과목명	데이터분석기초
주제	<에브리타임> 강의평 데이터를 이용한 강의 추천시스템
담당교수	정 덕 종 교수님
제출일	2022 년 6 월 21 일(화)
학과	통계학과
학번	2018580004
이름	김 다 희

서울시립대학교
UNIVERSITY OF SEOUL

목차

1. 서론

A. 주제선정 및 이유

매 학기, 대부분의 대학생들은 한 학기를 책임질 시간표를 완성하기 위해 신중하게 고민하며 자신에게 맞는 강의를 찾기 위해 많은 시간을 투자한다. 현재 대학생들이 시간표를 짜기 위해선 원하는 시간대의 강의 중에서 직접 찾아보거나, 선후배의 추천 강의를 고려하는 방법 등이 있다. 하지만 이런 방법으로는 강의 탐색시간이 많이 걸리며, 특히 어느 정도 정보를 얻기 쉬운 전공과목을 제외하면 자신에게 맞는 교양과목을 찾기는 상당히 어렵다. 따라서 SNS의 친구 추천, 유튜브의 영상 추천, 쇼핑 사이트의 제품 추천 등 일상 속에 자연스럽게 자리 잡은 추천시스템을 이제는 강의에 접목시켜 맞춤형 강의를 찾는 데 어려움을 겪는 모든 대학생들을 위한 알고리즘을 개발하고자 한다. 즉, 본 연구에서는 대학교 커뮤니티 및 시간표 서비스를 제공하는 애플리케이션인 **<에브리타임> 강의평 데이터를 이용한 콘텐츠 기반 추천시스템**의 구현을 주제로 한다.

B. 선행연구 조사

i. 토픽모델링 기반 비대면 강의평 분석 및 딥러닝 분류 모델 개발

해당 논문에서는 코로나로 인해 기존 대면 수업 형태에서 새로운 비대면 수업 형태로 바뀌며 새롭게 강조되는 수업 평가 요인과 강의 만족도 영향 요인에 대한 연구를 진행했다. 연세대학교 <에브리타임>의 강의평 데이터에 LDA 토픽모델링을 적용해 강의를 평가하는 속성에 대해서 파악했으며, KoBERT 기반의 토픽 자동 분류 모델을 구축해 강의 만족도가 어떤 요인으로부터 기인한 것인지 확인했다. 본 연구에서는 코로나 전후의 만족도 요인 변화에 중점을 두지 않을 것이므로 이와 관련된 분석은 진행하지 않지만, <에브리타임> 데이터를 이용하여 강의 추천 모델을 개발할 수 있겠다는 아이디어를 얻었다.

ii. 사용자 평점과 리뷰 유사도를 이용한 협업 필터링 기반 영화 추천시스템

리뷰 정보를 이용한 강의 추천시스템을 구현하기 위해 관련된 논문을 확인하여 개인의 리뷰 데이터를 어떻게 추천시스템에 반영할지 확인했다. 해당 연구에서는 사용자 평점에 리뷰 감성수치를 반영하여 새로운 평점 매트릭스를 생성하고, 이를 이용한 협업 필터링 기반 추천을 진행한다. 하지만 본 연구에서 사용할 <에브리타임> 리뷰 데이터에는 유저 정보가 식별 불가능하므로 평점 테이블을 만들 수 없다고 판단하여 아이템 정보만으로도 구현할 수 있는 콘텐츠 기반 추천시스템 알고리즘을 적용하기로 했다.

iii. NLP를 활용한 국회의원 시각화 및 추천시스템

Word2Vec을 이용해 국회의원의 발의한 법안을 바탕으로 국회의원 벡터를 생성하여 사용자가 관심있는 법안과 유사한 방향을 가지는 국회의원을 추천하는 내용의 프로젝트이다. 이를 본 연구에 맞게 수정 및 개선하여 강의 리뷰 및 강의를 임베딩하고,

사용자 입력을 받아 사용자가 원하는 바와 가장 유사한 강의를 추천해주는 알고리즘을 개발하고자 한다.

C. 분석 절차

i. 데이터 수집

파이썬 웹스크래핑 라이브러리 BeautifulSoup과 Selenium을 이용하여 서울시립대학교 <에브리타임> 강의목록 및 강의리뷰 데이터를 크롤링하는 단계이다.

ii. 데이터 탐색

크롤링한 데이터의 기본적인 전처리와 요약통계량 및 EDA 등을 통해 기본적인 데이터 탐색을 진행하는 단계이다.

iii. 데이터 분석

정제된 리뷰 데이터를 이용하여 단어 corpus를 생성한 후, Skip-gram 모델로 학습한 단어 임베딩을 바탕으로 강의 벡터를 생성하여 콘텐츠 기반 추천시스템을 구현하는 단계이다.

2. 본론

A. 데이터 수집

i. <에브리타임> 데이터 수집 이유

기존 자기기입식 설문조사 및 대학에서 실시하는 강의평가는 설문조사를 기획하는 사람의 의도가 반영될 수 있고, 실제 평가보다 덜 솔직하다는 단점이 있다. 결정적으로 수업평가 결과가 학생들에게 공개되지 않으므로 대학 수업평가 내용을 바탕으로 한 강의 추천은 불가능하다. 차선책으로 수강생들만 작성할 수 있는 <에브리타임>의 강의평 데이터를 크롤링하여 보다 직설적이며 정제되지 않은 실제 학생들의 의견을 이용한 추천을 진행하고자 한다.

ii. <에브리타임> 데이터 수집 방법

<에브리타임>에서 강의평 데이터를 수집하기 위해선 시간표 탭에서 강의목록과 강의정보를 크롤링하고, 수집한 강의정보에 존재하는 링크 정보를 통해 강의리뷰 탭으로 이동하여 강의리뷰를 수집하는 구조이다. 즉, 먼저 강의목록 크롤러(lecture_crawler)를 정의해 서울시립대학교 내 모든 강의에 대한 정보를 수집했으며, 추천의 목적을 고려했을 때 전공과목보다는 교양과목의 추천이 적절하다고 판단하여 교양선택 과목에 대해서 강의리뷰 크롤러(review_crawler)를 정의해 추가적으로 강의리뷰 데이터를 생성하였다. 이때, 원칙상 <에브리타임>은 크롤링을 자유롭게 할 수 없는 사이트이므로 코드 내에 time.sleep을 랜덤으로 주어 차단을 방지했다. 데이터 수집 기준일은 2022.05.21이다.

iii. 강의목록 크롤링

<에브리타임> 시간표 탭에서 학년, 교과번호, 교과목명, 학점, 교수, 강의시간, 강의유형, 담은인원, 정원, 강의평점, 리뷰링크 데이터를 수집하였고 최종적으로 2,807개의 행과 11개의 열을 가지는 데이터프레임을 생성했다. 사용한 코드와 생성된 데이터프레임은 다음과 같다.

```
#강의목록 크롤러 정의
def lecture_cralwer():

    #라이브러리 로드
    import pandas as pd
    import numpy as np
    from bs4 import BeautifulSoup
    from selenium import webdriver
    from selenium.webdriver.common.keys import Keys
    from tqdm import tqdm
    from tqdm.notebook import tqdm
    import re
    from time import sleep
    import time
    import warnings
    import random
    from openpyxl import Workbook,load_workbook
    warnings.filterwarnings(action='ignore')

    #드라이버 정의
    url = 'https://everytime.kr/timetable'
    driver = webdriver.Chrome('/Users/diekim/downloads/chromedriver')
    driver.get(url)
    driver.maximize_window()
    time.sleep(random.uniform(3,5))

    #에브리타임 로그인
    id_var = input('에브리타임 아이디를 입력하세요: ')
    pw_var = input('에브리타임 비밀번호를 입력하세요: ')
    driver.find_element_by_name('userid').send_keys(id_var)
    driver.find_element_by_name('password').send_keys(pw_var)
    driver.find_element_by_xpath('//*[@id="container"]/form/p[3]/input').click()
    time.sleep(random.uniform(3,5))

    #강의목록 스크롤
    driver.find_element_by_xpath('//*[@id="container"]/ul/li[1]').click()
    time.sleep(random.uniform(3,5))

    pre_cnt = 0
    while True: #끝까지 내리기
        #현재 페이지 요소 접근
        element = driver.find_elements_by_css_selector("#subjects > div.list > table > tbody > tr")
        time.sleep(random.uniform(3,5))
        #현재 페이지 끝까지 스크롤
        driver.execute_script('arguments[0].scrollIntoView(true);', element[-1])
        time.sleep(random.uniform(3,5))
        #같으면 루프 탈출
        current_cnt = len(element)
        if pre_cnt == current_cnt:
            break
        #같아질 때까지 반복
        pre_cnt = current_cnt
        time.sleep(random.uniform(3,5))

    #강의목록 크롤링
    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')
    trs = soup.select('#subjects > div.list > table > tbody > tr')
    results = []
    for tr in trs:
        result=[]
        tds = tr.select('#subjects > div.list > table > tbody > tr > td')
        score = tr.select('#subjects > div.list > table > tbody > tr > td')[7].a['title']
        link = tr.select('#subjects > div.list > table > tbody > tr > td')[7].a['href']
        #학교마다 구성 다름
        result.append(tds[0].text)  #학년
        result.append(tds[1].text)  #교과번호-분반
        result.append(tds[2].text)  #교과목명
        result.append(tds[3].text)  #학점
        result.append(tds[4].text)  #교수
        result.append(tds[5].text)  #강의시간/강의실
        result.append(tds[6].text)  #강의유형
        #result.append(tds[7].text) #강의평
        result.append(tds[8].text)  #담은인원
        result.append(tds[9].text)  #정원
        result.append(score)        #강의평
        result.append(link)         #링크
        results.append(result)

    #강의목록 저장
    columns = ['학년', '교과번호', '교과목명', '학점', '교수' ,'강의시간',
               '강의유형', '담은인원', '정원' ,'강의평', '링크']
    columnn = 11
    write_wb = Workbook()
    write_ws = write_wb.active
    write_ws.append(columns)
    for data in results:
        write_ws.append(data)
    write_wb.save('강의목록.xlsx')
    df = pd.read_excel("강의목록.xlsx")
    df.to_csv('강의목록.csv', encoding='utf-8-sig', index=False)

    driver.quit()
    return df
```

<강의목록 크롤러 정의>

```
#강의목록 크롤링
df = lecture_cralwer()
df
```

	학년	교과번호	교과목명	학점	교수	강의시간	강의유형	담은인원	정원	강의평	링크
0	1.0	01569-01	영문분석과활용	3	김선희	금02,03,04/4-204~6	교양선택	35	25	4.83	/lecture/view/2090902
1	1.0	01570-01	영어면접과발표	3	William Hart	수05/4-208~10, 목03,04/4-208~10	교양선택	26	28	4.67	/lecture/view/2090903
2	1.0	01573-01	영어말하기와토론	3	Joseph Van Dorn	월06,07,08/20-320	교양선택	17	25	0.00	/lecture/view/2379676
3	1.0	01573-02	영어말하기와토론	3	Steven Moore	수05/4-204~6, 목03,04/4-204~6	교양선택	26	28	4.54	/lecture/view/2090905
4	1.0	01668-01	시사영어	3	김선희	금07,08,09/4-204~6	교양선택	27	25	4.57	/lecture/view/2090906
...	...	...	...	...	...	...	...	...	...	...	...
2802	NaN	I4900-02	논문연구	0	박병은	NaN	전공선택	0	1	0.00	/lecture/view/1376285
2803	NaN	521-02	윤강	2	한인섭	목10,11/4-432	전공선택	0	6	0.00	/lecture/view/2380242
2804	0.0	40.900-01	논문연구	0	이수일	NaN	전공선택	0	0	0.00	/lecture/view/296156
2805	0.0	49.900-02	논문연구	0	김한준	NaN	전공선택	0	1	0.00	/lecture/view/296191
2806	0.0	31.900-01	논문연구	0	NaN	NaN	전공선택	0	0	0.00	/lecture/view/390722

2807 rows × 11 columns

<강의목록 크롤링 결과>

iv. 강의리뷰 크롤링

수집한 강의정보 데이터의 리뷰링크 칼럼을 활용하여 강의리뷰를 크롤링했으며 교양선택 과목의 추천을 위해 강의유형이 교양선택이 과목에 대해서만 강의명, 교수명, 개설학기, 평균평점, 과제, 조모임, 성적, 출결, 시험, 유저평점, 수강학기, 리뷰내용 데이터를 수집했다. 최종적으로 7,186개의 행과 12개의 열을 가지는 데이터프레임을 얻었고 사용한 코드와 생성된 데이터프레임은 다음과 같다.

```
#강의리뷰 크롤러 정의
def review_cralwer(URL):

    #라이브러리 로드
    import pandas as pd
    import numpy as np
    from bs4 import BeautifulSoup
    from selenium import webdriver
    from selenium.webdriver.common.keys import Keys
    from tqdm import tqdm
    from tqdm.notebook import tqdm
    import re
    from time import sleep
    import time
    import warnings
    import random
    from openpyxl import Workbook,load_workbook
    warnings.filterwarnings(action='ignore')

    #전체 강의 목록에 대해서 반복
    res = pd.DataFrame()
    driver = webdriver.Chrome('/Users/diekim/downloads/chromedriver')
    for k, url in enumerate(URL):
        print('{}번째 강의의 리뷰데이터를 수집합니다...'.format(k+1))
        url = 'https://everytime.kr/' + url

        #드라이버 정의
        driver.get(url)
        driver.maximize_window()
        time.sleep(random.uniform(3,5))

        #에브리타임 로그인
        if k==0:
            id_var = input('에브리타임 아이디를 입력하세요: ')
            pw_var = input('에브리타임 비밀번호를 입력하세요: ')
            driver.find_element_by_name('userid').send_keys(id_var)
            driver.find_element_by_name('password').send_keys(pw_var)
            driver.find_element_by_xpath('//*[@id="container"]/form/p[3]/input').click()
            time.sleep(random.uniform(3,7))
        else:
            pass

        #공통정보
        x1=[];x2=[];x3=[];x4=[];x5=[];x6=[];x7=[];x8=[];x9=[]
        x1.append(driver.find_element_by_xpath('/html/body/div[1]/div[2]/h2').text) #강의명
        try:
            x2.append(driver.find_element_by_xpath('/html/body/div[1]/div[2]/p[1]/span').text) #교수명
        except:
            pass
        try:
            x3.append(driver.find_element_by_xpath('/html/body/div[1]/div[2]/p[2]/span').text) #개설학기
        except:
            pass
        try:
            x4.append(driver.find_element_by_xpath('/html/body/div[1]/div[4]/div[1]/div[1]/span/span[1]').text)  #평균평점
```

```
            x6.append(driver.find_elements_by_xpath('/html/body/div[1]/div[4]/div[1]/div[2]/p[2]/span')[0].text) #조모임
            x7.append(driver.find_elements_by_xpath('/html/body/div[1]/div[4]/div[1]/div[2]/p[3]/span')[0].text) #성적
            x8.append(driver.find_elements_by_xpath('/html/body/div[1]/div[4]/div[1]/div[2]/p[4]/span')[0].text) #출결
            x9.append(driver.find_elements_by_xpath('/html/body/div[1]/div[4]/div[1]/div[2]/p[5]/span')[0].text) #시험
        except:
            pass
        time.sleep(random.uniform(3,5))

        #개별정보
        i=0; x10=[]; x11=[]; x12=[]
        while True:
            i += 1
            print('{}번째 리뷰 크롤링 중...'.format(i))
            time.sleep(random.uniform(3,5))
            try:
                #유저평점
                x10.append(driver.find_elements_by_xpath(
                    '/html/body/div[1]/div[4]/div[2]/article['+str(i)+']/p[1]/span/span'
                )[0].get_attribute('style'))
                #수강학기
                x11.append(driver.find_elements_by_xpath(
                    '/html/body/div[1]/div[4]/div[2]/article['+str(i)+']/p[2]/span'
                )[0].text)
                #리뷰내용
                x12.append(driver.find_elements_by_xpath(
                    '/html/body/div[1]/div[4]/div[2]/article['+str(i)+']/p[3]'
                )[0].text)
            except:
                break

        #데이터프레임 저장
        tmp = pd.DataFrame()
        tmp['강의명'] = x1*(i-1)
        tmp['교수명'] = x2*(i-1)
        tmp['개설학기'] = x3*(i-1)
        tmp['평균평점'] = x4*(i-1)
        tmp['과제'] = x5*(i-1)
        tmp['조모임'] = x6*(i-1)
        tmp['성적'] = x7*(i-1)
        tmp['출결'] = x8*(i-1)
        tmp['시험'] = x9*(i-1)
        tmp['유저평점'] = x10
        tmp['수강학기'] = x11
        tmp['리뷰내용'] = x12
        res = pd.concat([res, tmp])
        time.sleep(random.uniform(3,5))

    driver.quit()
    return res
```

<강의리뷰 크롤러 정의>

```
#전체 강의 중 교양선택 과목에 대해서 진행할 예정
df = df[df['강의유형']=='교양선택']
df.to_csv('강의목록_교양선택.csv', encoding='utf-8-sig', index=False)
```

	학년	교과번호	교과목명	학점	교수	강의시간	강의유형	담은인원	정원	강의평	링크
0	1.0	01569-01	영문분석과활용	3	김선희	금02,03,04/4-204~6	교양선택	35	25	4.83	/lecture/view/2090902
1	1.0	01570-01	영어면접과발표	3	William Hart	수05/4-208~10, 목03,04/4-208~10	교양선택	26	28	4.67	/lecture/view/2090903
2	1.0	01573-01	영어말하기와토론	3	Joseph Van Dorn	월06,07,08/20-320	교양선택	17	25	0.00	/lecture/view/2379676
3	1.0	01573-02	영어말하기와토론	3	Steven Moore	수05/4-204~6, 목03,04/4-204~6	교양선택	26	28	4.54	/lecture/view/2090905
4	1.0	01668-01	시사영어	3	김선희	금07,08,09/4-204~6	교양선택	27	25	4.57	/lecture/view/2090906
...	...	...	...	...	...	...	...	...	...	...	...
260	1.0	01330-04	사회봉사I	1	차한솔	월08/20-205/206	교양선택	91	100	4.74	/lecture/view/2252671
261	1.0	01330-01	사회봉사I	1	송지호	금05/19-108/109	교양선택	91	100	4.65	/lecture/view/2172206
262	1.0	01330-02	사회봉사I	1	차한솔	월07/4-120/121	교양선택	91	100	4.74	/lecture/view/2252671
263	1.0	01331-02	사회봉사II	1	송지호	금08/19-227	교양선택	12	100	2.92	/lecture/view/2172207
264	1.0	01512-01	인간과언어	3	김천학	화02,03,04/20-321	교양선택	19	22	5.00	/lecture/view/2380212

265 rows × 11 columns

<전체 강의목록 중 교양선택 강의목록>

```
#강의리뷰 크롤링
df = review_cralwer()
df
```

	강의명	교수명	개설학기	평균평점	과제	조모임	성적	출결	시험	유저평점	수강학기	리뷰내용
0	영문분석과활용	김선희	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	두번	width: 100%;	20년 1학기 수강자	영어 잘하시면 성적 잘받을수 있습니당 ㅎㅎ (토플라이팅 29점 기준!)
1	영문분석과활용	김선희	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	두번	width: 80%;	21년 1학기 수강자	교수님 강의력은 정말 뛰어나시고 질문도 잘 받아주심.\n그런데 매주 문제 풀어오는 ...
2	영문분석과활용	김선희	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	두번	width: 100%;	21년 1학기 수강자	우선 교수님 너무 친절하시고 좋으세요 ㅠㅠㅠ 최고
3	영문분석과활용	김선희	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	두번	width: 100%;	20년 1학기 수강자	매 수업시간마다 편하게 다가오려 하시고 수업자체도 열정 넘치게 하셔서 너무 좋았다....
4	영문분석과활용	김선희	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	두번	width: 100%;	20년 1학기 수강자	교수님 정말 좋으시고 이수업을 통해서 많은 걸 배울 수 있다고 생각합니다. 김선희 ...
...	...	...	...	...	...	...	...	...	...	...	...	...
7181	사회봉사III	송지호	2022-1, 2021-2, 2021-1, 2020-2	2.92	보통	보통	너그러움	전자출결	없음	width: 80%;	20년 2학기 수강자	이번학기부터 기본교육, 소양교육, 그룹코칭 등 뭔가 합계 많아져서 귀찮아진 부분이 ...
7182	사회봉사III	송지호	2022-1, 2021-2, 2021-1, 2020-2	2.92	보통	보통	너그러움	전자출결	없음	width: 60%;	20년 2학기 수강자	봉사시간 30시간 채우는 것 외에도 그룹코칭 등 온라인 교육이 많아요
7183	사회봉사III	송지호	2022-1, 2021-2, 2021-1, 2020-2	2.92	보통	보통	너그러움	전자출결	없음	width: 20%;	20년 2학기 수강자	그저 좋은 말만 나열하는데 시간 다 쓰는 의미없는 교육. 웨이렇게 이수도 복잡하게 ...
7184	사회봉사III	송지호	2022-1, 2021-2, 2021-1, 2020-2	2.92	보통	보통	너그러움	전자출결	없음	width: 80%;	20년 2학기 수강자	사회봉사 2는 헤아할게 생각보다 많더라구요 이점 참고하세요
7185	인간과언어	김천학	2022-1	5.00	보통	보통	보통	전자출결	두번	width: 100%;	22년 1학기 수강자	새로 열린 강의여서 그런지 강의 평이 하나도 없어서 걱정 많이 했는데 아주 좋습니다...

7186 rows × 12 columns

<강의리뷰 크롤링 결과>

B. 데이터 탐색

i. 데이터 전처리

강의정보 데이터와 강의리뷰 데이터를 교과목명 기준으로 병합했으며, 결측값과 중복값은 없는 것을 확인했다. 이후 유저평점 칼럼과 시험 칼럼을 형식에 맞게 텍스트에서 수치형으로 변환하고, 리뷰 수 칼럼을 추가하여 분석에 사용할 (7186, 21) 사이즈의 최종 데이터프레임을 얻었다. 데이터프레임과 변수 목록은 다음과 같다.

```
#유저평점 칼럼 수정
print(df['유저평점'].value_counts())

def for_user_rating(x):
    if x=='width: 100%;':
        return 5.
    elif x=='width: 80%;':
        return 4.
    elif x=='width: 60%;':
        return 3.
    elif x=='width: 40%;':
        return 2.
    else:
        return 1.

df['유저평점'] = df['유저평점'].apply(for_user_rating)
df[['유저평점']]
```

```
width: 100%;    3625
width: 80%;     1782
width: 60%;     1110
width: 20%;      366
width: 40%;      303
Name: 유저평점, dtype: int64
```

	유저평점
0	5.0
1	4.0
2	5.0
3	5.0
4	5.0

<유저평균 칼럼 전처리>

```
#시험 칼럼 수정
print(df['시험'].value_counts())

def for_test(x):
    if x=='없음':
        return 0
    elif x=='한 번':
        return 1
    elif x=='두 번':
        return 2
    elif x=='세 번':
        return 3
    else:
        return 4

df['시험'] = df['시험'].apply(for_test)
df[['시험']]
```

```
두 번          5072
한 번           928
없음            846
네 번 이상      328
세 번            12
Name: 시험, dtype: int64
```

	시험
0	2
1	2
2	2
3	2
4	2

<시험 칼럼 전처리>

```
#리뷰수 칼럼 추가
df['리뷰수'] = df.groupby('링크').transform('count').iloc[:, -1]
df[['리뷰수']]
```

	리뷰수
0	6
1	6
2	6
3	6
4	6
...	...
7181	24
7182	24
7183	24
7184	24
7185	1

7186 rows × 1 columns

<리뷰 수 칼럼 생성>

```
#최종 데이터 프레임
df = pd.read_csv('에브리타임_크롤링_최종.csv')
df.head(5)
```

	강의명	교수명	학년	교과번호	학점	강의시간	강의유형	담은인원	정원	링크	개설학기	평균평점	과제	조모임	성적	출결	시험	유저평점	수강학기	리뷰내용	리뷰수
0	영문분석과활용	김선희	1.0	01569-01	3	금02,03,04/4-204~6	교양선택	35	25	/lecture/view/2090902	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	2	5.0	20년 1학기 수강자	영어 잘하시면 성적 잘받을수 있습니당 ㅎㅎ (토플 라이팅 29점 기준!)	6
1	영문분석과활용	김선희	1.0	01569-01	3	금02,03,04/4-204~6	교양선택	35	25	/lecture/view/2090902	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	2	4.0	21년 1학기 수강자	교수님 강의력은 정말 뛰어나시고 질문도 잘 받아주심.\n그런데 매주 문제 풀어오는 ...	6
2	영문분석과활용	김선희	1.0	01569-01	3	금02,03,04/4-204~6	교양선택	35	25	/lecture/view/2090902	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	2	5.0	21년 1학기 수강자	우선 교수님 너무 친절하시고 좋으세요 ㅠㅠㅠ 최고	6
3	영문분석과활용	김선희	1.0	01569-01	3	금02,03,04/4-204~6	교양선택	35	25	/lecture/view/2090902	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	2	5.0	20년 1학기 수강자	매 수업시간마다 편하게 다가오려 하시고 수업자체도 열정 넘치게 하셔서 너무 좋았다....	6
4	영문분석과활용	김선희	1.0	01569-01	3	금02,03,04/4-204~6	교양선택	35	25	/lecture/view/2090902	2022-1, 2021-1, 2020-1	4.83	보통	보통	너그러움	직접호명	2	5.0	20년 1학기 수강자	교수님 정말 좋으시고 이 수업을 통해서 많은 걸 배울 수 있다고 생각합니다. 김선희 ...	6

<최종 데이터프레임>

```
#최종 변수 목록
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7186 entries, 0 to 7185
Data columns (total 21 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   강의명      7186 non-null   object 
 1   교수명      7186 non-null   object 
 2   학년       7186 non-null   float64
 3   교과번호     7186 non-null   object 
 4   학점       7186 non-null   int64  
 5   강의시간     7186 non-null   object 
 6   강의유형     7186 non-null   object 
 7   담은인원     7186 non-null   int64  
 8   정원       7186 non-null   int64  
 9   링크       7186 non-null   object 
 10  개설학기     7186 non-null   object 
 11  평균평점     7186 non-null   float64
 12  과제       7186 non-null   object 
 13  조모임      7186 non-null   object 
 14  성적       7186 non-null   object 
 15  출결       7186 non-null   object 
 16  시험       7186 non-null   int64  
 17  유저평점     7186 non-null   float64
 18  수강학기     7186 non-null   object 
 19  리뷰내용     7186 non-null   object 
 20  리뷰수      7186 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.2+ MB
```

<변수 정보>

ii. 요약통계량

최종 데이터프레임 내 수치형 변수에 대한 요약통계량을 확인했다. 특히 강의의 평균 평점을 나타내는 평균평점 칼럼의 평균이 4.1070, 표준편차가 0.5787이었으며, 유저별 평균을 나타내는 유저평균 칼럼의 평균이 4.1129, 표준편차가 1.1284였다. 즉, 두 칼럼의 평균은 비슷하나 유저평균 칼럼의 표준편차가 더 큰 것을 보아 분포가 더 퍼져 있음을 예상할 수 있다.

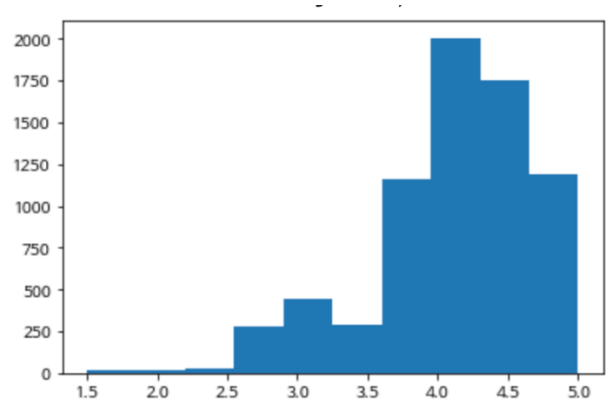
```
df.describe()
```

	학년	학점	담은인원	정원	평균평점	시험	유저평점	리뷰수
count	7186.000000	7186.000000	7186.000000	7186.000000	7186.000000	7186.000000	7186.000000	7186.000000
mean	1.014333	2.771918	74.084609	70.684525	4.107040	1.728361	4.112858	91.317840
std	0.118869	0.631043	43.961365	47.713384	0.578676	0.842979	1.128456	64.341363
min	1.000000	1.000000	12.000000	0.000000	1.500000	0.000000	1.000000	1.000000
25%	1.000000	3.000000	39.000000	36.000000	3.810000	2.000000	4.000000	38.000000
50%	1.000000	3.000000	58.000000	50.000000	4.200000	2.000000	5.000000	76.000000
75%	1.000000	3.000000	103.000000	100.000000	4.580000	2.000000	5.000000	146.000000
max	2.000000	4.000000	190.000000	198.000000	5.000000	4.000000	5.000000	200.000000

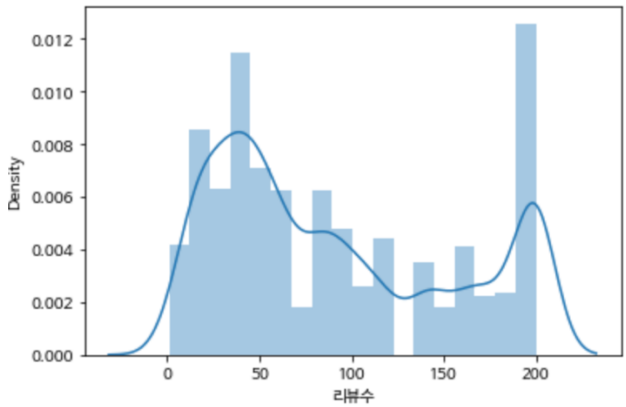
<요약통계량 표>

iii. EDA

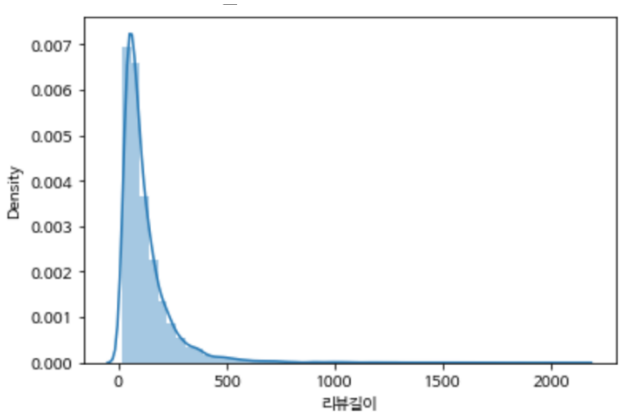
본격적인 분석에 앞서 변수를 좀 더 직관적으로 이해하기 위해 간단한 EDA를 진행했다. 평점평균의 분포를 matplotlib의 hist를 통해 확인한 결과, 주로 4~5점으로 상향 평준화된 분포 양상임을 확인했다. 또한 리뷰 수와 리뷰 길이의 분포를 seaborn의 displot을 통해 확인한 결과, 리뷰 수는 쌍봉 모양의 곡선으로 리뷰가 극단적으로 적거나 많게 치우친 분포였으며 리뷰 길이는 오른쪽으로 꼬리가 긴, 양의 왜도를 가지는 분포임을 확인할 수 있었다.



<평점평균 분포 확인>



<리뷰 수 분포 확인>



<리뷰 길이 분포 확인>

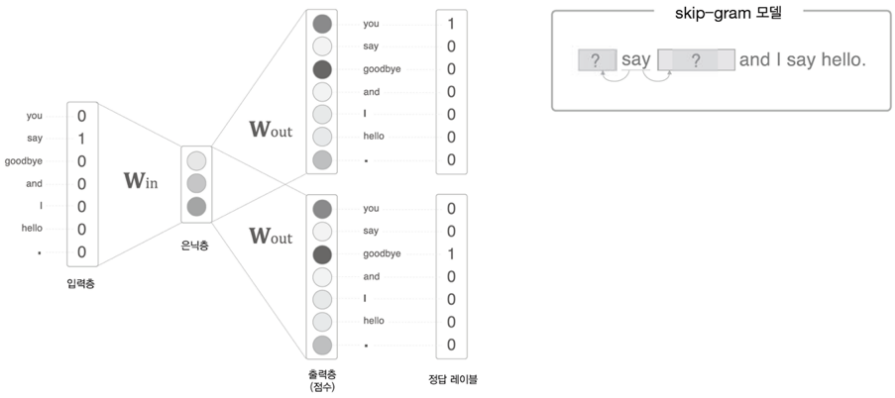
C. 데이터 분석

i. 콘텐츠 기반 추천시스템이란

본 연구에서 사용한 콘텐츠 기반 추천시스템 알고리즘에 대한 이론을 먼저 설명하면, 이는 콘텐츠를 벡터 형태로 표현하여 사용자가 이미 좋아하는 혹은 좋아할 것이라 예상하는 콘텐츠와 유사한 콘텐츠를 추천하는 가장 전통적인 추천시스템 알고리즘이다. 콘텐츠 도메인에 따라 벡터화 방법이 다양한데, 텍스트의 경우 TF-IDF나 Word2Vec 알고리즘을 주로 이용하며 이미지의 경우 CNN 모델을 이용해 벡터화한 후 유사도를 계산한다. 본 연구에서는 각 강의를 하나의 콘텐츠로 보고 강의 리뷰의 텍스트를 Word2Vec 알고리즘을 이용해 임베딩하여 유사도를 계산하고 컨텐츠 추천을 진행하고자 한다.

ii. Word2Vec이란

단어를 임베딩(Embedding)한다는 것은 단어를 실수값을 갖는 벡터로 표현하는 것이다. 텍스트라는 비정형데이터를 컴퓨터 상에서 계산하고 다루기 위해선 계산가능한 벡터 공간에 투영할 필요가 있다. Word2Vec은 대표적인 Word Embedding 방법론이며 그 방식에 따라 CBOW와 Skip-Gram으로 나뉜다. CBOW 모델은 주변(Context) 단어를 이용해 중심(Center) 단어를 예측하며 Skip-Gram 모델은 반대로 중심(Center) 단어를 이용해 주변(Context) 단어를 예측하는 방식이다. 두 모델 모두 은닉층이 1개인 얕은 신경망 모형이며 은닉층에 활성함수가 존재하지 않는 구조이다. 이는 특정 단어가 나올 가능성을 가지고 모델링하는 일종의 Multinomial Logistic Regression Model로서 모형 학습은 Cross-entropy loss function을 최소화 하는 방향으로 진행된다. 이때 CBOW는 중심단어 하나에 대한 손실을 계산한다면 Skip-Gram은 주변단어마다 손실을 구하므로 각 맥락에서 구한 손실의 총합을 최소화 해야 한다. 두 모델 중 단어 사이의 패턴을 더 잘 학습하며 특히 자주 등장하지 않는 단어 유추에서도 성능이 좋다고 알려진 Skip-Gram 모델을 이용하면 corpus 내 단어들을 골고루 잘 학습할 것으로 기대하고 본 연구에서는 Skip-Gram 모델을 이용한 임베딩을 진행했다.



<Skip-Gram 모델 아키텍처>

$$L = -\frac{1}{T}\sum_{t=1}^{T}(\log P(w_{t-1} \mid w_t) + \log P(w_{t+1} \mid w_t))$$

<Skip-Gram 모델 손실함수>

iii. 단어벡터 생성

가장 먼저 강의평에 있는 단어들의 corpus를 iteration 1,000의 Skip-gram 모델로 학습하여 단어벡터를 생성하였다. 이때 정제된 corpus를 생성하기 위한 형태소 분석기로는 mecab을 사용해 젊은 대학생들의 어휘 습관을 보다 잘 반영하고자 했고, 한국어 불용어 텍스트 파일을 이용해 기본적인 불용어 처리를 진행했다. 다음은 사용한 함수와 모델 적합 결과의 일부이다. 유사한 단어가 높은 코사인 유사도를 보이며 임베딩이 잘 됐음을 확인할 수 있다.

```
] #형태소 분석
  def tokenized_mecab(data):
      result = re.sub('\n','', data)
      result = re.sub('\u200b','',result)
      result = re.sub('\xa0','',result)
      result = re.sub('([a-zA-Z])','',result)
      result = re.sub('[ㄱ-ㅎㅏ-ㅣ]+','',result)
      result = re.sub('[-=+,#/\?:^$.@*\"※~&%·!』\\‘|\(\)\[\]\<\>`\'…》]','',result)
      result = mecab.morphs(result)
      result = ' '.join(result)
      return result

  df['tokenized'] = df['리뷰내용'].apply(tokenized_mecab)
  df[['tokenized']].head(5)
```

	tokenized
0	영어 잘 하 시 면 성적 잘 받 을 수 있 습니당 토플 라이팅 29 점 기준
1	교수 님 강의 력 은 정말 뛰어나 시 고 질문 도 잘 받 아 주 심 그런데 매주 문...
2	우선 교수 님 너무 친절 하 시 고 좋 으세요 최고
3	매 수업 시간 마다 편하 게 다가오 려 하 시 고 수업 자체 도 열정 넘치 게 하 ...
4	교수 님 정말 좋 으시 고 이 수업 을 통해서 많 은 걸 배울 수 있 다고 생각 합...

<형태소 분석>

```
#불용어 처리
def out_stopwords(data, list_stopwords):
    data = data.split(" ")
    list_stopwords = list_stopwords
    data = [token for token in data if token not in list_stopwords]
    return data

stopwords = list(open('stopwords.txt', 'r'))
list_stopwords = []
for stopword in stopwords:
  list_stopwords.append(stopword)

word_list = []
df['cleared'] = df['tokenized'].apply(lambda x: out_stopwords(x, list_stopwords=list_stopwords))
for i in df['cleared']:
  tmp = ','.join(i)
  word_list.append(tmp)

df['cleared2'] = word_list
df[['cleared2']].head(5)
```

	cleared2
0	영어,잘,하,시,면,성적,잘,받,을,수,있,습니당,토플,라이팅,29,점,기준
1	교수,님,강의,력,은,정말,뛰어나,시,고,질문,도,잘,받,아,주,심,그런데,매주,문...
2	우선,교수,님,너무,친절,하,시,고,좋,으세요,최고
3	매,수업,시간,마다,편하,게,다가오,려,하,시,고,수업,자체,도,열정,넘치,게,하,...
4	교수,님,정말,좋,으시,고,이,수업,을,통해서,많,은,걸,배울,수,있,다고,생각,합...

<불용어 처리>

```
#corpus 확인
direc ='corpus ver1.0 (전체, stopword 제거).txt'
corpus = [sent.strip().split(",") for sent in open(direc).readlines()]
corpus[:5]
```

```
[['영어',
  '잘',
  '하',
  '시',
  '면',
  '성적',
  '잘',
  '받',
  '을',
  '수',
  '있',
  '습니당',
  '토플',
  '라이팅',
  '29',
  '점',
  '기준'],
```

<생성된 corpus 일부>

```
#word2vec 학습
w2v_model = Word2Vec(corpus, size=100, window=5, min_count=5,
                     workers=12, iter=1000, hs=0, sg=1)
```

<Word2Vec 모델 학습>

```
#모델 테스트
print(w2v_model.wv.most_similar("강의"))
print(w2v_model.wv.most_similar('출석'))
print(w2v_model.wv.most_similar('시험'))
```

```
[('수업', 0.6693397760391235), ('력', 0.5731416940689087), ('교양', 0.5041673183441162), ('녹화', 0.4914613366127014), ('교수', 0.479758620262146),
[('출결', 0.7220022678375244), ('퀴즈', 0.6862040162086487), ('체크', 0.6476608514785767), ('책', 0.5115859508514404), ('처리', 0.5047526955604553),
[('기말', 0.6616092920303345), ('기말고사', 0.6295183897018433), ('중간', 0.5891581773757935), ('객관식', 0.5750774145126343), ('중간고사', 0.567284464
```

<모델 적합 결과 일부>

iv. 리뷰벡터 생성

앞에서 생성한 단어 벡터들을 바탕으로 리뷰에 들어있는 단어들의 벡터를 전부 더하여 리뷰벡터를 생성했다. 다음 코드와 결과를 통해 리뷰벡터가 잘 생성되었음을 확인할 수 있다.

```
[ ] #리뷰벡터 생성
    dict_review_vector = {}
    for idx in tqdm(df.index):
        list_vector = []
        for word in df.loc[idx]['cleared'].split():
            if word in word_dict.keys():
                list_vector.append(word_dict[word])
        dict_review_vector[df.loc[idx]['리뷰내용']] = np.sum(list_vector, axis=0).tolist()

    df['vector'] = df['리뷰내용'].map(dict_review_vector)
    df[['vector']].head(5)
```

100% 7186/7186 [00:07<00:00, 995.87it/s]

	vector
0	[0.37148723006248474, -0.1974397897720337, 1.0...
1	[3.085845708847046, -0.25097426772117615, 1.87...
2	[0.922540545463562, -0.7411057949066162, -1.83...
3	[2.0812458992004395, -0.09567178040742874, 1.5...
4	[3.8295540809631348, 1.352896809577942, 1.5570...

<리뷰벡터 생성>

v. 강의벡터 생성

최종적으로 강의 추천을 위한 강의벡터를 생성하는 단계로, 강의별 리뷰벡터들을 더하여 강의벡터를 생성했다. 강의벡터가 2차원 공간 상에 어떻게 표현되는지 확인하기 위해 t-SNE를 이용한 2차원으로의 축소 시각화를 진행했고 결과는 아래와 같다. 참고로 t-SNE는 feature extraction을 통한 비선형 차원축소를 바탕으로 고차원 데이터의 저차원 시각화에 많이 사용하는 알고리즘이며 복잡한 데이터를 한 눈에 확인하여 데이터 구조를 이해하는데 도움을 준다.

```
[ ] #강의벡터 생성
    dict_lecture_vector = {}
    for idx in tqdm(df.index):
        list_vector = []
        for word in df.loc[idx]['cleared'].split():
            if word in word_dict.keys():
                list_vector.append(word_dict[word])
        dict_lecture_vector[df.loc[idx]['교과번호']] = np.sum(list_vector, axis=0).tolist()

    df['vector2'] = df['교과번호'].map(dict_lecture_vector)
    df[['vector2']].head(5)
```

100% 7186/7186 [00:07<00:00, 739.91it/s]

	vector2
0	[12.018617630004883, 29.162105560302734, 15.53...
1	[12.018617630004883, 29.162105560302734, 15.53...
2	[12.018617630004883, 29.162105560302734, 15.53...
3	[12.018617630004883, 29.162105560302734, 15.53...
4	[12.018617630004883, 29.162105560302734, 15.53...

<강의벡터 생성>

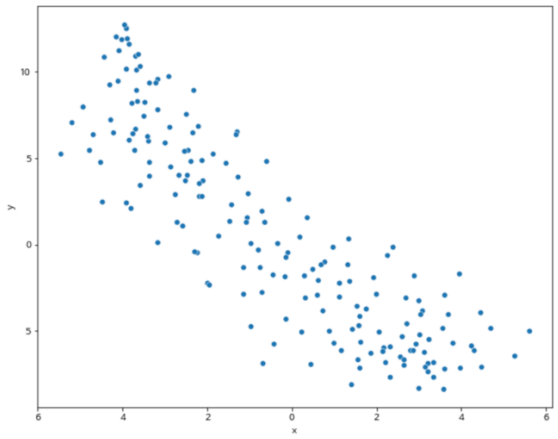
```
16] #2차원 축소
    lecture_tsne = tsne.fit_transform(vec_df.iloc[:, 1:])
    lec_df = pd.DataFrame(lecture_tsne, index=vec_df['교과번호'].tolist(), columns=['x', 'y']).reset_index()
    lec_df.head()
```

	index	x	y
0	01033-01	7.904090	0.268844
1	01034-01	4.170717	-0.713167
2	01035-01	-7.800597	-2.025708
3	01091-01	-6.075416	-1.725595
4	01092-01	-11.006075	-2.866262

<t-SNE를 이용한 차원축소>

```
#2차원 시각화
fig, ax = plt.subplots(figsize=(10, 8))
sns.scatterplot(ax=ax, data=lec_df, x="x", y="y")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f696a954090>
```



<2차원 시각화>

vi. 클러스터링

생성한 강의벡터가 유사한 특성의 강의끼리 잘 뭉쳐 있는지 확인하기 위해 추가적으로 Hierarchical Clustering과 K-Means Clustering을 진행했다. ward 연결법을 이용해 계산한 거리에 따라 그룹을 묶어가며 계층적 클러스터링 결과를 시각적으로 보여주는 Dendrogram을 통해 7~15개 정도가 적정 군집 개수라고 판단했으며, 군집 내 거리와 군집 간 거리를 이용해 같은 군집끼리 잘 뭉쳐지고 다른 군집끼리는 잘 구별되는지 클러스터링을 평가하는 척도인 Silhouette 계수를 통해 최종 클러스터 개수를 7개로 결정하였다. 최종적으로 K-Means Clustering 알고리즘을 통해 클러스터링 한 시각화는 다음과 같고 클러스터별로 잘 군집화 되어 있음을 확인할 수 있다. 즉, 강의별 특성을 잘 나타내며 임베딩 됐다고 판단할 수 있다.

```
#Hierarchical clustering
import scipy.cluster.hierarchy as shc
data = df.iloc[:, 1:3].values  #nparray 형식으로 반환

import matplotlib.pyplot as plt
plt.figure(figsize=(10, 7))
plt.title("Lectures Dendograms")
dend = shc.dendrogram(shc.linkage(data, method='ward'))    # dendrogram에서 적정 군집 수를 파악
```

<Dendrogram을 통한 적정 군집 수 파악>

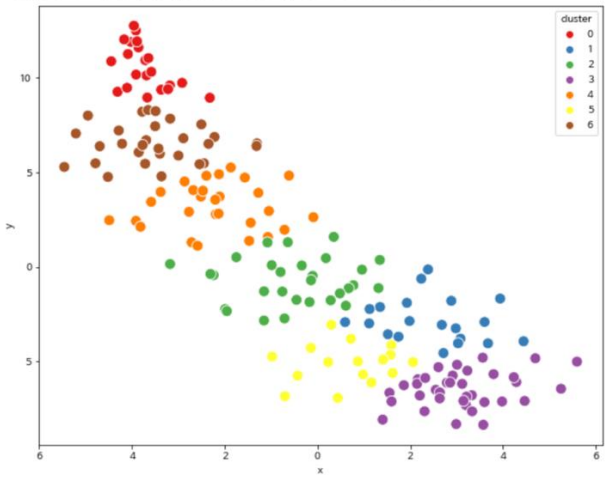
```
#Silhouette 분석
range_n_clusters = range(7, 15)

for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters,
                       max_iter=1000,
                       n_init=10)
    preds = clusterer.fit_predict(df[['x','y']])
    centers = clusterer.cluster_centers_

    score = silhouette_score(df[['x','y']], preds)
    print("For n_clusters = {}, silhouette score is {}".format(n_clusters, round(score, 4)))
```

```
For n_clusters = 7, silhouette score is 0.4066
For n_clusters = 8, silhouette score is 0.3924
For n_clusters = 9, silhouette score is 0.3742
For n_clusters = 10, silhouette score is 0.3539
For n_clusters = 11, silhouette score is 0.3572
For n_clusters = 12, silhouette score is 0.3481
For n_clusters = 13, silhouette score is 0.3489
For n_clusters = 14, silhouette score is 0.3538
```

<Silhouette 분석을 통한 군집 수 결정>

```
#시각화
fig, ax = plt.subplots(figsize=(10, 8))
sns.scatterplot(ax=ax, data=df, x="x", y="y",
                hue='cluster', palette='Set1', s=120)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f71285fc910>
```



<클러스터링 결과 시각화>

vii. 추천시스템

클러스터링을 통해 강의벡터가 잘 생성된 것을 확인했고, 궁극적으로 이를 활용해 추천시스템을 구현했다. 최종 추천시스템은 먼저 사용자의 텍스트 인풋을 받아 input2vec 함수를 통해 사용자 인풋을 벡터화 한다. 이때 텍스트 인풋은 사용자가 원하는 강의에 대한 문장형 텍스트 형식으로 받는다. 다음으로는 similar_lectures 함수를 통해 생성한 사용자 벡터와 가장 유사한 Top 5의 강의벡터를 추출해 강의 정보를 얻고, similar_reviews 함수를 통해 해당 강의 내에서 특히 유사한 리뷰 Top 3개를 추출한다. 최종 추천함수, lecture_recommendation이 반환하는 결과는 두 가지인데 하나는 강의명, 교수명, 키워드, TOP3리뷰를 칼럼으로 가지는 간결한 데이터프레임이며 다른 하나는 처음 데이터가 가지고 있던 모든 강의정보를 반환하는 자세한 데이터프레임이다. 이때 간결한 데이터프레임의 키워드 칼럼은 KR-WordRank 알고리즘을 이용한 키워드추출을 통해 발견한 강의 정보를 담는다. KR-WordRank는 TextRank나 PageRank 같이 그래프 기반 순위 알고리즘을 사용하는 키워드추출 방법론으로 기존 일본어나 중국어를 대상으로 했던 WordRank을 개선해 비지도학습 기반 한국어 단어를 추출할 수 있게 한 알고리즘이다.

```python
#인풋 텍스트 벡터화 함수
def input2vec(user_input, stopword_list, word2vec_word_dict):
    tokenized_input = tokenized_mecab(user_input)
    tokenized_input_outStopwords = out_stopwords(tokenized_input, stopword_list)

    list_vector = []
    for word in tokenized_input_outStopwords:
        if word in word_dict.keys():
            list_vector.append(word_dict[word])
    user_vector = (np.sum(list_vector, axis=0) / len(list_vector)).tolist()
    return user_vector

#비슷한 강의 찾아주는 함수
def similar_lectures(user_vector, df_w2v):

  #강의별 유사도 top5
  similarity = {}
  for idx in df_w2v.index:
    sim = cosine_similarity(np.array(user_vector).reshape(1,-1), np.array([float(i) for i in df_w2v.loc[idx]['vector'][1:-1].split(', ')]).res
    similarity[str(df_w2v.loc[idx]['교과번호'])] = float(sim)
    similarity = {key: value for key, value in sorted(similarity.items(), key=lambda item: item[1], reverse=True)}
    rating = [key for key, value in sorted(similarity.items(), key=lambda item: item[1], reverse=True)]
    top_5 = rating[:5]

  #top5 강의 및 확률 반환
  result = {}
  for i in top_5:
      result[i] = str(abs(round((similarity[i]*100), 2))) + "%"

  return result

#비슷한 강의평 찾아주는 함수
def similar_reviews(user_vector, df_w2v, lec_ids):
  lectures = {}
  #top5 강의에 대해서 반복
  for id_ in tqdm(lec_ids):
    lec_name = df_w2v[df_w2v['교과번호']==id_]['강의명'].iloc[0]
    lec_reviews = df_w2v[df_w2v['강의명'].str.contains(lec_name)]['리뷰내용'].values
    #강의리뷰 유사도 계산
    similarity = {}
    for review in lec_reviews:
      sim = cosine_similarity(np.array(user_vector).reshape(1,-1), np.array([float(i) for i in df_w2v[df_w2v.리뷰내용==review].iloc[0]['vector']
      similarity[df_w2v[df_w2v['리뷰내용']==review]['리뷰내용'].iloc[0]] = float(sim)
    similarity = {key: value for key, value in sorted(similarity.items(), key=lambda item: item[1], reverse=True)}
    rating = [str(key) for key, value in sorted(similarity.items(), key=lambda item: item[1], reverse=True)]
    top_3 = rating[:3]
    #가장 유사한 top3 리뷰 확인
    result = {}
    for i in top_3:
        result[i] = str(abs(round((similarity[i]*100), 2))) + "%"
    lectures[id_] = result
  return lectures

#추천 강의 정보를 제공하는 함수
def show_infos(dict_lec_rev, df_w2v):
  result = []
  for lec_id in dict_lec_rev:
    name = df_w2v[df_w2v['교과번호'] == lec_id]['강의명'].iloc[0]
    for rev_id in list(dict_lec_rev[lec_id]):
      dict_temp = {}
      dict_temp['강의명'] = name
      dict_temp['교수명'] = df_w2v[df_w2v['리뷰내용']==rev_id]['교수명'].iloc[0]
      dict_temp['키워드'] = df_w2v[df_w2v['리뷰내용']==rev_id]['키워드'].iloc[0]
      dict_temp['TOP3리뷰'] = df_w2v[df_w2v['리뷰내용']==rev_id]['리뷰내용'].iloc[0]
      result.append(dict_temp)
  return result
```

<필요한 함수 정의>

```python
#최종 추천 함수
def lecture_recommendation(user_input):

  #list_stopword
  stopwords = list(open('stopwords.txt', 'r'))
  list_stopwords = []
  for stopword in stopwords:
    list_stopwords.append(stopword)

  #word_dict
  w2v_model = Word2Vec.load('word2vec ver1.0 (iter500).model')
  word_dict = {}
  for vocab in tqdm(w2v_model.wv.index2word):
      word_dict[vocab] = w2v_model.wv[vocab]

  #recommendation
  user_vector = input2vec(user_input, list_stopwords, word_dict)
  dict_lec = similar_lectures(user_vector, df_w2v)
  dict_lec_rev = similar_reviews(user_vector, df_w2v, dict_lec)
  lec_rev_info = show_infos(dict_lec_rev, df_w2v)
  res_short = pd.DataFrame(lec_rev_info)

  #return
  res_long = pd.DataFrame()
  for i in range(5):
    lec_no = list(dict_lec_rev.items())[i][0]
    tmp = df_final[df_final['교과번호']==lec_no]
  res_long = pd.concat([res_long, tmp]).reset_index(drop=True)

  return res_short, res_long
```

<최종 추천 함수 정의>

```python
#키워드 추출 함수 정의
!pip install KRWordRank
def lecture_keyword(reviewlist, stopwords):
  from krwordrank.word import summarize_with_keywords
  keywords = summarize_with_keywords(reviewlist, min_count=3, max_length=10,
             beta=0.85, max_iter=10, stopwords=stopwords, verbose=True)
  return keywords
```

<키워드추출 함수 정의>

```python
#키워드 칼럼 생성
name_list = df['교과번호'].unique().tolist()
key_list = []
for i, name in enumerate(name_list):
  try:
    keywords = lecture_keyword(locals()['review_list{}'.format(i+1)], stopwords)
    top5 = sorted(keywords)[:5]
  except:
    top5 = np.nan
  for j in range(len(df[df['교과번호']==name])):
    key_list.append(top5)
df['키워드'] = key_list
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: KRWordRank in /usr/local/lib/python3.7/dist-packages (1.0.3)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.4.1)
Requirement already satisfied: numpy>=1.18.4 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.21.6)
Requirement already satisfied: scikit-learn>=0.22.1 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.0.2)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.1->KRWordRank) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.1->KRWordRank) (3.1.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
scan vocabs ...
num vocabs = 43
done
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: KRWordRank in /usr/local/lib/python3.7/dist-packages (1.0.3)
Requirement already satisfied: numpy>=1.18.4 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.21.6)
Requirement already satisfied: scikit-learn>=0.22.1 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.0.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from KRWordRank) (1.4.1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.1->KRWordRank) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.1->KRWordRank) (3.1.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use
scan vocabs ...
num vocabs = 60
done
```

<키워드 칼럼 생성>

viii. 추천 예시

구축한 추천시스템 모델을 통해 직접 강의 추천을 해본 결과이다. 예시1은 사용자 인풋으로 "교수님 강의력이 좋고 열정적인 분이면 좋겠어요"를 받아 ['화학및실험1', '물리학및실험1', '스포츠산업과스포츠마케팅', '현대행정법과법의체계', '사회학에의초대'] 5개의 강의를 추천한 모습이다. 대부분 강의력과 교수님의 친절, 열정에 대한 긍정적인 평가를 가진 강의가 잘 추천됐음을 볼 수 있었다. 예시2는 "학점을 잘 주는 꿀강의를 원합니다"라는 사용자 인풋을 받아 ['시민의자유와헌법', '물리학및실험1', '일

반물리학및실습', '문화속의사회', '회계의이해'] 5개의 강의를 추천했다. 학점을 잘 받을 수 있다는 평가를 가진 강의도 있지만 '꿀강의'라는 단어는 corpus에 존재하지 않았기 때문에 제대로 반영되지 않은 면도 있다. 추천의 특성상 정확도를 정확히 파악할 수 없으나 사용자 인풋과 강의리뷰의 내용이 상당 부분 유사한 것을 보아 적절한 추천이 이뤄졌다고 볼 수 있다.

```
res_short, res_long = lecture_recommendation('교수님 강의력이 좋고 열정적인 분이면 좋겠어요!')
```

```
100% 3086/3086 [00:00<00:00, 10356.16it/s]
100% 5/5 [00:01<00:00, 4.15it/s]
```

```
res_short
```

	강의명	교수명	키워드	TOP3리뷰
0	화학및실험1	조대행	['강의', '공부', '과제', '교수', '내용']	내용이 많이 쉽긴 하지만 교수님 강의력이 무난해서 이럭저럭 재미있게 들음
1	화학및실험1	권해두	['개인', '공부', '과제', '교수', '기말']	진짜 권해두 교수님 너무 좋습니다. 근데 실험 윤미경 교수님이 좀 빡세게 채점하시는...
2	화학및실험1	윤미경	['강의', '공부', '과제', '교수', '기말']	교수님이 열정적이시고 전달력이 좋으십니다\n그것과는 별개로 좀 깐깐하십니다\n시험도...
3	물리학및실험1	유병덕	['교과', '교수', '기말', '내용', '레포']	출석은 직접 호명하시는데 교수님이 부르시고 싶으신 날과 시간에 부르세요... 시험은...
4	물리학및실험1	김민아	['강의', '고등', '공부', '과목', '과제']	시험을 말아 먹었지만 보고서라도 열심히 작성했더니 B+…\n교수님 강의력도 괜찮으시...
5	물리학및실험1	이나영	['강의', '강추', '고등', '공부', '과목']	교수님 강의력이 너무 좋으시고 친절하게 가르쳐주심\n1학년 수업이라 애들이 좀 떠드...
6	스포츠산업과스포츠마케팅	김성규	['강의', '개꿀', '개인', '경기', '공부']	운동 좋아하시면 꼭 들으세요 내년에 수업 방식이 어떨진 모르겠지만 비대면이면 정말 ...
7	스포츠산업과스포츠마케팅	김성규	['강의', '개꿀', '개인', '경기', '공부']	꿀꿀꿀꿀강의~\n짱인강의입니다 교수님도 너무 좋습니다
8	스포츠산업과스포츠마케팅	김성규	['강의', '개꿀', '개인', '경기', '공부']	스포츠를 좋아한다면 꼭 들으세요. 교수님 친절하시고 강의 내용도 적어요! 점수 잘받...
9	현대행정과법의체계	최윤영	['강의', '개인', '걱정', '공부', '과목']	교수님 정말 친절하시고 강의력도 좋아요! 다만 나긋나긋하셔서 중간중간 졸았던 적도 ...
10	현대행정과법의체계	최윤영	['강의', '개인', '걱정', '공부', '과목']	교수님께서 정말 꼼꼼하게 설명해주신다. 시험 난이도도 적절했고 강의력이 정말 좋으심...
11	현대행정과법의체계	최윤영	['강의', '개인', '걱정', '공부', '과목']	너무너무너무 좋으신 교수릠,,,👍👍👍꼭 들으세요...학생들 배려도 많이 해주시고 과...
12	사회학에의초대	최가영	['강의', '공부', '과목', '과제', '교수']	들을 만해요. 교수님 강의력은 별로지만 조별토론이 재미있었어요
13	사회학에의초대	박상희	['강의', '개인', '공부', '과목', '과제']	교수님이 열정이 너무 가득하셔서 수업을 빡세게 진행하심 그래도 학점은 굿
14	사회학에의초대	최가영	['강의', '공부', '과목', '과제', '교수']	매우매우매우 만족이라 아래 수강평보고 놀람.. 솔직히 과제 많은거 때문인가 싶을정도...

```
res_long
```

	강의명	교수명	학년	교과번호	학점	강의시간	강의유형	담은인원	정원	링크	개설학기	평균평점	과제	조모임	성적	출결	시험	유저평점	수강학기	리뷰내용	리뷰수
0	사회학에의초대	최가영	1.0	01236-01	3	목06,07,08/15-118/119	교양선택	21	27	/lecture/view/1940569	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2019-1	2.96	많음	없음	보통	직접호명	2	4.0	21년 1학기 수강자	장황한 설명때문에 지루하긴 하지만, 사회학에 대해 1도 모르던 사람으로썬 나름 흥미...	26
1	사회학에의초대	최가영	1.0	01236-01	3	목06,07,08/15-118/119	교양선택	21	27	/lecture/view/1940569	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2019-1	2.96	많음	없음	보통	직접호명	2	3.0	20년 2학기 수강자	빡세게 들을 수 있는 교양.. 토론, 발표 다 하고 질문 점수도 있어서 부담스러울 ...	26
2	사회학에의초대	최가영	1.0	01236-01	3	목06,07,08/15-118/119	교양선택	21	27	/lecture/view/1940569	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2019-1	2.96	많음	없음	보통	직접호명	2	1.0	21년 1학기 수강자	수업에서 매주 과제를 내주십니다 그리고 기말 발표를 하는데 준비할때 부담이 꽤 됨니...	26
3	사회학에의초대	최가영	1.0	01236-01	3	목06,07,08/15-118/119	교양선택	21	27	/lecture/view/1940569	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2019-1	2.96	많음	없음	보통	직접호명	2	4.0	21년 1학기 수강자	중으로 수업. 개인적으로 사회학에 관심이 많아서 정말 재미있었고, 교수님 강의력도 ...	26
4	사회학에의초대	최가영	1.0	01236-01	3	목06,07,08/15-118/119	교양선택	21	27	/lecture/view/1940569	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2019-1	2.96	많음	없음	보통	직접호명	2	1.0	21년 1학기 수강자	교수님 강의력도 좋지 않고 매주 뭘 해야함 3학점 교양이 아닌 전공보다 빡센듯하게 ...	26

<예시1. "교수님 강의력이 좋고 열정적인 분이면 좋겠다.">

```
] res_short, res_long = lecture_recommendation('학점을 잘 주는 꿀강의를 원합니다.')

100% 3086/3086 [00:00<00:00, 64818.58it/s]
100% 5/5 [00:01<00:00, 3.59it/s]
```

res_short

	강의명	교수명	키워드	TOP3리뷰
0	시민의자유와헌법	남정아	['감상', '강의', '개인', '객관식', '거부']	전 좋았습니다. 교수님이 비율을 꽉꽉 채워주셔서 그런지 학점받기가 수월한것 같습니다...
1	시민의자유와헌법	남정아	['감상', '강의', '개인', '객관식', '거부']	수업 잘 듣고 사례 잘 보면 좋은 성적 받을 수 있어요.\n법 잘 모르는데도 좋은 ...
2	시민의자유와헌법	남정아	['감상', '강의', '개인', '객관식', '거부']	교수님이 학생배려를 잘해주심. 일단 학점을 잘주시기때문에 불만 전혀없는수업.
3	물리학및실험1	남윤성	['강의', '공부', '과제', '교수', '기말']	수업을 재밌게 하려고 노력은 하시지만 ㅠㅠ 잘 모르겠다 물리 노베라는 가정하에 수업...
4	물리학및실험1	김민아	['강의', '고등', '공부', '과목', '과제']	다른교수님의 같은강의를 안들어봐서 상대적비교는 힘들지만 큰 장단점없이 무난합니다\n...
5	물리학및실험1	이나영	['강의', '강추', '고등', '공부', '과목']	교수님\n친절하시고 이해도 쉽게 설명해주세요!\n제가 공부를 안해서 학점은 별로지만...
6	일반물리학및실습	이나영	['강의', '공부', '교수', '기말', '레포']	교수님도 완전 친절하시고 모르는거 질문하면 친절하게 다 알려주심\n교수님이 물포자들...
7	일반물리학및실습	이나영	['강의', '공부', '교수', '기말', '레포']	갓나영님! 학생들이 물리를 좋아하지도 잘하지도 않는다는 것을 알고계셔서 최대한 맞춰...
8	일반물리학및실습	이나영	['강의', '공부', '교수', '기말', '레포']	교수님 너무 좋으세요..진짜 짱!!!! 제가 진짜 물리 못하고 시험도 매번 못봤는데...
9	문화속의사회	전인한	['강의', '공부', '과제', '교수', '교양']	싸강 기준으로 작성해보자면 중간고사를 과제 3개로 대체하면서 거의 한달 넘게 과제에...
10	문화속의사회	전인한	['강의', '공부', '과제', '교수', '교양']	정말 매 강의에 최선을 다하시는 교수님임. 강의력도 좋으시고 아는게 많으심. 원래 ...
11	문화속의사회	전인한	['강의', '공부', '과제', '교수', '교양']	일단 과제가 매우매우매우 많음 싸강으로 바뀌어서 그런지 조별과제 발표 하나랑 일반 ...
12	회계의이해	박종찬	['가능', '강의', '개념', '개인', '게시']	그저 갓갓 교수님 ... 성적 잘 주심.\n강의력도 좋고 질문도 잘 받아주심\n문제...
13	회계의이해	박종찬	['가능', '강의', '개념', '개인', '게시']	수업 정말 잘 가르쳐주시고 회계의 회자도 몰라도 다 이해가능해요! 공부좀만 하면 좋...
14	회계의이해	박종찬	['가능', '강의', '개념', '개인', '게시']	설명도 친절하시고 강의력 좋으십니다!\n과제가 많은건 시험에 도움이 되어서 괜찮았고...

res_long

	강의명	교수명	학년	교과번호	학점	강의시간	강의유형	담은인원	정원	링크	개설학기	평균평점	과제	조모임	성적	출결	시험	유저평점	수강학기	리뷰내용	리뷰수
0	시민의자유와헌법	남정아	1.0	01289-01	3	월08,09,10/20-207/208	교양선택	56	50	/lecture/view/1040639	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2018-1...	3.55	보통	없음	너그러움	직접호명	2	3.0	22년 1학기 수강자	헌법... 기본적으로 알고 있는 사람들에겐 제법 쉬운 강의 같습니다 정치와 법 공부...	164
1	시민의자유와헌법	남정아	1.0	01289-01	3	월08,09,10/20-207/208	교양선택	56	50	/lecture/view/1040639	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2018-1...	3.55	보통	없음	너그러움	직접호명	2	3.0	22년 1학기 수강자	노베한텐 열심히 공부 안하면 어려움. 공부량이 너무 많음..	164
2	시민의자유와헌법	남정아	1.0	01289-01	3	월08,09,10/20-207/208	교양선택	56	50	/lecture/view/1040639	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2018-1...	3.55	보통	없음	너그러움	직접호명	2	4.0	22년 1학기 수강자	나는 수험 헌법을 듣고 수업을 들어 따라오는 데 큰 무리가 없지만 강의가 쉽거나 내...	164
3	시민의자유와헌법	남정아	1.0	01289-01	3	월08,09,10/20-207/208	교양선택	56	50	/lecture/view/1040639	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2018-1...	3.55	보통	없음	너그러움	직접호명	2	5.0	20년 2학기 수강자	꿀강 중의 꿀강 .. 학교 다니면서 꿀이라는 강의들 들어봤는데 그 중 제일 좋았음 ...	164
4	시민의자유와헌법	남정아	1.0	01289-01	3	월08,09,10/20-207/208	교양선택	56	50	/lecture/view/1040639	2022-1, 2021-2, 2021-1, 2020-2, 2020-1, 2018-1...	3.55	보통	없음	너그러움	직접호명	2	5.0	21년 2학기 수강자	겨울계절학기로 수강했습니다. 과제 없이 시험만 2번 봤고, 중간은 서술형 기말은 객...	164

<예시2. "학점을 잘 주는 꿀강의를 원합니다.">

3. 결론

A. 의의

많은 대학생들의 입장에서 필요하다고 느낄만한 분야에 추천시스템을 도입해 실용적이고 도움이 될 수 있는 주제를 제시했다. 가장 기본적인 콘텐츠 기반 추천을 통해 쉽고 직관적인 추천을 진행해 '왜 이런 강의가 추천됐는지'에 대한 설명력이 높다는 장점이 있다. 또한 사용자가 원하는 강의의 조건에 단순히 맞춰서 추천하는 것이 아닌 텍스트 인풋을 받아 유사도가 높은 강의를 추천하므로 좀 더 섬세한 추천이 가능하며 기존 유저 정보가 필요하지 않으므로 추천시스템에서 주로 대두되는 cold start 문제에 강건하다는

장점이 있다. 또한 현재는 학생들의 입장에서 강의 추천에 집중해 리뷰를 다뤘지만 반대로 강의자의 입장에서 학생들이 만족하는 수업의 요인을 분석하고 파악해볼 수도 있을 것이다.

B. 한계

사용자 정보가 없는 리뷰 정보를 이용하기 위해 콘텐츠 기반 추천이라는 주제에 갇혀 추천시스템을 구현하다 보니 강의 추천의 본질을 놓쳤다는 아쉬움이 남는다. 특히 기존 데이터프레임에 존재하는 유저평점이나 시험, 과제, 출결 등의 강의를 구성하는 정량 및 정성적 칼럼에 대한 정보 반영이 직접적으로 이뤄지지 않아 이를 포함한 알고리즘 향상을 기대한다. 또한 '꿀강의', '싸강' 등 사전에 등재되지 않은 단어를 사용자 정의 사전에 추가하여 텍스트 전처리를 더 꼼꼼히 할 필요성이 있으며, 사용자 인풋 텍스트가 길고 복잡해질수록 추천 성능이 좋지 않아 모델 개선의 여지가 남아있다.

4. 참고문헌

한지영, 허고은. (2021). 토픽 모델링 기반 비대면 강의평 분석 및 딥러닝 분류 모델 개발. *한국문헌정보학회지*, 55(4), pp. 267-291.

HE JINLU. (2021). *사용자 평점과 리뷰 유사도를 이용한 협업 필터링 기반 영화 추천시스템*(석사). 경희대학교 대학원, 서울.

NLP를 활용한 20대 국회 법안&국회의원 시각화 및 국회의원 추천시스템. (2020). https://dacon.io/codeshare/1985

X-senators (국회의원 추천 Web). (2020). https://github.com/hw79chopin/X-senators

5. 부록

보고서 상에 모든 코드를 담지 못하여 사용한 코드를 올려놓은 github 링크를 공유드립니다.
https://github.com/DieKim/everytime_lecture_recommendation