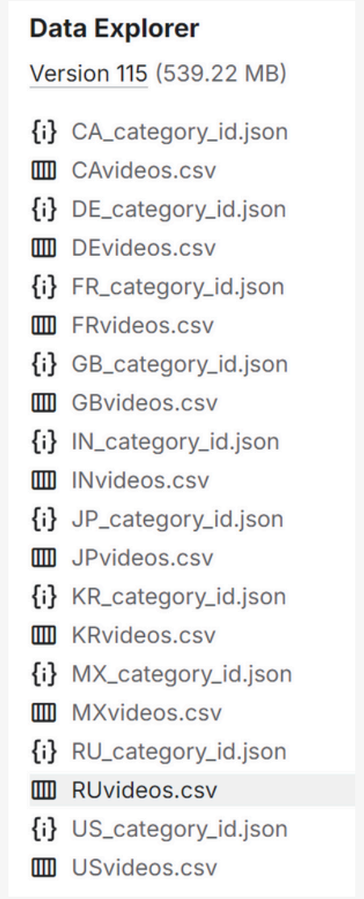
컨텐츠 주제와 타겟 국가 선정에 따른 성장가능성 예측 3

컴퓨터공학과 2022108115
이우석

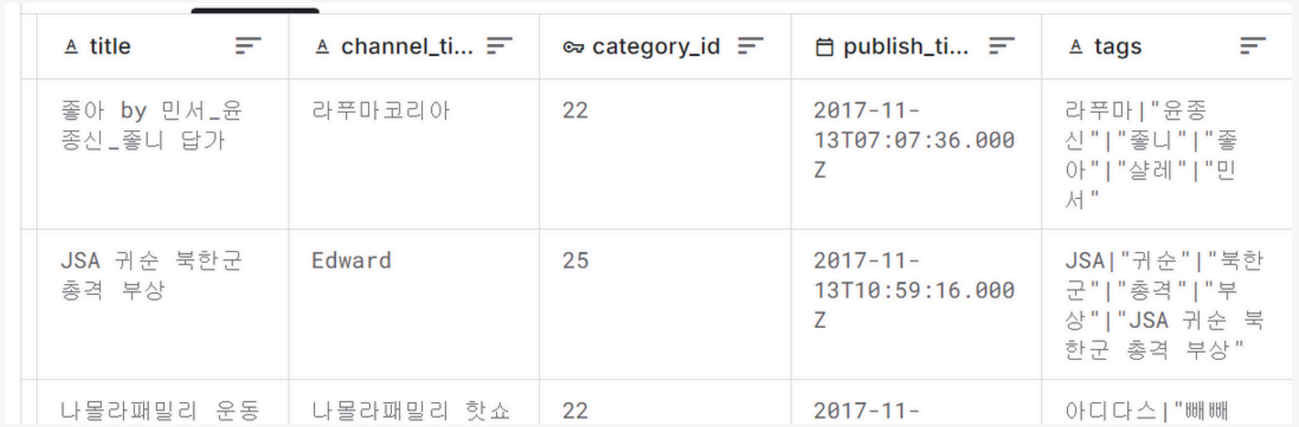
01 데이터셋

Trending YouTube Video Statistics

해당 데이터는 **미국, 영국, 독일, 캐나다, 프랑스, 러시아, 멕시코, 한국, 일본, 인도**
총 10개국의 유튜브 동영상에 대한 데이터를 포함합니다.
데이터는 하루의 최대 200개의 인기 동영상을 기록합니다.



01 데이터셋



title	channel_ti...	category_id	publish_ti...	tags
좋아 by 민서_윤종신_좋니 답가	라푸마코리아	22	2017-11-13T07:07:36.000Z	라푸마|"윤종신"|"좋니"|"좋아"|"샬레"|"민서"
JSA 귀순 북한군 충격 부상	Edward	25	2017-11-13T10:59:16.000Z	JSA|"귀순"|"북한군"|"충격"|"부상"|"JSA 귀순 북한군 충격 부상"
나몰라패밀리 운동	나몰라패밀리 핫쇼	22	2017-11-	아디다스|"빼빼

Trending YouTube Video Statistics

해당 데이터의 column은 총 10개로 구성되어 있습니다.
비디오 id, 인기동영상으로 게시된 날짜, 제목, 채널명,
카테고리, 업로드 날짜, 태그, 시청수, 좋아요, 싫어요로 구성되어 있습니다.

이 중 모델 학습에 사용할 항목은 **카테고리, 좋아요, 시청수**와
국가별로 구분된 파일로 존재하는 csv파일을 하나로 합치고 부여한 **국가명**을 사용할 예정입니다.

01 데이터셋

데이터셋의 카테고리 id

해당 데이터셋의 카테고리는 숫자로된 id로 구분됩니다.
숫자별 카테고리 분류는 다음과 같습니다.

1,Film & Animation
2,Autos & Vehicles
10,Music
15,Pets & Animals
17,Sports
18,Short Movies
19,Travel & Events
20,Gaming
21,Videoblogging
22,People & Blogs
23,Comedy
24,Entertainment
25,News & Politics
26,Howto & Style
27,Education
43,Shows

28,Science & Technology
29,Nonprofits & Activism
30,Movies
31,Anime/Animation
32,Action/Adventure
33,Classics
34,Comedy
35,Documentary
36,Drama
37,Family
38,Foreign
39,Horror
40,Sci-Fi/Fantasy
41,Thriller
42,Shorts
43,Shows

02 데이터 전처리

데이터에서 필요한 항목만 남기고
글자를 숫자로 변환하고, 결측값처리 등의
전처리 과정을 진행하였습니다.
전처리 과정을 진행하며 국가 Column을 추가하였습니다

모든 국가의 csv파일을 하나로 합쳐
하나의 파일로 만들었습니다.

```python
import pandas as pd
import os

# 특정 국가의 CSV 파일 경로
file_path = '/content/RUvideos.csv'  # 파일 경로를 실제 경로로 변경

# 전처리된 데이터를 저장할 리스트
processed_data = []

# 파일을 읽어오기 (인코딩 오류가 있을 수 있으므로 encoding을 시도해봄)
try:
    # CSV 파일 읽기
    chunk = pd.read_csv(file_path, encoding='utf-8', low_memory=False)

    # 국가 이름을 추출
    country_name = 'Russia'

    # 필요한 열만 선택하기
    filtered_chunk = chunk[['category_id', 'views', 'likes']].copy()

    # 결측값 처리
    filtered_chunk['views'] = filtered_chunk['views'].fillna(0)
    filtered_chunk['likes'] = filtered_chunk['likes'].fillna(0)

    # category_id를 숫자형으로 변환 (문제가 있을 경우 0으로 처리)
    filtered_chunk['category_id'] = pd.to_numeric(filtered_chunk['cate

    # 'country' 컬럼에 해당 국가명을 추가
    filtered_chunk['country'] = country_name

    # 처리된 데이터를 리스트에 추가
```

02 데이터 전처리

합쳐진 데이터셋이 국가별로 몰려있어서 파일의 행을 무작위로 섞은 후 저장하였습니다.

	category_id	views	likes	country
0	22	62408	334	Russia
1	22	330043	43841	Russia
2	24	424596	49854	Russia
3	22	112851	3566	Russia
4	24	243469	36216	Russia

```
[32] # 전체 데이터를 무작위로 섞기
     shuffled_data = final_data.sample(frac=1).reset_index(drop=True)

     # 무작위로 섞은 데이터를 CSV 파일로 저장
     shuffled_data.to_csv('/content/shuffled_data.csv', index=False)

     # 결과 확인
     print("파일이 무작위로 섞여 저장되었습니다.")
```

```
파일이 무작위로 섞여 저장되었습니다.
```

```
rows_10003_10009 = shuffled_data.iloc[10002:10009]

print(rows_10003_10009)
```

```
       category_id    views    likes country
10002           22     9275      373  Mexico
10003           22    46082     1102  Mexico
10004           10  7518990  1573046  Russia
10005           10   300850    32066   Japan
10006           26    46347     2606  Russia
10007           10    12498     1018  Mexico
10008           24  1327815    32895     USA
```

03 모델 학습

이렇게 확보한 데이터를 사용할 모델을
Pycaret을 이용하여 결정 및 학습하였습니다.

```
from pycaret.regression import *

# 필요한 컬럼만 사용
data_view = data[['category_id', 'country', 'views', 'likes']]

# PyCaret 설정 (목표 변수: 'views')
reg_view_setup = setup(data=data_view,
                       target='views',
                       numeric_features=['category_id'],
                       categorical_features=['country'],
                       session_id=123,
                       verbose=False)  # verbose로 대화형 입력 방지

# 모델 비교
best_model_view = compare_models()

# 최적 모델 학습
final_model_view = finalize_model(best_model_view)

# 모델 평가
print("View Prediction Results:")
print(predict_model(final_model_view))
```

03 모델 학습

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	745563.9190	15572558682939.9863	3923091.9047	0.6937	1.3751	3.9924	19.4460
lightgbm	Light Gradient Boosting Machine	719237.2197	18993199439212.2148	4338084.6927	0.6256	1.0815	2.0984	
catboost	CatBoost Regressor	711252.9810	19016025371416.7422	4340249.4483	0.6252	1.0659	2.1354	
xgboost	Extreme Gradient Boosting	710239.2050	19056222782231.6484	4345213.6281	0.6244	1.0615	3.3084	
lasso	Lasso Regression	925483.8922	19847268130520.1992	4432926.8746	0.6089	1.6700	5.3597	10.3420
ridge	Ridge Regression	925480.2592	19847268092068.6523	4432926.8628	0.6089	1.6699	5.3597	0.4200
lar	Least Angle Regression	925486.6583	19847268135017.2695	4432926.8790	0.6089	1.6701	5.3599	0.4180
llar	Lasso Least Angle Regression	925484.3302	19847268139121.5938	4432926.8788	0.6089	1.6701	5.3597	0.4390
lr	Linear Regression	925486.6583	19847268135017.6328	4432926.8790	0.6089	1.6701	5.3599	1.3600
en	Elastic Net	841849.4349	19993973852618.5703	4449408.7995	0.6060	1.4518	2.5821	0.7290
rf	Random Forest Regressor	724008.5710	20034070708911.2266	4443571.8256	0.6059	1.0149	1.8435	96.9780
ada	AdaBoost Regressor	1031721.4144	19721062985087.9102	4434327.2412	0.6054	2.1536	15.9208	3.1970
omp	Orthogonal Matching Pursuit	822119.1860	20071299371434.5234	4458175.9963	0.6045	1.1684	2.0912	0.7940
br	Bayesian Ridge	822118.3317	20071277987734.7266	4458173.5294	0.6045	1.1684	2.0911	0.5680
knn	K Neighbors Regressor	791691.4786	20182614460308.3203	4465861.5959	0.6028	1.0867	2.1725	1.6560
et	Extra Trees Regressor	772326.5043	22452479123500.2422	4713154.9781	0.5563	1.0743	1.8894	67.9940
huber	Huber Regressor	762149.4702	22542765317909.0898	4725266.8426	0.5559	1.1233	1.3180	1.2740
dt	Decision Tree Regressor	815647.5516	27466397016017.5508	5210805.8810	0.4571	1.1335	1.9269	1.7480
dummy	Dummy Regressor	1825923.4783	50466835780142.8984	7086069.1170	-0.0000	2.8104	39.2687	0.4520
par	Passive Aggressive Regressor	1413803.0699	64491430501496.2578	7158156.2553	-0.3184	1.1736	1.4602	1.3020

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Gradient Boosting Regressor	739902.8029	13342278017995.4922	3652708.3127	0.7343	1.3692	3.9044

모델 선정 결과 Gradient Boosting Regressor가
선정되었고, R^2 (결정계수)의 값이 0.73으로
데이터 변동성의 약 73.43%를 모델이 설명할 수 있음

04 모델 활용 방안

이렇게 학습된 모델을 통해 사용자는
다음과 같은 문제를 해결할 수 있습니다.

우리의 음악 유튜브 채널을 해외로
확장 시키고 싶은데 어느 국가를
타겟팅 하는게 좋을까?

이번에 새로 유튜브 채널을 개설하려고
하는데 어느 분야가 성장하기 쉬울까?

05 모델 개선 방안

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Gradient Boosting Regressor	739902.8029	13342278017995.4922	3652708.3127	0.7343	1.3692	3.9044

선정된 모델의 MAE(Mean Absolute Error)와 RMSE(Root Mean Squared Error)를 확인해보면

- MAE(예측 값과 실제 값 간의 절대 오차의 평균): 739,902.8029
- RMSE(예측 오차 제곱의 평균의 제곱근): 3,652,708.3127

해당 모델에서는 조회수 예측에서 큰 오차를 줄이는 것이 중요하기 때문에 큰 오차에
민감하게 반응하는 **RMSE**를 개선하고자 합니다.

추후 이상값(Outliers)을 처리하고, 하이퍼파라미터를 튜닝하여 모델의 성능을 개선하고자 합니다.

Thank you

감사합니다.