네이버 영화 리뷰 자연어 처리 및 분석

목차

마블 영화 리뷰 분석

2022년에 개봉한 마블 영화 리뷰 분석

- 어벤져스 : 엔드게임 이후 마블 영화 평판이 안좋아지기 시작했다.
- 어벤져스 : 엔드게임(2019.04) 이전에는 전체적으로 마블 관련 영화들의 관객 수가 증가하는 추세였지만 이후에는 하락하는 추세이다.(시리즈 별 정리)
  - 아이언맨1(430만) → 아이언맨2(442만) → 아이언맨3(900만)
  - 퍼스트 어벤져(51만) → 윈터 솔져(396만) → 시빌 워(867만)
  - 가디언즈 오브 갤럭시(134만) → 가디언즈 오브 갤럭시2(273만)
  - 어벤져스(708만) → 에이지 오브 울트론(1050만) → 인피니티 워(1123만) → 엔드게임(1397만)
  - 앤트맨(284만) →앤트맨과 와스프(544만)
  - 스파이더맨 - 홈커밍(725만) → **파 프롬 홈(802만)** → **노 웨이 홈(755만)**
  - 닥터 스트레인지(544만) → **닥터 스트레인지: 대혼돈의 멀티버스(588만)**
  - 토르1(169만) → 토르2(304만) → 토르3(485만) → **토르4(271만)**
  - 블랙 팬서(540만) → **블랙 팬서 : 와칸다 포에버(210만)**
    - 굵은 글씨가 엔드 게임 이후 영화
- 2022년에 개봉한 3개의 영화 닥터 스트레인지: 대혼돈의 멀티버스, 토르 : 러브 앤 썬더, 블랙 팬서 : 와칸다 포에버에 대한 리뷰 분석을 통하여 부정적인 의미의 단어를 찾아 현재 마블 영화의 관객 감소 추세 원인에 대해 분석해보려고 한다.
  - 가설 : 2022년에 개봉한 3개의 마블 영화 리뷰에서는 부정적인 내용이 많을 것이다.
    - 각 영화 별 리뷰 첫 페이지에 있는 10개의 리뷰를 추출하여 분석 → 총 30개
    - 리뷰의 명사, 동사를 활용하여 부정적인 리뷰를 분류하기

▼ 웹 서버 요청

- 활용 페이지
  - 닥터 스트레인지 : 대혼돈의 멀티버스https://movie.naver.com/movie/bi/mi/review.naver?code=182016
  - 토르 : 러브 앤 썬더

    https://movie.naver.com/movie/bi/mi/review.naver?code=187347

- 블랙 팬서 : 와칸다 포에버https://movie.naver.com/movie/bi/mi/review.naver?code=184516

```
# 웹 서버 요청 함수
def web_server_request(code_number):
    # 네이버 영화 사이트 - 영화 리뷰 페이지
    url = "https://movie.naver.com/movie/bi/mi/review.naver?code={}".format(code_number)

    # reqeust.get
    resp = requests.get(url)
    return resp

doctor = web_server_request(182016) # 닥터 스트레인지 : 대혼돈의 멀티버스
thor = web_server_request(187347) # 토르 : 러브 앤 썬더
black = web_server_request(184516) # 블랙팬서 : 와칸다 포에버

print(doctor)
print(thor)
print(black)
```

```
<Response [200]>
<Response [200]>
<Response [200]>
```

응답 내용 확인

```
doctor.text[150:350]
```

ng="ko">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="imagetoolbar" content="no">
<title>닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화</title>

```
thor.text[150:350]
```

ng="ko">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="imagetoolbar" content="no">
<title>토르: 러브 앤 썬더 : 네이버 영화</title>

```
black.text[150:350]
```

ng="ko">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="imagetoolbar" content="no">
<title>블랙 팬서: 와칸다 포에버 : 네이버 영화</title>

▼ HTML 파싱

```
# 3개의 영화 서버 응답 텍스트 리스트 생성
movie_text = [doctor.text, thor.text, black.text]

# 파싱한 내용을 담을 리스트 생성
```

```
soup_list = []

# BeautifulSoup 함수로, HTML 문서 구조를 파싱
for text in movie_text:
    soup = BeautifulSoup(text, 'html.parser')
    soup_list.append(soup)
    # 파싱한 내용을 담고 있는 객체의 자료형 확인
    print(type(soup))
```

```
<class 'bs4.BeautifulSoup'>
<class 'bs4.BeautifulSoup'>
<class 'bs4.BeautifulSoup'>
```

- 영화 제목

```
for soup in soup_list:
    # <title> 영화 이름 : 네이버 영화</title>
    # title 태그 이름을 활용하여 영화 제목이 포함되어 있는 요소를 찾습니다.
    title_tag = soup.find("title")
    print(title_tag)

    # 텍스트 부분만 출력
    title_text = title_tag.get_text()
    print(title_text)
```

```
<title>닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화</title>
닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화
<title>토르: 러브 앤 썬더 : 네이버 영화</title>
토르: 러브 앤 썬더 : 네이버 영화
<title>블랙 팬서: 와칸다 포에버 : 네이버 영화</title>
블랙 팬서: 와칸다 포에버 : 네이버 영화
```

- 리뷰 갯수

```
for soup in soup_list:
    # <span class="cnt">총<em>건수</em>건</span>

    # span 태그의 class 속성값을 활용하여 리뷰 갯수가 포함되어 있는 요소를 찾습니다.
    count_tag = soup.find("span", attrs = {'class': 'cnt'})
    print("span 태그: ", count_tag)

    # count_tag 요소에서 em 태그 부분을 찾습니다.
    count_tag = count_tag.find('em')
    print("em 태그: ", count_tag)

    # 텍스트 부분만 추출합니다.
    count_text = count_tag.get_text()
    print("텍스트: ", count_text)
```

```
span 태그:  <span class="cnt">총<em>140</em>건</span>
em 태그:  <em>140</em>
텍스트:  140
span 태그:  <span class="cnt">총<em>67</em>건</span>
em 태그:  <em>67</em>
텍스트:  67
span 태그:  <span class="cnt">총<em>19</em>건</span>
em 태그:  <em>19</em>
텍스트:  19
```

- 리뷰 목록
  - 결과는 코드에 참조

```
code_number = [182016, 187347, 184516] # 각 영화별 페이지 code number
i = 0 # 초기 값 설정

# 30개의 영화 제목, 리뷰 제목, 사용자, url을 담을 리스트 생성
movie_list = []
title_list = []
uid_list = []
url_list = []

for soup in soup_list:
    # <ul class="rvw_list_area">

    # ul 태그의 class 속성값을 활용하여 리뷰 제목과 링크가 포함되어 있는 요소를 찾습니다.
    review_list_tag = soup.find('ul', attrs = {'class':'rvw_list_area'})

    # review_list_tag 요소에 포함된 li 태그를 모두 찾습니다.
    review_list_tags = review_list_tag.find_all('li')

    # 10개의 리뷰를 반복문으로 조회하면서, 영화제목, 리뷰 제목(rli.title), 사용자(rli.uid), 상세 페이지 url 값을 추출합니다.
    for li_tag in review_list_tags:
        title_text = soup.find("title").get_text()
        movie_list.append(title_text)

        review_title = li_tag.find_all('a')[0].get_text()
        title_list.append(review_title)

        review_uid = li_tag.find_all('a')[1].get_text()
        uid_list.append(review_uid)

        review_nid = re.findall('\d{7}', li_tag.find('a').get('onclick'))[0]
        review_url = f"https://movie.naver.com/movie/bi/mi/reviewread.naver?nid={review_nid}&code={code_number[i]}&order=#tab"
        url_list.append(review_url)
    i += 1 # 영화 리뷰 10개 정보 추가하면 다음 영화로 넘어가기

print(movie_list)
print(title_list)
print(uid_list)
print(url_list)
```

- 리뷰 상세 페이지
  - 결과는 코드에 참조

```
# 리뷰 상세페이지의 HTML 소스코드를 가져와서 파싱(parsing)
resp_text = requests.get(url_list[0])
soup_text = BeautifulSoup(resp_text.text, 'html.parser')

# 리뷰 본문의 텍스트를 추출합니다. / <div class = "user_tx_area">
review_text_tag = soup_text.find('div', attrs={'class':'user_tx_area'})

# 텍스트 부분만 추출합니다.
review_text = review_text_tag.get_text()
print(review_text)
```

- 30개의 리뷰 본문을 모두 수집 -> list로 정리

```
# url 반복하여 텍스트를 추출하고 리스트에 추가

text_list = []

for url in url_list:

    # 리뷰 상세 페이지의 HTML 소스코드를 가져와서 파싱(parsing)
    resp_text = requests.get(url)
    soup_text = BeautifulSoup(resp_text.text, 'html.parser')

    # 리뷰 본문의 텍스트를 추출합니다. / <div class = "user_tx_area">
    review_text_tag = soup_text.find('div',attrs={'class':'user_tx_area'})

    # 텍스트 부분만 추출합니다.
    review_text = review_text_tag.get_text()
```

```
    text_list.append(review_text)

    # 대기 시간을 추가합니다. (서버에 과도한 호출이 되지 않도록 유의)
    time.sleep(2)

# 추출된 아이템의 수량
print(len(text_list))
```

30

▼ 데이터 수집

```
# 딕셔너리 형식으로 항목별 리스트를 원소로 추가
dict_data = {
    'movie' : movie_list,
    'title' : title_list,
    'user' : uid_list,
    'review' : text_list
}

# 판다스 데이터프레임으로 변환
df_data = pd.DataFrame(dict_data)

# 변환 결과를 확인
df_data
```

	movie	title	user	review
0	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	저는 이글을 2년뒤인 2021년에 볼 예정입니다	vita****	₩n2021년이면 내가 18살이구만.. 고2.. 과연그때까지 살아있을까... 제 3...
1	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	미래에서 왔습니다	idso****	₩n닥터 스트레인지는 타임스톤에 마법을 걸어 두었습니다.타노스에게 넘겨주기 직전 타...
2	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	탈모 스트레인지 2	s220****	₩r₩n₩t₩t₩t₩t이전 닥터 스트레인지 1편을 보신 분들은 아실 겁니다. 마법사...
3	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	(스포 有) 닥스2 캐릭터 아주 자세한 고찰: 닥터 스트레인지: 대혼돈의 멀티버스(...	aasd****	₩n드디어..마침내..닥터 스트레인지 2가 개봉했습니다.이 글은 닥터스트레인지2, ...
4	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	[조대장] 마블 영화 네이버 팬카페	mc_g****	₩n마블 시네마틱 네이버 카페본 카페는 MCU의 세계관을 다룬, 마블 스튜디오가 직...
5	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	[영화감상] 닥터 스트레인지: 대혼돈의 멀티버스 (Doctor Strange in ...	sakg****	₩n5월 4일 개봉하는 영화 '닥터 스트레인지: 대혼돈의 멀티버스'입니다.'마블'시...
6	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	개봉 언제죠... 개봉 하면 당장 달려갑니다...	hmin****	₩r₩n₩t₩t₩t₩t닥터스트레인지1 재밋었는데 2도 엄청 기대중 ㅠㅠㅠㅠㅠㅠㅠㅠㅠㅠ...
7	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	[영화 간단 리뷰] 닥터 스트레인지: 대혼돈의 멀티버스 (2022)	choj****	₩n오늘은 <닥터 스트레인지: 대혼돈의 멀티버스>를 봤습니다.2016년에 개봉한 <...
8	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	2년뒤에 (인워 스포일러 솔솔(개 소리	char****	₩n₩r₩n(스포일러있음)[세상에.. 일단 타노스가 손가락튕겨서 일단 별점 10...
9	닥터 스트레인지: 대혼돈의 멀티버스 : 네이버 영화	닥터 스트레인지2 대혼돈의 멀티버스 마블영화 쿠키영상 스포일러 후기	ptj0****	₩n ₩n₩n₩n닥터 스트레인지: 대혼돈의 멀티버스₩n₩n감독₩n샘 레이미₩n출연₩...
10	토르: 러브 앤 썬더 : 네이버 영화	형~ 채식 잘 하고 있죠?	choo****	₩n지난번에 형네 엄마가 그랬잖아요 ..채식하라구 ..맥주 끊기는 어렵겠지만. 형 ...
11	토르: 러브 앤 썬더 : 네이버 영화	2년뒤 나에게 쓰는 마블을 좋아하는 한 고2의 미래편지(과거에서 왔습돠!!)	jioa****	₩r₩n₩t₩t₩t현재는 2019년 8월 23일 금요일 나는 학교를 마치고 자습...
12	토르: 러브 앤 썬더 : 네이버 영화	<토르: 러브 앤 썬더> 역대급 애매함	jime****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 햄...
13	토르: 러브 앤 썬더 : 네이버 영화	토르: 러브 앤 썬더 (동화스런 신성모독을 위한 자유도 벨런스 붕괴 외전)-평점 4점	reno****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 햄...
14	토르: 러브 앤 썬더 : 네이버 영화	토르형~ 드디어 결혼 하는거야?	get1****	₩n원래 남녀가 헤어지고 만나고 헤어지고 만나고 그러는거니까이해하는거고 미리 축하해...
15	토르: 러브 앤 썬더 : 네이버 영화	브링미 따노수우ㅜㅜㅜㅜㅜ	rhkd****	₩r₩n₩t₩t₩t₩t₩r₩n성지순례하고간다아ㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏ...
16	토르: 러브 앤 썬더 : 네이버 영화	[영화감상] 토르: 러브 앤 썬더 (Thor: Love and Thunder, 2022)	sakg****	₩n 7월 6일 개봉하는 영화 '토르 : 러브 앤 썬더'입니다. '토르'시리즈의 네...
17	토르: 러브 앤 썬더 : 네이버 영화	토르 러브 앤 썬더 and Marvel Studios' Thor: Love and ...	kjgh****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 햄...
18	토르: 러브 앤 썬더 : 네이버 영화	후후 현재 고3	chan****	₩n₩r₩n현재 수시 원서 접수를 마친 고3이지.. 어벤져스 엔드게임 또 보며 눈물...
19	토르: 러브 앤 썬더 : 네이버 영화	<토르: 러브 앤 썬더> 후기 스포없음 재밋게 시간순삭	maro****	₩n정말이지 얼마만의 극장 나들이던가!!! 혼자 시사회를 가거나 했던게 전부였는데 ...
20	블랙 팬서: 와칸다 포에버 : 네이버 영화	[블랙팬서2] 굿바이 故채드윅 보스만 " 블랙팬서 : 와칸다 포에버 " (+스포,...	mapa****	₩n안녕하세요 신비에용.오늘 소개할 영화는 블랙팬서 2번째 시리즈," 블랙팬서 : ...
21	블랙 팬서: 와칸다 포에버 : 네이버 영화	[블랙팬서:와칸다 포에버]<예매권>(67번째) 조연들만 가득모여 블랙팬서를 그리워만...	acts****	₩n20221109의정부 CGV IMAX-E-12VIP 쿠폰2.5/5 "와칸다를 지...
22	블랙 팬서: 와칸다 포에버 : 네이버 영화	[블랙 팬서: 와칸다 포에버]를 보고	film****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
23	블랙 팬서: 와칸다 포에버 : 네이버 영화	[블랙팬서: 와칸다 포에버] 티저 예고편	kjgh****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
24	블랙 팬서: 와칸다 포에버 : 네이버 영화	Reflect on the cultural impact and legacy of B...	kjgh****	₩nMarvel Studios @MarvelStudiosReflect on the ...
25	블랙 팬서: 와칸다 포에버 : 네이버 영화	Marvel Future Fight @Marvel_FFightUS · Join ...	kjgh****	₩n#Marvel Future Fight @Marvel_FFightUSJoin M'...
26	블랙 팬서: 와칸다 포에버 : 네이버 영화	[스포, 영화리뷰] 블랙팬서 : 와칸다 포에버	gkal****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
27	블랙 팬서: 와칸다 포에버 : 네이버 영화	[CGV 전주 효자] 블랙 팬서: 와칸다 포에버 아이맥스 - 관람후기	jihu****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
28	블랙 팬서: 와칸다 포에버 : 네이버 영화	<영화> 블랙 팬서: 와칸다 포에버 2022.11.22	elle****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
29	블랙 팬서: 와칸다 포에버 : 네이버 영화	[블랙 팬서: 와칸다 포에버] ★★☆ 주제와 의미는 좋지만, 히어로 영화로써 실격이다.	zxc1****	₩n ₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티...

- movie 컬럼 처리
  - movie 컬럼이 ':'가 2개가 있지만 앞에 :는 뒤에만 띄어쓰기가 되어 있고 뒤에 :는 앞뒤로 띄어쓰기가 되어있기 때문에 ' : '로 분할

```
df_data['movie'] = df_data['movie'].str.split(' : ').str[0]
df_data
```

	movie	title	user	review
0	닥터 스트레인지: 대혼돈의 멀티버스	저는 이글을 2년뒤인 2021년에 볼 예정입니다	vita****	₩n2021년이면 내가 18살이구만.. 고2.. 과연그때까지 살아있을까... 제 3...
1	닥터 스트레인지: 대혼돈의 멀티버스	미래에서 왔습니다	idso****	₩n닥터 스트레인지는 타임스톤에 마법을 걸어 두었습니다.타노스에게 넘겨주기 직전 타...
2	닥터 스트레인지: 대혼돈의 멀티버스	탈모 스트레인지 2	s220****	₩r₩n₩t₩t₩t₩t이전 닥터 스트레인지 1편을 보신 분들은 아실 겁니다. 마법사...
3	닥터 스트레인지: 대혼돈의 멀티버스	(스포 有) 닥스2 캐릭터 아주 자세한 고찰: 닥터 스트레인지: 대혼돈의 멀티버스(...	aasd****	₩n드디어,,마침내,,닥터 스트레인지 2가 개봉했습니다.이 글은 닥터스트레인지2, ...
4	닥터 스트레인지: 대혼돈의 멀티버스	[초대장] 마블 영화 네이버 팬카페	mc_g****	₩n마블 시네마틱 네이버 카페본 카페는 MCU의 세계관을 다룬, 마블 스튜디오가 직...
5	닥터 스트레인지: 대혼돈의 멀티버스	[영화감상] 닥터 스트레인지: 대혼돈의 멀티버스 (Doctor Strange in ...	sakg****	₩n5월 4일 개봉하는 영화 '닥터 스트레인지: 대혼돈의 멀티버스'입니다.'마블'시...
6	닥터 스트레인지: 대혼돈의 멀티버스	개봉 언제죠... 개봉 하면 당장 달려갑니다...	hmin****	₩r₩n₩t₩t₩t₩t닥터스트레인지1 재밌었는데 2도 엄청 기대중 ㅠㅠㅠㅠㅠㅠㅠㅠㅠㅠ...
7	닥터 스트레인지: 대혼돈의 멀티버스	[영화 간단 리뷰] 닥터 스트레인지: 대혼돈의 멀티버스 (2022)	choj****	₩n오늘은 <닥터 스트레인지: 대혼돈의 멀티버스>를 봤습니다.2016년에 개봉한 <...
8	닥터 스트레인지: 대혼돈의 멀티버스	2년뒤에 (인워 스포일러 솔솔(개 소리	char****	₩n₩r₩n(스포일러있음)[세상에.. 일단 타노스가 손가락튕겨서 일단 범점 10...
9	닥터 스트레인지: 대혼돈의 멀티버스	닥터 스트레인지2 대혼돈의 멀티버스 마블영화 쿠키영상 스포일러 후기	ptj0****	₩n ₩n₩n₩n닥터 스트레인지: 대혼돈의 멀티버스₩n₩n감독₩n샘 레이미₩n출연₩...
10	토르: 러브 앤 썬더	형~ 채식 잘 하고 있죠?	choo****	₩n지난번에 형네 엄마가 그랬잖아요 ..채식하라구 ..맥주 끊기는 어렵겠지만, 형 ...
11	토르: 러브 앤 썬더	2년뒤 나에게 쓰는 마블을 좋아하는 한 고2의 미래편지(과거에서 왔습돠!!)	jioa****	₩r₩n₩t₩t₩t₩t현재는 2019년 8월 23일 금요일 나는 학교를 마치고 자습...
12	토르: 러브 앤 썬더	<토르: 러브 앤 썬더> 역대급 애매함	jime****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 헴...
13	토르: 러브 앤 썬더	토르: 러브 앤 썬더 (동화스런 신성모독을 위한 자유도 벌런스 붕괴 외전)-평점 4점	reno****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 헴...
14	토르: 러브 앤 썬더	토르형~ 드디어 결혼 하는거야?	get1****	₩n원래 남녀가 헤어지고 만나고 헤어지고 만나고 그러는거니까이해하는거고 미리 축하해...
15	토르: 러브 앤 썬더	브링미 따노수우ㅜㅜㅜㅜㅜ	rhkd****	₩r₩n₩t₩t₩t₩r₩n성지순례하고간다아ㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏㅏ...
16	토르: 러브 앤 썬더	[영화감상] 토르: 러브 앤 썬더 (Thor: Love and Thunder, 2022)	sakg****	₩n 7월 6일 개봉하는 영화 '토르 : 러브 앤 썬더'입니다. '토르'시리즈의 네...
17	토르: 러브 앤 썬더	토르 러브 앤 썬더 and Marvel Studios' Thor: Love and ...	kjgh****	₩n₩n₩n₩n토르: 러브 앤 썬더₩n₩n감독₩n타이카 와이티티₩n출연₩n크리스 헴...
18	토르: 러브 앤 썬더	후후 현재 고3	chan****	₩n₩r₩n현재 수시 원서 접수를 마친 고3이지.. 어벤져스 엔드게임 또 보며 눈물...
19	토르: 러브 앤 썬더	<토르: 러브 앤 썬더> 후기 스포없음 재밌게 시간순삭	maro****	₩n정말이지 얼마만의 극장 나들이던가!!! 혼자 시사회를 가거나 했던게 전부였는데 ...
20	블랙 팬서: 와칸다 포에버	[블랙팬서2] 굿바이 故채드윅 보스만, " 블랙팬서 : 와칸다 포에버 " (+스포,...	mapa****	₩n안녕하세요 신비에용.오늘 소개할 영화는 블랙팬서 2번째 시리즈," 블랙팬서 : ...
21	블랙 팬서: 와칸다 포에버	[블랙팬서:와칸다 포에버]<예매권>(67번째) 조연들만 가득모여 블랙팬서를 그리워만...	acts****	₩n20221109의정부 CGV IMAX-E-12VIP 쿠폰2.5/5 "와칸다를 지...
22	블랙 팬서: 와칸다 포에버	[블랙 팬서: 와칸다 포에버]를 보고	film****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
23	블랙 팬서: 와칸다 포에버	[블랙팬서: 와칸다 포에버] 티저 예고편	kjgh****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
24	블랙 팬서: 와칸다 포에버	Reflect on the cultural impact and legacy of B...	kjgh****	₩nMarvel Studios @MarvelStudiosReflect on the ...
25	블랙 팬서: 와칸다 포에버	Marvel Future Fight @Marvel_FFightUS · Join ...	kjgh****	₩n#Marvel Future Fight @Marvel_FFightUSJoin M'...
26	블랙 팬서: 와칸다 포에버	[스포, 영화리뷰] 블랙팬서 : 와칸다 포에버	gkal****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
27	블랙 팬서: 와칸다 포에버	[CGV 전주 호자] 블랙 팬서: 와칸다 포에버 아이맥스 - 관람후기	jihu****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
28	블랙 팬서: 와칸다 포에버	<영화> 블랙 팬서: 와칸다 포에버 2022.11.22	elle****	₩n₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티티...
29	블랙 팬서: 와칸다 포에버	[블랙 팬서: 와칸다 포에버] ★★☆ 주제와 의미는 좋지만, 히어로 영화로서 실격이다.	zxc1****	₩n ₩n₩n₩n블랙 팬서: 와칸다 포에버₩n₩n감독₩n라이언 쿠글러₩n출연₩n레티...

- csv 파일로 저장

```
# csv 파일로 저장
df_data.to_csv("/content/drive/MyDrive/data/marvel_2022_review.csv")
```

▼ 형태소 분석

- 네이버 영화 리뷰 데이터 다운로드
  - 출처: https://github.com/e9t/nsmc

```
# 다운로드 받을 폴더를 준비
DATA_DIR = "./data"
os.makedirs(DATA_DIR, exist_ok = True)

# 데이터 다운로드
torchtext.utils.download_from_url(url='https://raw.githubusercontent.com/e9t/nsmc/master/ratings_train.txt',path=os.path.join(DATA_DIR,

# txt 파일을 판다스 데이터프레임으로 읽어오기
data = pd.read_csv(os.path.join(DATA_DIR, 'review.txt'), sep='\t')

data.head()
```

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밋었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1

- KoNLPy 한글 형태소 분석

```
# 긍정 리뷰를 하나 선택 -> label 1
sample_text = data['document'].iloc[102629]
sample_text

# 트위터 형태소 분석기(Okt)를 활용
okt = Okt()

tokens_n = okt.nouns(sample_text) # 명사 추출
pprint(tokens_n)

tokens_v = [x for (x, y) in okt.pos(sample_text) if y == 'Verb'] # 동사 추출
pprint(tokens_v)
```

```
['초딩',
 '때',
 '기억',
 '다시',
 '만난',
 '유지인',
 '이덕화',
 '품',
 '대사',
 '영화',
 '제목',
 '집',
 '와이프',
 '이야기',
 '가끔',
 '멘트',
 '목소리',
 '네',
 '해',
 '나',
 '달이']
['봤던', '보내면서', '하는', '하다가', '써', '먹는', '하구요', '되면', '되는구나']
```

- 텍스트 전처리
  - 두 글자 이하인 경우 불필요한 단어(명사, 동사)가 많으므로 분석 대상에서 제외

```
# 누락 데이터를 제거
review_data = data['document'].dropna().values

# 1500개의 샘플을 선택하여 추출
review_data = review_data[:1500]

# 두 글자 단어 제외
cleaned_review_data = []

for review in tqdm(review_data):
    tokens_n = okt.nouns(review) # 명사
    tokens_v = [x for (x, y) in okt.pos(review) if y == 'Verb'] # 동사
    tokens = tokens_n + tokens_v
    cleaned_tokens = []
    for word in tokens:
        if len(word) >= 3:
            cleaned_tokens.append(word)
        else :
```

```
            pass
    cleaned_review = " ".join(cleaned_tokens)
    cleaned_review_data.append(cleaned_review)

print(len(cleaned_review_data))
print(cleaned_review_data[757]) # 758번째 데이터 추출
```

```
100%|██████████| 1500/1500 [00:10<00:00, 145.40it/s]1500
다니엘헤니 다니엘헤니 설경구
```

▼ 벡터화 표현

```
# TF-IDF 변환기 객체를 생성
tfid = TfidfVectorizer()

# TF-IDF 변환기에 데이터를 입력하여 변환
review_tfid = tfid.fit_transform(cleaned_review_data)

# 배열의 크기
print(review_tfid.shape)

# 첫 번째 데이터
print(review_tfid[757])
```

```
(1500, 1916)
  (0, 972)      0.4472135954999579
  (0, 333)      0.8944271909999159
```

```
# 단어 사전 확인 (딕셔너리 형태)
vocab = tfid.vocabulary_

# 단어 사전의 크기
print(len(vocab))

# 단어 사전 출력(앞에서 5개의 단어만 출력)
print({k:v for i, (k,v) in enumerate(vocab.items()) if i < 5})

# 단어들의 사전 인덱스를 이용하여 원래 단어를 검색하는 매핑 딕셔너리
index_to_word = { v:k for k, v in vocab.items()}

# 앞에서 5개의 단어를 출력
print({k:v for i, (k,v) in enumerate(index_to_word.items()) if i<5})
```

```
1916
{'목소리': 648, '포스터': 1714, '않구나': 1116, '보는것을': 780, '교도소': 93}
{648: '목소리', 1714: '포스터', 1116: '않구나', 780: '보는것을', 93: '교도소'}
```

```
# 758번째 리뷰를 구성하는 단어들의 사전 인덱스를 이용하여 원래 단어를 복원
original_text = " ".join([index_to_word[word_idx] for word_idx in review_tfid[757].indices])
print(original_text)
```

```
설경구 다니엘헤니
```

▼ 토픽 모델링

- LDA

```
# LDA 모델링 객체를 생성 (토픽 갯수를 2로 설정 : 긍정 / 부정)
lda1 = LatentDirichletAllocation(n_components = 2)
```

```
# LDA 모델링 객체를 생성 (토픽 갯수를 3로 설정)
lda2 = LatentDirichletAllocation(n_components = 3)

# TF-IDF 벡터를 입력하여 모델 학습
print(lda1.fit(review_tfid))
print(lda2.fit(review_tfid))

# 토픽 모델링 결과를 담고 있는 배열의 형태 : (2개의 토픽, 1916개의 단어)
print(lda1.components_.shape)

# 토픽 모델링 결과를 담고 있는 배열의 형태 : (3개의 토픽, 1916개의 단어)
print(lda2.components_.shape)
```

```
LatentDirichletAllocation(n_components=2)
LatentDirichletAllocation(n_components=3)
(2, 1916)
(3, 1916)
```

```
# 1916개의 단어중에서, 토픽 별로 가장 중요도가 높은 단어를 10개씩 출력(토픽 2개)
for idx, topic in enumerate(lda1.components_):
    print(f"토픽 유형 {idx+1}:", [(index_to_word[i], topic[i].round(3)) for i in topic.argsort()[:-11:-1]])
```

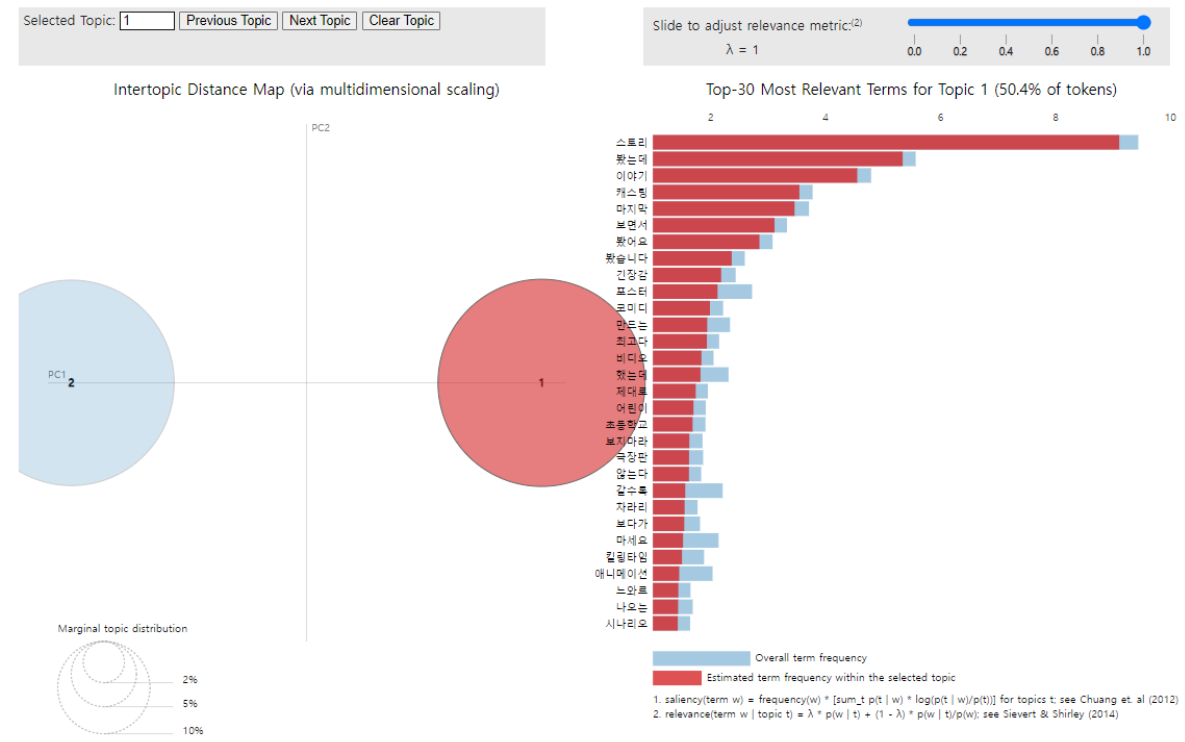
```
토픽 유형 1: [('드라마', 23.295), ('쓰레기', 19.063), ('주인공', 12.801), ('시리즈', 8.673), ('한국영', 7.056), ('합니다', 6.761), ('모르겠다', 5.754
토픽 유형 2: [('스토리', 21.95), ('봤는데', 12.774), ('이야기', 10.849), ('캐스팅', 8.398), ('마지막', 8.193), ('보면서', 7.344), ('봤어요', 6.706),
```

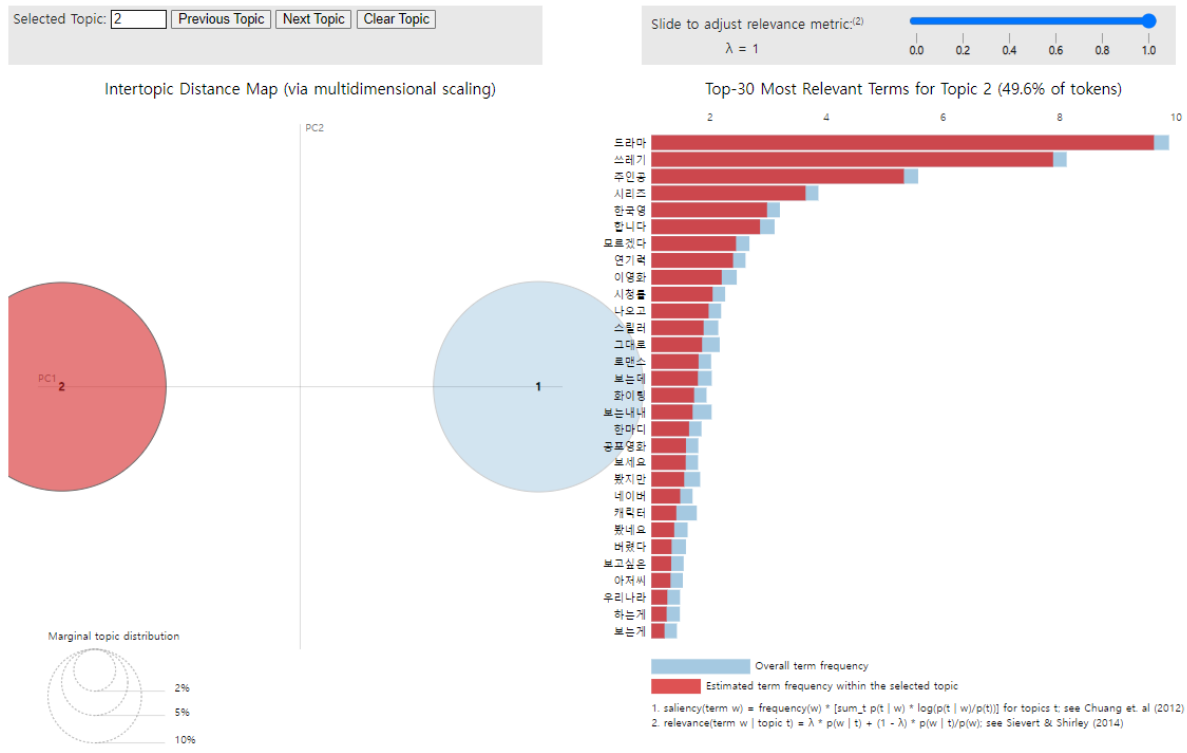
```
# 1916개의 단어중에서, 토픽 별로 가장 중요도가 높은 단어를 10개씩 출력(토픽 3개)
for idx, topic in enumerate(lda2.components_):
    print(f"토픽 유형 {idx+1}:", [(index_to_word[i], topic[i].round(3)) for i in topic.argsort()[:-11:-1]])
```

```
토픽 유형 1: [('스토리', 21.411), ('쓰레기', 18.93), ('한국영', 6.917), ('합니다', 6.365), ('포스터', 5.722), ('모르겠다', 5.56), ('시청률', 4.468),
토픽 유형 2: [('드라마', 22.53), ('봤는데', 9.278), ('주인공', 8.79), ('시리즈', 8.509), ('캐스팅', 8.284), ('보면서', 7.177), ('이야기', 6.704), ('
토픽 유형 3: [('봤어요', 6.567), ('마지막', 5.253), ('긴장감', 5.005), ('그대로', 4.395), ('나오고', 4.374), ('최고다', 4.329), ('스릴러', 4.321), (
```

- pyLDAvis 시각화

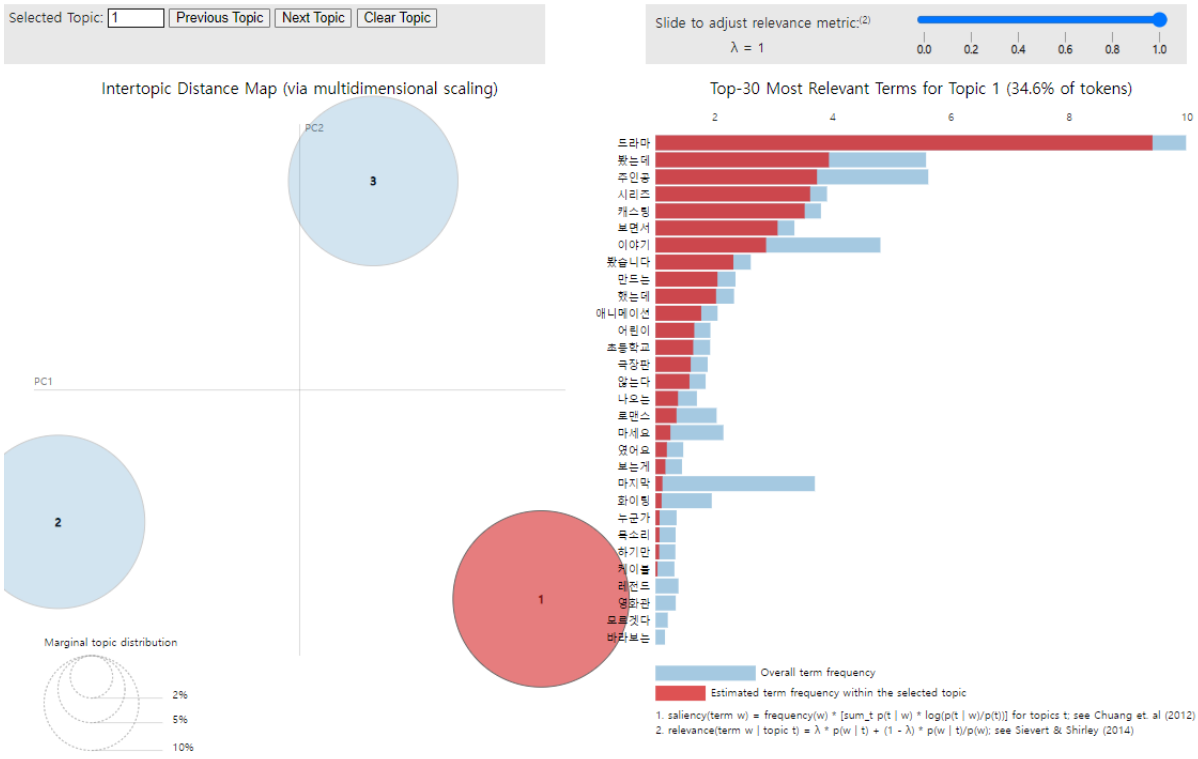
```
# LDA 토픽 모델링 결과를 시각화
# 토픽 2개
pyLDAvis.enable_notebook()
visualization = pyLDAvis.sklearn.prepare(lda1, review_tfid, tfid)
pyLDAvis.display(visualization)
```

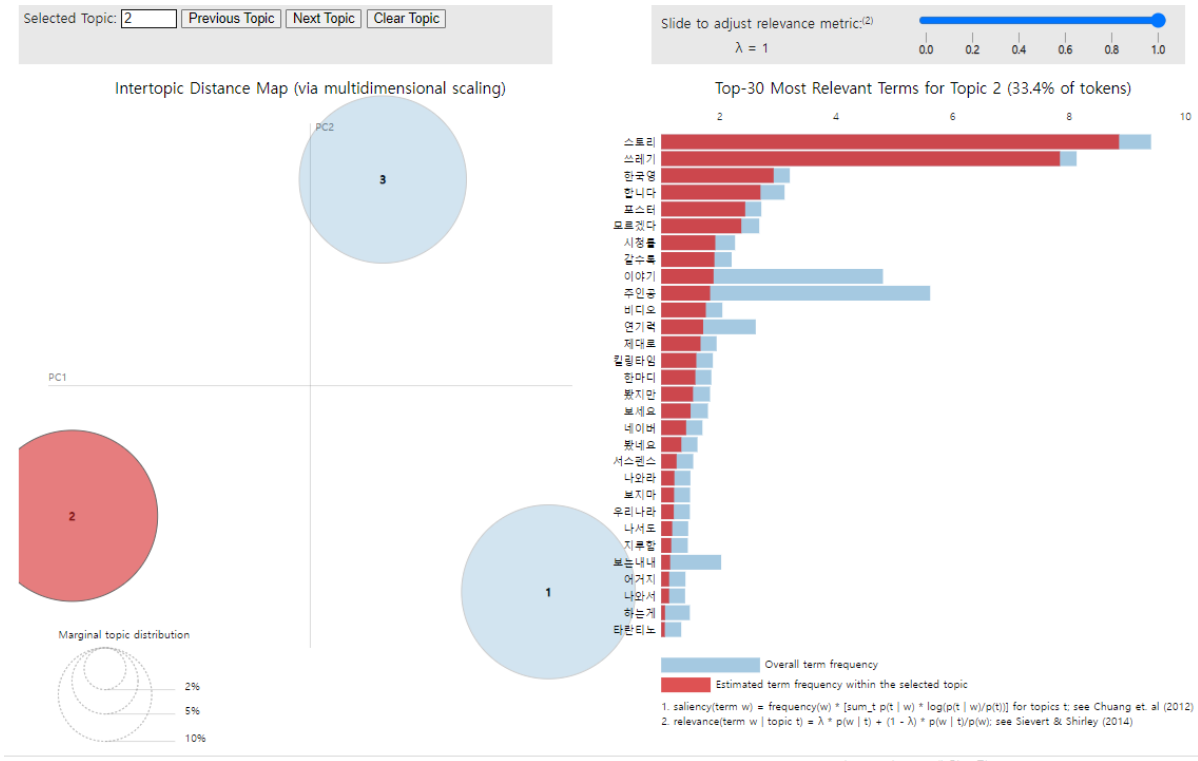


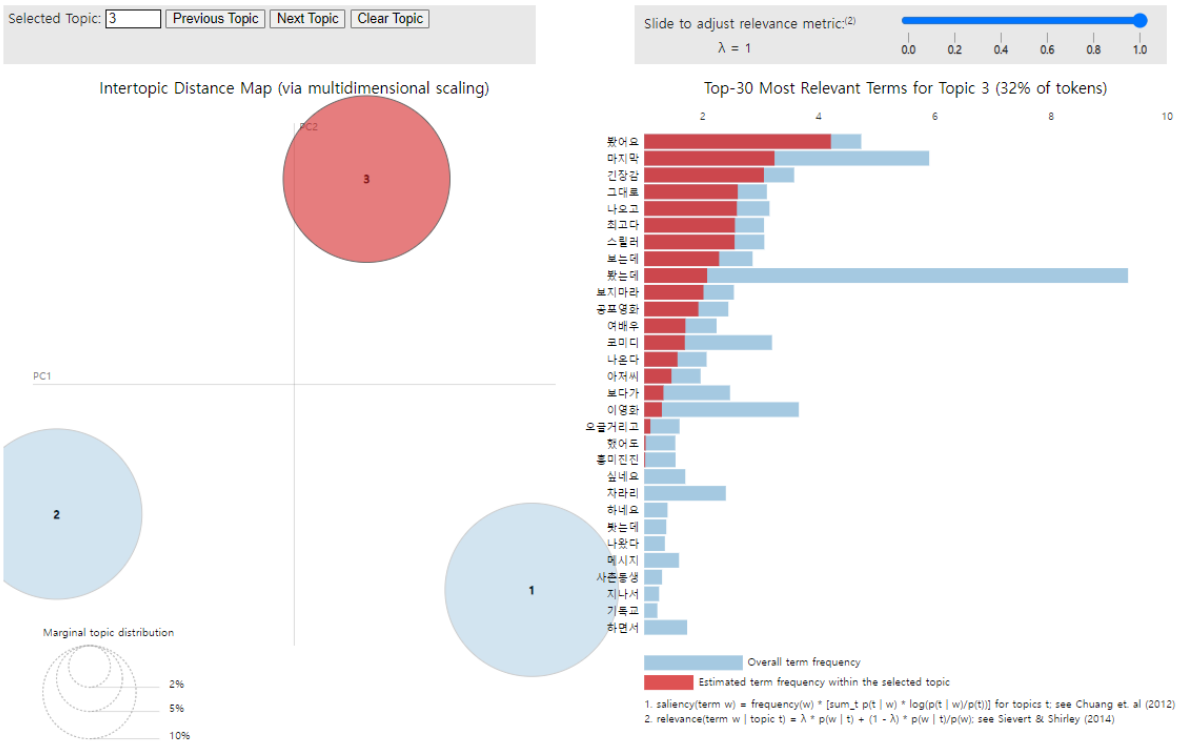



```
# 토픽 3개
pyLDAvis.enable_notebook()
```

```
visualization = pyLDAvis.sklearn.prepare(lda2, review_tfid, tfid)
pyLDAvis.display(visualization)
```







▼ 감성 분석

- 머신러닝 알고리즘을 활용하여, 긍정/부정 감성을 분류

```
labels = data['label'].iloc[:1500].values

# 로지스틱 분류 모델링 객체를 생성
lr = LogisticRegression()

# TF-IDF 벡터를 입력하여 모델 학습
lr.fit(review_tfid, labels)

# 첫 번째 샘플을 이용하여 모델 예측
pred = lr.predict(review_tfid[0])
print(pred)
```

[0]

- 2022 마블 영화 리뷰를 긍정, 부정으로 감성 분류 예측

```
test = pd.read_csv('/content/drive/MyDrive/data/marvel_2022_review.csv')
test['emotion'] = 0
for i in range(len(test)):
    test_sample = test['review'][i]

    # 세 글자 이상의 명사만을 추출
    tokens_n = okt.nouns(test_sample)
    tokens_v = [x for (x, y) in okt.pos(test_sample) if y == 'Verb'] # 동사
    tokens = tokens_n + tokens_v
    cleaned_tokens = []
    for word in tokens:
        if len(word) >= 3:
            cleaned_tokens.append(word)
        else:
            pass
```

```
    cleaned_review = " ".join(cleaned_tokens)

    # TF-IDF 변환기에 데이터를 입력하여 변환
    test_review_tfid = tfid.transform([cleaned_review])

    # 로지스틱 회귀 모델을 활용하여 분류 예측
    test_pred = lr.predict(test_review_tfid)[0]
    print("분석 결과 {}적인 리뷰로 예측됩니다.".format("긍정" if test_pred > 0 else "부정"))
    test['emotion'].iloc[i] = test_pred
```

```
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
분석 결과 긍정적인 리뷰로 예측됩니다.
분석 결과 부정적인 리뷰로 예측됩니다.
```

```
test.head()
```

	Unnamed: 0	movie	title	user	review	emotion
0	0	닥터 스트레인지: 대혼돈의 멀티버스	저는 이글을 2년뒤인 2021년에 볼 예정입니다	vita****	₩n2021년이면 내가 18살이구만.. 고2.. 과연그때까지 살아있을까... 제 3...	0
1	1	닥터 스트레인지: 대혼돈의 멀티버스	미래에서 왔습니다	idso****	₩n닥터 스트레인지는 타임스톤에 마법을 걸어 두었습니다.타노스에게 넘겨주기 직전 타...	1
2	2	닥터 스트레인지: 대혼돈의 멀티버스	탈모 스트레인지 2	s220****	₩r₩n₩t₩t₩t₩t이전 닥터 스트레인지 1편을 보신 분들은 아실 겁니다. 마법사...	1
3	3	닥터 스트레인지: 대혼돈의 멀티버스	(스포 有) 닥스2 캐릭터 아주 자세한 고찰: 닥터 스트레인지: 대혼돈의 멀티버스(...	aasd****	₩n드디어,,마침내,,닥터 스트레인지 2가 개봉했습니다.이 글은 닥터스트레인지2, ...	0
4	4	닥터 스트레인지: 대혼돈의 멀티버스	[조대장] 마블 영화 네이버 팬카페	mc_g****	₩n마블 시네마틱 네이버 카페본 카페는 MCU의 세계관을 다룬, 마블 스튜디오가 직...	0

```
test.groupby('movie')['emotion'].sum()
```

```
movie
닥터 스트레인지: 대혼돈의 멀티버스    4
블랙 팬서: 와칸다 포에버         5
토르: 러브 앤 썬더            6
Name: emotion, dtype: int64
```

- 30개의 리뷰 중에서 닥터 스트레인지: 대혼돈의 멀티버스는 6개, 블랙 팬서 : 와칸다 포에버는 5개, 토르 : 러브 앤 썬더는 4개의 부정적인 리뷰가 있었다.

분석을 마치며

- 생각보다 부정적인 리뷰가 압도적으로 많지는 않았다.

    - 영화별로 10개씩 추출하여 분석을 했기 때문에 확실하지 않다. 추후에 더 많은 리뷰를 추출하여 분석할 필요가 있다.
- github에서 제공하는 데이터가 아닌 마블 영화 데이터를 모아 분석을 해야 할 것 같다.
    - 마블 영화 데이터를 기반으로 한 것이 아니다 보니 정확한 분류가 되지 않은 것 같다.
- 사람들이 자신들의 의견을 적는 글이다 보니 사전에 없는 단어들이 많았다.
    - 분석에 방해가 되는 단어 요소들을 확실히 제거한 후 분석할 필요가 있다.