

SEVEN ELEVEN

세븐 일레븐



영

화

분

식

유진아, 김나연, 김동수, 안은지, 안태용

START



01 플젝 소개

02 분석 배경

03 분석 과정

04 분석 결과

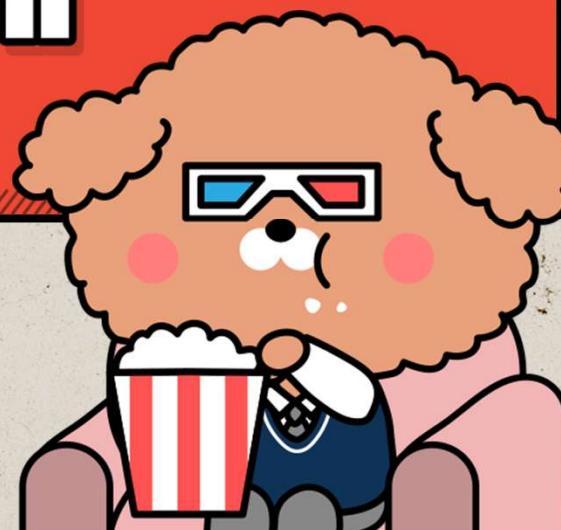
05 활용 방안

06 질의 응답

Contents

part.
01

주제 및 팀원 소개



Part.

01

팀원 소개



유진아



김나연



안태용



안은지



김동수



주제 소개

명절 인기 영화
줄거리 분석을 통한
영화 소재 트렌드 파악



Part.
02

분석 배경

주제 선정 과정



후보



뉴스 데이터 분석



영화 데이터 분석

Part.
03

분석 과정

데이터 수집 방법

크롤링, DataFrame 생성,
데이터 전처리, 텍스트 전처리



Part.
03-1

크롤링

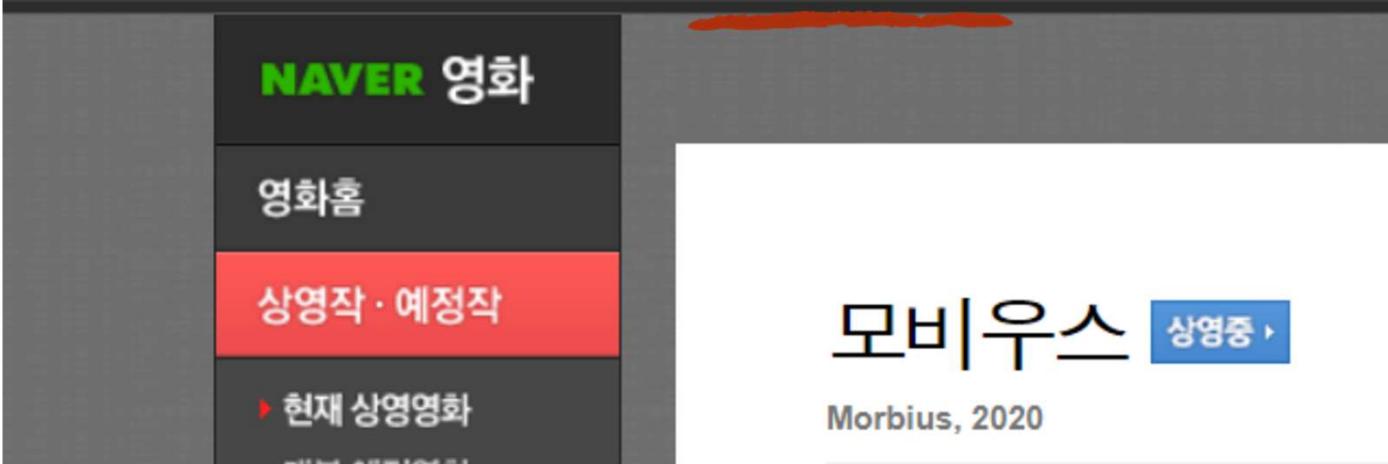


Part. 03-1

Part. 03-1

데이터 전처리 (DataFrame으로 변경) - 영화 코드

movie.naver.com/movie/bi/mi/basic.naver?code=190695#



URL ? 뒤 code=190695# 형태의 쿼리스트링
변수 code와 값 190695

Part. 03-1

```
1 from bs4 import BeautifulSoup  
2 from urllib.request import urlopen  
3 import pandas as pd  
4 import re
```

```
1 def clean_text(inputString):  
2     text_rmv = re.sub('[-+=#/₩?:^_.@*₩"※~+!„‘|₩₩₩₩₩₩`₩'…»₩₩₩₩₩₩-]', '', inputString)  
3     return text_rmv
```

- 필요한 라이브러리 불러옴
 - 기호 제거를 위한 함수 → 정규표현식

Part. 03-1

크롤링 준비

```
def movieinfo(): # 함수화
    movie_code_list = []
    url_list = []

    # (설, 추석) 박스오피스 상위 5개의 영화 코드 입력하기
    for i in range(5):
        movie_code = input("영화 코드 입력: ")
        movie_code_list.append(movie_code) # 입력한 영화 코드를 append
        url_list.append('https://movie.naver.com/movie/bi/mi/basic.naver?code=' + movie_code_list[i]) # 영화 정보 페이지 URL append

    # DataFrame의 컬럼이 될 리스트 생성
    title = [] #영화제목
    plot = [] #줄거리
    date = [] #개봉일
    holiday = [] #영절(설날/추석)
    genre = [] #영화 장르
    nation = [] #제작 국가

    for url in url_list:
        web = urlopen(url)
        web_page = BeautifulSoup(web, 'html.parser')
```

함수 선언, 크롤링 준비
DataFrame의 Column0이 될 list를 생성

Part. 03-1

데이터 전처리 (DataFrame으로 변경) - 크롤링

#영화 제목 가져오기

```
title_temp = web_page.find('h3', {'class': 'h_movie'}).find('a') # 제목 크롤링
title1 = title_temp.get_text() # 제목 중 텍스트 추출
title.append(title1) # 추출 후 리스트에 append
```

#영화 줄거리 가져오기

```
movie1 = web_page.find('p', {'class': 'con_tx'}) # 줄거리 크롤링
movie1_text = movie1.get_text() # 줄거리 중 텍스트 추출
movie2_text = movie1_text.replace("\n", "").replace("\xa0", "") # 이스케이프 시퀀스(\n,\n..), 유니코드(\xa0) 제거
movie3_text = clean_text(movie2_text) # 제거 후 리스트에 append
plot.append(movie3_text)
```

#영화 개봉일 가져오기

```
#년도 가져오기
date_origin = web_page.select('p span')[3] # 개봉일자 크롤링
date_temp = date_origin.get_text() # 개봉일자 중 텍스트 추출
date_temp1 = date_temp.replace("\t", "").replace("\n", "").replace("\r", "").replace("개봉", "").replace(" ", "") # '개봉', 공백, 이스케이프 시퀀스, 유니코드 제거
date1 = date_temp1.replace(".", "") # 제거 후 리스트에 append
date.append(date1)
```

#영화 장르 가져오기

```
genre_origin = web_page.find('dl', {'class': 'info_spec'}).find('span') # 장르 크롤링
new_genre = genre_origin.get_text() # 장르 중 텍스트 추출
genre1 = new_genre.replace("\t", "").replace("\n", "").replace("\r", "") # 이스케이프 시퀀스 제거
genre.append(genre1) # 제거 후 리스트에 append
```

#영화 국가정보 가져오기

```
nation_origin = web_page.select('p span')[1] # 제작 국가 크롤링
nation_temp = nation_origin.get_text() # 금어온 국가정보에서 텍스트만 추출
nation1 = nation_temp.replace("\t", "").replace("\n", "").replace("\r", "") # 이스케이프 시퀀스 제거
nation.append(nation1) # 제거 후 리스트에 append
```

```
# 제목 크롤링
# 제목 중 텍스트 추출
# 추출 후 리스트에 append
```

```
# 줄거리 크롤링
# 줄거리 중 텍스트 추출
# 이스케이프 시퀀스(\t,\n..), 유니코드(\xa0) 제거
# 제거 후 리스트에 append
```

```
# 개봉일자 크롤링
# 개봉일자 중 텍스트 추출
'\r', "").replace("개봉", "").replace(" ", "") # '개봉', 공백, 이스케이프 시퀀스, 유니코드 제거
# 제거 후 리스트에 append
```

```
) # 장르 크롤링
# 장르 중 텍스트 추출
") # 이스케이프 시퀀스 제거
# 제거 후 리스트에 append
```

제목, 줄거리, 개봉일, 장르, 제작국가
반복문 내에서 크롤링 및 텍스트만 남기고 리스트 append

Part.
03-2

DataFrame 생성



Part. 03-2

데이터 전처리 (DataFrame으로 변경) - DataFrame 생성

```
#데이터 프레임 형태로 만들기
global movie
movie = pd.DataFrame({'title': title, 'plot' : plot, 'genre': genre, 'nation': nation, 'date' : date})
# 날짜, 시간 데이터 형식을 datetime64로 변환
movie['date']=pd.to_datetime(movie['date'])
# strftime: 날짜와 시간을 문자열로 변경
movie['year'] = movie['date'].dt.strftime('%Y')
# %Y : 년도를 4자리 문자열로 추출 후 'year' column에 추가
movie['month'] = movie['date'].dt.strftime('%m')
# %m : 월을 2자리 문자열로 추출 후 'month' column에 추가

#엑셀로 저장하기
name = input("저장할 이름을 작성하세요(년도,명절구분) ex2009설날: ") # 엑셀 파일의 이름 사용자 지정
movie.to_excel(excel_writer = '{}.xlsx'.format(name), index = False) # index를 제외하고 엑셀(.xlsx) 파일로 저장
```

제목, 줄거리, 장르, 국가, 개봉일, 개봉일(연도), 개봉일(월)의
Column을 가진 DataFrame 생성 및 excel 저장

Part. 03-2

데이터 전처리 (DataFrame으로 변경) - DataFrame 확인

```
1 movie.info()
```

```
영화 코드 입력: 65669  
영화 코드 입력: 68063  
영화 코드 입력: 51306  
영화 코드 입력: 51143  
영화 코드 입력: 68952
```

저장할 이름을 작성하세요(년도,명절구분) ex2009설날: 키획

```
1 #출력 잘 됐나 확인  
2 movie
```

	title	plot	genre	nation	date	year	month
0	적벽대전 1부 - 거대한 전쟁의 시작	위 측 오 3국이 대립하던 서기 208년 중국 천하통일을 위해 중국대륙을 피로 물들...	전쟁, 액션, 모험, 드라마	중국	2008-07-10	2008	07
1	작전명 발키리	강직한 성품의 클라우스 폰 슈타펜버그 대령은 조국과 국민을 위하는 충성스런 장교이자...	스릴러, 드라마, 전쟁	미국, 독일	2009-01-22	2009	01
2	유감스러운 도시	강력계 근성이 숨쉬고 있는 교통 경찰 장충동 외부에 얼굴이 알려지지 않았다는 이유로...	액션, 범죄, 코미디	한국	2009-01-22	2009	01
3	과속스캔들	한때 아이돌 스타로 10대 소녀 팬들의 영원한 우상이었던 남현수차태현지금은 서른 중...	코미디	한국	2008-12-03	2008	12
4	잉크하트	소리 내어 읽으면 책 속의 인물을 현실 세계로 불러낼 수 있는 신비한 능력을 가진 ...	판타지, 모험	미국, 독일, 영국	2009-01-29	2009	01

Part.
03-3

결측값/값변경



Part. 03-3

결측값(NaN) 및 DF 수정 작업

```
df = pd.read_excel('movie.xlsx')
```

```
df = df.drop(['Unnamed: 0'], axis=1)
```

Unnamed: 0		title	plot
0	0	작벽대전 1부 - 거대한 전쟁의 시작	위 측 오 3국이 대립하던 서기 208년 중국고향 톈일을 위해 중국 대륙을 피로 물들...
1	1	작전명 발키리	강직한 성격의 클라우스 폰 슈타인하그 대령은 조국과 국민을 지향하는 충성스런 장교이지...
2	2	유감스러운 도시	강력계 노년이 숨쉬고 있는 교통 경찰 장충동 외부에 얼굴이 알려지지 않았다는 이유로...
3	3	기소스캔들	한때 아이돌 스타로 1대 소녀 팬들의 영원한 우상이었던 남현수차태현지금은 서른 중...
4	4	잉크하트	소리 내어 읽으면 책 속의 인물을 현실 세계로 불러낼 수 있는 신비한 능력을 가진 ...

Part. 03-3

결측값(NaN) 및 DF 수정 작업

	title	plot	genre	nation	date	year
0	적벽대전 2부 - 최후의 결전	유비의 책사 제갈량은 손권과의 동맹에 극적으로 성공하고 손권 휘하의 명장 주 유와 함...	전쟁, 드라마	중국	2009-01-22	2009
1	작전명 발기리	강직한 성품의 클라우스 폰 슈타펜버그 대령은 조국과 국민을 위하는 충성스런 장교이자...	스릴러, 드라마, 전쟁	미국, 독일	2009-01-02	2009
2	유감스러운 도시	강력계 근성이 숨쉬고 있는 교통 경찰 장충동 외부에 얼굴이 알려지지 않았다는 이유로...	액션, 범죄, 코미디	한국	2009-01-01	2009
3	과속스캔들	한때 아이돌 스타로 10대 소녀 팬들의 영원한 우상이었던 남현수차태현지금은 시는 중...	극	국	2008-12-03	2008
4	잉크하트	소리 내어 읽으면 책 속의 인물을 현실 세계로 불러낼 수 있는 신비한 능력을 가진 ...	판타지, 도정	미국, 독일, 영국	2009-01-29	2009
...				
95	안시성	우리는 물러서는 법을 배우지 못했다 우리는 무를 끓는 법을 배우지 못했다 우리는 항...	액션	한국		
96	명당	땅의 기운을 점쳐 이간의 운명을 바꿀 수 있는 천재지과 박재상조수우은연다을 이용해 ...	Nan	한국		
97	협상	어떠한 상황에서도 냉철함을 잊지 않던 최고의 협상가 하채윤은 긴급 투입된 현장에서 인...	범죄	한국		

#데이터 전처리

```
df.at[96, 'genre'] = '사극, 드라마'
```

```
df.at[3, 'year'] = 2009
```

Part. 03-3

결측값(NaN) 및 DF 수정 작업

	title	plot	genre	nation	date	year	month
0	적벽대전 2부 - 최후의 결전	유비의 책사 제갈량은 손권과의 동맹에 극적으로 성공하고 손권 휘하의 명장 주유와 함...	전쟁, 액션	중국	2009-01-22	2009	1
1	작전명 발기리	강직한 성품의 클라우스 폰 슈타펜버그 대령은 조국과 국민을 위하는 충성스런 장교이지...	스릴러, 드라마, 전쟁	미국, 독일	2009-01-22	2009	1
2	유감스러운 도시	강력계 근성이 숨쉬고 있는 교통 경찰 장충동 외부에 얼굴이 알려지지 않았다는 이유로...	액션, 범죄, 코미디	한국	2009-01-22	2009	1
3	과속스캔들	한때 아이돌 스타로 10대 소녀 팬들의 영원한 우상이었던 남현수자태현지금은 서른 중...	코미디	한국	2008-12-03	2009	12 21
4	잉크하트	소리 내어 읽으면 책 속의 인물을 현실 세계로 불러낼 수 있는 신비한 능력을 가진 ...	판타지, 모험	미국, 독일, 영국	2009-01-29	2009	1
...
95	안시성	우리는 물러서는 법을 배우지 못했다 우리는 무릎 끓는 법을 배우지 못했다 우리는 항...	액션	한국	2018-09-19	2018	9
96	명당	땅의 기운을 점쳐 인간의 운명을 바꿀 수 있는 천재지관 박재상조승우은명당을 이용해 ...	사극, 드라마	한국	2018-09-19	2018	9
97	협상	어떠한 상황에서도 냉철함을 잊지 않던 최고의 협상가 하채윤은 긴급 투입된 현장에서 인...	범죄	한국	2018-09-19	2018	9
98	더 네	루마니아의 젊은 수녀가 자살하는 사건을 의뢰 받아 바티칸에서 파견된 버크 신부와 아...	공포, 미스터리, 스릴러	미국	2018-09-19	2018	9
99	원더풀 고스트	딸 앞에선 바보지만 남의 일에는 1도 관심 없는 유도 관장 장수마동석에게 의욕과다 ...	코미디, 드라마, 범죄	한국	2018-09-26	2018	9

100 rows x 8 columns

Part.
03-4

텍스트 전처리



Part.
03-4

불용어 추가

Korean Stopwords

아	어찌됐든	하기보다는
휴	그위에	차라리
아이구	게다가	하는 편이 낫다
아이쿠	점에서 보아	흐흐

RANKS NL의 Korean Stopwords 활용

Part.
03-4

불용어 추가

```
lst = cleaned_word_list_result[3]
word_dic = {}
for word in lst:
    if word not in word_dic:
        word_dic[word] = 1 # changed from "0" to "1"
    else:
        word_dic[word] += 1
word_dic
```

```
{'공률': 1,
'도전': 1,
'시라노': 1,
'에이전시': 3,
'연애': 5,
...}
```

연도별 줄거리 확인 후 불용어 추가

Part. 03-4

불용어 정리

2017-2018 불용어

In [25]: #2017년

```
list_str.extend(['엠 브렐라','동판','민재','중인','장첸','타키','츠맨','윤계상','스테이','진태','골든','철 흥','박해일','곳도','2004년','조재윤','석도','마동석','태련','멀린','마크','스트롬','켄터키','줄리','무어','후
```

executed in 9ms, finished 15:17:33 2022-04-05

In [26]: #2018년

```
list_str.extend(['시빌','와칸','티찰라','채드윅','비브라늄','블랙','팬서','김민','데모','건우','강동원','동규','김대','금철','김성균','선영','한효주','조혁','조항리','코난','미란','보라','VS','5천','20만','40','조승우','장동','김씨','13년','김좌근','채윤','10일','그로','민태구','위해','12시간','버크','김영광','고스
```

executed in 10ms, finished 15:17:35 2022-04-05

Part.
03-5

토론회



Part. 03-5

코드 소개

```
1 lst_1 = list(df['plot'])
2 con_tx=[]
3 cleaned_word_list_result=[]
4 for i in range(10):
5     str_1 = str(i*2009)
6     for j in range(2):
7         str_2 = str_1+'_'+str(j+1)
8
9     print(str_2)
10
11    txt = lst_1[((i*2)+j)*5:((i*2)+j)*5+5]
12
13    con_tx.append('')
14    for k in range(5):
15        con_tx[((i*2)+j)] = str(con_tx[((i*2)+j)])+str(txt[k])
16
17    tokenizer = Okt()
18    raw_pos_tagged = tokenizer.pos(con_tx[((i*2)+j)], norm=True, stem=True)
19
20    word_cleaned = []
21    for word in raw_pos_tagged:
22        if word[1] in ["Noun"]:
23            if (len(word[0]) != 1) & (word[0] not in del_list):
24                word_cleaned.append(word[0])
25
26    cleaned_word_list_result.append(word_cleaned)
27
28
```

2009_설날, 2010_추석 등 기간 별
반복 구문 만들기

DataFrame에서 해당 기간 1~5위
줄거리를 con_tx에 저장하기

konlpy를 이용한 한글 단어 토큰화

Part. 03-5

코드 소개

```
1 lst_1 = list(df['plot'])
2 con_tx=[]
3 cleaned_word_list_result=[]
4 for i in range(10):
5     str_1 = str(i+2009)
6     for j in range(2):
7         str_2 = str_1+'_'+str(j+1)
8
9     print(str_2)
10
```

str_2의 결과 예시

2009_1
2009_2
2010_1
2010_2
2011_1
2011_2
2012_1
2012_2
2013_1
2013_2
2014_1
2014_2
2015_1
2015_2
2016_1
2016_2
2017_1
2017_2
2018_1
2018_2

활용 예시

2009_1.xlsx
2009_2.xlsx
2010_1.xlsx
2010_2.xlsx
2011_1.xlsx
2011_2.xlsx
2012_1.xlsx
2012_2.xlsx
2013_1.xlsx
2013_2.xlsx
2014_1.xlsx
2014_2.xlsx
2015_1.xlsx
2015_2.xlsx
2016_1.xlsx
2016_2.xlsx
2017_1.xlsx
2017_2.xlsx
2018_1.xlsx
2018_2.xlsx

Part.
03-5

코드 소개

```
17 tokenizer = Okt()  
18 raw_pos_tagged = tokenizer.pos(con_tx[(i*2)+j], norm=True, stem=True)  
19  
20 word_cleaned = []  
21 for word in raw_pos_tagged:  
22     if word[1] in ["Noun"]:  
23         if (len(word[0]) != 1) & (word[0] not in del_list):  
24             word_cleaned.append(word[0])  
25 cleaned_word_list_result.append(word_cleaned)  
26  
27
```



Part.
04

분석 결과



Part. ●
04

결과 도출을 위한 Word Cloud 사용

```
word_cloud = WordCloud(font_path="C:/Windows/Fonts/malgun.ttf",
                       width=2000, height=1000,
                       max_words=100,
                       background_color='white')
word_cloud.generate_from_frequencies(word_dic)

plt.figure(figsize=(15,15))
plt.imshow(word_cloud)
plt.axis("off")
plt.tight_layout(pad=0)
plt.show()
```

Part. 04

2009~2010년 추석/설날 영화 키워드



실종, 특수, 조직, 수사, 살해, 살인

범죄 / 액션 / 스릴러

아내, 엄마, 사랑, 인생, 인류 등

가족 / 드라마 / 코미디
/ 멜로/ 로맨스

Part. ●
04

2011~2012년 추석/설날 영화 키워드



고구려, 전쟁, 세상, 조선, 평양성, 신라



역사/ 전쟁

장화, 고양이, 펭귄, 탐험대, 마법, 모험



애니메이션/ 모험

가족, 아버지, 아들



드라마/ 가족

Part. ●
04

2013~2014년 추석/설날 영화 키워드



아들, 가족, 남자, 운명, 마음, 사랑, 관상

→ 가족/ 드라마/ 로맨스/ 멜로

악당, 몬스터, 슈퍼, 챔피언, 왕국

→ 애니메이션/ 모험 / 코미디

작전, 테러, 음모, 악마, 발생, 국제

→ 액션 / 스릴러/ 범죄 / SF

Part. 04

2015~2016년 추석/설날 영화 키워드

아버지, 아들, 아이돌, 음악



가족 / 모험 / 코미디 / 뮤지컬

수사, 살인, 암호, 조직, 누명, 검사



액션 / 스릴러 / 범죄

조선, 전설, 세자, 시절, 유배, 전쟁



역사/ 전쟁

Part. 04

2017~2018년 추석/설날 영화 키워드

조직, 계획, 수사, 형사, 작전, 강력반

액션/ 범죄/ 스릴러

친구, 사람, 운명

드라마/ 가족/ 멜로/ 코미디

백성, 남한, 북한, 안시성, 조선, 군사

▶ 역사/ 액션/ 모험/ 드라마

**Part.
04**

키워드로 보는 명절 영화장르 top3



가족/드라마/코미디/멜로



액션/ 범죄/ 스릴러/역사



애니메이션 / 모험 / 뮤지컬
판타지 / SF

아쉬운 점

**불용어 추가
영화코드 수집 자동화
Kobis에서 API 데이터 받기**







```
25 import requests
26 import json
27 import pandas as pd
28
29 targetDt = 20110203
30 targetDt_data = str(targetDt)
31
32
33 url = 'http://kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchWeeklyBoxOfficeList.json#'
34 ?key=fb522be7e75eba7a47e8d046a0180b73&targetDt='+targetDt_data      # KOBIS api를 사용하여 원하는 날짜의 주간 박스오피스 접속
35 #&weekGb="0"#
36 #&multiMovieYn="N"
37
38
39 res = requests.get(url)                                     # 요청 성공 <Response [200]>
40 text = res.text
41
42 json_data = json.loads(text)                                # json 데이터를 크롤링 진행
43
44 week_movie_name_list=[]
45 for i in range(len(json_data['boxOfficeResult']['weeklyBoxOfficeList'])):
46     week_movie_name_list.append(json_data['boxOfficeResult']['weeklyBoxOfficeList'][i]['movieNm'])
47                                         # 반복문으로 박스오피스 상위 10개의 영화 제목 저장
48
49 week_movie_name_list_DF = pd.DataFrame(week_movie_name_list)
50 week_movie_name_list_DF.to_csv('week_movie_name_list_DF+.csv', mode='w', encoding='UTF-8', index=False)
51                                         # list -> Dataframe 변환 후 csv로 저장
52 print('week_movie_name_list_DF :',week_movie_name_list_DF )
```



```
52 # print(res) #<Response [200]>
53 # print(type(text)) #str
54
55 json_data = json.loads(text)                                # json으로 정보 변환
56
57
58 week_movie_name_list=[]
59 movie_search_list=[]
60
61                                         # 반복문으로 박스오피스 상위 10개의 영화 제목 저장
62 for i in range(len(json_data['boxOfficeResult']['weeklyBoxOfficeList'])):
63     week_movie_name_list.append(json_data['boxOfficeResult']['weeklyBoxOfficeList'][i]['movieNm'])
64
65 movie_search = week_movie_name_list[i].replace(" ","")           # " " 공백을 ""로 변경(공백삭제)
66 movie_search_list.append(movie_search)
67
68 naver_search_url = 'https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94'+movie_search_
69 print(naver_search_url)                                         # 네이버 영화 + 영화 제목 검색 URL
70
```

```
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94조선명탐정:각시투구꽃의비밀
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94걸리버여행기
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94평양성
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94글러브
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94상하이
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94그린호넷
https://search.naver.com/search.naver?sm=tab_hty.top&where=nexearch&query=%EC%98%81%ED%99%94메가마인드
```



주간/주말 박스오피스 API 서비스

특정 일자가 속한 주차의 주간/주말/주중 상영작들의 박스오피스 정보를 영화구분(다양성영화,상업영화), 한국/외국 구분, 상영지역등의 조건을 통해 조회합니다.

REST/SOAP 방식 중 선택적으로 호출가능하며 REST 방식의 응답형식은 XML과 JSON을 지원합니다.(URI의 extension으로 구분)

1. REST 방식

- 기본 요청 URL : <http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchWeeklyBoxOfficeList.xml> (또는 .json)
- 요청 parameter : 3번항의 요청 인터페이스 정보를 참조하여 GET 방식으로 호출

SEVEN ELEVEN



Q & A

SEVEN ELEVEN

THE END



주제 명절 인기 영화 줄거리 분석을 통한 국내
영화 소재 트렌드 파악

팀장 유진아

협찬 멋쟁이샤자처럼

주인공 유진아, 김나연, 김동수, 안은지, 안태용