2019301066 이지용

도서 리뷰 분석 및 추천

Contents

목 차

Chapter.01

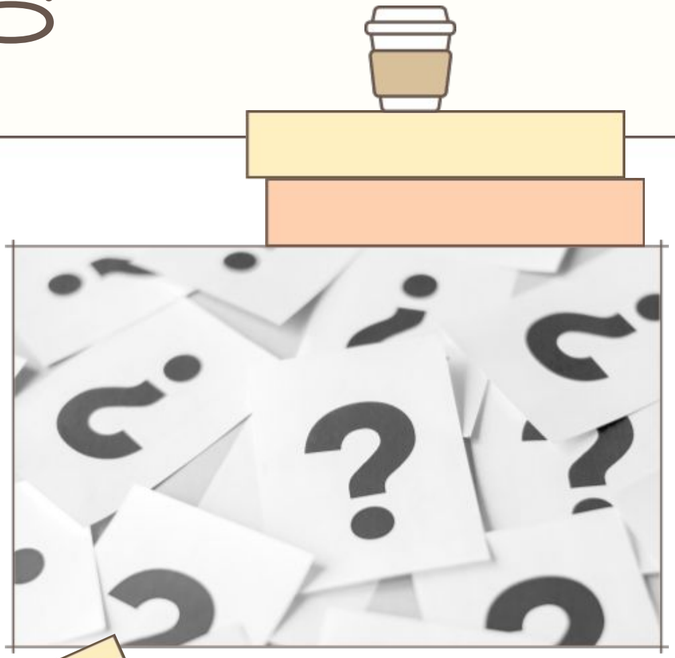
주세 선정 배경

저조한 독서율

사고력과 창의력 저하
언어 능력 저하

방대한 도서 리뷰의 정리 필요성

독자들이 남긴 리뷰는 방대하나, 이를 종합적으로 파악하기 어려운 상황이다.



도서 선택의 어려움

독자들은 방대한 양의 도서 중에서 자신에게 적합한 도서를 선택하는 데 어려움을 겪고 있다.

Chapter.02

크롤링


```python
# 페이지 네비게이션 요소를 찾습니다.
tmp = driver.find_element(By.CSS_SELECTOR, '.megaseller_rank')
pages = tmp.find_elements(By.CSS_SELECTOR, '[class$="naviRank"]')


for page_index in range(len(pages)):
    if(page_index == 10):
        next = driver.find_elements(By.CSS_SELECTOR, '.endse01 a')[1]
        ActionChains(driver).move_to_element(next).click().perform()
    try:
        # 페이지 네비게이션 버튼을 다시 찾아 클릭 (동적 로드 문제 해결)
        if(page_index != 0):
            next_page = driver.find_elements(By.CSS_SELECTOR, '[class$="naviRank"]')[page_index]
            next_page = next_page.find_element(By.CSS_SELECTOR, 'a')
            ActionChains(driver).move_to_element(next_page).click().perform()

        # 현재 페이지의 책 URL 크롤링
        page_urls = crolling_page_url(driver)
        book_urls.extend(page_urls)

    except Exception as e:
        print(f"{page_index}페이지 클릭 에러: {e}")
        continue
```

https://www.aladin.co.kr/home/welcome.aspx
알라딘 사이트를 사용함

도서 url 크롤링

1위~1000위까지 각각 버튼을 눌러 동적으로 로드

ActionChains을 통해서 동적 로드 문제 해결

인덱스를 통해서 순위 접근

책 상세페이지 url 크롤링

Chapter.02

크롤링


```python
# 스크롤하면서 찾기
def scroll_down_to_end(driver):
    while True:
        # 페이지 끝까지 스크롤
        driver.execute_script("window.scrollBy(0, 2000);")
        time.sleep(0.2)  # 스크롤 후 로딩 대기

        try:
            target_element = driver.find_element(By.ID, "tabOrderer")
            print("태그를 찾았습니다")
            break  # 요소를 찾았으면 반복 종료
        except:
            pass
            #print("ID를 가진 요소를 찾지 못했습니다.")

        current_scroll_position = driver.execute_script("return window.pageYOffset;")
        page_height = driver.execute_script("return document.body.scrollHeight")


        if current_scroll_position >= page_height-2000:  # 더 이상 스크롤되지 않을 때 종료
            print("마지막 페이지까지 스크롤 완료")
            break
```

```python
while True:
    # "더보기" 버튼 클릭
    try:
        more_button = driver.find_element(By.ID, "divReviewPageMore")  # 버튼 ID
        more_button.find_element(By.TAG_NAME, "a").click()  # 버튼 클릭
        time.sleep(0.5)  # 페이지 로드 대기
    except Exception as e:
        print("더보기 버튼이 더 이상 존재하지 않습니다.")
        break
```

도서 정보 크롤링

리뷰가 있는 곳에 화면이 가야 로드가 되기 때문에관련 태그나 나올때까지 스크롤을 내리면서 찾음

모든 리뷰가 로드가 되어야하기 때문에 더보기 버튼을 계속 누름

Chapter.02

크롤링

```python
# 책 정보와 리뷰 데이터를 담은 구조 예제
book_data = {
    "제목": "책 제목",
    "저자": "저자 이름",
    "출판사": "출판사 이름",
    "출판일": "YYYY-MM-DD",
    "정가": "가격",
    "분류": "카테고리",
    "리뷰": [
        {
            "작성자": "리뷰 작성자",
            "내용": "리뷰 내용",
            "별점": 5,  # 1부터 5까지의 숫자
            "작성 날짜": "YYYY-MM-DD"
        },
        {
            "작성자": "리뷰 작성자",
            "내용": "리뷰 내용",
            "별점": 4,
            "작성 날짜": "YYYY-MM-DD"
        }
        # 추가 리뷰 ...
    ]
}
```

도서 정보 크롤링

제목, 저자, 출판사, 정가, 분류, 리뷰에 대한 정보를 크롤링함

Chapter.03

리뷰 분석 및 시각화

```python
# 특수 문자 제거
def preprocess_text(text):
    return re.sub(r'[^\uAC00-\uD7A3a-zA-Z\s]', '', text)

# wordcloud 시각화
def visualize_reviews_wordcloud(reviews):
    # 형태소 분석기 초기화
    okt = Okt()

    # 모든 리뷰에서 명사 추출
    nouns = []
    for review in reviews:
        clean_review = preprocess_text(review)
        nouns.extend(okt.nouns(clean_review))

    # 단어 빈도수 계산
    word_counts = Counter(nouns)

    # 워드 클라우드 생성
    wordcloud = WordCloud(
        font_path='/usr/share/fonts/truetype/nanum/NanumGothic.ttf',
        background_color='white',
        width=800,
        height=400
    ).generate_from_frequencies(word_counts)

    # 시각화
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title("Review Word Cloud", fontsize=20)
    plt.show()
```

konlpy와 wordcloud 라이브러리 사용

특수 문자를 다 제거한후

okt.nouns() 사용하여 명사를 추출

WordCould()를 사용하여 시각화

Chapter.04

도서 추천 알고리즘

```python
# 리뷰와 메타데이터를 벡터화하는 함수
def vectorize_data(reviews, metadata):
    # TF-IDF를 사용하여 리뷰와 메타데이터를 벡터화
    review_vectorizer = TfidfVectorizer()
    review_vectors = review_vectorizer.fit_transform(reviews)

    meta_vectorizer = TfidfVectorizer()
    meta_vectors = meta_vectorizer.fit_transform(metadata)

    return review_vectors, meta_vectors

# 책 추천 함수
def recommend_books(book_name, books, review_vectors, meta_vectors, ratings, review_weight=0.4, metadata_weight=0.4, rating_weight=0.2, top_n=5):
    if book_name not in books:
        return f"'{book_name}'는 데이터에 존재하지 않습니다."

    target_index = books.index(book_name)

    # 유사도 계산
    review_similarity = cosine_similarity(review_vectors[target_index], review_vectors).flatten()
    metadata_similarity = cosine_similarity(meta_vectors[target_index], meta_vectors).flatten()
    normalized_ratings = np.array(ratings) / max(ratings)  # 별점 정규화

    # 가중치를 적용한 유사도 점수 계산
    combined_scores = (
        review_similarity * review_weight +
        metadata_similarity * metadata_weight +
        normalized_ratings * rating_weight
    )

    # 상위 N개의 추천 책 선택
    similar_indices = combined_scores.argsort()[-(top_n + 1):][::-1]
    recommendations = [(books[i], combined_scores[i]) for i in similar_indices if i != target_index]

    return recommendations[:top_n]
```

코사인 유사도를 사용하여 추천 알고리즘 구현

라이브러리 : sklearn -> TfidfVectorizer, cosine_similarity

TfidfVectorizer를 사용하여 리뷰와 메타데이터를 벡터화

cosine_similarity를 사용하여 유사도를 계산

가중치 부여 (리뷰 4 : 메타데이터 4 : 별점 2)

유사도 상위 5개를 리턴

Chapter.05

결과

소년이 온다
제목 : 소년이 온다
저자 : 한강
출판사 : 창비
출판일 : 2014-05-19
분류 : 한국소설
정가 : 15,000원
리뷰 평균 : 4.74

'소년이 온다'와(과) 유사한 책 추천:
1. 채식주의자 (리마스터판) (유사도 점수: 0.61)
2. 디 에센셜 한강 (무선 보급판) (유사도 점수: 0.48)
3. 노랑무늬영원 (유사도 점수: 0.47)
4. 희랍어 시간 (유사도 점수: 0.46)
5. 한강 스페셜 에디션 (작별하지 않는다 + 흰 + 검은 사슴 + 필사 노트) (유사도 점수: 0.45)

Chapter.05

결과

이처럼 사소한 것들
제목 : 이처럼 사소한 것들
저자 : 클레어
출판사 : 키건
출판일 :
분류 : 세계의 소설
정가 : 13,800원
리뷰 평균 : 4.48

'이처럼 사소한 것들'와(과) 유사한 책 추천:
1. 맡겨진 소녀 (유사도 점수: 0.64)
2. 푸른 들판을 걷다 (유사도 점수: 0.63)
3. 예언자의 노래 (유사도 점수: 0.40)
4. 참을 수 없는 존재의 가벼움 (유사도 점수: 0.37)
5. 타임 셸터 (유사도 점수: 0.36)

Chapter.05

결과

축구의 시대
제목 : 축구의 시대
저자 : 정몽규
출판사 : 브레인스토어
출판일 : 2024-07-26
분류 : 명사에세이
정가 : 29,000원
리뷰 평균 : 2.33

'축구의 시대'와(과) 유사한 책 추천:
1. 나의 인생만사 답사기 (유사도 점수: 0.31)
2. 쓸 만한 인간 (유사도 점수: 0.30)
3. 태도에 관하여 (20만 부 기념 완결판) (유사도 점수: 0.30)
4. 내가 한 말을 내가 오해하지 않기로 함 (유사도 점수: 0.28)
5. 겨울철 한정 봉봉 쇼콜라 사건 - 하 (유사도 점수: 0.21)

Chapter.05

결과

차가운 자본주의
제목 : 차가운 자본주의
저자 : 윤후카스
출판사 : RISE(떠오름)
출판일 : 2023-07-21
분류 : 경제학/경제일반
정가 : 17,500원
리뷰 평균 : 1.78

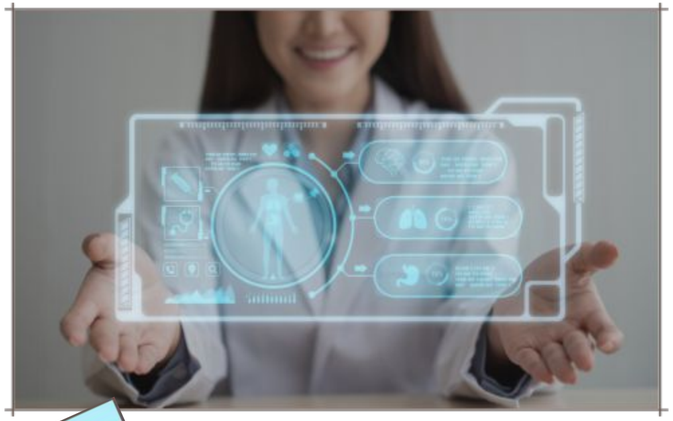
'차가운 자본주의'와(과) 유사한 책 추천:
1. 시대예보: 호명사회 (유사도 점수: 0.31)
2. 댄 애리얼리 미스빌리프 (유사도 점수: 0.31)
3. 트럼프 2.0 시대 (유사도 점수: 0.30)
4. 당신이 모르는 진짜 농업 경제 이야기 (유사도 점수: 0.30)
5. 불변의 법칙 (유사도 점수: 0.30)

Chapter.06

어려웠던 점

동적 크롤링

데이터 크롤링 하는 데 있어서 동적으로 로드 되는 곳이 많아서 그 부분을 처리하는 것



데이터 구성

데이터 구성을 어떻게 할 것인지에 대한 고민과 있으면 좋겠다 싶어서 다시 크롤링한 부분

리뷰 분석 및 도서 추천

어떤 방식으로 어떤 도구를 사용할 것인지

감사합니다