

BookCurator

: AI 기반 맞춤형 도서 추천 시스템

기존 문제점

1

기존 공공도서관 도서 검색
시스템은 키워드 기반 작동
→ 직관적 검색의 한계

2

독자의 특성과 관심사를
고려한 ‘머신러닝+LLM’
하이브리드 방식의
도서 추천 시스템 부재

3

사용자의 위치를 기반으로,
해당 추천 도서를 소장하고
있는 도서관의 정보를
제공하는 시스템의 부재

1

사용자가 사용하기 편리한
LLM 프롬프트를 결합해
비교적 쉽게 맞춤 도서를
추천 받을 수 있음

2

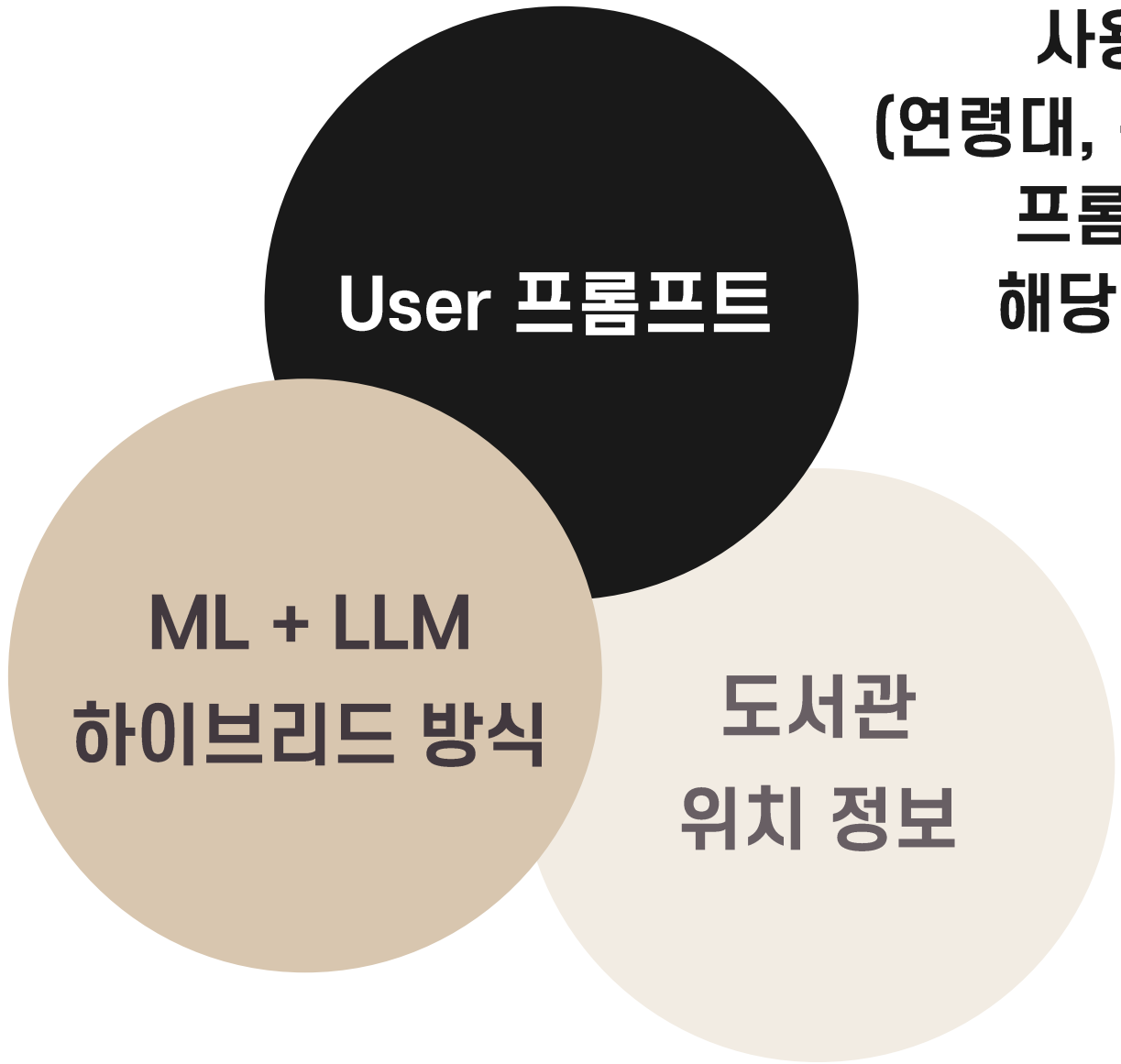
ML 기반의 단순 도서 추천에
LLM을 결합해
해당 도서의 추천 이유 및
읽는 순서까지 제공

3

사용자의 위치를 기반으로,
해당 추천 도서를 소장하고
있는 도서관의 정보를
제공하는 시스템의 부재

주요 기능

- ML: 사용자의 프로필에 맞는 최적의 도서 후보군 선정
- LLM: GPT 모델을 활용한 상세 추천 이유와 설명 제공



사용자가 자신의 프로필 정보
(연령대, 관심사, 도서 목적, 도서 수준)를
프롬프트에 문장으로 입력하면
해당 사용자의 맞춤 도서를 추천

사용자의 위치에 기반해
추천 도서가 소장되어 있는
가까운 도서관 정보 제공

사용한 데이터셋

인기대출 데이터

loan_information.csv

- 사용자의 성별, 연령, 제목, 출판사, 저자명, 도서별 정보를 위해 사용

도서관 정보 데이터

library.csv

- 사용자의 위치와 가까운 도서관의 정보를 위해 사용





데이터 전처리 (1) - 인기 대출 도서

1

칼럼
mapping

- 'RANK_CO': 'rank',
- 'BOOK_TITLE_NM': 'title',
- 'AUTHR_NM': 'author',
- 'BOOK_INTRCN_CN': 'description',
- 'PUBLISHER_NM': 'publisher',
- 'AGE_FLAG_NM': 'age_group',
- 'SEXDSTN_FLAG_NM': 'gender'

2

결측치
처리

- title, author, publisher가 없는 행 제거
- description, age_group, gender 없으면 '설명 없음', '전체'로 채워 넣음

3

데이터
정제

- 문자열에서 공백 및 특수문자 제거
- 시리즈 여부 추정
- 도서 장르 8개 카테고리 생성(이 외엔 '기타' 장르로 분류)
- 예상 독자층, 도서 장르, 연령대 추정 (도서 설명 text에서 키워드 추출)

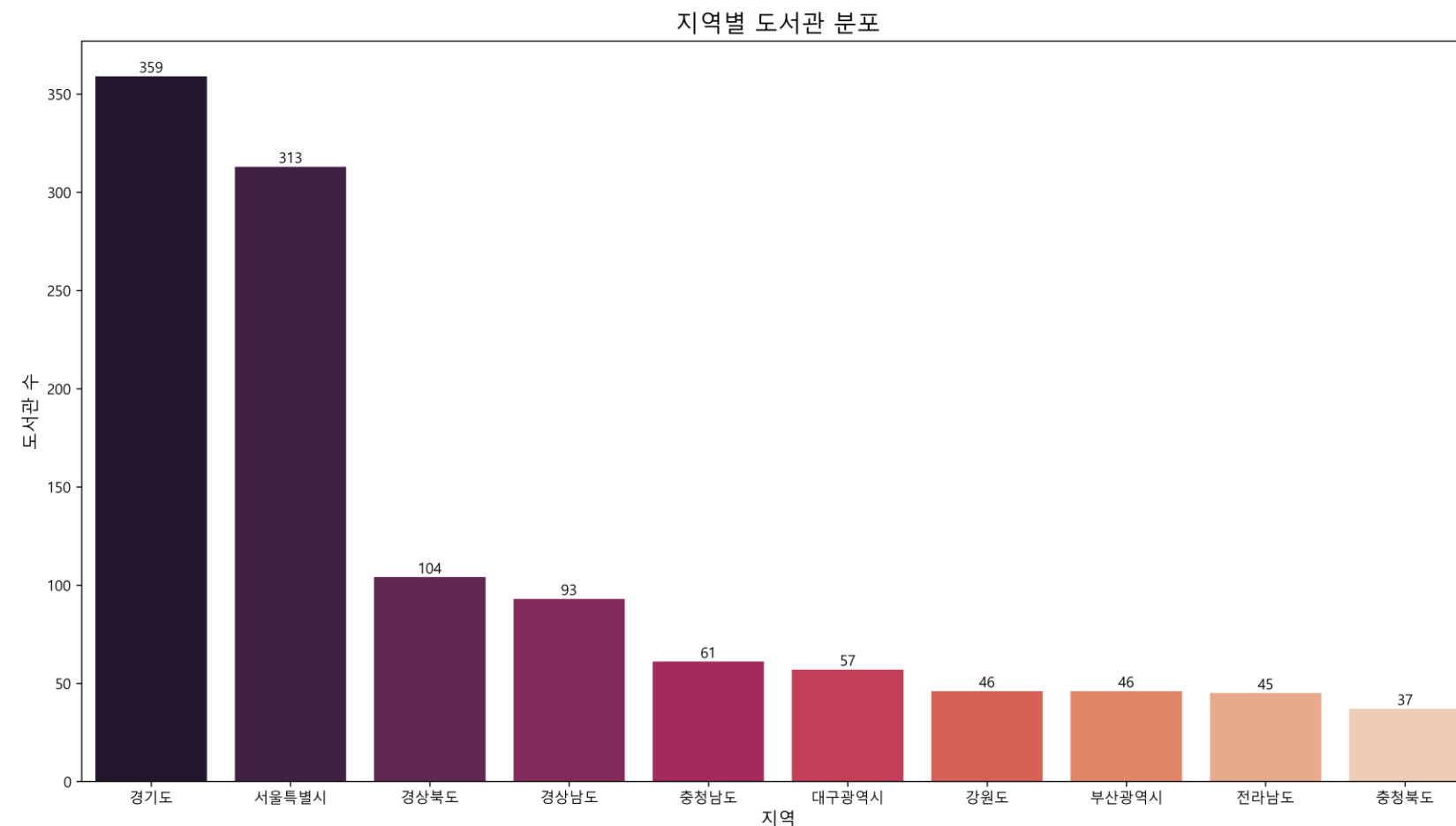
4

새로운 csv
파일로 저장

```
if __name__ == "__main__":  
    csv_path =  
    './loan_information_no_duplicates.csv'  
    processed_df =  
    preprocess_book_data(csv_path)  
    processed_df.to_csv('loan_information_  
_processed.csv', index=False)
```

데이터 전처리 (2) – 도서관 정보

- 📌 **선별적 특성 추출:** 11개 핵심 컬럼만 선택하여 데이터 간소화 (도서관 ID, 이름, 주소, 좌표 등)
- 📌 **데이터 정규화:** 컬럼명 영문화 및 일관된 네이밍 규칙 적용으로 코드 가독성 향상
- 📌 **지오데이터 최적화:** 위도/경도 수치형 변환 및 결측치 제거로 위치 기반 서비스 안정성 확보
- 📌 **통합 위치 정보 생성:** 도/시 정보를 결합한 location 필드 생성으로 지역 검색 효율성 향상



Machine Learning

TF-IDF

1

- 도서 설명(clean_description)에서 중요 단어 추출 및 벡터화
- 단순 빈도수가 아닌 문서 내 중요도를 반영한 가중치 부여
- 불용어 제거로 '등', '및', '을', '를' 등 의미 없는 조사 필터링
- 차원 축소(max_features=1000)로 계산 효율성 확보

2

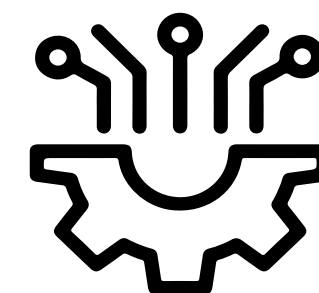
RandomForestClassifier

- 장르(estimated_category)가 정해진 데이터만 추려서 학습
- ‘기타’ 장르로 분류된 도서들도 예측해서 장르 채워 넣음
- input X: clean_description (TF-IDF vector),
- output y: estimated_category
- supervised learning

3

추천 도서 점수 계산(rule-based)

- 사용자 프로필 기반 가중치 모델 구현
(연령 40%, 관심사 40%, 인기도 20%)
- 다양성 보장 알고리즘으로 저자/출판사 편중 방지
- 콘텐츠 기반 필터링으로 콜드 스타트 문제 해결
- 폴백 메커니즘 적용: 필터링 결과 부족 시 조건 자동 완화



LLM

1

LLM 프롬프트 템플릿 생성

- OpenAI API키를 가져와 프롬프트를 생성하고, LLM 모델 호출
- 프롬프트에는 각 도서의 '제목', '저자', '설명', '장르' 등을 상세하게 나열

```
def create_recommendation_prompt(user_request, top_candidates_df, library_info=None):
    """
    사용자 요청과 후보 도서 정보를 바탕으로 LLM 프롬프트 생성
    """
    candidates_text = ""

    for i, (_, book) in enumerate(top_candidates_df.iterrows(), 1):
        candidates_text += f"후보 {i}:\n"
        candidates_text += f"제목: {book['title']}\n"
        candidates_text += f"저자: {book['main_author']}\n"
        candidates_text += f"출판사: {book['publisher']}\n"
        candidates_text += f"카테고리: {book['estimated_category']}\n"
        candidates_text += f"대상 독자: {book['age_group']}\n"

    # 설명이 너무 길면 짧게 잘라서 표시
    description = book['description']
    if isinstance(description, str) and len(description) > 200:
        description = description[:200] + "..."
    candidates_text += f"설명: {description}\n\n"
```

2

LLM 기반 추천

- run_hybrid_recommendation(): ML 후보 중 3권 추천
- 추천 이유 설명, 주요 내용 요약, 추천 대상, 읽기 순서 제공



```
prompt = f"""
사용자 요청: {user_request}

아래는 사용자 요청에 기반하여 선별된 도서 후보 목록입니다:

{candidates_text}

이제 다음 조건을 충족하도록 답변을 생성해주세요:

1. 위 도서 목록 중에서 **사용자 요청에 가장 적합한 3권**을 선택하세요.
2. 각 도서에 대해 다음 항목을 포함해 상세하게 작성하세요:
   - 추천 이유: **사용자의 니즈와 요청 배경에 적합한 이유를 구체적으로 설명**해주세요.
   - 주요 내용: 도서가 다루는 **핵심 개념, 주요 챕터, 대표 사례, 저자의 관점** 등을 간결하고 명확하게 정리해주세요.
   - 이런 분께 추천합니다: **어떤 상황에 처한 사람**, 또는 **어떤 관점이나 수준의 독자**에게 특히 유익한지 설명해주세요.

3. 3권의 책을 어떤 순서로 읽는 것이 가장 효과적인지 간단하게 제안해주세요.
   - **이해도, 난이도, 주제의 흐름**을 고려해 순서를 정하고, 그 이유도 덧붙여 설명해주세요.

※ 응답은 아래 형식에 맞춰 정돈된 문장으로 작성해주세요:

1. [도서 제목] (저자)
   - 추천 이유:
   - 주요 내용:
   - 이런 분께 추천합니다:
```

Streamlit

✓ 캐싱 메커니즘 구현

- `@st.cache_resource`: 추천 시스템 모델 로딩 최적화 (메모리 유지)
- `@st.cache_data`: 도서관 데이터 처리 결과 캐싱 (재계산 방지)

✓ 반응형 UI/UX

- 사이드바: 데이터 경로, 모델 선택, 후보 수 설정
- 확장 패널(Expander): 상세 정보 접근성 향상
- 다중 컬럼 레이아웃: `st.columns()`을 활용한 입력 폼 최적화

✓ 비동기 처리

- `with st.spinner()`: LLM 호출 및 데이터 처리 중 사용자 피드백
- 세션 상태 관리: `st.session_state`로 모델 인스턴스 유지

✓ 데이터 시각화

- 데이터프레임 표시: `st.dataframe()`으로 후보 도서 목록 시각화
- JSON 표시: `st.json()`으로 사용자 프로필 정보 구조화

설정

데이터 파일 경로:
loan_information_processed.csv

LLM 모델 선택:
gpt-4o

ML 추천 후보 수:
5 15 20

추천 시스템 로드 완료: 54110 권의 도서 정보

데이터 통계
총 도서 수: 54110
연령대 구분: 13
카테고리 구분: 8
출판사 수: 5169
인기 카테고리:

- 소설: 38513권
- 어린이: 4816권
- 과학: 3622권
- 역사: 2091권
- 경제/경영: 1632권

북큐레이터(BookCurator): AI 기반 맞춤형 도서 추천 시스템

북큐레이터(BookCurator)는 머신러닝과 LLM 모델을 결합한 하이브리드 방식으로 도서를 추천합니다.

1. 머신러닝으로 사용자 프로필에 맞는 도서 후보군 선정
2. LLM 모델로 최종 추천 및 상세한 추천 이유 생성

추천 요청 문장을 입력하세요:

중학생 딸이 과학에 관심을 보이기 시작했어요. 청소년이 쉽게 읽을 수 있는 과학책 추천해주세요.

추가 정보 입력 (선택사항)

연령대:
자동 감지

독서 목적:
학습

관심사:
Choose an option

독서 수준:
☒ 자동 감지 ☐ 초급 ☐ 중급 ☐ 고급

하이브리드 추천

하이브리드 추천 시스템 정보



Web Demo 실행 영상

https://drive.google.com/file/d/1hgbFzDfQ1uPKr0t-JVcGEqwACRjFr_oj/view?usp=sharing

기대효과

- 사용자 만족도 향상 : 맞춤형 추천으로 독서 경험 최적화
- ML+LLM 하이브리드 방식의 기술적 우수성: 추천 정확도 확보
- 도서관 자원 활용 증대: 다양한 도서관 콘텐츠에 대한 접근성 개선
- 사용자 위치 기반으로 편리성 확대: 위치 기반 서비스로 도서관 접근성 개선



Q&A

감사합니다😊🌸

