

2023 2학기 인공지능시스템 3조 · 1차 과제

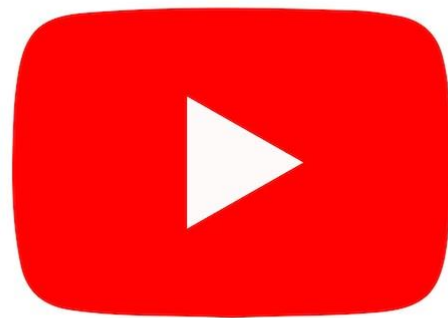
유튜브 영상 조회수 예측

산업정보시스템공학과 _____ 이휘영
인공지능응용학과 21102__ 이예랑
인공지능응용학과 21102392 최세민

차례

1. Project overview
2. Data collection
3. Preprocessing process
4. Model architecture
5. Results

Project overview (what, how, why)



상위에 노출된 댓글의 내용으로
영상의 조회수*를 예측할 수 있을까?

*또는 영상의 흥행 성공 여부

Project overview (what, how, why)

→ '보다' 채널의 영상과 댓글을 이용해
동일 채널에 게시된 영상의 조회수를 예측할 수 있는지 확인



보다 BODA

@boda100 구독자 153만명 동영상 982개

누구나, 쉽게, 재밌게 볼 수 있는 영상 >

smartstore.naver.com/aweplay 외 링크 3개



조회수 60만회

Data Collection

youtube_replies.xlsx

	A	B	C	D	E	F	G	H
1	view	194914	319933	608346	98384	402196	150218	232800
2	0	이 교수님	OCN에 컨	보다 멤버	교수님 강	영화 <크	우와 영상	BODA X
3	1	김기범 교	이 네 분	기노..	학창시절	크리에이	터제가 이	때 마지막
4	2	ㅋㅋㅋㅋ	ㅋㅋ 이	건 멤버	십분들	역사든	전 말씀	하신 ※
5	3	이 고양이	@user-ig					
6	4	교수님 목	철학과	문학도	추가시	≤≤유명	지스카이	넷드제가
7	5	그렇네요	역시 허	준형님	의 진행	그냥 답	글 평소	에 과학
8	6	더더욱	올려준	님 진	게스트	새 @user	-dik	인간들이
9	7	공항 짓	는 0 0	를 보	@analyst	역사를	보다 ai	에 대한
10	8	@RuRan	항상 재	밋이 영	상은 ≤	≤유명	지Oh good	맥락이
11	9	올름도	화산	광민수	소각저	랑같은	생BODA	늘 AI에
12	10	올름도	에서점	점 네	분 몇	달 전	부 믿고	보는
13	11	궁금한	내과	학을	보다	는 흥	미롭 ≤	≤유명
14	12	전달력	이 역	사를	보	다 김	범준 교	와 진짜
15	13	멋지세	요 역	사를	보	다	진짜	학창

(51rows*201columns)

최근 200개 영상의
상위 10개* 댓글 이용

*오류 감안해 50개 수집

Data Collection

youtube_replies.xlsx

	A	B	C	D	E	F	G	H
1	view	194914	319933	608346	98384	402196	150218	232800
2	0	이 교수님	OCN에 컨	보다 멤버	교수님 강	영화 <크	우와 영상	BODA X
3	1	김기범 교	이 네 분	기노..	학창시절	크리에이	터제가 이	때 마지막
4	2	ㅋㅋㅋㅋ	ㅋㅋ	이건	멤버십분	들역사든	전말씀하	신 {※ 이
5	3	교수님 목	철학과 문	학도 추가	시≤유명	지스카이	넷드제가	들어공부
6	4	그렇네요	역시 허준	형님의 진	행그냥 답	글평소에	과학자역	학적무지
7	5	더더욱 올	허준님 진	게스트 새	@user-di	인간들이	기보다채	널 ㅋㅋ
8	6	공항 짓는	○○○를 보	@analyst	역사를보	다ai에 대	한파동의	형태 @jo
9	7	@RuRan	항상 재밌	이 영상은	≤유명지	Oh good	맥락이 없	@joshuat
10	8	올름도화	산광민수	소저랑같	은생BODA	늘AI에 대	한사실 실험	(멍청한 사
11	9	올름도에서	점점 네분	몇 달 전	부믿고 보	는크리에	이터Blue	Box 웅배
12	10	궁금한 내	과학을 보	는 흥미롭	≤유명지	인류 VS	A.그럼 블	랙연대화
13	11	전달력이	역사를 보	다김범준	교와 진짜	페정말, 정	영어렵게	설≤유명
14	12	멋지세요	. 역사를보	다진짜 학	창페르시아	기정영진	매무슨말	인지과학
15	13	화산에 대	정말 재밌	학창시절	어몽골이	그과 42분	짜그니까	중외계인

→ 조회수

→ 댓글 내용

(51rows*201columns)

Data Collection

Selenium, BeautifulSoup 라이브러리 이용

```
from selenium import webdriver
import time
from openpyxl import Workbook
import pandas as pd
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
import requests

import warnings
warnings.filterwarnings('ignore')
```

리스트(*url_list*)의 영상 전부에 대해 실행됨

*유튜브 api 이용 시 발생하는 오류를 고치지 못해 링크는 직접 복사함

```
while True:
    if ii == len(url_list):
        break
```

```
url = url_list[ii]
```

```
driver = webdriver.Chrome("chromedriver.exe")
driver.get(url)
driver.implicitly_wait(3)
```

```
# ----- #
```

```
url_list = ["https://youtu.b
https://youtu.be
https://youtu.be
https://youtu.be
```

Data Collection

```
# ----- #
for i in range(5):
    driver.execute_script("window.scrollTo(0,document.documentElement.scrollHeight);")
    time.sleep(2)

time.sleep(2)

# ----- #

try:
    driver.find_element_by_css_selector("#dismiss-button").click()
except:
    pass

# ----- #

buttons = driver.find_elements(By.CSS_SELECTOR, "#more-replies")
time.sleep(3)

for button in buttons:
    button.click()
    time.sleep(2)

# ----- #
```

댓글 확인을 위한
스크롤(5회)

광고 확인
(dismiss)

답글 확인
(more replies)

```
# ----- #

while True:
    try:
        more_replies = driver.find_element(By.XPATH, "//span[text()='자세히 보기']")
        if more_replies.is_displayed():
            more_replies.click()
            time.sleep(2)
        else:
            break
    except Exception as e:
        break

time.sleep(3)

# ----- #

html_source = driver.page_source
soup = BeautifulSoup(html_source, 'html.parser')

comment_list = soup.select("yt-formatted-string#content-text")
comment_final = []

# ----- #
```

댓글 더보기
(자세히 보기)
확인

BeautifulSoup 이용
→ html 파싱

```
for i in range(len(comment_list)):
    temp_comment = comment_list[i].text
    temp_comment = temp_comment.replace('\n', '')
    temp_comment = temp_comment.replace('\t', '')
    temp_comment = temp_comment.replace(' ', '')
    comment_final.append(temp_comment)
```

공백 제거

```
# ----- #

view_count_element = soup.find("span", class_="view-count")
view_count_text = view_count_element.get_text()
view_count = ''.join(filter(str.isdigit, view_count_text))
print("현재 조회수:", view_count)
```

조회수 수집

```
# ----- #

if len(comment_final) > 50:
    del comment_final[50:]
```

50개 이후의 댓글 제거

```
youtube_pd[view_count] = comment_final
```

```
# ----- #

print("받은 영상 수: " + str(youtube_pd.shape[1]))
ii += 1
print("index: " + str(ii))
print("# ----- #")
```

저장 후 반복

Data Preprocessing

	Views	comments	label
0	194914	이 교수님 정말 재미있게 강의해주시네요... 자주 봐왔음 좋겠습니다.\n김기범 교수...	Good
1	319933	OCN에 컨김에 왕까지에서 허준씨를 처음 봤을때는 이렇게 될 줄은 상상도 못 했습니...	Good
3	98384	교수님 강의를 이렇게 쉽게 들을 수 있다니 행운이고 감사합니다.\n학창시절엔 과학이...	Good
4	402196	영화 <크리에이터> 예매권 이벤트 이 영상에 영화에 대한 기대평 댓글을\n남겨주신...	Good
5	150218	우와 영상 재밌게봤어요! 넘넘 차분하게 전달력 있게 말씀해주시네요.. 하도 슈뢰딩거...	Good
...
195	287723	다양한 천문학 이야기 전할 수 있는 좋은 기회 주신 BODA 분들께 감사드립니다! ...	Good
196	2298575	곧 안전하고 값싼 우주 여행이 실현되길 바랍니다 우주 여행 관련 이야기할 수 있는...	Great
197	362675	오늘도 초대해 주셔서 감사합니다!!!! 자주 놀러와서 축구 이야기 해볼게요!!영상 ...	Good
198	353262	여자를 위해, 여자를 가장 잘 아는 산부인과 의사 언니 김지연 원장이 개발한 우먼스37...	Good
199	744042	스스로 타는 천체, 스타 개그는 제가 만든 거 아닙니다. 저도 배운거예요.. \n어...	Great

198 rows × 3 columns

views: 조회수
comments: 댓글 내용
label: 조회수 분류

조회수에 따른 label 부여
600,001 이상: Great
그 외: Good

Data Preprocessing

	Views	comments	label
0	194914	이 교수님 정말 재미있게 강의해주시네요... 자주 봐있음 좋겠습니다.\n김기원 교수...	Good
1	319933	OCN에 컨셉에	
3	98384	교수님 강의를 이	
4	402196	영화 <크리에이	
5	150218	우와 영상 재밌	
...	...		
195	287723	다양한 천문	
196	2298575	곧 안전하고 강	
197	362675	오늘도 초대	
198	353262	여자를 위해, 여	
199	744042	스스로 타는 천체, 스타 개그는 제가 만든 거 아닙니다. 저도 뉴배운거예요.. \n어...	Great

198 rows × 3 columns

```
new_df = pd.DataFrame([])
new_df["Views"] = df_trans["view"]
new_df["comments"] = df_trans.iloc[:,1] #df_trans.iloc[:,1:11]
```

```
new_df = pd.DataFrame([])
new_df["Views"] = df_trans["view"]
new_df["comments"] = df_trans.iloc[:,1] #df_trans.iloc[:,1:11]
for i in range(2,NOC+1):
    new_df["comments"] += "\n" + df_trans.iloc[:,i]
#60만회 초과인지 기준으로
new_df["label"] = new_df["Views"].apply(lambda x: 'Great' if int(x)>10e4*6 else 'Good')
#new_df.fillna("\n", inplace=True)
new_df = new_df.dropna()
new_df
```

views: 조회수

댓글 내용
= 분류

큰 label 부여
Great

Data Preprocessing

comments

이 교수님 정말 재미있게 강의해주시네요 자주 보았음 좋겠습니다 김기범 교수님 ...

OCN에 컨김에 왕까지에서 허준씨를 처음 봤을때는 이렇게 될 줄은 상상도 못 했습니...

교수님 강의를 이렇게 쉽게 들을 수 있다니 행운이고 감사합니다 학창시절땀 과학이나...

영화 크리에이터 예매권 이벤트 이 영상에 영화에 대한 기대평 댓글을 남겨주신 ...

우와 영상 재밌게봤어요 넘넘 차분하게 전달력 있게 말씀해주시네요 하도 슈뢰딩거고...

...

다양한 천문학 이야기 전할 수 있는 좋은 기회 주신 BODA 분들께 감사드립니다 ...

곧 안전하고 값싼 우주 여행이 실현되길 바랍니다 우주 여행 관련 이야기할 수 있는...

오늘도 초대해 주셔서 감사합니다 자주 놀러와서 축구 이야기 해볼게요 영상 재미있게...

여자를 위해 여자를 가장 잘 아는 산부인과 의사 언니 김지연 원장이 개발한 우먼스37...

스스로 타는 천체 스타 개그는 제가 만든 거 아닙니다 저도 배우거든요 어려울...

1) Punctuation 및 특수문자 제거

```
#punctuation 제거, 특수문자 제거, Lowercasing
def remove_white_space(text):
    text = re.sub(r'[\t\r\n\f\v]', ' ', str(text))
    return text

def remove_special_char(text):
    text = re.sub('[^가-힣 0-9 a-z A-Z]+', ' ', str(text))
    text.lower()
    return text

new_df.comments = new_df.comments.apply(remove_white_space)
new_df.comments = new_df.comments.apply(remove_special_char)
new_df
```

Data Preprocessing

token_final

[교수, 님, 정말, 재미있다, 강의, 해주다, 자주, 뵈다, 좋다, 김기범, 교수...

[OCN, 커다, 김, 왕, 까지에, 서다, 허준, 씨, 처음, 보다, 때, 이렇게...

[교수, 님, 강의, 이렇게, 쉬다, 들다, 수, 있다, 행운, 이고, 감사하다, ...

[영화, 크리에이터, 예매, 권, 이벤트, 영상, 영화, 대한, 기, 대, 평, 댓...

[우와, 영상, 재밌다, 보다, 넘다, 넘다, 차분하다, 전달, 력, 있다, 말씀...

...

[다양하다, 천문학, 이야기, 전, 수, 있다, 좋다, 기회, 주신, BODA, 분...

[곧, 안전하다, 값싸다, 우주, 여행, 실현, 되다, 바라다, 우주, 여행, 관련...

[오늘, 초대, 주다, 감사하다, 자주, 놀러와, 서, 축구, 이야기, 해보다, 영...

[여자, 위해, 여자, 가장, 자다, 알다, 산부인과, 의사, 언니, 김지연, 원장...

[스스로, 타다, 천체, 스타, 개그, 제, 만들다, 거, 아니다, 저, ㅠ, 배우...

2) Stopwords 제거 및 Tokenize

불용어 정의

```
stopwords = ['의', '가', '이', '은', '들', '는', '좀', '잘', '강', '과', '도', '를', '으로', '자', '에', '와', '한', '하다', 'ㅋㅋ']
```

```
okt = Okt()
```

```
tokenized_data = []
```

```
for sentence in tqdm(new_df['comments']):
```

```
    tokenized_sentence = okt.morphs(sentence, norm=True, stem=True) # 토큰화
```

```
    stopwords_removed_sentence = [word for word in tokenized_sentence if not word in stopwords] # 불용어 제거
```

```
    tokenized_data.append(stopwords_removed_sentence)
```

tqdm(진행 상황 표시), konlpy(Okt()), 한국어 분석) 이용

Data Preprocessing

단어 임베딩(Word2Vec)

```
embedding_model = Word2Vec(tokenized_df['token_final'],
                             sg = 2, # skip-gram
                             vector_size = 100,
                             window = 2,
                             min_count = 1,
                             workers = 8
                             )

print(embedding_model)
```

Word2Vec<vocab=18138, vector_size=100, alpha=0.025>

```
embedding_model.wv.save_word2vec_format('week3pj_tokens_eng_w2v')
loaded_model = KeyedVectors.load_word2vec_format('week3pj_tokens_eng_w2v') # 모델 로드

model_result = loaded_model.most_similar("AI")
print(model_result)
```

[('방향', 0.9742190837860107), ('군대', 0.9709122180938721), ('과거', 0.9689631462097168), ('입장', 0.9677556753158569), ('생물', 0.9669126868247986), ('감정', 0.9665217399597168), ('결과', 0.9659112095832825), ('혹시', 0.9650603532791138), ('실험', 0.9631057381629944), ('경험', 0.9629468321800232)]

Data Preprocessing

단어 임베딩(Word2Vec)

```
embedding_model = Word2Vec(tokenized_df['token_final'],  
                             sg = 2, # skip-gram  
                             vector_size = 100,  
                             window = 2,  
                             min_count = 1,  
                             workers = 8  
                             )
```

- Skip-gram 이용
- 100차원 벡터
- 인근 2개 단어와의 관계 학습(window=2)
- 등장 단어 전부 학습에 이용

데이터 분할(8:2)

```
rng = RandomState()  
  
tr = df.sample(frac=0.8, random_state=rng)  
val = df.loc[~df.index.isin(tr.index)]  
  
tr.to_csv('train.csv', index=False, encoding='utf-8-sig')  
val.to_csv('validation.csv', index=False, encoding='utf-8-sig')
```

- Train 8 : Val 2 로 분할