A large, horizontal, pink brushstroke shape with irregular, feathered edges, serving as a background for the text.

노래가사 프로젝트

프로젝트 목적

- 노래들을 장르별로 쉽게 분류하고 싶다.
- 가사로만 노래의 장르를 맞출 수 있을까?
- 노래가사만 입력하면 노래의 장르를 예측해주는 모델 생성해보자.

1. 크롤링(데이터 수집)

박스

<https://music.bugs.co.kr/> 에서 노래가사를 크롤링하였습니다.

The screenshot shows the Bugs! music website interface. On the left is a navigation menu with options like '박스 VIP 멤버십 안내', '이용권 구매', '상품권 등록', '박스차트', '최신 음악', '뮤직4U', '장르', '뮤직포스트', '테마', '뮤직PD 앨범', '추천앨범 리뷰', '연도별', '커넥트', '라디오', '박스TV', '스페셜 라이브', and '내 음악'. The main content area displays the album 'END THEORY' by YOUNHA. It includes the album cover, artist name '윤하 (Younha/ユンナ)', and details about the 5th Album Repackage. Below the album info, there are buttons for '듣기', '재생목록에 추가', '내 앨범에 담기', '다운로드', and '라디오 듣기'. At the bottom, the lyrics (가사) are displayed in Korean.

Bugs!

연말 파티 인싸 준비 됐어? Q

로그인 사용자: seulgi han (seulgi980@naver.com)

앨범 정보

아티스트: 윤하 (Younha/ユンナ)
참여 정보: 보컬 윤하 (Younha/ユンナ) | 작곡 윤하 (Younha/ユンナ), JEWNO (손준호) | 작사 윤하 (Younha/ユンナ) 전체 보:
앨범: YOUNHA 5th Album Repackage 'END THEORY: Final Edition'
재생 시간: 05:00
고음질: FLAC 16bit, 24bit | 하이 레졸루션

가사

생각이 많은 건 말이야
당연히 해야 할 일이야
나에겐 우리가 지금 1순위야
안전한 유리병을 핑계로
바람을 가둬 둔 것 같지만
거역나? 그날의 우리가
잠았던 그 손엔 말이야
설레임보다 커다란 믿음이 담겨서
난 행복웃음을 지었지만
울음이 날 것도 같았어
소중한 건 언제나 두려움이니까
문을 열면 들리던 목소리
너로 인해 변해있던 따뜻한 공기
여전히 자신 없지만 안녕히

1. 크롤링

url requests로 응답을 받고, beautifulsoup으로 dom화



```
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
import requests
```

```
URL = 'https://music.bugs.co.kr/track/6179174?wl_ref=list_tr_08_search'
```

```
# response로 접근 요청하고 혹시 모르니 header도 달아주기
```

```
headers = {'user-agent' : UserAgent().chrome}
```

```
response = requests.get(URL, headers=headers)
```

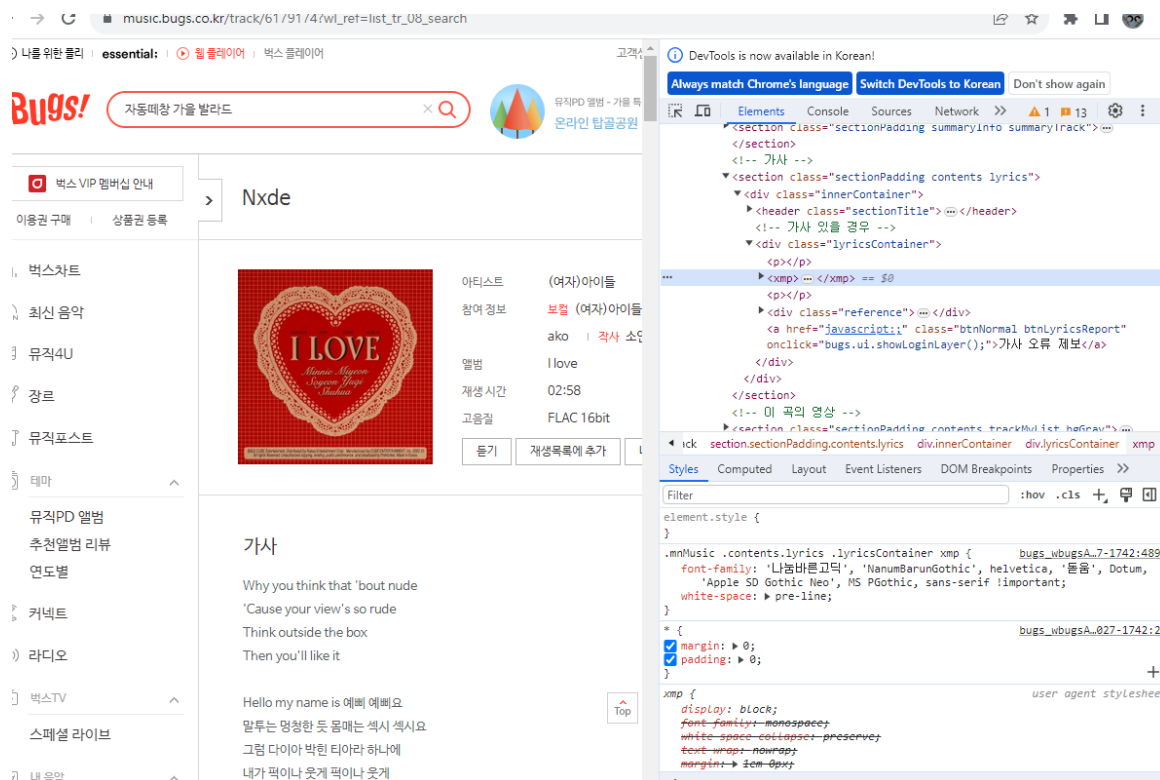
```
# DOM화 시키기
```

```
page = response.content
```

```
soup = BeautifulSoup(page, 'html.parser')
```

1. 크롤링

Soup에서 가사가 있는 부분의 경로를 찾기 위해 F12를 눌러 경로찾기
아래의 가사는 lyricsContainer의 class에 존재하는 것으로 확인
select_one으로 경로를 사용해 원하는 가사 추출 시작



The screenshot shows the music.bugs.co.kr website. The album 'Nxde' by (여자)아이들 is displayed. The lyrics are shown in a container with the class 'lyricsContainer'. The DevTools console shows the HTML structure of the lyrics container, confirming the class attribute.

```
soup.select_one('.lyricsContainer')
```

```
<div class="lyricsContainer">
<p><xmp>Why you think that 'bout nude
'Cause your view's so rude
Think outside the box
Then you'll like it
```

Hello my name is 예뻐 예뻐요
말투는 멍청한 듯 몸매는 섹시 섹시요
그럼 다이아 박힌 티아라 하나에
내가 찍이나 웃게 찍이나 웃게

뒤틀러버린 로렐라이 Don't need no man
철학에 미친 독서광 Self-made woman
싸가지없는 이 Story에 무지 황당한
야유하는 관객들 You tricked me you're a liar

마 발가벗겨져 버린 Movie star
마 별빛이 깨져버린 밤
꿀이 볼품없대도 망가진다 해도
다신 사랑받지 못한대도

1. 크롤링

단, 완벽히 노래가사 데이터만을 필요로 하기 때문에 세부 경로 탐색
완벽히 가사만 찾은후, text로 가사데이터 추출

```
Nude</xmp></p>
<div class="reference">
<cite class="writer">Bugs</cite>님이 등록해 주신 가사입니다.
</div>
<a class="btnNormal btnLyricsReport" href="javascript:;" onclick="bugs.ui.showLoginLayer();">가사 오류 제보</a>
</div>
```

```
▶ soup.select_one('.lyricsContainer xmp')
3 ❏ 마유하는 관객들 You tricked me you're a liar
```

아 발가벗겨져 버린 Movie star
아 별빛이 깨져버린 밤
꽃이 볼품없대도 망가진다 해도
다신 사랑받지 못한대도

Yes I'm a nude
Nude 따따랏따라
Yes I'm a nude
Nude I don't give a love

Baby how do I look, how do I look
아리따운 날 입고 따따랏따라
Baby how do I look, how do I look
아리따운 날 입고 따따랏따라

```
▶ lyric1 = soup.select_one('.lyricsContainer xmp').text
lyric1
```

```
3 ❏ 'Why you think that 'bout nude'Cause your view's so rudeThink outside the boxThen you'll like itHello my name is 예뻐 예뻐요
한 듯 몸매는 섹시 섹시요그럼 다이아 박힌 티아라 하나에내가 찍이나 웃게 찍이나 웃게뒤틀러버린 로렐라이 Don't need no man철학에 미친
If-made woman싸가지없는 이 Story에 무지 황당한마유하는 관객들 You tricked me you're a liar아 발가벗겨져 버린 Movie star아 별빛이 )
꽃이 볼품없대도 망가진다 해도다신 사랑받지 못한대도Yes I'm a nudeNude 따따랏따라Yes I'm a nudeNude I don't give a love
ow do I look, how do I look아리따운 날 입고 따따랏따라Baby how do I look, how do I look아리따운 날 입고 따따랏따라(Ouch!)실례;
게신 모두야한 작품을 기대하셨다면Oh I'm sorry 그딴 건 없어요환불은 저쪽 대중은 흥미 없는 정보그 팔콘을 던져도 엄덤행복과 반비례 폼
my 정점 멋대로 낸 편견은 토할 거 같지아 발가벗겨져 버린 Movie star아 더 부끄러울 게 없는 밤꽃이 볼품없대도 어찌면 네게도다신 사랑!
도Yes I'm a nudeNude 따따랏따라Yes I'm a nudeNude I don't give a loveBaby how do I look, how do I look아리따운 날 입고
라#...
```

2. 전처리

개행 문자 split을 이용하여 삭제 후 다시 join

개행문자 삭제해주기

- 개행문자가 \n으로 되어있어서 '\n'으로 스플릿 하였습니다.

```
[ ] lyric1 = lyric1.split('\r\n')
lyric1
```

```
["Why you think that 'bout nude",
 "'Cause your view's so rude",
 'Think outside the box',
 "Then you'll like it",
 '..',
 'Hello my name is 예뻐 예뻐요',
 '말투는 멍청한 듯 몸매는 섹시 섹시요',
 '그럼 다이아 박힌 티아라 하나에',
 '내가 찍이나 웃게 찍이나 웃게',
 '..',
 "뒤들려버린 로젤라이 Don't need no man",
 '철학에 미친 독서광 Self-made woman',
 '싸가지없는 이 Story에 무지 황당한',
 "야유하는 관객들 You tricked me you're a liar",
 '..']
```

```
[ ] lyric1 = ''.join(lyric1)
lyric1
```

```
'Why you think that 'bout nude'Cause your view's so rudeThink outside the boxThen you'll like itHello my name is 예뻐 예뻐요말투는 멍청한
그럼 다이아 박힌 티아라 하나에내가 찍이나 웃게 찍이나 웃게뒤들려버린 로젤라이 Don't need no man철학에 미친 독서광 Self-made woman싸가지없
한야유하는 관객들 You tricked me you're a liar아 발가벗겨져 버린 Movie star아 별빛이 깨져버린 밤꽃이 불품없대도 망가진다 해도다신 사랑받
eNude 따따랏따라Yes I'm a nudeNude I don't give a loveBaby how do I look, how do I look아리따운 날 입고 따따랏따라Baby how do I look, ho
입고 따따랏따라(Ouch!)실례합니다 여기 계신 모두마한 작품을 기대하셨다면Oh I'm sorry 그만 건 없어요환불은 저쪽 대장은 흥미 없는 정보그 팔
반비례 평점But my 정점 멋대로 낸 편견은 토할 거 같지아 발가벗겨져 버린 Movie star아 더 부끄러울 게 없는 밤꽃이 불품없대도 어찌면 네게도C
I'm a nudeNude 따따랏따라Yes I'm a nudeNude I don't give a loveBaby how do I look, how do I look아리따운 날 입고 따따랏따라Baby how do I
따운 날 입고 따따랏따라Um ha um ha umYes I'm a nudeYes I'm a nudeNow I draw a luxury nudeWhy you think that 'bout nude'Cause your
ide the box'라고 말해...'
```

2. 전처리

Konlpy의 Okt.morphs를 이용하여 형태소 별로 분류해준다.
Stem = True를 이용해 어간추출도 수행해준다.
최종적으로 어간들을 다시 join

```
[ ] from konlpy.tag import Okt
    okt = Okt()

    # stem=True : 어간 추출 , norm=True : 정규화
    lyric1 = okt.morphs(lyric1, norm=True, stem=True)
    lyric1
```

```
[ ] ['Why',
     'you',
     'think',
     'that',
     '',
     'bout',
     'nude',
     '',
     'Cause',
     'your',
     'view',
     '',
     's',
     'so',
     'rudeThink',
     'outside',
     'the',
     'boxThen',
```

```
[ ] lyric1 = ' '.join(lyric1)
```


3. 과정 단순화

위의 과정은 노래가사 1개를 가져오는 과정이었기 때문에,
한 번에 여러 곡을 가져올 수 있도록 함수를 생성필요

```
X = []
y = []

def crol(url, label):
    # 1. 크롤링
    headers = {'user-agent' : UserAgent().chrome}
    response = requests.get(url, headers=headers)

    page = response.content
    soup = BeautifulSoup(page, 'html.parser')

    lyric = soup.select_one('.lyricsContainer xmp').text

    # 2. 전처리
    lyric = lyric.split('\r\n')
    lyric = ''.join(lyric)

    okt = Okt()
    lyric = okt.morphs(lyric, norm=True, stem=True)

    # 3. 다시조립
    lyric = ''.join(lyric)

    X.append(lyric)
    y.append(label)
```



이런식으로 하나하나 넣어주는 방법
위의 방법은 원하는 데이터를 골라넣거나, 하나하나 추가할때 유용할 것 같습니다.

```
crol('https://music.bugs.co.kr/track/6179174?wl_ref=list_tr_08_search', 0)
```

3. 과정 단순화

모델링은 인기차트 100의 데이터를 이용하였습니다.

- 편리성을 위해 페이지에 있는 차트를 통째로 가져오는 방법을 선택하였습니다.

3. 과정 단순화

인기차트 100의 데이터를 이용하기 위한 코드
여러 데이터를 가져올 것이므로 select_one이 아니라 select로
인기차트 100개의 노래의 url을 url_lst에 담아준 후 앞의 함수 사용

```
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
import requests
from konlpy.tag import Okt

X = []
y = []

def crol(url, genre):
    URL_lst = []
    headers = {'user-agent' : UserAgent().chrome}
    response = requests.get(url, headers=headers)
    page = response.content
    soup = BeautifulSoup(page, 'html.parser')

    for i in range(100):
        URL = soup.select("a.trackInfo")[i]['href']
        URL_lst.append(URL)

    for url in URL_lst:
        headers = {'user-agent' : UserAgent().chrome}
        response = requests.get(url, headers=headers)
        page = response.content
        soup = BeautifulSoup(page, 'html.parser')

        lyric = soup.select_one('.lyricsContainer xmp').text
```

```
# 2. 전처리
lyric = lyric.split('\r\n')
lyric = ''.join(lyric)

okt = Okt()
lyric = okt.morphs(lyric, norm=True, stem=True)

# 3. 다시조립
lyric = ''.join(lyric)

X.append(lyric)

if genre == '발라드':
    for g in range(100):
        y.append(1)

elif genre == '댄스':
    for g in range(100):
        y.append(0)
```

3. 과정 단순화

최종 사용한 데이터 X y, feature와 label

```
[ ] crol('https://music.bugs.co.kr/genre/chart/kpop/ballad/total/day', '발라드')
```

```
[ ] crol('https://music.bugs.co.kr/genre/chart/kpop/dance/total/day', '댄스')
```

```
[ ] len(X), X[:3]
```

(200,
['다 잊다 거짓말 또 해 버리다 내 마음 에 그 대란 사람 없다 하다 너무나 쉬다 잊혀지다 이젠 남 이라고 서둘다 내 사랑 에 지치다 떠나다 그대 너무 많이 울리다 잡 을 용기 조차 낼
없다 미안하다 내 사랑 아 다시다 나 같다 사람 만나다 마 요 혹시 찾아가다 두 번 달다 허락 하다 주지 마 요 그 대다 여리고 너무 착하다 싫다 말 도 자다 못 하다 많이 부족하다 나르다
사랑 한 그 대 이 거 면 돼다 더 이상은 그대 불행하다 앓다 도록 나 이쯤 에서 없어지다 게그 대다 위 한 나 의 사랑 인 걸 요 너무 투명하다 때론 불안하다 제멋대로 살아오다 나르다 같
하다 수 없다 것 같다 미안하다 내 사랑 아 다시다 나 같다 사람 만나다 마 요 혹시 찾아가다 두 번 달다 허락 하다 주지 마 요 그 대다 여리고 너무 착하다 싫다 말 도 자다 못 하다 고민
나르다 사랑 하다 주다 서나 같다 사람 이 두 번 다시 감히 발다 수 없다 사랑 그대 때문 에 행복하다 울 지 마 요 그대 자다 생각 하다 보다 나쁘다 일 들 만 가득하다 우리 다 잊다 해
다시다 나 같다 사람 만나다 마 요 혹시 찾아가다 두 번 달다 허락 하다 주지 마 요 그 대다 여리고 너무 착하다 싫다 말 도 자다 못 하다 많이 부족하다 나르다 사랑 한 그 대 이 거 면
다 더 이상은 그대 불행하다 앓다 도록 나 이쯤 에서 없어지다 게그 대다 위 한 나 의 사랑 인 걸 요',
'이윽고 내 가 한눈 에 너 를 알아보다 때 모든 건 분명 달라지다 있다 내 세상 은 널 알다 전과 후 로 나뉜다 숨 쉬 면 따스하다 바람 이 불어오다 웃다 눈부시다 햇살 이 비추다 있다
게 너 라서 가끔 내 어깨 에 가만히 기 대 주며 서나 는 있다 정말 빈틈 없이 행복하다 너 를 따라서 시간 은 흐르다 멈추다 물끄러미 너 를 들여다보다 해 그것 말고는 아무 것 도 하다
없다 서너 의 모든 순간 그게 나다 좋다 생각 만 해도 가슴 이 차오르다 나 는 온통 너 로 보고 있다 웬지 꿈 처럼 아득하다 몇몇 광년 동안 날 향 하다 날아오다 별빛 또 지금 의 너거 기
있다 그게 너 라서 가끔 나 에게 조용하다 안기다 나 는 있다 정말 남김없이 고맙다 너 를 따라서 시간 은 흐르다 멈추다 물끄러미 너 를 들여다보다 해너 를 보다 게 나 에게는 사랑 이니
너 의 모든 순간 그게 나다 좋다 생각 만 해도 가슴 이 차오르다 나 는 온통 너 로 니 모든 순간 나다',
'밤하늘 손 을 잡다 기분 이 좋다 열다 웃음 띠 며 나 에게 말 하다 너 는 슬프다 노래 를 부르다 그제 우리 둘 의 주제곡 같다 하루 는 순진하다 눈 으로 나르다 바라보다 매일 두 손
으다 하늘 에 비다 고우리 둘 의 시간 이 영원하다 그제 차다 아프다 그리다 기도 하다 둘 이 가다 두다 알다 수많은 거리 그 위로 하루하루 쌓이다 가다 소중하다 추억 그 위로 우리 다시 돌아가다 수 있다 알 아 결코 닿다 수 없다 시절 의 우리 그때 는 이해 하다 수 없다 너 의 그 마지막 이젠 선명하다 다 알 것 같다 어그 래 우린 그렇게 사랑 하다 뻔하다 이별 마저도
리 다 왔던 거 야 둘 이 가다 두다 알다 수많은 거리 그 위로 하루하루 쌓이다 가다 소중하다 추억 그 위로 우리 다시 돌아가다 수 있다 알 아 결코 닿다 수 없다 시절 의 우리 그때 의 두
처럼 둘 이 가다 두다 알다 며칠따다 풍경 그 위로 자라나다 쌓이다 가다 머리숙하다 모습 그대로 이제 다시 돌아가다 여쭈다 그렇다 결코 보내다 수 없다 시절 의 우리'])

```
[ ] len(y), y[:3]
```

```
[ ] (200, [1, 1, 1])
```

4. 모델링

X, y를 train/ valid / test셋으로 분리

```
▶ from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size = 0.2,
    stratify = y,
    random_state=42
)
```

```
[ ] X_train, X_valid, y_train, y_valid = train_test_split(
    X_train,
    y_train,
    test_size = 0.2,
    stratify = y_train,
    random_state=42
)
```

```
[ ] len(X_train), len(X_valid), len(X_test)

(128, 32, 40)
```

4. 모델링

영어와 'a, the' 한국어의 '의, 는' 같은 조사는 모델링에 방해가 된다고 생각하여 이러한 문제를 해결해주는 Tfidfvectorizer를 사용하였습니다.

```
# 모델에 numpy배열로 넣어주기 위해 y를 넘파이변환
import numpy as np
```

```
y_train = np.array(y_train)
y_valid = np.array(y_valid)
y_test = np.array(y_test)
```

```
[ ] y_train.shape, y_valid.shape, y_test.shape

((128,), (32,), (40,))
```

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfidf_vect= TfidfVectorizer()
```

```
X_train_tfidf_vect = tfidf_vect.fit_transform(X_train)
```

```
# 검증세트는 꼭 transform만해주기★
```

```
X_valid_tfidf_vect = tfidf_vect.transform(X_valid)
```

```
X_test_tfidf_vect = tfidf_vect.transform(X_test)
```

```
X_train_tfidf_vect
```

```
<128x3415 sparse matrix of type '<class 'numpy.float64'>'
  with 10683 stored elements in Compressed Sparse Row format>
```

```
[ ] X_train_tfidf_vect.shape, X_valid_tfidf_vect.shape, X_test_tfidf_vect.shape

((152, 3644), (38, 3644), (48, 3644))
```

4. 모델링

1. 메모리 사용량이 적고
2. 속도가 빠르고
3. parameter를 조절할 수 있는 lightgbm을 사용해보았습니다.

(단, 너무 적은 데이터(1만건 이하)는 오히려 과대적합을 불러일으킬 수 있는 단점이 존재하기 때문에 이럴경우 Xgboost를 사용해 보면 될 것이라고 생각합니다.)

```
] import lightgbm
   from lightgbm import LGBMClassifier

   lgbm_wrapper = LGBMClassifier(
       n_estimators=1000,
       learning_rate=0.05,
       max_depth = 5
   )
```

4. 모델링

모델 훈련 `early_stopping_rounds` 옵션을 걸어주어
Loss 변화가 너무 없으면 조기종료하도록 설정

```
] evals = [  
    (X_train_tfidf_vect, y_train),  
    (X_valid_tfidf_vect, y_valid)  
]  
  
lgbm_wrapper.fit(  
    X_train_tfidf_vect,  
    y_train,  
    early_stopping_rounds=500,  
    eval_metric='logloss',  
    eval_set=evals  
)
```

```
/usr/local/lib/python3.8/dist-packages/lightgbm/sklearn.py:726: UserWarning: 'early_stopping  
_log_warning("'early_stopping_rounds' argument is deprecated and will be removed in a futu
```

[1]	training's binary_logloss: 0.644596	valid_1's binary_logloss: 0.645199
[2]	training's binary_logloss: 0.624695	valid_1's binary_logloss: 0.627752
[3]	training's binary_logloss: 0.606749	valid_1's binary_logloss: 0.61138
[4]	training's binary_logloss: 0.59051	valid_1's binary_logloss: 0.5976
[5]	training's binary_logloss: 0.574918	valid_1's binary_logloss: 0.583062
[6]	training's binary_logloss: 0.561367	valid_1's binary_logloss: 0.572005
[7]	training's binary_logloss: 0.548192	valid_1's binary_logloss: 0.560254
[8]	training's binary_logloss: 0.536649	valid_1's binary_logloss: 0.550784
[9]	training's binary_logloss: 0.525391	valid_1's binary_logloss: 0.540953
[10]	training's binary_logloss: 0.515028	valid_1's binary_logloss: 0.5311
[11]	training's binary_logloss: 0.504073	valid_1's binary_logloss: 0.520425
[12]	training's binary_logloss: 0.495222	valid_1's binary_logloss: 0.513609
[13]	training's binary_logloss: 0.485594	valid_1's binary_logloss: 0.504322
[14]	training's binary_logloss: 0.475967	valid_1's binary_logloss: 0.497083

4. 모델평가

Train set으로 모델을 평가해보니 83.5%의 정확도를 알 수 있었습니다.
분류 모델이라 지표로 정확도 사용

```
[>] from sklearn.metrics import accuracy_score  
  
pred = lgbm_wrapper.predict(X_test_tfidf_vect)  
  
accuracy_score(y_test , pred)
```

⇒ 0.8354430379746836

5. 모델확인

개별적으로 모델을 넣어서 결과를 확인해볼 수 있었습니다.

```
def music_genre(pred):  
    if max(pred) == 0:  
        print("음악 장르는 댄스입니다.")  
    else:  
        print("음악 장르는 발라드입니다.")
```

```
# 발라드  
test1 = '''미칠 것 같아 기다림 내게 아직도 어려워  
  
보이지 않는 니가 미웠어  
  
참을 수밖에 내게 주어진 다른 길 없어  
  
속삭여 불러보는 네 이름,  
  
머두운 바다를 떠돌아 다니는 부서진 조각배 위에 누워 내 작은 몸  
언젠가 그대가 날 아무말 없이 안아 주겠죠  
  
그 품안에 아주 오래도록  
  
나에게 지워진 시간의 무게가 견디기 힘이 들도록 쌓여간다 해도  
언젠가 그대가 날 아무말없이 안아 주겠죠  
  
그댄 나를 아무말 없이 안아주겠죠  
  
그 품안에 아주 오래도록'''
```

```
▶ pred = lgbm_wrapper.predict(tfidf_vect.transform([test1]))  
music_genre(pred)
```

☞ 음악 장르는 발라드입니다.

```
[ ] lgbm_wrapper.predict_proba(tfidf_vect.transform([test1]))  
  
array([[0.37045309, 0.62954691]])
```

6. 프로젝트 결과 및 후기

결과) 노래의 가사만으로 장르를 구분하는 모델 생성
박스 사이트만 뿐 아니라, url만 새로 입력해준다면 멜론, 지니 같은 다른 사이트도 이용가능한 편리성 모델

아쉬운점)

lightgbm특성상 데이터를 추가하거나
Parameter를 조절하면 더 좋은 정확도의 모델을 생성할 수 있을 것

Xgboost같은 다른 여러 모델을 사용해 보았어도 좋았을 것 같다.
프로젝트 시간제한 상 다양한 시도를 해보지 못한 것이 아쉽다.