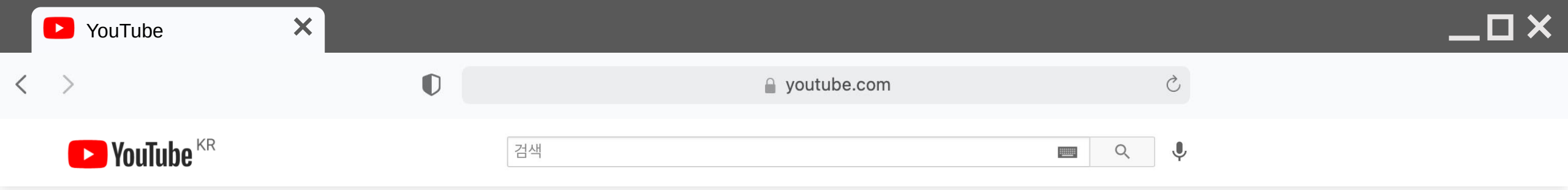


YouTube Trend 분석 및 긍/부정 모델 구축

하하호호

강민지, 김정연,
김형준, 문지원

youtube.com

YouTube KR

0. Content

youtube.com

YouTube KR

1. 개요_ YouTube Boom의 예견

코로나19로 집콕한 3월, 유튜브로 이용자 몰렸다

동영상 플랫폼 순 방문자 수 TOP 10*

	APP	
01	유튜브	2,887.1
02	네이버 밴드	1,585.6
03	인스타그램	1,105.8
04	페이스북	929.5
05	넷플릭스	342.5
06	웨이브	256.2
07	트위터	231.0
08	틱톡	209.4
09	U+ 모바일 tv	176.2
10	네이버TV	172.3

	WEB	
01	유튜브	1,340.0
02	페이스북	564.1
03	네이버TV	436.6
04	카카오TV	282.5
05	인스타그램	267.4
06	트위터	187.5
07	넷플릭스	157.1
08	티빙	152.9
09	밴드	137.1
10	아프리카TV	126.4

동영상 플랫폼 평균 실행 횟수 TOP 10*

	APP	
01	트위터	290.7
02	페이스북	131.2
03	인스타그램	123.0
04	유튜브	111.7
05	틱톡	76.7
06	트위치	70.1
07	네이버밴드	62.6
08	아프리카TV	53.4
09	웨이브	32.5
10	넷플릭스	31.7

	WEB	
01	유튜브	99.6
02	트위터	87.3
03	라프텔	74.4
04	웨이브	50.9
05	아프리카TV	32.9
06	넷플릭스	32.8
07	트위치	23.7
08	밴드	21.0
09	티빙	18.7
10	인스타그램	18.0

동영상 플랫폼 평균 체류시간 TOP 10*

	APP	
01	유튜브	1,464.5
02	트위터	966.7
03	트위치	713.3
04	웨이브	625.5
05	아프리카TV	571.2
06	페이스북	543.1
07	넷플릭스	494.8
08	틱톡	452.0
09	티빙	354.3
10	인스타그램	328.6

	WEB	
01	유튜브	124.3
02	라프텔	78.8
03	트위터	60.9
04	웨이브	36.5
05	넷플릭스	26.9
06	트위치	24.4
07	아프리카TV	22.1
08	네이버TV	19.1
09	밴드	18.9
10	티빙	13.6

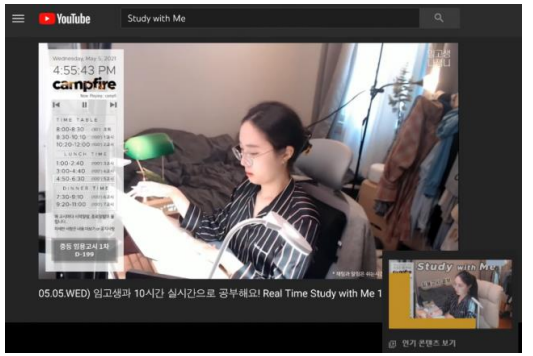
incross 인크로스 미디어 데이터 클리핑 4월 호 <동영상 플랫폼> 편

*2020. 03. 데이터 기준
*App, Web 주요 동영상 플랫폼 기준 순 방문자 수 내림차순 기준으로 분석함
**App은 평균 실행 횟수, Web은 페이지뷰로 각각 분석함

이재원 인크로스 대표는 "최근 디지털 동영상 소비가 유튜브 위주로 이루어지고 있는 한편 온라인동영상서비스(OTT), 소셜 미디어 등 모바일 중심의 다양한 동영상 플랫폼들 역시 상승세를 나타내고 있다"며 "특히 코로나19 영향으로 실내 활동이 증가하면서 3월 동영상 소비가 유의미하게 늘었다는 점에서 마케터들은 광고·마케팅 활동에서 디지털 동영상 매체에 주목할 필요가 있다"고 말했다.

-출처: 아주경제. 차현아 기자 (2020.05.02)

온라인 독서실 160%·출산 브이로그 250%↑…코로나 시대, 유튜브 스트리밍으로 소통한다



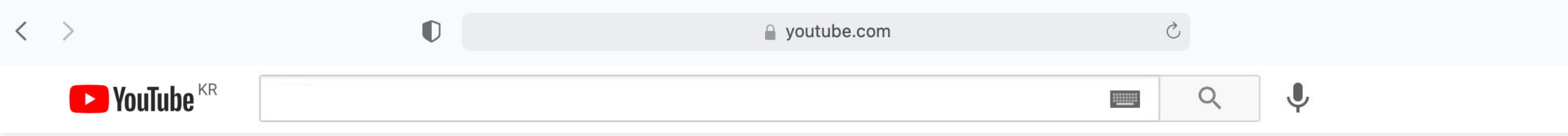
-출처: 서울경제. 차현아 기자 (2021.06.21)

열정의 '쌤튜버'…연예인 쌤 아닌 우리 선생님 채널요~

경기지역 '쌤튜버' 600명…수업 영상부터 마스크 송까지
MZ세대 교사 등장과 코로나 비대면수업으로 관심 급증
국내 최고 '쌤튜버' 악플에 교직 포기 후유증도
경기도 교육청 "교사 품위 유지 등 지침"

경기도 광주의 광수중학교 교사인 '과학교사K'는 과학 학습 영상을 올리는 쌤튜버로 누적 조회수만 80만회에 이른다.

-출처: 한겨레. 차현아 기자 (2021.07.12)



2. EDA 분석

youtube.com

YouTube KR

2. EDA 분석_

	categoryId	view_count	likes	dislikes	comment_count
count	64154.000000	6.415400e+04	6.415400e+04	64154.000000	6.415400e+04
mean	20.404511	1.551901e+06	7.745969e+04	1354.628067	1.165818e+04
std	6.486183	6.778900e+06	4.810992e+05	13603.911092	1.497904e+05
min	1.000000	0.000000e+00	0.000000e+00	0.000000	0.000000e+00
25%	17.000000	2.945252e+05	4.908250e+03	112.000000	6.170000e+02
50%	24.000000	5.905640e+05	9.962500e+03	229.000000	1.384000e+03
75%	24.000000	1.209719e+06	2.326900e+04	491.000000	3.325000e+03
max	29.000000	2.963142e+08	1.646425e+07	879358.000000	6.939302e+06

- 인기 동영상의 평균 조회수: 1.551901e+06회
- 조회수의 중앙값: 5.905640e+05회

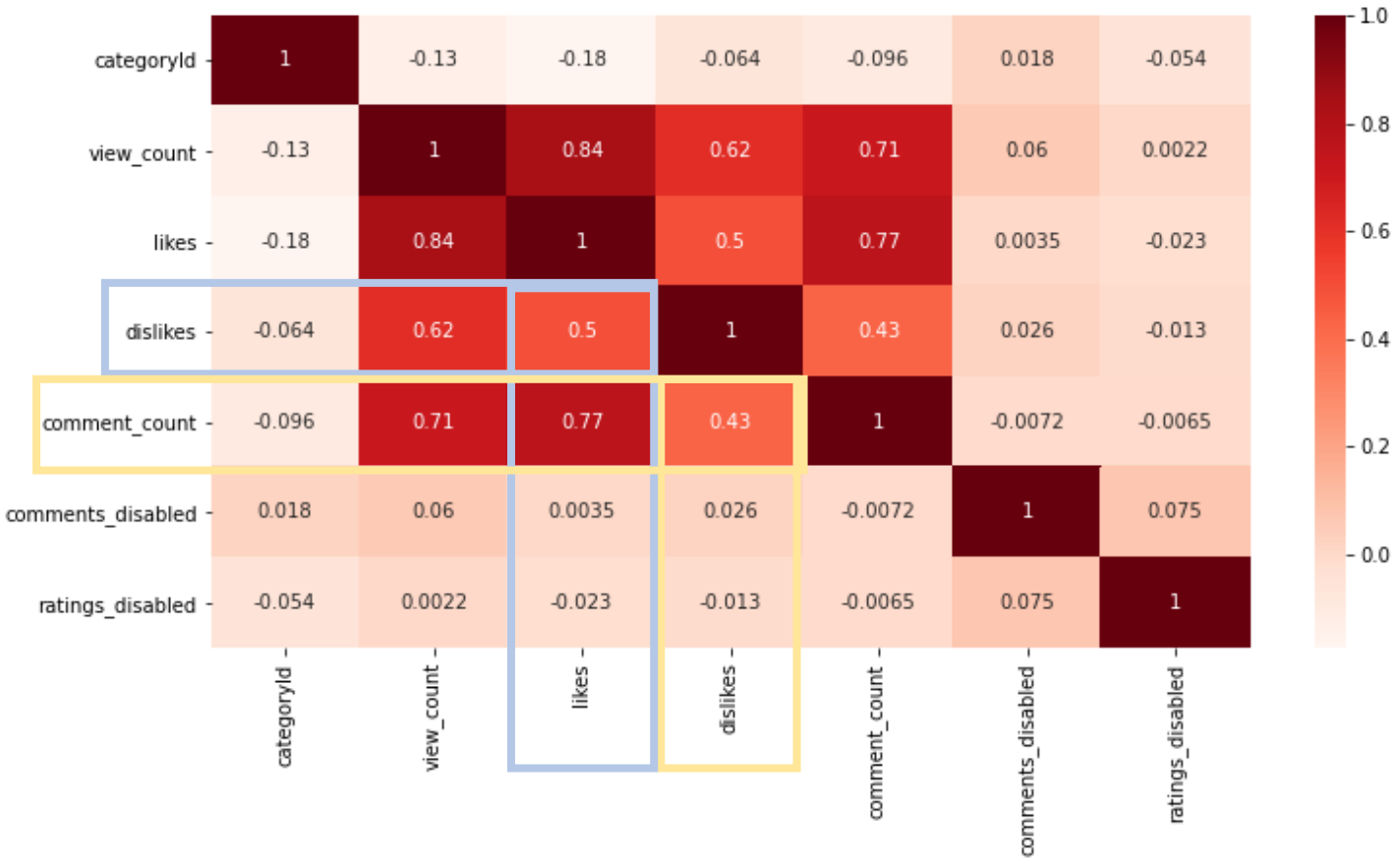
→ 인기 동영상의 절반은 조회수가 중앙값보다 낮다.

- 인기 동영상의 평균 좋아요수: 7.745969e+04개
- 인기 동영상의 평균 싫어요수: 1354.628067개
- 댓글 수의 평균: 1.165818e+04개
- 댓글 수의 중앙값: 1.384000e+03개

youtube.com

YouTube KR

2. EDA 분석_ 상관계수 *(heatmap)*



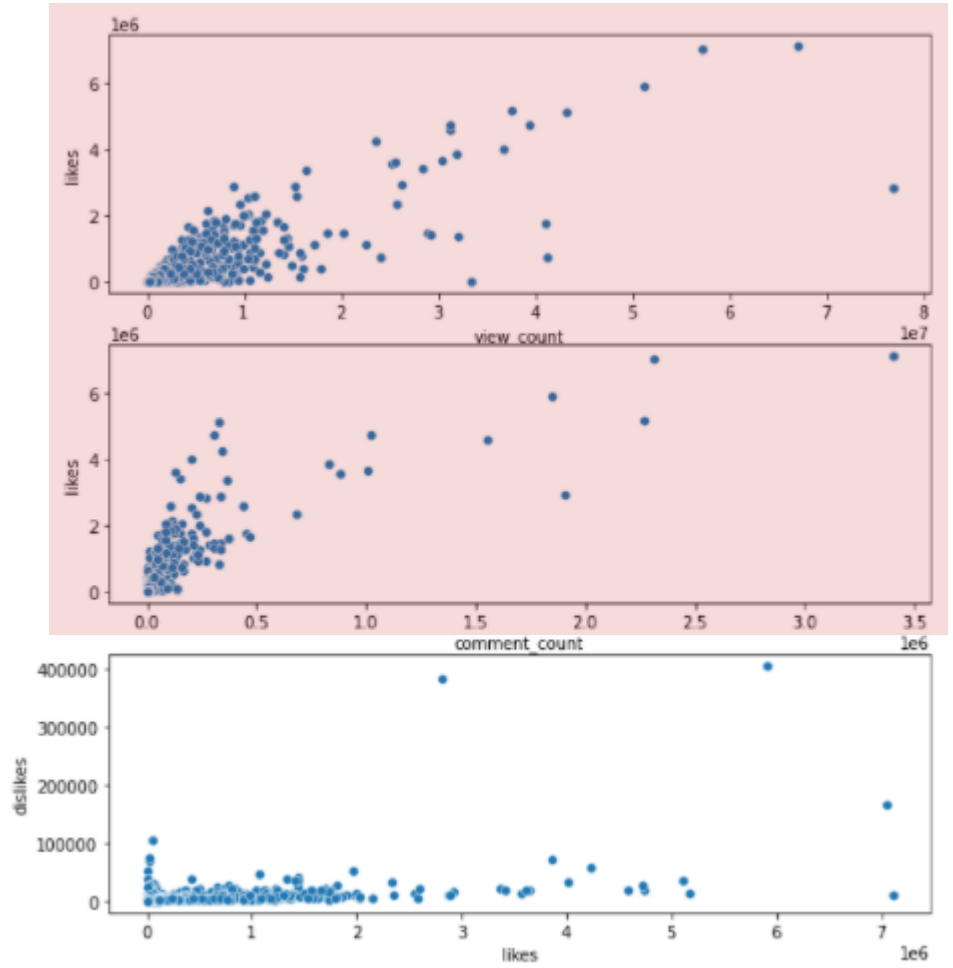
좋아요 / 싫어요: 0.5

싫어요 / 댓글 수: 0.43

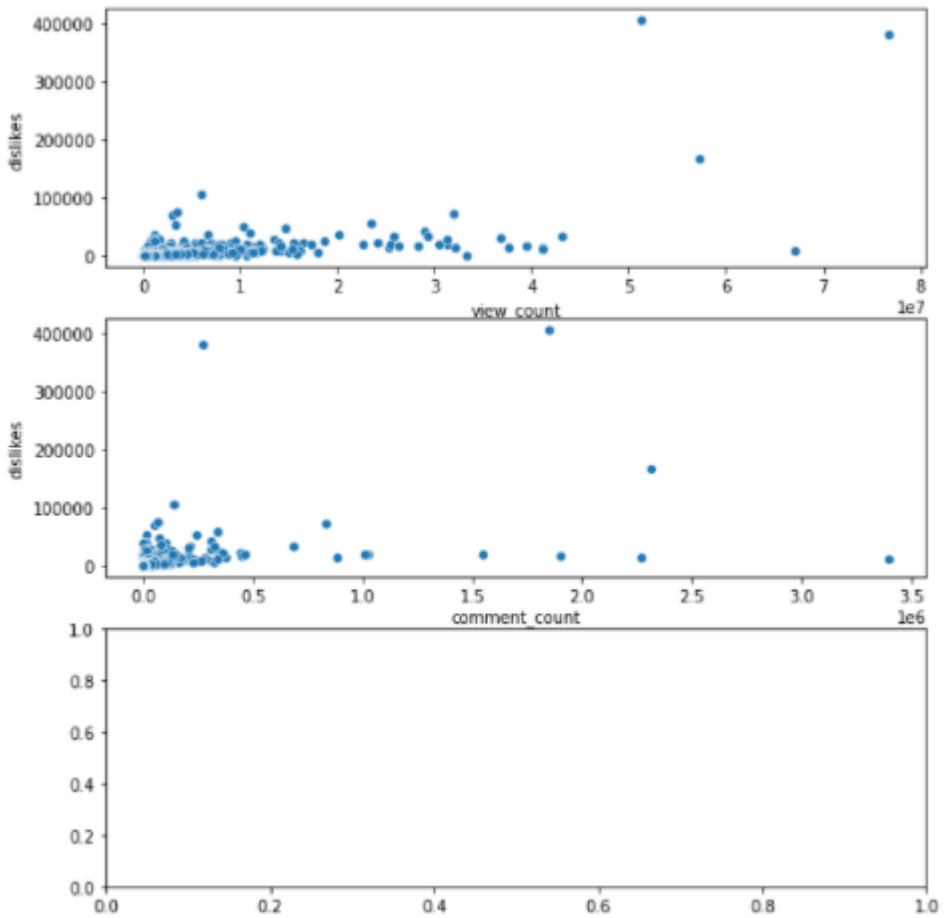
무의미

youtube.com

2. EDA 분석_ 상관계수 산점도(*scatter plot*)



‘좋아요’는 높은 상관도를 보임

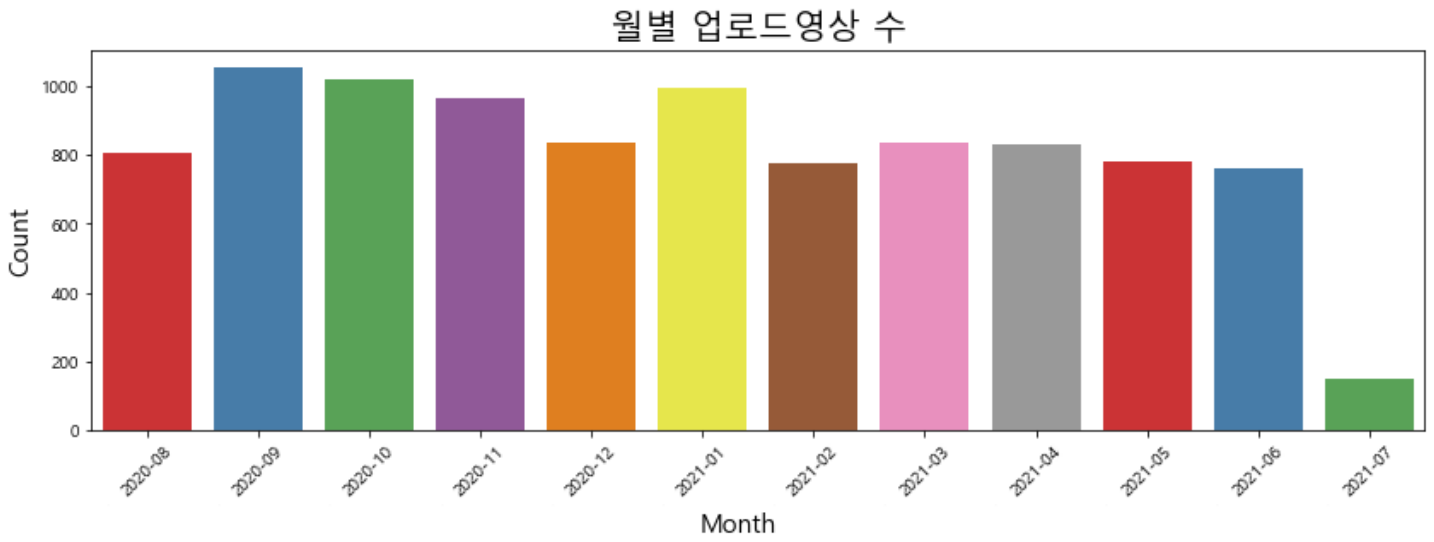


‘싫어요’는 낮은 상관도를 보임

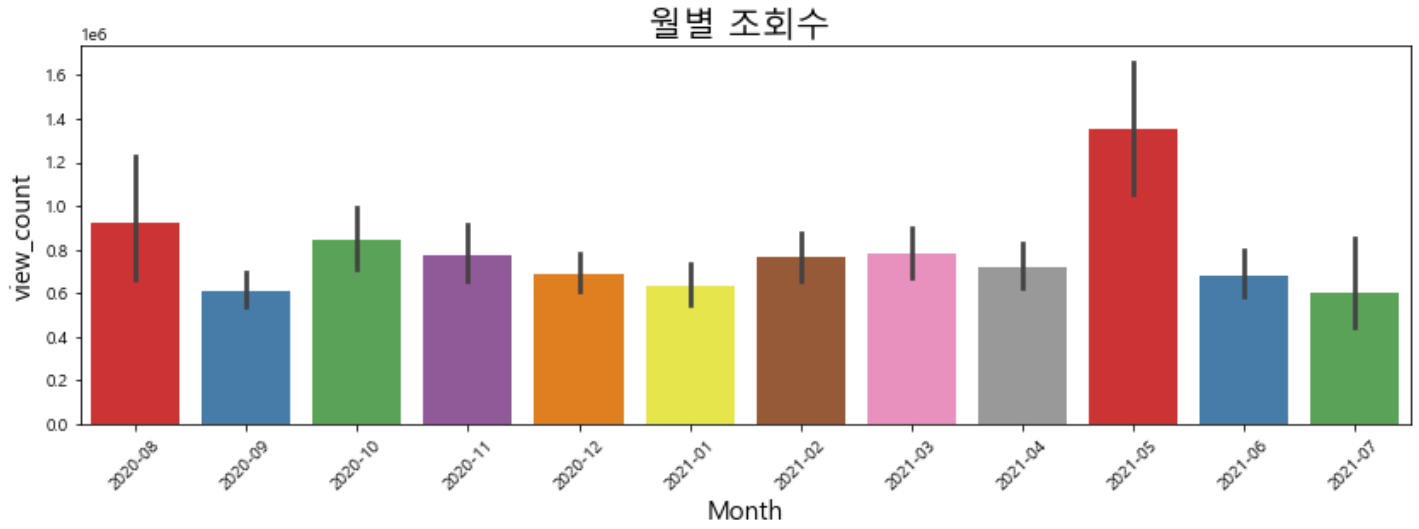
youtube.com

YouTube KR

2. EDA 분석_ 월 별 업로드와 조회수



가장 높은 업로드 횟수:
2020년 9월



가장 높은 조회수:
2021년 5월

youtube.com

YouTube KR

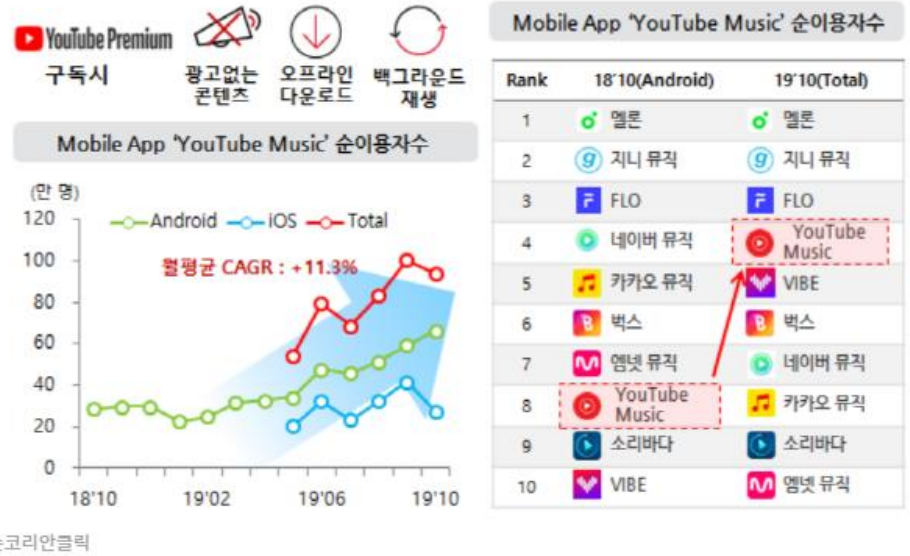
2. EDA 분석_ 월 별 업로드와 조회수의 분석

YouTube MUSIC

멜론·지니·플로 출혈경쟁…'유튜브뮤직' 뜬다

유튜브뮤직 순이용자수 4위, 전년 대비 4단계 올라

지니 월 100원 등 음원시장 가격 경쟁 치열…"차별화 서비스 마련해야"



Mobile App 'YouTube Music' 순이용자수

Rank	18'10(Android)	19'10(Total)
1	멜론	멜론
2	지니 뮤직	지니 뮤직
3	FLO	FLO
4	네이버 뮤직	YouTube Music
5	카카오 뮤직	VIBE
6	벅스	벅스
7	엠넷 뮤직	네이버 뮤직
8	YouTube Music	카카오 뮤직
9	소리바다	소리바다
10	VIBE	엠넷 뮤직

-출처: EBN. 황준익 기자 (2019.11.27)

방탄소년단 신곡 '버터', 20시간 만에 조회수 1억뷰 '신기록'

방탄소년단(BTS)의 신곡 '버터' 뮤직비디오가 공개된 지 21시간 만에 유튜브에서 조회수 1억건을 넘겼다. 공개 24시간 만에 1억뷰를 돌파했던 '다이너마이트'보다 3시간 가량 앞서 자체 기록을 경신한 것이다.

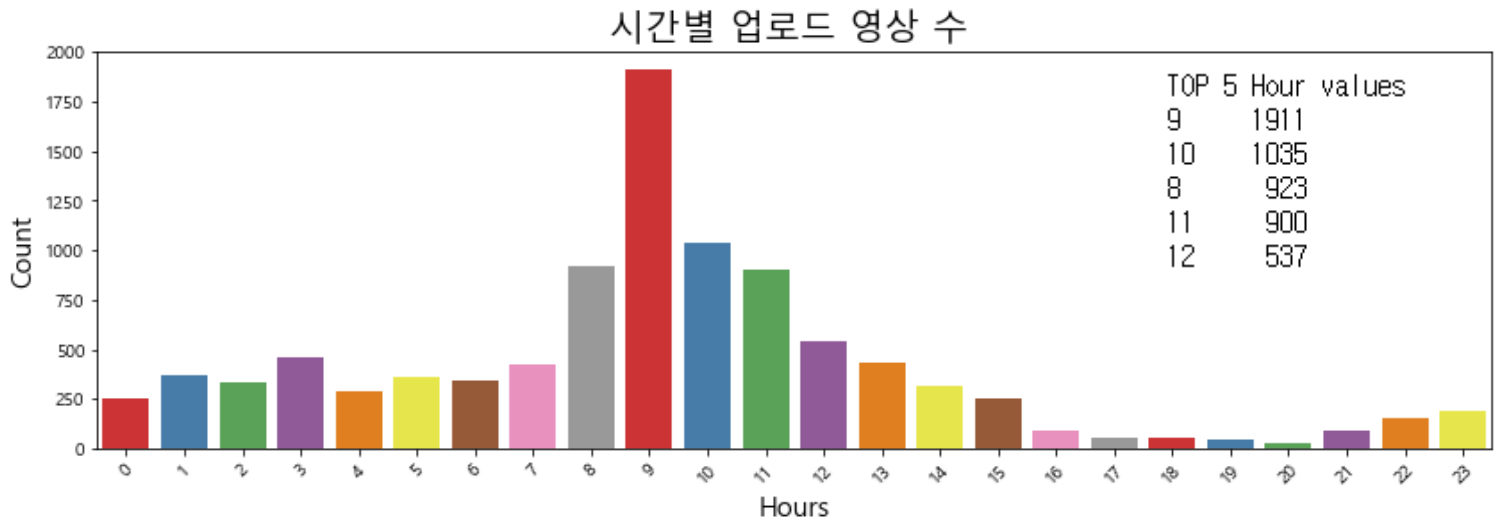
-출처: 조선비즈. 연선옥 기자 (2021.05.22)

에스파, 오늘 '넥스트 레벨'로 컴백…'분노의 질주' 재해석

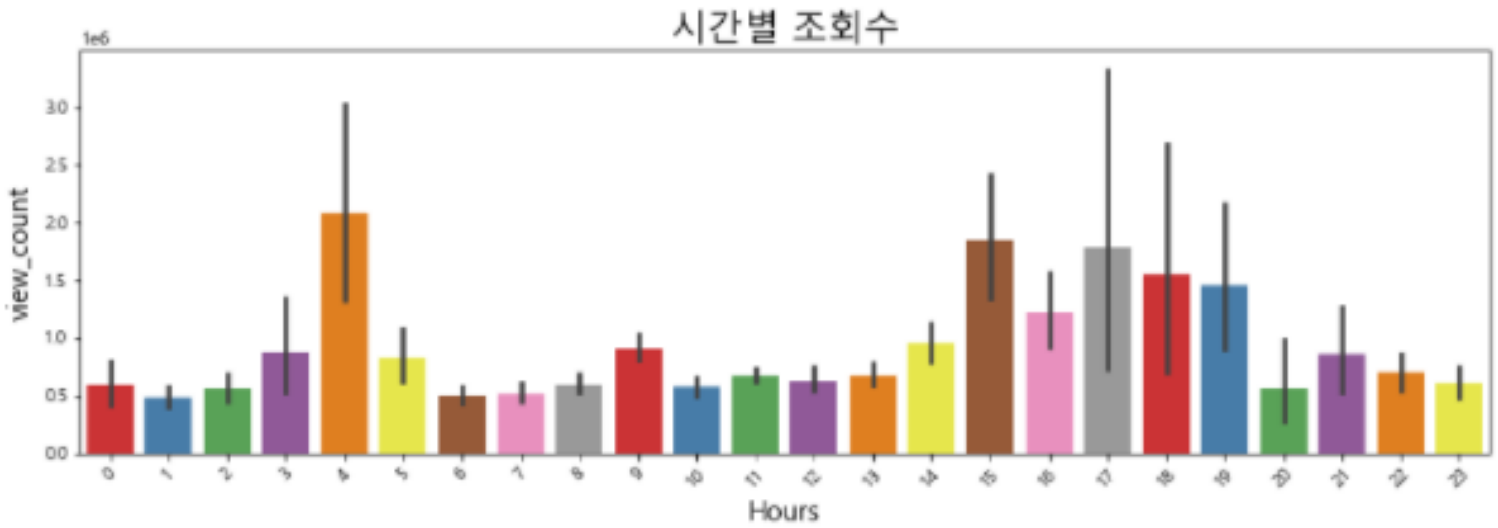
-출처: 동아일보. 뉴시스 기자 (2021.05.17)

youtube.com

2. EDA 분석_ 시간 별 업로드와 조회수



가장 많은 영상 업로드 시간:
오전 9시



가장 조회수가 많은 시간:
오전 4시

youtube.com

YouTube KR

2. EDA 분석_ 각 항목 별 Top10 도출

인기 동영상 지정 횟수, 기간을 포함한 csv

	video_id	trending_count	days_trend	first_trend_date	last_trend_date	views	likes	dislikes	comments
0	--8IwTcvN98	9	9	2020-12-19	2020-12-27	565697	19575	260	3997
1	--MvOWR16L8	7	7	2021-01-24	2021-01-30	345912	9678	415	1689
2	--ixRiohjh4	2	2	2020-09-02	2020-09-03	76183	1407	113	91
3	-0dbirUY_uk	5	5	2021-07-02	2021-07-07	2078098	287028	1138	10838
4	-1HbFmXHKWs	9	9	2021-01-02	2021-01-10	1249804	27828	353	7630

Category ID

1 영화/애니메이션
2 자동차/ 교통
10 음악
15 애완동물/동물
17 스포츠
19 여행/이벤트
20 게임
22 인물/블로그
23 코미디
24 엔터테인먼트
25 뉴스/정치
26 노하우/스타일
27 교육
28 과학기술
29 비영리/사회운동

youtube.com

2. EDA 분석_ 각 항목 별 Top10 도출

영상 조회수 Top10

	title	channelTitle	categoryId	views
5110	BTS (방탄소년단) 'Butter' Official MV	HYBE LABELS	10	296314174
6788	BTS (방탄소년단) 'Dynamite' Official MV	Big Hit Labels	10	262319276
9084	BLACKPINK - 'Ice Cream (with Selena Gomez)' M/V	BLACKPINK	10	184778248
15	BTS (방탄소년단) 'Life Goes On' Official MV	Big Hit Labels	10	161912058
6385	BLACKPINK – 'Lovesick Girls' M/V	BLACKPINK	10	161416953
2010	ROSÉ - 'On The Ground' M/V	BLACKPINK	10	117461115
5224	TWICE Alcohol-Free M/V	JYP Entertainment	10	104119672
2014	TWICE I CAN'T STOP ME M/V	JYP Entertainment	10	99382749
853	aespa 에스파 'Next Level' MV	SMTOWN	10	84077957
6979	Cardi B - WAP feat. Megan Thee Stallion [Offic...	Cardi B	10	82765322

좋아요 Top10

	title	channelTitle	categoryId	likes
5110	BTS (방탄소년단) 'Butter' Official MV	HYBE LABELS	10	16464253
6788	BTS (방탄소년단) 'Dynamite' Official MV	Big Hit Labels	10	16254784
9084	BLACKPINK - 'Ice Cream (with Selena Gomez)' M/V	BLACKPINK	10	11795696
15	BTS (방탄소년단) 'Life Goes On' Official MV	Big Hit Labels	10	11650994
6385	BLACKPINK – 'Lovesick Girls' M/V	BLACKPINK	10	9537518
2010	ROSÉ - 'On The Ground' M/V	BLACKPINK	10	7508707
9708	BTS (방탄소년단) 'Film out' Official MV	HYBE LABELS	10	7137434
8120	BTS (방탄소년단) 'Dynamite' Official Teaser	Big Hit Labels	10	6185031
1893	BTS (방탄소년단) 'Dynamite' Official MV (B-side)	Big Hit Labels	10	6032032
3917	BTS (방탄소년단) 'Butter' Official Teaser	HYBE LABELS	10	5696952

싫어요 Top10

	title	channelTitle	categoryId	dislikes
9084	BLACKPINK - 'Ice Cream (with Selena Gomez)' M/V	BLACKPINK	10	879358
6788	BTS (방탄소년단) 'Dynamite' Official MV	Big Hit Labels	10	770144
6979	Cardi B - WAP feat. Megan Thee Stallion [Offic...	Cardi B	10	408064
5110	BTS (방탄소년단) 'Butter' Official MV	HYBE LABELS	10	181350
8120	BTS (방탄소년단) 'Dynamite' Official Teaser	Big Hit Labels	10	159659
6385	BLACKPINK – 'Lovesick Girls' M/V	BLACKPINK	10	148435
15	BTS (방탄소년단) 'Life Goes On' Official MV	Big Hit Labels	10	141033
8979	안녕하세요 보겸입니다	보겸 BK	24	115494
2010	ROSÉ - 'On The Ground' M/V	BLACKPINK	10	109304
1893	BTS (방탄소년단) 'Dynamite' Official MV (B-side)	Big Hit Labels	10	100054

댓글 수 Top10

	title	channelTitle	categoryId	comments
5110	BTS (방탄소년단) 'Butter' Official MV	HYBE LABELS	10	6939302
6788	BTS (방탄소년단) 'Dynamite' Official MV	Big Hit Labels	10	6303708
15	BTS (방탄소년단) 'Life Goes On' Official MV	Big Hit Labels	10	4225989
520	EXO 엑소 'Don't fight the feeling' MV	SMTOWN	10	3119817
9084	BLACKPINK - 'Ice Cream (with Selena Gomez)' M/V	BLACKPINK	10	2735999
9708	BTS (방탄소년단) 'Film out' Official MV	HYBE LABELS	10	2240915
2010	ROSÉ - 'On The Ground' M/V	BLACKPINK	10	1610298
6385	BLACKPINK – 'Lovesick Girls' M/V	BLACKPINK	10	1541326
3917	BTS (방탄소년단) 'Butter' Official Teaser	HYBE LABELS	10	1177546
5224	TWICE Alcohol-Free M/V	JYP Entertainment	10	1070335

- **공통적으로 보이는 채널:**

HYBE LABELS,

Big Hit Labels,

BLACKPINK,

JYP Entertainment,

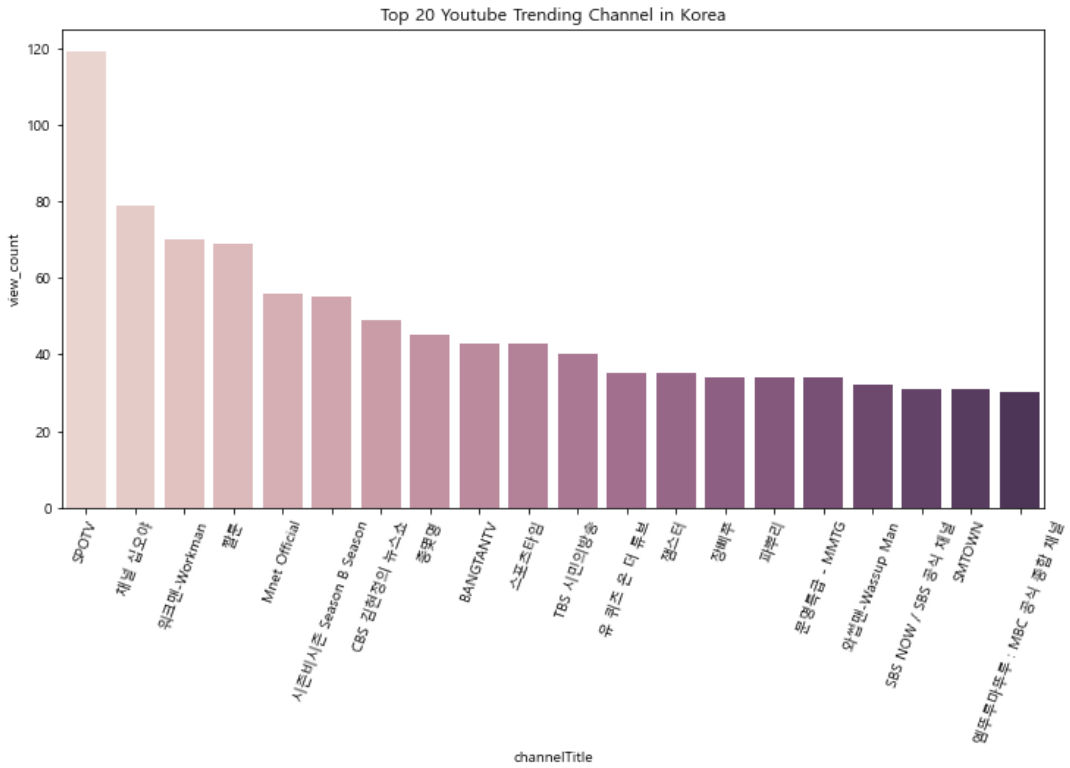
SMTOWN

youtube.com

YouTube KR

2. EDA 분석_ 각 항목 별 Top10 도출 및 분석

한국의 인기 있는 채널 Top20

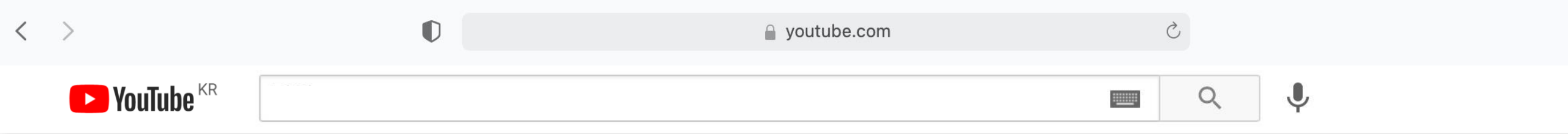


인기 동영상 지정 횟수 Top10

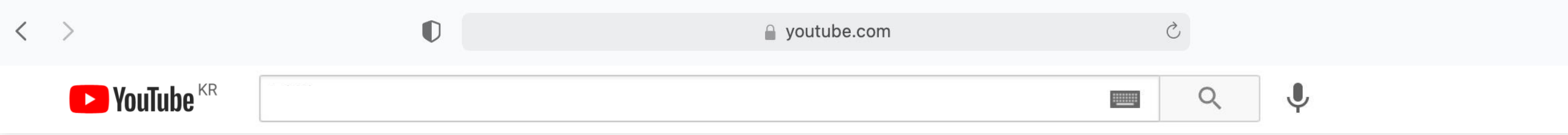
	title	channelTitle	categoryId	trending_count
3191	대한민국 VS 투르크메니스탄 : FIFA 카타르 월드컵 2차 예선 하이라이트 - 2...	KFATV_한국 축구 국가대표팀	17	24
1926	컵라면 먹을 때 절대 일어날리 없는 상황들 ㅋㅋㅋ	웃소 Wootso	23	24
2627	칼로리 계산 킹받네..?!ㅋㅋㅋ ▶엔조이 커플 고칼로리 먹방 VS 비 헬스💪 I 시즌비...	시즌비시즌 Season B Season	24	23
2628	[EN] 칼로리 계산 킹받네..?!ㅋㅋㅋ ▶엔조이 커플 고칼로리 먹방 VS 비 헬스...	시즌비시즌 Season B Season	24	23
6618	2주더놀릴예정	런닝맨 - 스브스 공식 채널	24	23
6570	대박... 멧돼지를 아들로 삼은 할아버지	KBS 주주클럽 050410 방송	KBS동물티비 : 애니멀포유 animal4u	22	23
1186	극과 극을 살고 있는 인도의 갑부와 빈곤층들!	지식한입	24	23
1652	[진돗개 진솔쓰] 차박 일기. #Shorts	진솔쓰Jinsolss	15	22
6160	[짐승친구들] 예비군	짤툰	1	22
4574	밤낮으로 우리집 비번 눌러대는 길고양이 좀 어떻게 해주세요 I Stray Cat Ent...	SBS TV동물농장x애니멀봐	15	22

인기 있는 채널, 지정 횟수

Top1 모두 스포츠 컨텐츠



3. 모델 구축 및 분석



3-1. 한국

YouTube 영상의 긍/부정 모델 구축

youtube.com

YouTube KR

3-1. 한국 YouTube_ 전처리

	view_count	likes	dislikes	comment_count	tags	time	categoryId	comments_disabled	ratings_disabled
0	5947503	53326	105756	139946	보겸|bokyem	09:32:48	24	False	False
1	963384	28244	494	3339	흥몃명|재밌는 만화|부락토스|루시퍼|흥몃명 프리퀄|흥몃명 스토리	09:00:08	1	False	False
2	2950885	17974	68898	50688	양팡|양팡유튜브|팡튜브|가족시트콤|양팡가족|양팡가족시트콤|양팡언니|현실남매|현실자매...	09:54:13	22	False	False
3	1743374	36893	1798	8751	꽈뚜룹|한국여행기|quaddurup|뚜룹이|korea|southkorea|vlog|...	15:00:58	24	False	False
4	3433885	353337	9763	23405	JYP Entertainment|JYP|J.Y.Park|JYPark|박진영|선미|S...	09:00:13	10	False	False
...	...	...	...	...	...	...	...	...	...
63994	132759	3499	54	785	올티|olltii|마이크스웨거|micswagger|밸런스게임|balancegame|...	09:00:15	10	False	False
63995	413747	8303	112	1506	하알라|코미디|코믹|개그|아프리카TV|아프리카|가족시트콤|엽캠|하선우|모델|야방|B...	10:21:39	24	False	False
63996	484332	22515	186	1905	[None]	10:56:45	22	False	False
63997	251454	6489	181	1298	귀촌|귀촌일기|힐링|시골영상|자연|서울 부부의 귀촌일기|시골 풍경|시골 경치|귀촌 ...	04:24:07	22	False	False
64033	193005	2747	37	223	트러블러|트래블러|이용진|이진호|랄랄|이상준|나이50살|20학번|첫엠티|첫MT|녹대...	09:00:15	24	False	False

9816 rows × 9 columns

	view_count	likes	dislikes	comment_count	categoryId	comments_block	ratings_block	pub_time	tag_yn
0	5947503	53326	105756	139946	24	0	0	9	1
1	963384	28244	494	3339	1	0	0	9	1
2	2950885	17974	68898	50688	22	0	0	9	1
3	1743374	36893	1798	8751	24	0	0	15	1
4	3433885	353337	9763	23405	10	0	0	9	1
...	...	...	...	...	...	...	...	...	...
63779	848777	13868	254	2502	22	0	0	9	0
63795	333929	11619	126	2166	22	0	0	9	0
63981	121362	6122	39	254	24	0	0	10	0
63987	113794	2486	56	675	24	0	0	6	0
63996	484332	22515	186	1905	22	0	0	10	0

9680 rows × 9 columns

- 댓글 막기, 평가 막기
  → 범주 0,1 변환
- 출간 시간대 변수 변환
- 태그 유무 전처리
- 좋아요, 싫어요 값 0 제거
- 필요한 변수만 추출

YouTube KR

3-1. 한국 YouTube_ 전처리

Rate = likes / dislikes

	view_count	likes	dislikes	comment_count	categoryId	comments_block	ratings_block	pub_time	tag_yn	target	real_target	rate
0	5947503	53326	105756	139946	24	0	0	9	1	True	1	0.504236
1	963384	28244	494	3339	1	0	0	9	1	False	0	57.174089
2	2950885	17974	68898	50688	22	0	0	9	1	True	1	0.260878
3	1743374	36893	1798	8751	24	0	0	15	1	False	0	20.518910
4	3433885	353337	9763	23405	10	0	0	9	1	False	0	36.191437
...	...	...	...	...	...	...	...	...	...	...	...	...
63779	848777	13868	254	2502	22	0	0	9	0	False	0	54.598425
63795	333929	11619	126	2166	22	0	0	9	0	False	0	92.214286
63981	121362	6122	39	254	24	0	0	10	0	False	0	156.974359
63987	113794	2486	56	675	24	0	0	6	0	False	0	44.392857
63996	484332	22515	186	1905	22	0	0	10	0	False	0	121.048387

9680 rows × 12 columns

youtube.com

3-1. 한국 YouTube_ 전처리

- 긍정: 94 이상
- 보통: 3.5이상 94미만
- 부정: 3.5미만

```
data6.rate.describe()
```

```
count    9680.000000
mean       94.058110
std       139.575934
min         0.132710
25%        29.249581
50%        50.112994
75%        94.000000
max      1910.400000
Name: rate, dtype: float64
```

Target

보통	7173
긍정	2421
부정	86

```
data7 = data6[data6.rate>=94]
```

```
data7['Target'] = '긍정'
```

```
<ipython-input-32-6ddd65f5915f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data7['Target'] = '긍정'
```

```
data8 = data6[(data6.rate < 94) & (data6.rate>=3.5)]
```

```
data8['Target'] = '보통'
```

```
<ipython-input-35-1a834e072575>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data8['Target'] = '보통'
```

```
data9 = data6[data6.rate < 3.5]
```

```
data9['Target']='부정'
```

```
<ipython-input-39-aa96bfc9c773>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data9['Target']='부정'
```

```
data7.shape,data8.shape,data9.shape
```

```
((2421, 13), (7173, 13), (86, 13))
```

youtube.com

YouTube KR

3-1. 한국 YouTube_ 머신러닝

머신러닝 분석 - 3:1로 train, test 분리

```
X_train.shape,X_test.shape
```

```
((7260, 7), (2420, 7))
```

로지스틱 회귀 분석

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:₩Users₩CPBO6GameN₩anaconda3₩lib₩site-packages₩sklearn₩utils₩validat
or y was passed when a 1d array was expected. Please change the shape
().
  return f(**kwargs)
(0.7454545454545455, 0.7590909090909091)
```

예측값

```
su = lr.predict(X_test[1000:2000])
```

```
su
```

```
array(['보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '긍정', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
```

실제값

```
np.array(y_test[1000:2000]).flatten()
```

```
array(['긍정', '긍정', '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '긍정', '긍정', '보통',
       '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '긍정',
       '긍정', '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '긍정', '긍정',
       '보통', '보통', '보통', '긍정', '긍정', '보통', '보통', '보통', '보통', '보통', '보통', '긍정',
       '보통', '긍정', '보통', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '긍정', '보통',
       '보통', '긍정', '긍정', '보통', '긍정', '보통', '보통', '보통', '보통', '보통', '긍정', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '긍정', '보통', '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통', '보통', '보통',
       '보통', '보통', '보통', '보통', '보통', '보통', '긍정', '긍정', '보통', '보통', '보통', '보통',
       '보통', '보통', '긍정', '보통', '보통', '보통', '보통', '보통', '긍정', '보통', '긍정', '보통',
```

예측값과 실제값 맞춘 값: 755

youtube.com

YouTube KR

3-1. 한국 YouTube_ 머신러닝

최근접 분류

```
from sklearn.neighbors import KNeighborsClassifier
for i in range(1,500):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    knn.score(X_test, y_test)
    print(knn.score(X_test, y_test),i)
    i+=1
```

```
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
0.6570247933884298 1
0.5958677685950413 2
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
0.7016528925619835 3
0.6776859504132231 4
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
<ipython-input-124-ba8ca5dbdd15>:3: DataConversionWarnin
d. Please change the shape of y to (n_samples, ), for ex
  knn.fit(X_train,y_train)
0.7260330578512396 5
0.7144628099173553 6
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=229)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-129-a1bd534a785b>:3: DataConversionWarning
d. Please change the shape of y to (n_samples, ), for exa
  knn.fit(X_train,y_train)
(0.7411845730027549, 0.753305785123967)
```

229의 수치가 가장 비율이 높음

Train의 정확도가 test의 정확도 보다 높기 때문에 부적합 하다고 판단

youtube.com

YouTube KR

3-1. 한국 YouTube_ 머신러닝

의사결정 나무

```
for i in range(0,500):
    tree = DecisionTreeClassifier(random_state=i)
    tree.fit(X_train, y_train)
    tree.score(X_test, y_test)
    print(tree.score(X_train, y_train),tree.score(X_test, y_test),i,tree.score(X_test, y_test)>0.737)
    i+=1
```

```
1.0 0.721900826446281 0 False
1.0 0.7214876033057851 1 False
1.0 0.7235537190082645 2 False
1.0 0.7268595041322314 3 False
1.0 0.7264462809917356 4 False
1.0 0.7338842975206612 5 False
1.0 0.7227272727272728 6 False
1.0 0.7285123966942149 7 False
1.0 0.7243801652892562 8 False
1.0 0.7272727272727273 9 False
1.0 0.7301652892561984 10 False
1.0 0.7264462809917356 11 False
1.0 0.7285123966942149 12 False
1.0 0.7347107438016529 13 False
```

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=283)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

(1.0, 0.737603305785124)

Train의 정확도가 100%의 성능을 보임 -
과적합, 분석하기에 부적합하다고 판단

3-1. 한국 YouTube_ 머신러닝

랜덤 포레스트

```
for i in range(1,500):
    forest = RandomForestClassifier(n_estimators=i,random_state=3)
    forest.fit(X_train,y_train)
    forest.score(X_train, y_train) , forest.score(X_test, y_test)
    print(forest.score(X_train, y_train),forest.score(X_test, y_test),i)
    i+=1
    #70,3
```

```
<ipython-input-135-7f5d950b6fa8>:3: DataConversionWarning: A column-vector y
d. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
<ipython-input-135-7f5d950b6fa8>:3: DataConversionWarning: A column-vector y
d. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
<ipython-input-135-7f5d950b6fa8>:3: DataConversionWarning: A column-vector y
d. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
<ipython-input-135-7f5d950b6fa8>:3: DataConversionWarning: A column-vector y
d. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
<ipython-input-135-7f5d950b6fa8>:3: DataConversionWarning: A column-vector y
d. Please change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
0.8899449035812672 0.7177685950413223 1
0.8826446280991735 0.6619834710743802 2
0.9475206611570248 0.7442148760330578 3
0.95 0.7363636363636363 4
0.9665289256198347 0.7706611570247934 5
0.971625344352617 0.762809917355371 8 6
0.977961432506887 0.7768595041322314 7
```

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=70,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-136-5c093a678eb4>:3: DataConversionWarning: A column
d. Please change the shape of y to (n_samples,), for example using
  forest.fit(X_train,y_train)
(0.9995867768595041, 0.8024793388429752)
```

Train의 정확도가 99%의 성능을 보임 -
과적합, 분석하기에 부적합하다고 판단

youtube.com

YouTube KR

3-1. 한국 YouTube_ 머신러닝

로지스틱 회귀 분석

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:₩Users₩CPB06GameN₩anaconda3₩lib₩site-packages₩sklearn₩utils₩validat
or y was passed when a 1d array was expected. Please change the shape
().
  return f(**kwargs)
(0.7454545454545455, 0.7590909090909091)
```

KNN 모델

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=229)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-129-a1bd534a785b>:3: DataConversionWarning
d. Please change the shape of y to (n_samples, ), for exa
  knn.fit(X_train,y_train)
(0.7411845730027549, 0.7533057851239674)
```

의사 결정 나무

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=283)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

```
(1.0, 0.737603305785124)
```

랜덤 포레스트

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=70,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-136-5c093a678eb4>:3: DataConversionWarning: A column
d. Please change the shape of y to (n_samples,), for example using
  forest.fit(X_train,y_train)
(0.9995867768595041, 0.8024793388429752)
```

youtube.com

YouTube KR

3-1. 한국 YouTube_ 미국 + 한국 YouTube 모델 구축

로지스틱 회귀 분석

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:₩Users₩CPBO6GameN₩anaconda3₩lib₩site-packages₩sklearn₩utils₩validat
or y was passed when a 1d array was expected. Please change the shape
().
  return f(**kwargs)
(0.7454545454545455, 0.759090909090909)
```

KNN 모델

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=229)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-129-a1bd534a785b>:3: DataConversionWarning
d. Please change the shape of y to (n_samples, ), for exa
  knn.fit(X_train,y_train)
(0.7411845730027549, 0.753305785123967)
```

의사 결정 나무

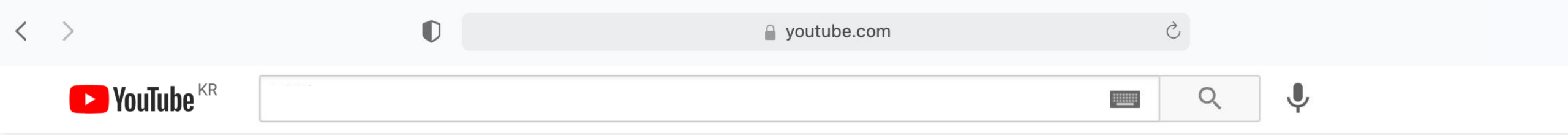
```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=283)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

```
(1.0, 0.737603305785124)
```

랜덤 포레스트

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=70,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-136-5c093a678eb4>:3: DataConversionWarning: A column
d. Please change the shape of y to (n_samples,), for example using
  forest.fit(X_train,y_train)
(0.9995867768595041, 0.8024793388429752)
```



3-2. 일본, 미국

YouTube 영상의 긍/부정 모델 구축

3-2. 일본 YouTube 긍·부정 모델 분석_ 전처리

한국 유튜브 전처리 방식과 동일한 순서로 진행

```
data6.rate.describe()
```

```
count    11945.000000
mean        39.632377
std         82.652137
min          0.015493
25%          5.908397
50%         15.000000
75%         38.888889
max       1935.000000
Name: rate, dtype: float64
```

```
data10.Target.value_counts()
```

```
보통    7097
긍정    3040
부정    1808
Name: Target, dtype: int64
```

```
data7 = data6[data6.rate>=38]
```

```
data7['Target'] = '긍정'
```

```
<ipython-input-31-6ddd65f5915f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data7['Target'] = '긍정'
```

```
data8 = data6[(data6.rate < 38) & (data6.rate>=3.5)]
```

```
data8['Target'] = '보통'
```

```
<ipython-input-33-1a834e072575>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data8['Target'] = '보통'
```

```
data9 = data6[data6.rate < 3.5]
```

```
data9['Target']='부정'
```

```
<ipython-input-35-aa96bfc9c773>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data9['Target']='부정'
```

youtube.com

3-2. 일본 YouTube 긍·부정 모델 분석_ 전처리

로지스틱 회귀 분석

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:\Users\CPB06GameN\anaconda3\lib\site-packages\sklearn\u
passed when a 1d array was expected. Please change the sh
  return f(**kwargs)
(0.5951105157401205, 0.6036156678942083)
```

KNN 모델

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=229)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-53-a1bd534a785b>:3: DataConversionWarning:
hange the shape of y to (n_samples, ), for example using
  knn.fit(X_train,y_train)
(0.5916499218575575, 0.601606963508537)
```

의사 결정 나무

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=283)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

```
(1.0, 0.547706729159692)
```

랜덤 포레스트

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=70,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-55-5c093a678eb4>:3: DataConversionWarning: A colum
hange the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
(0.9997767358785443, 0.6360897221292267)
```

도출된 값의 정확도가 60%내의 성능을 보임 - 한국 csv 의 파일과 합쳐서 분석하기에 부적합하다고 판단

3-2. 미국 YouTube 긍·부정 모델 분석_ 전처리

한국 유튜브 전처리 방식과 동일한 순서로 진행

```
data6.rate.describe()
```

```
count    6344.000000
mean       51.282600
std        62.528626
min         0.041583
25%        13.635737
50%        32.653130
75%        67.413635
max      1737.333333
Name: rate, dtype: float64
```

```
data10.Target.value_counts()
```

```
보통    4311
긍정    1607
부정     426
Name: Target, dtype: int64
```

```
data7 = data6[data6.rate>=67]
```

```
data7['Target'] = '긍정'
```

```
<ipython-input-31-6ddd65f5915f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data7['Target'] = '긍정'
```

```
data8 = data6[(data6.rate < 67) & (data6.rate>=3.5)]
```

```
data8['Target'] = '보통'
```

```
<ipython-input-33-1a834e072575>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data8['Target'] = '보통'
```

```
data9 = data6[data6.rate < 3.5]
```

```
data9['Target']='부정'
```

```
<ipython-input-35-aa96bfc9c773>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data9['Target']='부정'
```

youtube.com

YouTube KR

3-2. 미국 유튜브 긍·부정 모델 분석

로지스틱 회귀 분석

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:\Users\CPB06GameN\anaconda3\lib\site-packages\sklearn
passed when a 1d array was expected. Please change the
  return f(**kwargs)
```

(0.6782261454392602, 0.6885245901639344)

KNN 모델

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=229)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-49-a1bd534a785b>:3: DataConversionWarning
hange the shape of y to (n_samples, ), for example using
  knn.fit(X_train,y_train)
```

(0.6763345943673813, 0.6891551071878941)

의사 결정 나무

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=283)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

(1.0, 0.6134930643127364)

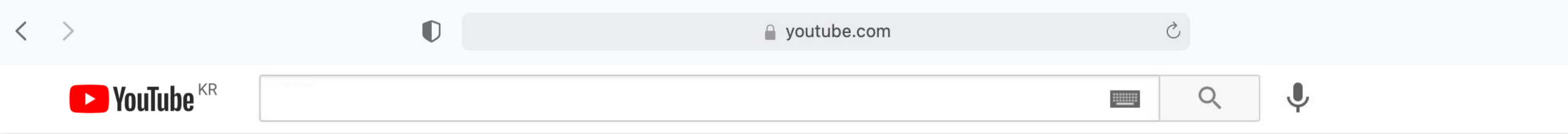
랜덤 포레스트

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=70,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-51-5c093a678eb4>:3: DataConversionWarning: A colum
hange the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
```

(0.9995796553173603, 0.6784363177805801)

도출된 값의 정확도가 70%내의 성능을 보임 – 한국 csv 의 파일과 합쳐서 분석하기에 합당하다고 판단



3-3. 한국 + 미국

YouTube 영상의 긍/부정 모델 구축

3-3. 미국 + 한국 유튜브 모델 구축

미국 + 한국 유튜브 모델 구축

```
df = pd.concat([data_kr, data_am])
```

```
df.Target.value_counts()
```

```
보통    11484
긍정     4028
부정      512
Name: Target, dtype: int64
```

```
X_train.shape,X_test.shape
```

```
((12018, 7), (4006, 7))
```

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:\Users\CPB06GameN\anaconda3\lib\site-packages\sklear
passed when a 1d array was expected. Please change the
  return f(**kwargs)
(0.7170910301214845, 0.7221667498751873)
```

```
from sklearn.neighbors import KNeighborsClassifier
list,a = [],[]
for i in range(1,100):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    knn.score(X_test, y_test)
    print(knn.score(X_test, y_test),i)
    list.append(knn.score(X_test, y_test))
    a.append(i)
    i+=1
nei = pd.concat([pd.DataFrame(np.round(list,4),columns=['값']),pd.DataFrame(a,columns=['순서'])],axis=1)
```

```
<ipython-input-82-1ce69e506531>:5: DataConversionWarning: A column-vector y was passed when a 1d array was
hange the shape of y to (n_samples, ), for example using ravel().
  knn.fit(X_train,y_train)
0.6225661507738393 1
<ipython-input-82-1ce69e506531>:5: DataConversionWarning: A column-vector y was passed when a 1d array was
hange the shape of y to (n_samples, ), for example using ravel().
  knn.fit(X_train,y_train)
0.5639041437843235 2
<ipython-input-82-1ce69e506531>:5: DataConversionWarning: A column-vector y was passed when a 1d array was
hange the shape of y to (n_samples, ), for example using ravel().
  knn.fit(X_train,y_train)
0.6645032451323015 3
<ipython-input-82-1ce69e506531>:5: DataConversionWarning: A column-vector y was passed when a 1d array was
hange the shape of y to (n_samples, ), for example using ravel().
  knn.fit(X_train,y_train)
0.6385421867199201 4
<ipython-input-82-1ce69e506531>:5: DataConversionWarning: A column-vector y was passed when a 1d array was
hange the shape of y to (n_samples, ), for example using ravel().
  knn.fit(X_train,y_train)
0.6834747878182726 5
```

youtube.com

YouTube KR

3-3. 미국 + 한국 유튜브 모델 구축

로지스틱 회귀 분석

```
X_train.shape,X_test.shape
```

```
((12018, 7), (4006, 7))
```

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr.score(X_train, y_train),lr.score(X_test, y_test)
```

```
C:\Users\CPB06GameN\anaconda3\lib\site-packages\sklear
passed when a 1d array was expected. Please change the
  return f(**kwargs)
(0.7170910301214845, 0.7221667498751873)
```

의사 결정 나무

```
tre_max = tre.값.max()
tre[tre.값 == tre_max]
```

	값	순서
40	0.6665	41

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=41)
tree.fit(X_train, y_train)
tree.score(X_train, y_train) , tree.score(X_test, y_test)
```

```
(1.0, 0.6665002496255616)
```

KNN 모델

```
nei_max = nei.값.max()
nei[nei.값 == nei_max]
```

	값	순서
55	0.7222	56

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=56)
knn.fit(X_train,y_train)
knn.score(X_train, y_train), knn.score(X_test, y_test)
```

```
<ipython-input-84-b009afcdd304>:3: DataConversionWarning:
hange the shape of y to (n_samples, ), for example using
  knn.fit(X_train,y_train)
(0.7153436511898819, 0.7221667498751873)
```

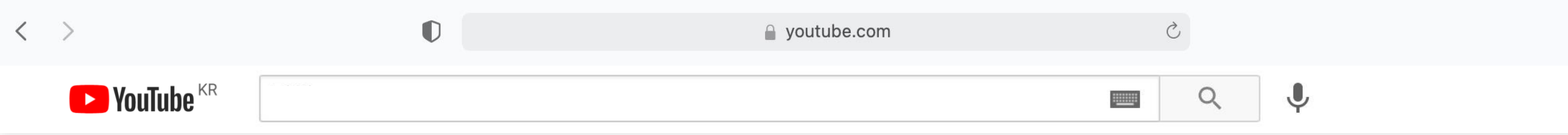
랜덤 포레스트

```
ran_max = ran.값.max()
ran[ran.값 == ran_max]
```

	값	순서
55	0.7466	56

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=56,random_state=3)
forest.fit(X_train,y_train)
forest.score(X_train, y_train) , forest.score(X_test, y_test)
```

```
<ipython-input-100-66f6d774d551>:3: DataConversionWarning: A colum
change the shape of y to (n_samples,), for example using ravel().
  forest.fit(X_train,y_train)
(1.0, 0.7466300549176236)
```



4. 시사점 및 한계점

youtube.com

YouTube KR

4. 시사점 및 한계점

시사점

- 유튜브 전반의 트렌드 분석을 통해 이해도를 높일 수 있다.
- 긍·부정 모델을 통해 자신의 유튜브의 영상의 여론을 어느정도 짐작해 볼 수 있다.
- 미국 시장의 반응까지 궁금하면 미국 + 한국의 긍·부정 모델을 사용해 파악이 가능하다

한계점

- 정확한 모델 구축을 위해서 필요한 영상 길이, 구독자 수 등의 데이터가 부족했다.
- YouTube 알고리즘에 영향을 미치는 시청 시간, 광고 수 등의 데이터가 부족했다.
- 다양한 매체와 사회적 상황의 영향을 받기때문에 정확한 예측이 어려울 수 있다.