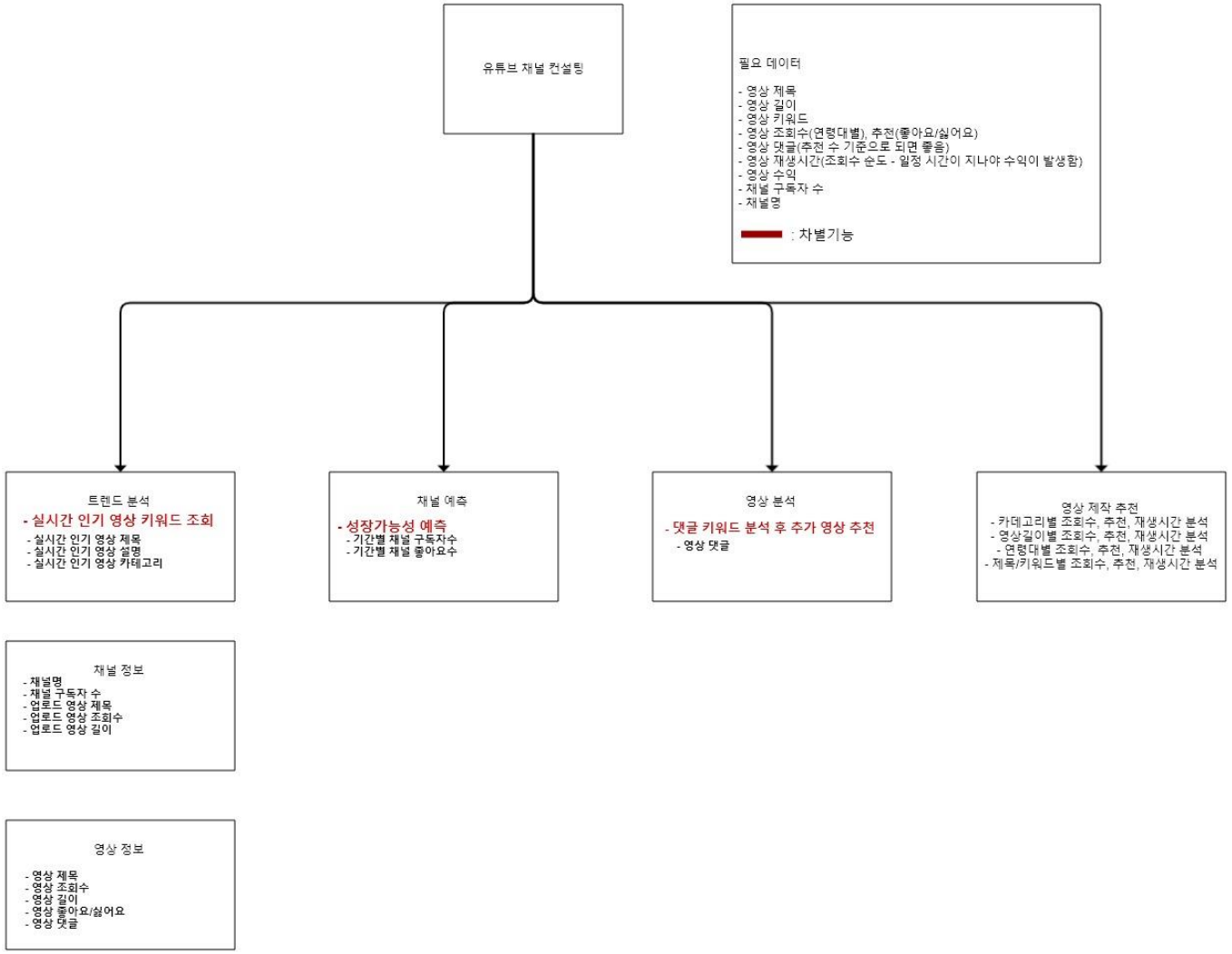

DIGITAL
VINYL
DESIGNED
B Y
L@RGO
ADSTORE

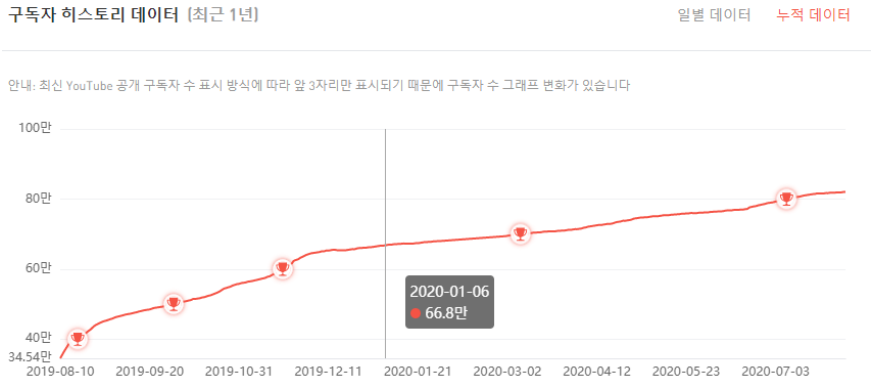
INDEX

- 목적
- 웹 크롤링
- 분석 결과

Flow chart


유튜브 채널 컨설팅
필요 데이터
- 영상 제목
- 영상 길이
- 영상 키워드
- 영상 조회수(연령대별), 추천(좋아요/싫어요)
- 영상 댓글(추천 수 기준으로 되면 좋음)
- 영상 재생시간(조회수 순도 - 일정 시간이 지나야 수익이 발생함)
- 영상 수익
- 재널 구독자 수
- 재널명
: 차별기능
트렌드 분석
- 실시간 인기 영상 키워드 조회
- 실시간 인기 영상 제목
- 실시간 인기 영상 설명
- 실시간 인기 영상 카테고리
채널 예측
- 성장가능성 예측
- 기간별 채널 구독자수
- 기간별 채널 좋아요수
영상 분석
- 댓글 키워드 분석 후 추가 영상 추천
- 영상 댓글
영상 제작 추천
- 카테고리별 조회수, 추천, 재생시간 분석
- 영상길이별 조회수, 추천, 재생시간 분석
- 연령대별 조회수, 추천, 재생시간 분석
- 제목/키워드별 조회수, 추천, 재생시간 분석
채널 정보
- 채널명
- 채널 구독자 수
- 업로드 영상 제목
- 업로드 영상 조회수
- 업로드 영상 길이
영상 정보
- 영상 제목
- 영상 조회수
- 영상 길이
- 영상 좋아요/싫어요
- 영상 댓글

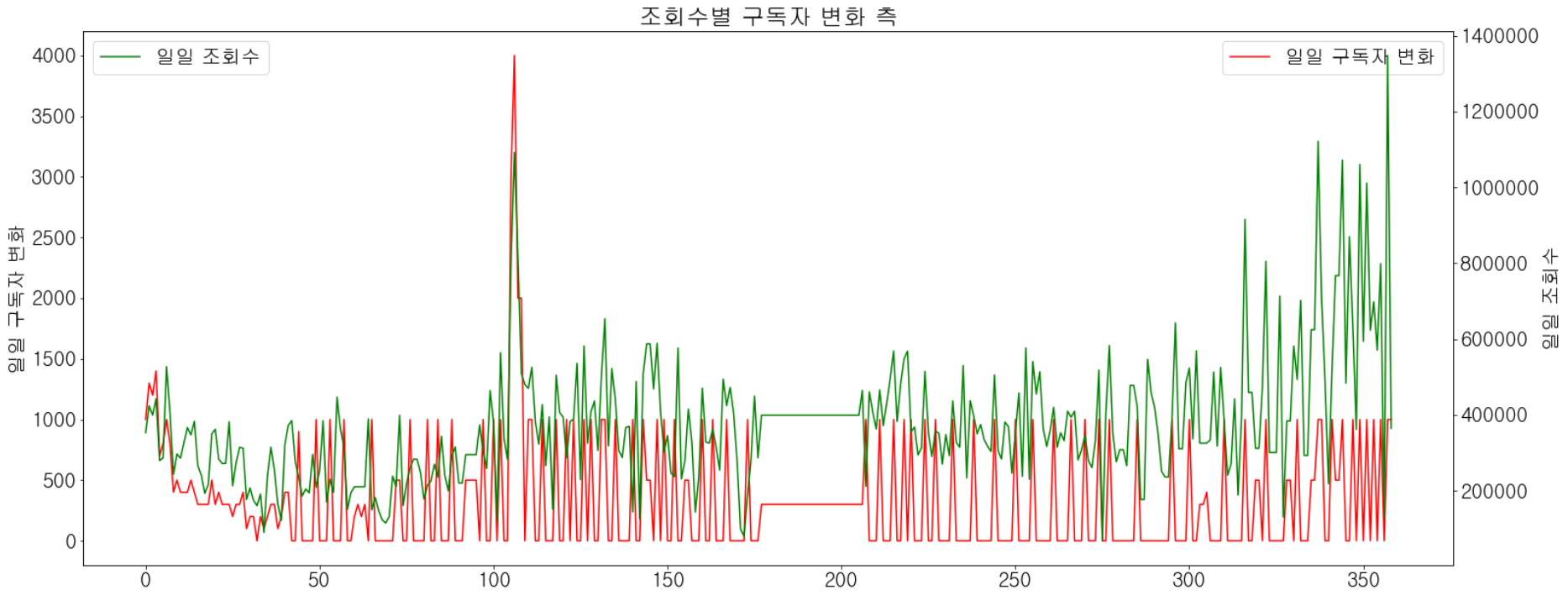
채널 정보 웹크롤링

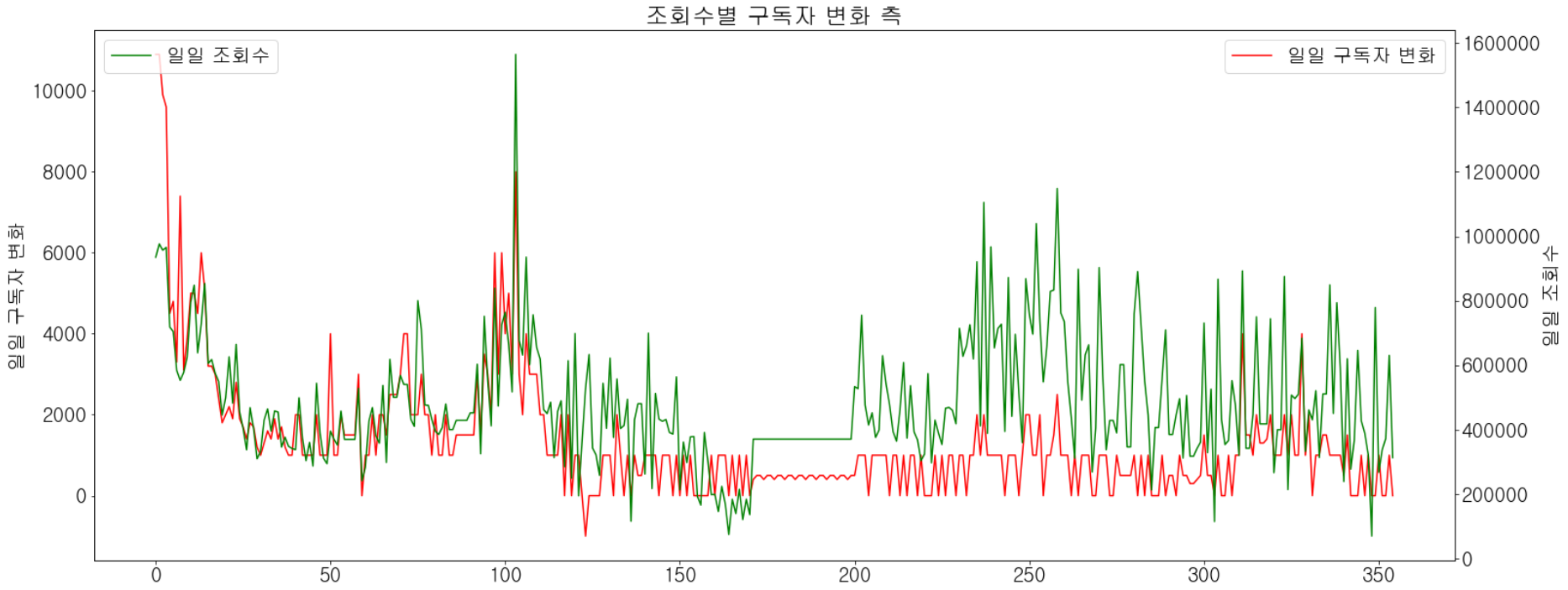



```python
# 구독자 누적 차트에 마우스 호버 실행
element_to_hover_over = driver.find_element_by_xpath('//*[@id="channel-history-sub-chart"]/div[1]/canvas')
hover = ActionChains(driver).move_to_element(element_to_hover_over) # 차트의 가운데로 마우스 호버
hover.perform()

# 차트 사이즈 가져오기
element_size = element_to_hover_over.size
element_height = element_size['height']
element_width = element_size['width']

# 차트의 좌측으로 이동
x_offset = (element_width / 2)
move = ActionChains(driver).move_by_offset(-x_offset, 0)
move.perform()

# 차트의 우측으로 이동하면서 날짜/누적 구독자수 저장
'''
soup = BeautifulSoup(html, 'lxml') -> 사용해서 크롤링 시간 단축 해보기!
'''
list_sub_date = []
list_subscribe_cnt = []
for idx in range(int(element_width / 2)):
    # 마우스 이동
    move = ActionChains(driver).move_by_offset(2, 0)
    move.perform()

    chart_info = driver.find_element_by_xpath('//*[@id="channel-history-sub-chart"]/div[2]').text
    if chart_info.find('\n') != -1 :
        list_sub_date.append(chart_info.split('\n')[0])
```

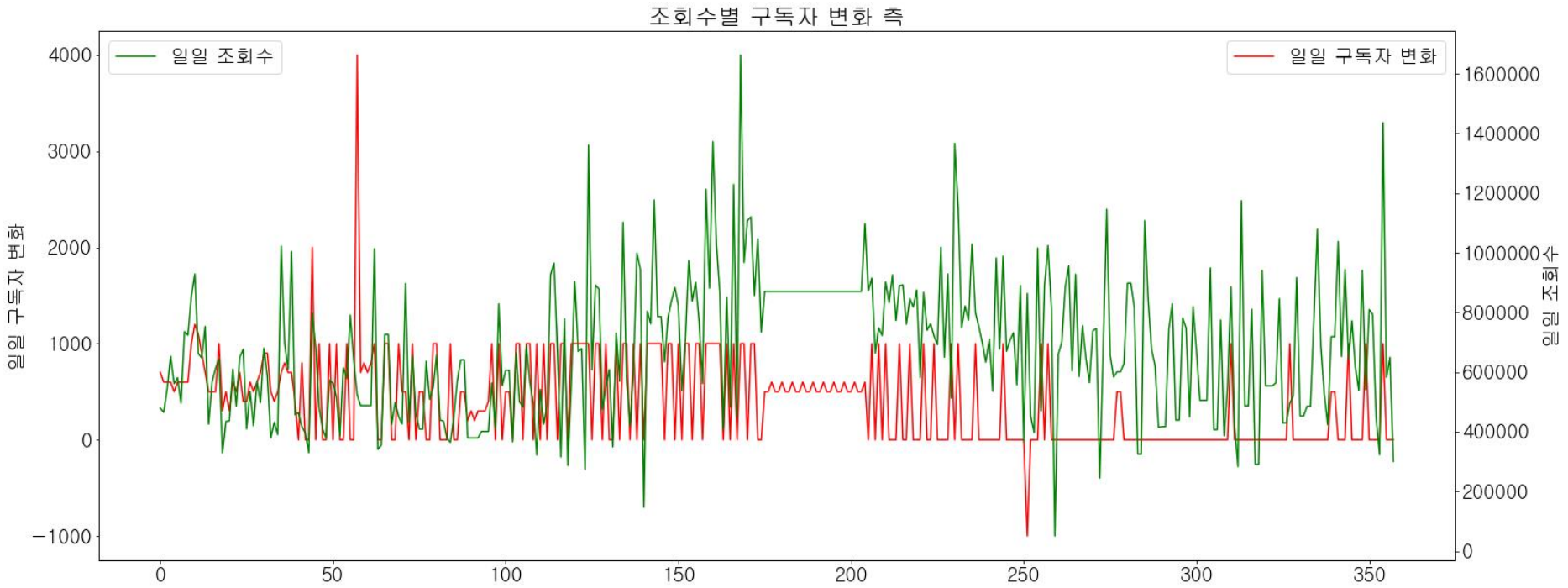
웹 크롤링 결과

A	B	C	D	E
	date	**subscribe count**	**daily subscribe count**	**daily view count**
0	2019-08-06	168700	1000	353200
1	2019-08-07	169700	1300	423900
2	2019-08-08	171000	1200	399300
3	2019-08-09	172200	1400	442300
4	2019-08-10	173600	700	279500
5	2019-08-11	174300	800	286200
6	2019-08-12	175100	1000	527200
7	2019-08-13	176100	800	404500
8	2019-08-14	176900	400	243200
9	2019-08-15	177300	500	296800
10	2019-08-16	177800	400	285700
11	2019-08-17	178200	400	326400
12	2019-08-18	178600	400	366700
13	2019-08-19	179000	500	347100

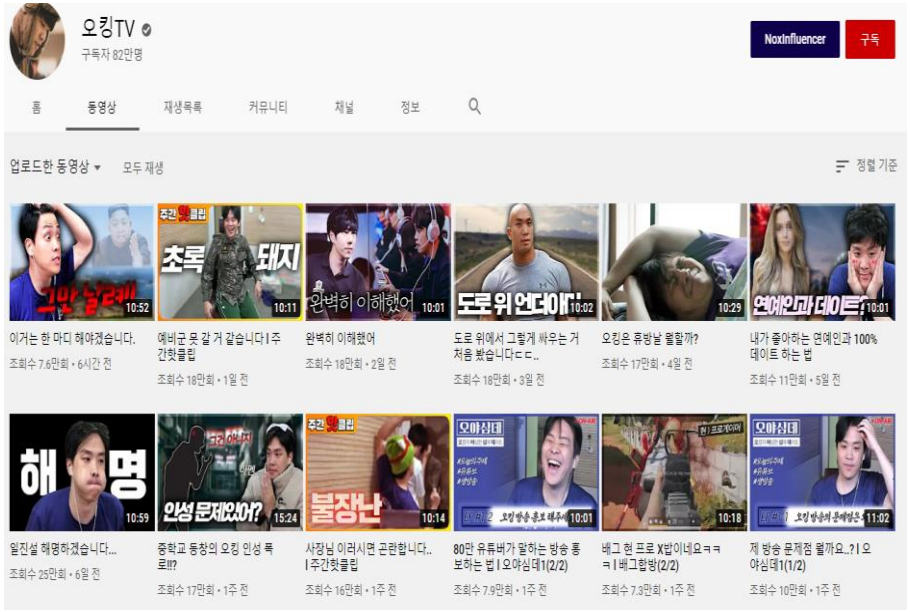

조회수별 구독자 변화 측
일일 조회수
일일 구독자 변화
일일 구독자 변화
일일 조회수
4000
3500
3000
2500
2000
1500
1000
500
0
1400000
1200000
1000000
800000
600000
400000
200000
0
50
100
150
200
250
300
350


조회수별 구독자 변화 측
일일 조회수
일일 구독자 변화
일일 구독자 변화
일일 조회수
10000
8000
6000
4000
2000
0
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
0
50
100
150
200
250
300
350

두찌와뿌꾸 채널


조회수별 구독자 변화 측
일일 조회수
일일 구독자 변화
일일 구독자 변화
일일 조회수
4000
3000
2000
1000
0
−1000
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
0
50
100
150
200
250
300
350

채널 업로드 정보 웹크롤링

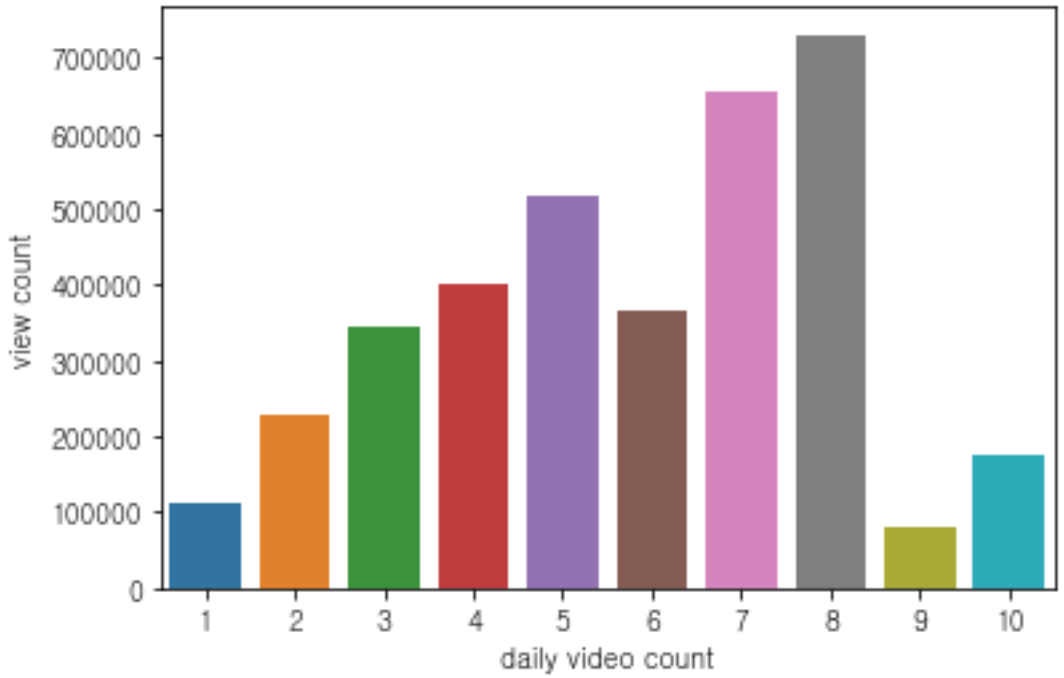



```python
# 동영상 페이지 제일 밑으로 스크롤
while True:
    last_height = driver.execute_script('return document.documentElement.scrollHeight')
    # 현재 화면의 길이를 리턴 받아 last_height에 넣음
    for i in range(10):
        body.send_keys(Keys.END)
        # body 본문에 END키를 입력(스크롤내림)
        time.sleep(SCROLL_PAUSE_TIME)
    new_height = driver.execute_script('return document.documentElement.scrollHeight')
    if new_height == last_height:
        break;

page = driver.page_source
soup = BeautifulSoup(page, 'lxml')

# 채널명 크롤링
username = soup.find('div', {'id' : 'text-container'}).text
username= username.strip()

# 제목, url 크롤링
all_videos = soup.find_all(id='dismissable')

list_title = []
list_url = []
list_video_length = []
for video in all_videos:
    title = video.find(id='video-title')
    if len(title.text.strip())>0: # 공백을 제거하고 글자수가 0보다 크면 append
        list_title.append(title.text)

    #find('a',{'id':'thumbnail'})['href']
    url = video.find(id='video-title')['href'] # url append
    list_url.append(url)

    video_lenth = video.find('span',{'class' : 'style-scope ytd-thumbnail-overlay-time-status-renderer'})
    list_video_length.append(video_lenth.text.strip())
```

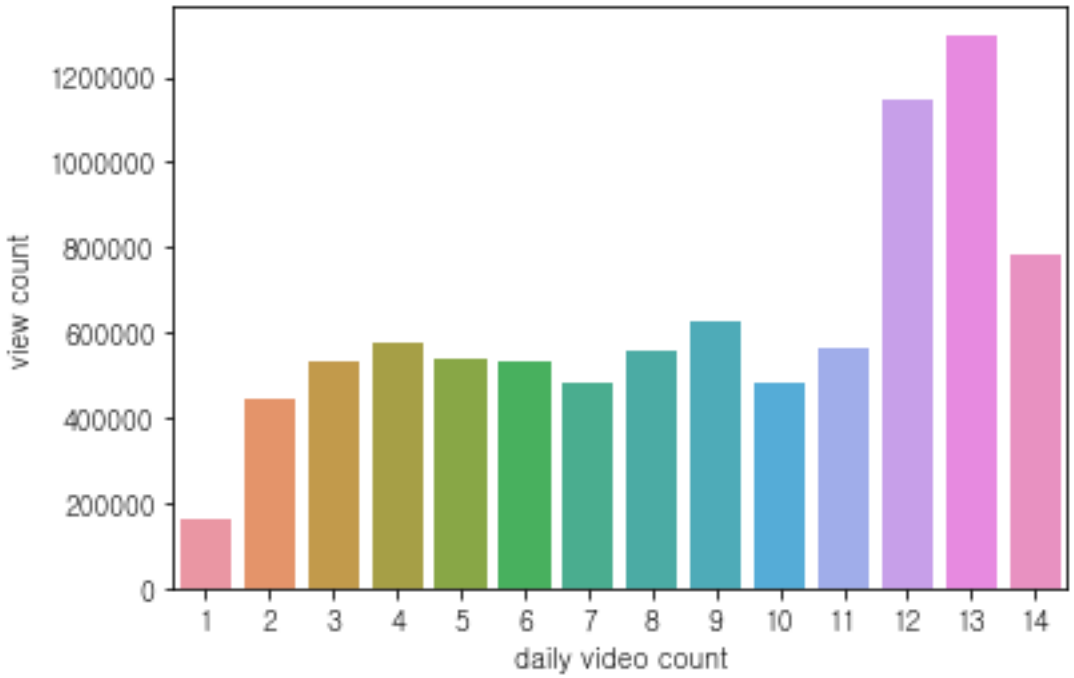
웹 크롤링 결과

	A	B
1	upload_date	daily video count
2	2017-12-09	1
3	2017-12-19	1
4	2017-12-30	1
5	2018-01-01	1
6	2018-01-04	1
7	2018-01-07	1
8	2018-01-11	1
9	2018-01-14	1
10	2018-02-13	1
11	2018-02-18	1
12	2018-02-25	1
13	2018-03-02	1
14	2018-03-15	1
15	2018-03-17	1
16	2018-03-22	1
17	2018-03-30	1
18	2018-04-03	1

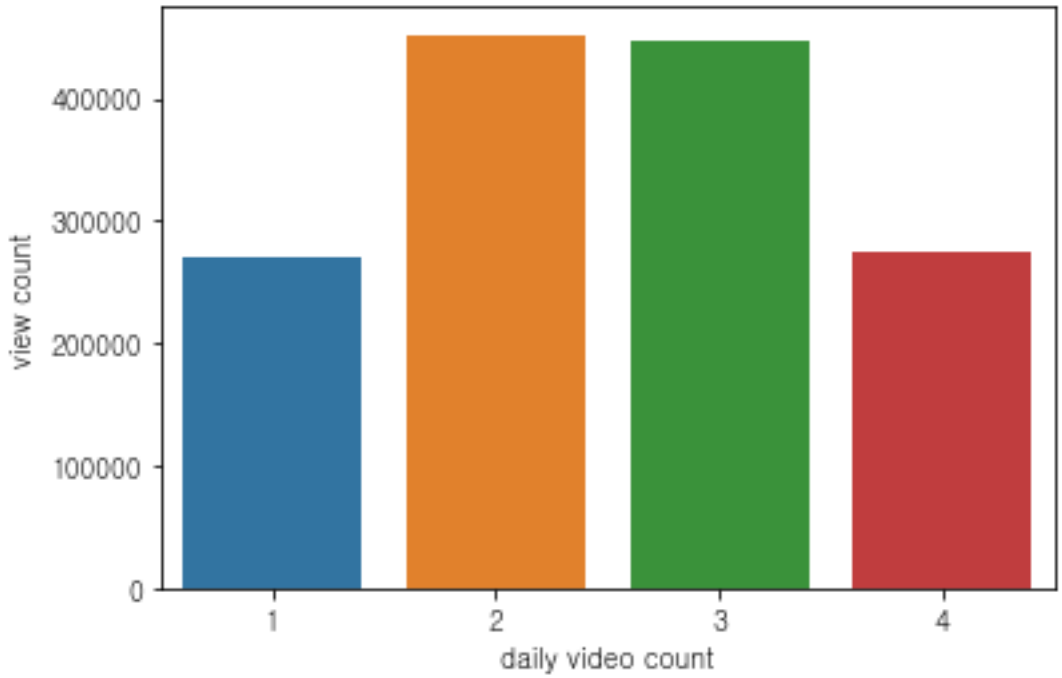
이스타TV 채널
영상개수별 조회수 시각화



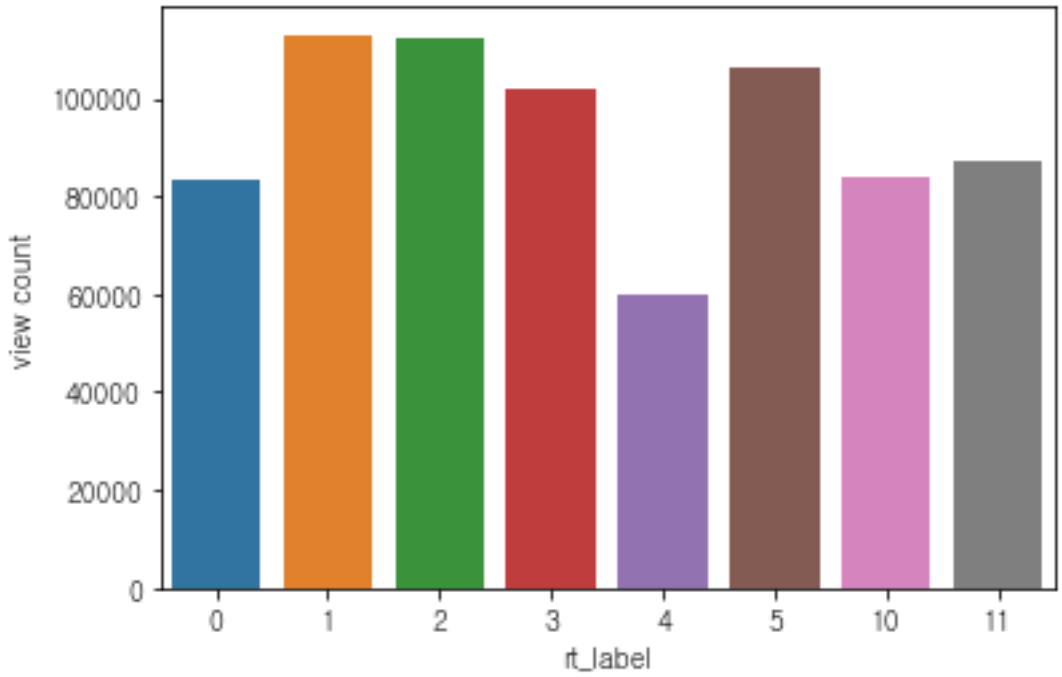
두찌와뿌꾸 채널 영상개수별 조회수 시각화



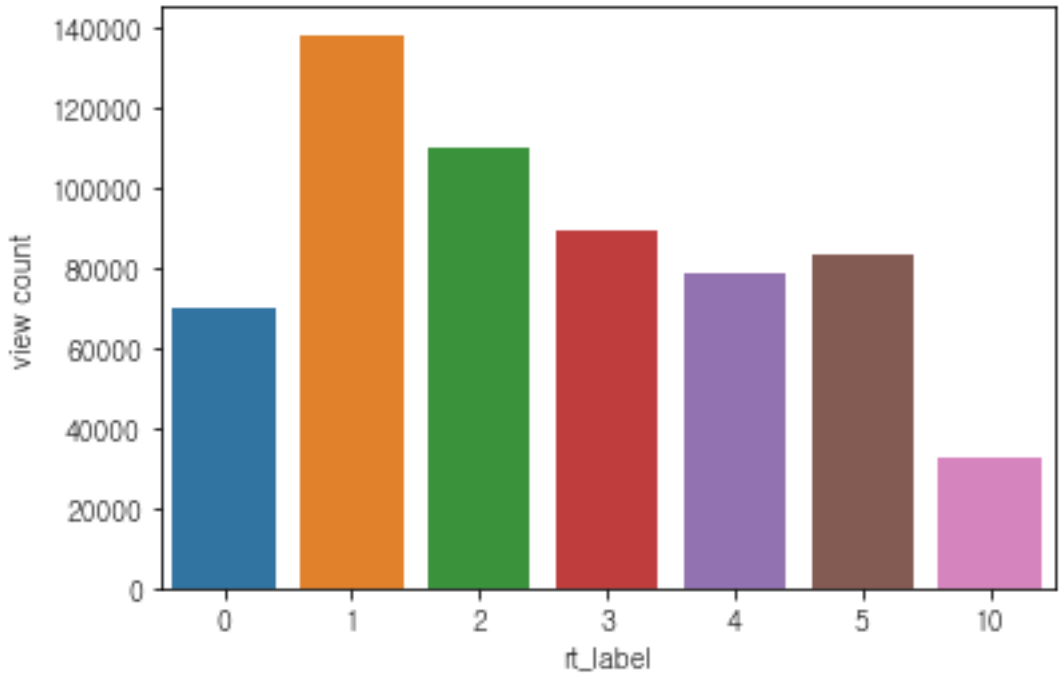
오킹TV 채널
영상개수별 조회수 시각화



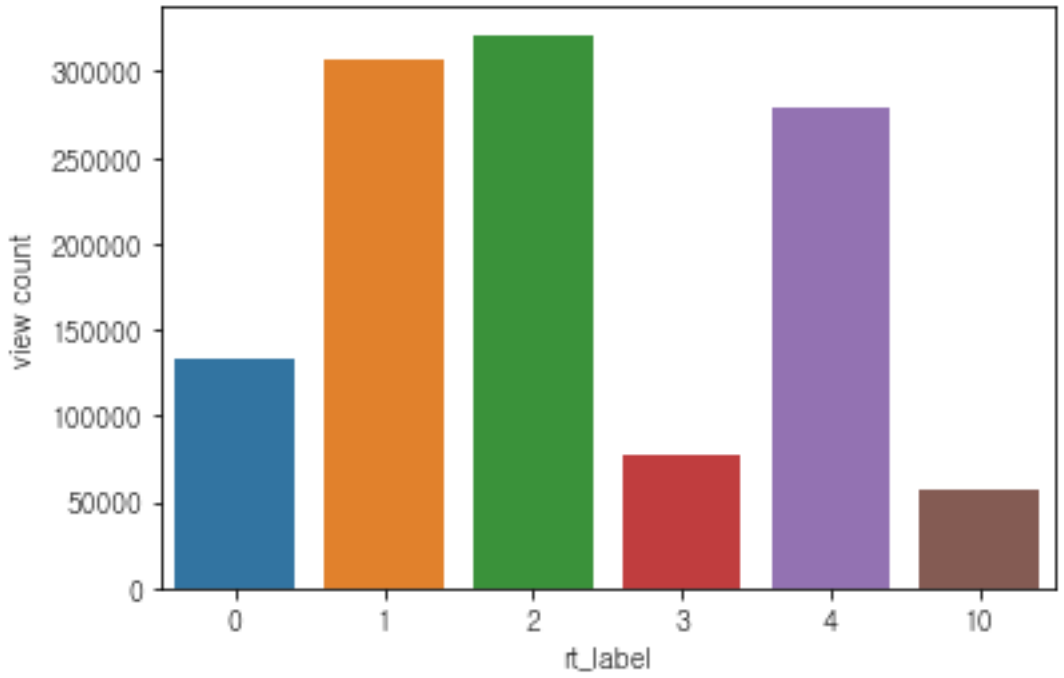
이스타TV 채널
영상 길이별 조회수 시각화



두찌와뿌꾸 채널
영상 길이별 조회수 시각화



오킹TV 채널
영상 길이별 조회수 시각화

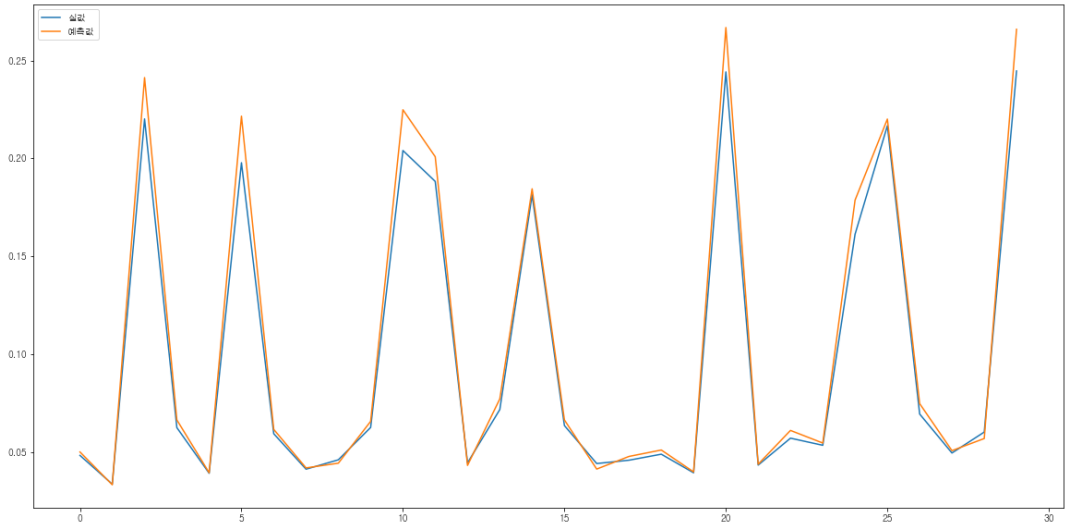


누적 구독자 변화 예측 시각화

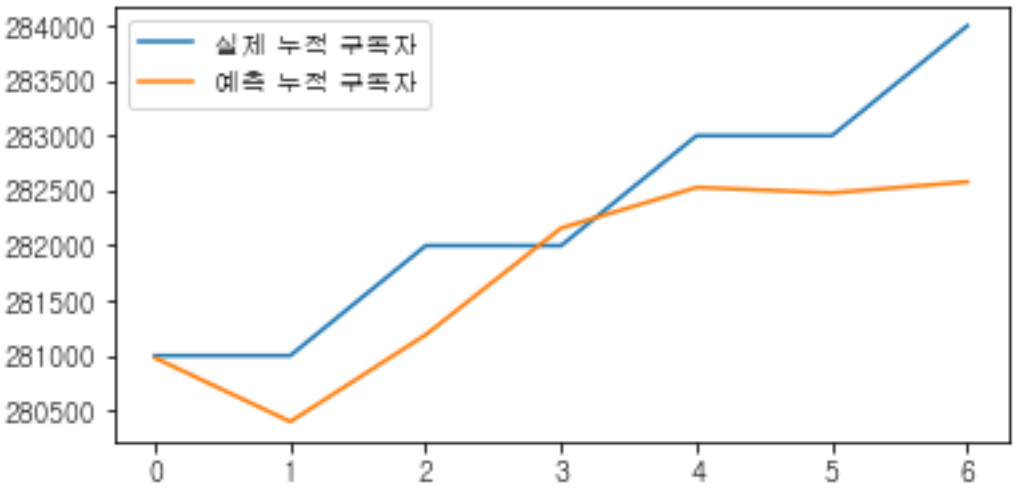
```
subscribe = list(df_read_channel_info['subscribe count'].values) # 누적 구독자 데이터 추출
seq_len = 50 # 기준 날짜 설정(50 -> 최근 50일)
# 데이터 분리 및 정규화
X, Y = normalized_df(subscribe, seq_len)

# 데이터 분할
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,
                                                    Y,
                                                    test_size = 0.1,
                                                    random_state = 0)
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

# 모델 구성
model = Sequential()
model.add(LSTM(100, return_sequences=True, input_shape=(seq_len, 1))) # 왜 input_shape를 2
model.add(LSTM(50, return_sequences=False))
model.add(Dense(1, activation='linear'))
model.compile(loss='mse', optimizer='adam')

model.fit(X_train, Y_train,validation_data=(X_test, Y_test),batch_size=1,epochs=10)

# 예측 값과 실제 값의 비교
Y_prediction = model.predict(X_test).flatten()
for i in range(10):
    label = Y_test[i]
    prediction = Y_prediction[i]
    print("실제가격: {:.3f}, 예상가격: {:.3f}".format(label, prediction))

from sklearn.metrics import r2_score
print('결정계수',r2_score(Y_test, Y_prediction))
```

누적 구독자 변화 예측 시각화

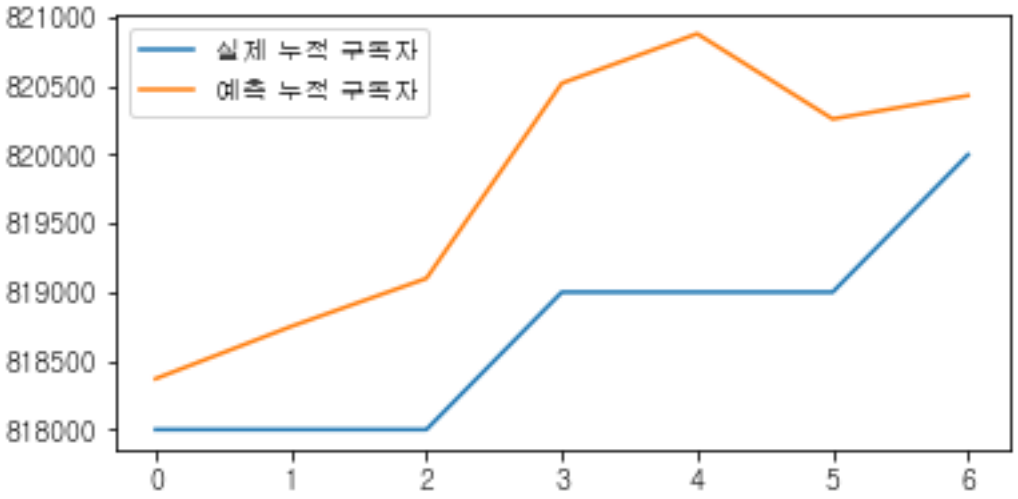
```
val_loss: 1.0165e-04
실제가격: 0.048, 예상가격: 0.050
실제가격: 0.034, 예상가격: 0.033
실제가격: 0.220, 예상가격: 0.241
실제가격: 0.062, 예상가격: 0.066
실제가격: 0.039, 예상가격: 0.039
실제가격: 0.198, 예상가격: 0.222
실제가격: 0.059, 예상가격: 0.062
실제가격: 0.041, 예상가격: 0.042
실제가격: 0.046, 예상가격: 0.044
실제가격: 0.062, 예상가격: 0.066
결정계수 0.9809205139992156
```



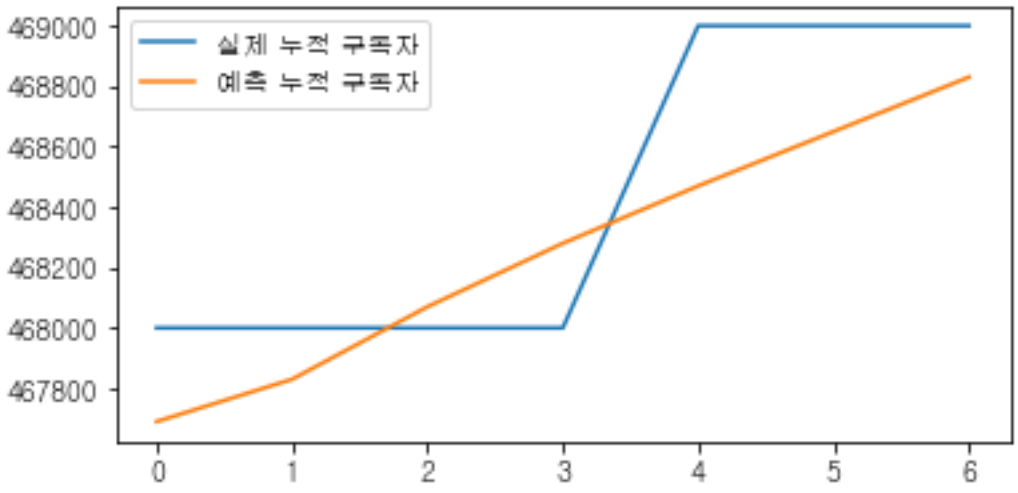
이스타TV 채널
영상개수별 조회수 시각화



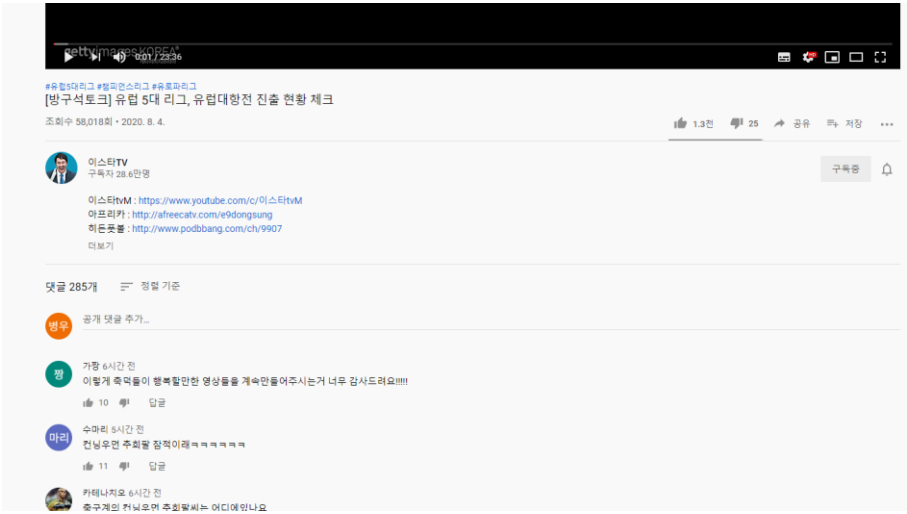
오킹TV 채널
누적 구독자 변화 예측



두찌와뿌꾸 채널 누적 구독자 변화 예측



댓글 웹크롤링




```
SCROLL_PAUSE_TIME = 0.5 # 한번 스크롤 하고 멈출 시간 설정
# 더보기 버튼 클릭을 위해 최초 1회 스크롤 다운
#body.send_keys(Keys.PAGE_DOWN)
#time.sleep(SCROLL_PAUSE_TIME)

# 더보기 클릭
more_button = driver.find_element_by_xpath('//*[@id="more"]/yt-formatted-string')
more_button.click()

# 크롤링을 위해 화면 맨 아래까지 스크롤 내리기
while True:
    last_height = driver.execute_script('return document.documentElement.scrollHeight')
    # 현재 화면의 길이를 리턴 받아 last_height에 넣음
    for i in range(10):
        body.send_keys(Keys.PAGE_DOWN)
        # body 본문에 END키를 입력(스크롤내림)
        time.sleep(SCROLL_PAUSE_TIME)
    new_height = driver.execute_script('return document.documentElement.scrollHeight')
    if new_height == last_height:
        break;

html = driver.page_source
soup = BeautifulSoup(html, 'lxml')

# 댓글 id 추출
all_comment_id = soup.find_all('a', {'id' : 'author-text'})
comment_id = [soup.find_all('a', {'id' : 'author-text'})[n].text for n in range(0,len(all_comment_id))]
comment_id = [i.strip() for i in comment_id]

# 댓글 추출(답글 제외)
all_comment_contents = soup.find_all('yt-formatted-string', {'id' : 'content-text'})
comment_contents = [soup.find_all('yt-formatted-string', {'id' : 'content-text'})[n].text for n in range(len(all_comment_contents))
comment_contents = [i.strip() for i in comment_contents]
```

웹 크롤링 결과

	A	B	
		comment id	
0		milk 0423	섭종각이라던데..형 이제라도 돈아끼고 키티누님이랑 데이트하는영상쫌올려줘
1		・・・	5:55 농 호나우두 금카
2		최준우	1:25 홀리~~ 쓋!!4:29 홀리~~ 쓋!!
3		조정현	5:55 채팅 예언보소
4		이름	5:00실수하네
5		ᄋ.	6:18 ㅆㅂㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅌㅌㅌ
6		으으윽	뿌꾸형 거만흑우일때 넘나 기여워욬ㅋㅋㅋ 개욱겨 증말 홀리쉣~! 홀리 쉿~! 개재밋서
7		차재근	5:48 믹슈 손가락 까닥까닥하는거 호나우두 셀레브레이션 복션 지렸다,...;ㄷㄷ
8		환	피자먹방 보고 온사람
9		김덕현	2:28 날강두 그 발언..
10		노해석	와 호나우두 8카가 나오긴 하는구나...난 EP 500만원 받고 끝인데...
11		유영빈	2:29 실수하는 소리하네
12		김민수	8:08 TB음바페 8:25 라이브음바페진짜 어제 생방으로 봤는데 쌍바페 레전드다 ㅋㅋㅋ
13		추배	믹슈형 아이콘 포함 88팩에서도 아이콘 띄우고호나우두 금카도 띄우고 금손인가?

이모티콘, 불용어 처리 전

df_read_comment - DataFrame

Index	ame	comment id	comment
0	0	이스타TV	제모크림 구매 링크 : http://s.godo.kr/lzgk
1	1	새참	첼시는 뭐 갈락티코도 아니고 존나 공격쪽만 보강하네ㅋㅋㅋㅋㅋ
2	2	khm	케파를 이스타가 깜 > 케파 안정적임 > 케파 잘하는거보고 이…
3	3	이쁜우리 누님	이야 첼시팬들은 우동사리 나가고 나서, 완전 기쁨 그자체네요. 축하드립니다
4	4	차감홍	
5	5	ㅎㅎㅎ	
6	6	anonymous	이 영상을 장지현해설이 좋아합니다
7	7	김동형	첼시팬은 행복합니다😭
8	8	이현수	ㅋㅋㅋ 공격 보다가 수비보니까 무게감 많이 떨어지네
9	9	탁원준	이게 될거라 생각 못했는데 진짜 되면 대박이다
10	10	[Hari st…	하베르츠까지 지르면 첼시 진짜 pl 너무 재밌어지겠네요 ㄷㄷ...
11	11	봉 가	뤼디거 장사 잘한다.국대 친구들 모으네
12	12	파드램	첼시 챔스 진출성공시 정확힌 모르겠지만 800억~900억의 이적 자금을 확보 할 수 있습니다
13	13	Supreme love	첼시 미쳤네.. 0입 한시즌 햇다고 이렇게 폭풍영입하나..리버풀도 미나미노만…

Format Resize Background col Column min/m Save and Clos Close

이모티콘, 불용어 처리 후

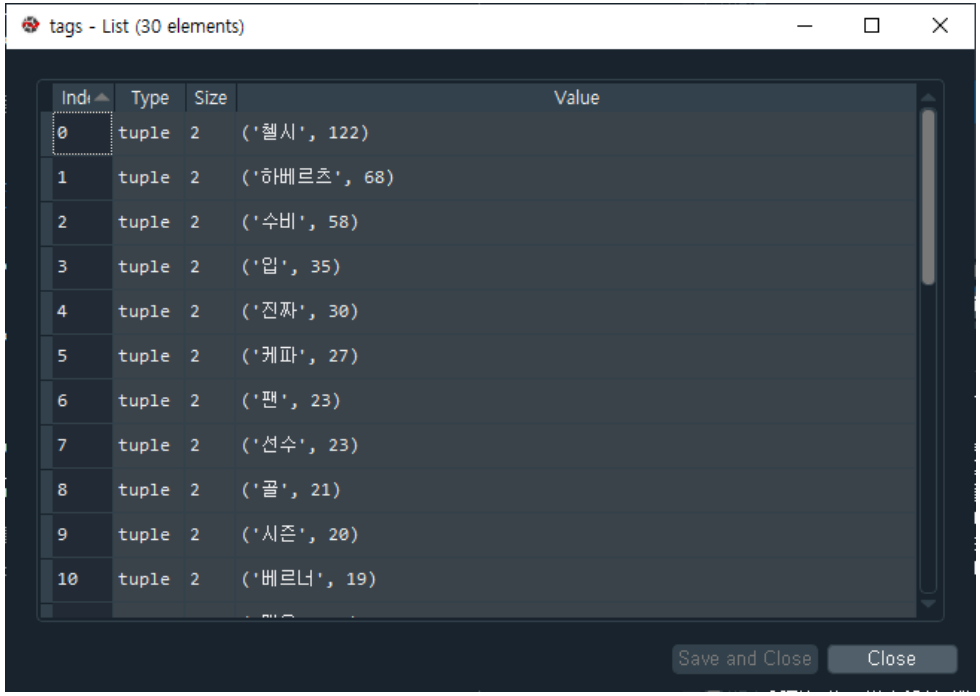

comment_result - List (352 elements)
Index
Type
Size
Value
0 str 1 제모크림 구매 링크 : http://sgodokr/lzgk
1 str 1 첼시는 뭐 갈락티코도 아니고 존나 공격쪽만 보강하네
2 str 1 케파를 이스타가 깜 > 케파 안정적임 > 케파 잘하는거보고 이스타가 사과함 > 케파가 그뒤로 못함 결론: 이 ...
3 str 1 이야 첼시팬들은 우동사리 나가고 나서 완전 기쁨 그자체네요 축하드립니다
4 str 1 마 아스날 한번 와라 어 어디서 좋은 일만 할려고 젊을 때 고생해야지
5 str 1 옛날 마드리드 느낌나네공격은 초호화 수비는 초토화
6 str 1 이 영상을 장지현해설이 좋아합니다
7 str 1 첼시팬은 행복합니다
8 str 1 공격 보다가 수비보니까 무게감 많이 떨어지네
9 str 1 이게 될거라 생각 못했는데 진짜 되면 대박이다
10 str 1 하베르츠까지 지르면 첼시 진짜 pl 너무 재밌어지겠네요
Save and Close
Close

명사 추출

```
def get_noun(comment_txt):
    twitter = Twitter()
    noun = []

    if len(comment_txt)>0:
        tw = twitter.pos(comment_txt)
        for i,j in tw:
            if j == 'Noun':
                noun.append(i)
    return noun
```

명사 빈도수 확인

```
noun_list = []
for i in range(len(df_comment_result)):
    for j in range(len(df_comment_result['token'].iloc[i])):
        noun_list.append(df_comment_result['token'].iloc[i][j])

counts = Counter(noun_list) # 추출된 명사 빈도수 확인
tags = counts.most_common(30) # 빈도수 상위 30개 추출
#%%
```



워드 클라우드

Question