 개선 및 추론 이유 설명

- CoT(Chain of Thought)방법론은 모델에게 정답을 내기 전 rationale을 이용한 추론 과정을 거치도록 하는 방법입니다
- 해당 방법론을 이용해 추천의 정확도와 추론 과정을 output으로 내도록 합니다. 이를 통해 유저의 취향을 분석하고 보다 정확한 추론을 할 수 있도록 합니다.

(2) hallucination 감소

- 대형 언어 모델의 입력으로 전달되는 영화 후보군을 추천시스템에서 생성하고 해당 영화 정보를 DB에서 가져와 잘못된 정보를 제공하는 경우를 없애 hallucination을 줄입니다.

5.5.2. 프롬프트 실험

- 5.5.1에 언급한 내용을 통해 두가지 버전의 프롬프트를 GPT-4O에서 테스트 했습니다. original은 전체 내용을 자세하게 작성하여 결과를 상세하게 받을 수 있었고 본래 목적인 추천에 대한 이유와 페르소나에 대한 적절한 추천을 얻을 수 있었습니다.
- optimized 버전은 실제 서비스에 적용될 때 비용 및 생성 시간을 고려해 짧게 만들었습니다. 그 결과를 확인했을 때 한번에 납득하기 어려운 짧고 간단한 설명뿐이라 실제 서비스에 적용시켰을 때 차이를 보고 수정할 계획입니다.

- original 버전이 적절한 검증 질문을 생성합니다. 이를 적절히 가공해서 사용자에게 제공하면 본인의 취향을 분석하고 다음 영화를 선택하는데 도움이 될거라 기대합니다.

(1) 실험 결과 및 전달 내용

페르소나 , 영화 후보 내용	<p>사용자 페르소나: 이름: 민수 (35세, 남성) 직업: IT 개발자 영화 취향: SF, 스릴러, 심리 드라마를 선호. 복잡한 줄거리와 예측하기 어려운 반전이 있는 영화를 좋아함. 다큐멘터리나 지나치게 느린 영화는 선호하지 않음. 여가 시간: 주말에 주로 영화를 감상하며, 기술적인 부분이나 철학적 주제가 포함된 작품을 좋아함.</p> <p>영화 후보 리스트:</p> <p>Inception (인셉션) - SF, 스릴러, 심리</p> <p>The Matrix (매트릭스) - SF, 액션, 철학</p> <p>Shutter Island (셔터 아일랜드) - 스릴러, 심리</p> <p>The Social Network (소셜 네트워크) - 드라마, 전기</p> <p>Interstellar (인터스텔라) - SF, 드라마, 모험</p> <p>Se7en (세븐) - 스릴러, 범죄</p> <p>Blade Runner 2049 (블레이드 러너 2049) - SF, 드라마, 미스터리</p> <p>Memento (메멘토) - 스릴러, 미스터리, 심리</p> <p>Gone Girl (나를 찾아줘) - 스릴러, 드라마</p> <p>Ex Machina (엑스 마키나) - SF, 스릴러, 드라마 추천 수: 5편</p>
--------------------------	---

original	optimized
당신은 영화 추천 전문가입니다. 미리 정의된 사용자 페르소나와 추천 시스템에서 생성된 영화 후보 목록을 바탕으로 최적의 영화를 추천해야 합니다. 다음 단계를 따라 추천 과정을 진행하세요:	당신은 영화 추천 전문가입니다. 미리 정의된 사용자 페르소나와 추천 시스템에서 생성된 영화 후보 목록을 바탕으로 최적의 영화를 추천해야 합니다. 다음 단계를 따라 추천 과정을 진행하세요: 사용자 페르소나: {사용자_페르소나} (이미

<p>사용자 페르소나: {사용자_페르소나} (이미 정의됨) 영화 추천 후보: {영화_후보_리스트} 추천할 영화 수: {N}</p> <p>1. 후보 영화 평가: 각 후보 영화에 대해 간단히 설명하고, 페르소나와의 적합성을 평가하세요. 적합성 점수를 1-10 사이로 매기고, 그 이유를 간단히 설명하세요. 2. 최종 추천: - 평가를 바탕으로 가장 적합한 {N}개의 영화를 선정하세요. - 각 추천 영화에 대해 왜 이 사용자에게 적합한지 간단히 설명하세요. 3. 검증 질문: - 추천한 영화들이 사용자 페르소나와 일치하는지 확인하는 질문을 스스로에게 하세요. - 질문에 답하며 추천의 타당성을 검증하세요. 4. 최종 출력: - 검증을 거친 최종 추천 영화 {N}개를 리스트로 제시하세요. 각 단계를 상세히 설명하고, 최종 추천 결과를 명확하게 제시해 주세요.</p>	<p>정의됨) 영화 후보: {영화_리스트} 추천 수: {N}</p> <p>1. 각 영화 적합성 점수 매기기 (1-10) 2. 상위 {N}편 선택 및 간단한 이유 제시 3. 빠른 검증 최종 추천 영화 {N}편을 리스트로 제시하세요.</p>
<p style="text-align: center;">결과</p>	
<p>1. 후보 영화 평가</p> <p>1. Inception (인셉션)</p> <ul style="list-style-type: none"> ○ 장르: SF, 스릴러, 심리 ○ 설명: 꿈속의 꿈을 다루는 복잡한 줄거리와 독창적인 설정이 특징입니다. 민수의 SF와 심리적 요소를 선호하는 취향에 잘 맞습니다. 또한, 예측 불가능한 반전이 포함되어 있어 더욱 흥미를 끌 것입니다. ○ 적합성 점수: 10/10 ○ 이유: 민수의 영화 취향에 완벽히 부합하는 영화로, SF와 심리적 요소가 강하게 결합되어 있으며, 복잡한 	<p>1. 각 영화 적합성 점수 매기기 (1-10)</p> <ul style="list-style-type: none"> ● Inception (인셉션) - 10 복잡한 줄거리와 심리적인 요소가 돋보이는 작품으로, 민수가 좋아하는 SF와 스릴러 요소를 모두 갖추고 있습니다. ● The Matrix (매트릭스) - 10 SF와 철학적 주제를 잘 결합한 영화로, 민수의 취향에 매우 부합합니다. 기술적인 부분도 강조되어 있습니다. ● Shutter Island (셔터 아일랜드) - 9 심리 스릴러의 대표작으로, 예측할 수 없는 반전과 복잡한 스토리가 매력적입니다. ● The Social Network (소셜 네트워크) - 6 IT와 관련된 전기 영화로, 민수의 직업과 어느 정도 연관이 있지만, SF나

줄거리가 민수의 흥미를 끌
것입니다.

2. The Matrix (매트릭스)

- 장르: SF, 액션, 철학
- 설명: 가상현실과 철학적 주제를 다루며, 혁신적인 비주얼과 스토리로 유명합니다. 기술적 요소와 철학적 질문들이 민수에게 매력적으로 다가올 것입니다.
- 적합성 점수: 9/10
- 이유: 민수가 선호하는 SF와 철학적 주제를 다루며, 혁신적인 스토리라인이 민수의 지적 호기심을 자극할 것입니다.

3. Shutter Island (셔터 아일랜드)

- 장르: 스릴러, 심리
- 설명: 미스터리와 심리적 요소가 강한 영화로, 예측 불가능한 반전이 특징입니다. 민수가 좋아하는 심리적 긴장감과 복잡한 줄거리가 돋보입니다.
- 적합성 점수: 9/10
- 이유: 민수가 선호하는 스릴러와 심리 요소가 중심이 되어 있으며, 영화의 반전이 강한 인상을 남길 것입니다.

4. The Social Network (소셜 네트워크)

- 장르: 드라마, 전기
- 설명: 페이스북 창립 과정과 관련된 이야기로, 기술적 배경이 있지만 드라마적 요소가 중심입니다. 민수의 취향과는 다소 거리가 있을 수 있습니다.
- 적합성 점수: 6/10
- 이유: 기술적 배경이 있지만, 드라마적 요소가 강해 민수의 주요 영화 취향과는 다소 다를 수 있습니다.

스릴러 요소가 부족합니다.

- **Interstellar (인터스텔라)** - 9
SF와 철학적 주제가 결합된 작품으로, 깊이 있는 줄거리와 감정적인 여정이 돋보입니다.
- **Se7en (세븐)** - 8
어두운 범죄 스릴러로, 긴장감 넘치는 스토리와 예측할 수 없는 결말이 인상적입니다.
- **Blade Runner 2049 (블레이드 러너 2049)** - 9
SF와 철학적 주제가 잘 녹아 있으며, 느린 템포에도 불구하고 시각적 효과와 심오한 주제가 매력적입니다.
- **Memento (메멘토)** - 10
복잡한 내러티브와 심리적 요소가 결합된 작품으로, 민수가 좋아하는 스릴러와 미스터리 요소가 강합니다.
- **Gone Girl (나를 찾아줘)** - 8
심리 스릴러로, 강력한 반전과 복잡한 캐릭터가 인상적이지만, 다른 후보들에 비해 SF 요소가 부족합니다.
- **Ex Machina (엑스 마키나)** - 9
인공지능과 철학적 주제를 다룬 SF 스릴러로, 기술적 요소와 심리적 긴장감이 돋보입니다.

2. 상위 5편 선택 및 간단한 이유 제시

1. **Inception (인셉션)** - 복잡한 줄거리와 심리적 요소가 민수의 취향과 완벽하게 맞아떨어집니다.
2. **The Matrix (매트릭스)** - SF와 철학적 주제를 결합한 영화로, 기술적인 요소와 철학적 고찰이 매력적입니다.
3. **Memento (메멘토)** - 비선형적인 이야기 구조와 강력한 심리적 스릴이 민수의 영화 취향과 부합합니다.
4. **Interstellar (인터스텔라)** - SF와 철학적 주제가 결합되어 있으며, 감정적 깊이와 스토리텔링이 뛰어납니다.
5. **Ex Machina (엑스 마키나)** - 인공지능과 인간의 심리를 탐구하는 영화로, 철학적이고 기술적인 주제가 돋보입니다.

<p>5. Interstellar (인터스텔라)</p> <ul style="list-style-type: none"> ○ 장르: SF, 드라마, 모험 ○ 설명: 우주 탐사를 중심으로 한 SF 영화로, 철학적 질문과 기술적 요소가 깊이 있게 다뤄집니다. 민수의 SF 취향과 잘 맞으며, 복잡한 주제와 감성적인 요소가 결합된 작품입니다. ○ 적합성 점수: 8/10 ○ 이유: SF와 철학적 주제를 다루며, 민수의 취향에 부합하지만, 다소 감성적인 부분이 있어 취향에 따라 평가가 갈릴 수 있습니다. <p>6. Se7en (세븐)</p> <ul style="list-style-type: none"> ○ 장르: 스릴러, 범죄 ○ 설명: 연쇄살인범을 추적하는 이야기를 다루며, 심리적 긴장감과 암울한 분위기가 특징입니다. 민수가 선호하는 예측 불가능한 스토리라인을 제공합니다. ○ 적합성 점수: 8/10 ○ 이유: 민수의 스릴러 선호도에 부합하지만, 다소 어두운 범죄 요소가 강해 취향에 따라 평가가 갈릴 수 있습니다. <p>7. Blade Runner 2049 (블레이드 러너 2049)</p> <ul style="list-style-type: none"> ○ 장르: SF, 드라마, 미스터리 ○ 설명: 미래의 인간과 인공지능의 관계를 탐구하는 영화로, 시각적으로 놀라운 연출과 깊이 있는 철학적 질문이 특징입니다. 민수의 SF와 철학적 주제 선호에 적합합니다. ○ 적합성 점수: 9/10 ○ 이유: 민수가 좋아하는 SF와 철학적 주제를 심도 있게 	<p>3. 빠른 검증</p> <p>이 영화들은 모두 민수가 선호하는 SF, 스릴러, 심리 드라마 장르에 속하며, 복잡한 줄거리와 예측하기 어려운 반전 요소를 포함하고 있습니다. 또한, 철학적 주제와 기술적인 요소가 포함되어 있어 민수의 영화 취향에 적합합니다.</p> <p>최종 추천 영화 리스트 (5편)</p> <ol style="list-style-type: none"> 1. Inception (인셉션) 2. The Matrix (매트릭스) 3. Memento (메멘토) 4. Interstellar (인터스텔라) 5. Ex Machina (엑스 마키나)
--	---

다루며, 기술적 연출도 뛰어난 작품입니다.

8. **Memento (메멘토)**

- 장르: 스릴러, 미스터리, 심리
- 설명: 주인공의 기억 상실을 중심으로 전개되는 비선형적인 이야기 구조가 특징입니다. 복잡한 스토리와 예측 불가능한 전개가 민수의 취향에 잘 맞을 것입니다.
- 적합성 점수: 10/10
- 이유: 비선형적인 스토리와 강한 심리적 요소가 민수의 취향에 완벽히 부합합니다.

9. **Gone Girl (나를 찾아줘)**

- 장르: 스릴러, 드라마
- 설명: 복잡한 인물 관계와 반전이 특징인 심리 스릴러입니다. 민수의 스릴러와 심리적 요소 선호에 맞는 영화입니다.
- 적합성 점수: 8/10
- 이유: 복잡한 인물 관계와 심리적 긴장감이 민수의 취향에 부합하지만, 드라마적 요소가 다소 많을 수 있습니다.

10. **Ex Machina (엑스 마키나)**

- 장르: SF, 스릴러, 드라마
- 설명: 인공지능과 인간의 관계를 탐구하는 영화로, 기술적 주제와 심리적 긴장감이 잘 조화된 작품입니다. 민수의 SF와 심리적 요소 선호에 적합합니다.
- 적합성 점수: 9/10
- 이유: SF와 심리적 요소가 결합된 영화로, 민수의 취향에 매우 적합합니다.

2. 최종 추천

1. Inception (인셉션)

- 이유: 복잡한 줄거리와 예측 불가능한 반전, SF와 심리적 요소가 결합된 완벽한 작품으로, 민수의 취향에 가장 잘 맞습니다.

2. Memento (메멘토)

- 이유: 비선형적인 스토리 전개와 강한 심리적 요소로, 민수의 지적 호기심을 자극할 것입니다.

3. The Matrix (매트릭스)

- 이유: SF와 철학적 주제를 다루며, 혁신적인 스토리와 시각적 요소가 민수에게 매력적일 것입니다.

4. Shutter Island (셔터 아일랜드)

- 이유: 예측 불가능한 반전과 심리적 긴장감이 특징인 스릴러로, 민수의 스릴러 취향에 부합합니다.

5. Blade Runner 2049 (블레이드 러너 2049)

- 이유: SF와 철학적 주제를 심도 있게 탐구하며, 시각적으로도 뛰어난 작품으로, 민수에게 인상적인 경험을 제공할 것입니다.

3. 검증 질문

- 이 영화들이 SF, 스릴러, 심리 드라마에 대한 민수의 선호에 부합하는가?
 - 예, 모든 영화는 민수가 선호하는 장르와 요소들을 포함하고 있으며, 복잡한 줄거리와 예측 불가능한 전개를 제공합니다.
- 민수의 기술적 관심과 철학적 주제를 다루는 영화가 포함되어 있는가?

<ul style="list-style-type: none"> ○ 예, 'The Matrix', 'Blade Runner 2049', 'Ex Machina' 등이 이러한 요소를 잘 다루고 있습니다. ● 민수가 다큐멘터리나 느린 전개를 선호하지 않음을 고려했는가? <ul style="list-style-type: none"> ○ 예, 선택된 영화들은 모두 긴장감 있는 전개와 빠른 속도로 진행됩니다. <p>4. 최종 출력</p> <ul style="list-style-type: none"> ● 추천 영화 리스트: <ol style="list-style-type: none"> 1. Inception (인셉션) 2. Memento (메멘토) 3. The Matrix (매트릭스) 4. Shutter Island (셔터 아일랜드) 5. Blade Runner 2049 (블레이드 러너 2049) 	
--	--