통계최강자전 Spotify 장르 예측

- 최종 보고서 -

헌진스

김대연 | 응용통계학과 201811830
김찬영 | 응용통계학과 201811838
이원종 | 응용통계학과 201811866
전효진 | 응용통계학과 201811870
천지은 | 응용통계학과 201811875

목차

1. 프로젝트 개요

21 세기 현대인들은 음악과 함께 살아간다. 집안일을 하거나 샤워할 때, 출근할 때 또는 공부할 때 많은 사람들의 귀에는 이어폰이 꽂혀있다. 이렇듯 음악은 서민들의 삶 속에 녹아 들었으나, 모든 음악을 좋아하는 사람은 찾기 힘들다. Pop, Hiphop, Dance 등 다양한 "장르"를 기준으로 음악이 나뉘어질 수 있으며 사람들의 취향은 장르를 따라간다.

현재는 Melon, Spotify, Sound Cloud, Youtube music, Genie 등 많은 음악 플랫폼이 존재한다. 음악을 듣고 싶어하는 현대인들을 위해 플랫폼들은 장르를 분류하는 매커니즘을 가지고 있으며 장르 분류 서비스가 잘 이루어질수록 고객들은 편의를, 플랫폼 회사는 경쟁력을 갖춘다.

장르를 나누는 것은 무궁무진하고 다양한 음악을 체계적으로 정리한다는 것을 의미한다. 빅데이터를 활용하여 이용자에 맞는 노래를 제시하는 추천시스템 개발은 음악 플랫폼 회사들 사이에서 가장 핫한 이슈인데, 해당 기술이 선호하는 장르를 최우선적으로 고려하여 개발되는 것도 같은 맥락인 것이다.

본 프로젝트는 스웨덴의 음악 스트리밍 및 미디어 서비스 제공 업체인 Spotify 에서 제공하는 데이터를 활용하였다. 수집된 데이터를 바탕으로 리듬, 음향, 가사의 선정성 등 다양한 음악적 요소를 통해 2000 개의 음악 데이터를 장르별로 구분하고자 한다.

그러나 대회 특성상 한정된 데이터를 가지고 분석에 진행하기에 분석 모델을 설계하는 부분에 있어 False 에 비해 True 가 현저히 적은 장르가 다수 존재한다. 장르의 종류 또한 굉장히 다양하기 때문에 이를 모두 분석하기에는 시간적인 한계가 있다. 따라서 본 프로젝트에서는 장르의 빈도수가 가장 높은 순으로 4 개의 장르(Pop, Hiphop, R&B, Dance/Electronic)를 각 모델의 기준으로 하여 4 개의 이진분류 모델을 설계하도록 하였다.

2. 데이터 탐색

2-A. 데이터 설명

분석에 사용할 데이터는 음악 스트리밍 사이트인 Spotify 의 인기곡 정보를 담은 데이터이다.

관측치의 개수는 총 2000 개이며 18 개의 변수로 이루어져 있으며 종속 변수는 genre 이다. 각 변수들의 의미와 타입은 아래의 표와 같다.

변수명	변수의 의미	타입
artist	아티스트의 이름	STR
song	곡의 이름	STR
duration_ms	밀리초 단위의 곡의 길이	INT
explicit	어린이에게 불쾌감을 주거나 부적절하다고 간주될 수 있는 것의 여부	BOOL
year	곡의 출시 연도	INT
popularity	곡의 인기도	INT
danceability	곡이 춤에 얼마나 적합한지	FLOAT
energy	곡의 강도와 활동성의 척도	FLOAT
key	곡의 키(음역대)	INT
loudness	곡 전체의 데시벨	FLOAT
mode	곡의 장조와 단조의 여부	INT
speechiness	곡의 음성 단어의 존재	FLOAT
acousticness	곡이 음향인지 여부	FLOAT
instrumentalness	곡에 보컬이 없는지 여부	FLOAT
liveness	녹음에서 청중의 존재 감지	FLOAT
valence	곡이 전달하는 음악적 긍정성	FLOAT
tempo	곡의 BPM	FLOAT
genre	곡의 장르	OBJECT

[표 1] 특성 의미 및 자료형

2-B. 데이터 전처리 및 시각화

2-B-가. 데이터 전처리

각 변수는 모두 2,000 개의 행을 갖고 있으므로 확인되는 결측치는 없었다.

genre 를 확인했을 때 두 개 이상의 장르가 혼합되어 있는 경우가 많았다. 곡의 장르를 분류하는 것이 목적이므로 여러 개의 장르가 합쳐져 있는 형태가 아닌 하나의 장르만을 확인할 필요가 있었다. 따라서 이를 처리하기 위해서 genre 를 one-hot encoding 하여 15 개의 열을 추가했다.

또한 분석에 앞서 BOOL 타입인 explicit 를 label encoding 하여 INT 타입으로 변경했다.
앞선 전처리 과정을 거친 데이터 셋은 아래와 같다. 데이터 셋은 2000 개의 관측치와 33 개의 변수를 가진다.

앞선 전처리 과정을 거친 데이터 셋은 아래와 같다. 데이터 셋은 2000 개의 관측치와 33 개의 변수를 가진다.

	artist	song	duration_ms	explicit	year	popularity	danceability	energy	key	loudness	...
0	Britney Spears	Oops!...I Did It Again	211160	False	2000	77	0.751	0.834	1	-5.444	...
1	blink-182	All The Small Things	167066	False	1999	79	0.434	0.897	0	-4.918	...
2	Faith Hill	Breathe	250546	False	1999	66	0.529	0.496	7	-9.007	...
3	Bon Jovi	It's My Life	224493	False	2000	78	0.551	0.913	0	-4.063	...
4	*NSYNC	Bye Bye Bye	200560	False	2000	65	0.614	0.928	8	-4.806	...

5 rows × 33 columns

[그림 1]

장르명	Pop	Hip hop	R&B	Dance/Electronic	Rock
개수	1633	778	452	390	234
장르명	Metal	Latin	Set()	Country	Folk/Acoustic
개수	66	64	22	21	20
장르명	World/Traditional	Easy listening	Blues	Jazz	Classical
개수	10	7	4	2	1

[표 2]

전처리를 마친 데이터 셋을 확인했을 때 15 개의 장르가 확인되었지만 장르별로 곡의 수가 균일하지 않다. 또한 대조집단에 비해서 사례집단의 수가 현저히 떨어지는 장르도 다수 존재해 분류를 위한 모델이 유의미하게 학습하지 못할 가능성이 우려된다. 이와 같은 이유로 본 프로젝트에서는 모든 장르가 아닌 사례집단과 대조집단의 비율이 1:5 를 넘어서지 않는 4 개의 장르들(Pop, Hiphop, R&B, Dance/Electronic)만을 각 모델의 기준으로 설정하고 분석한다.

2-B-나. 데이터 시각화

앞서 정한 4 개의 장르별 변수들의 히스토그램과 상자그림을 확인해보았다. 그림은 보고서의 부록에 첨부하였다.

Pop 장르는 대중의 귀를 쉽게 잡아 끌 수 있는 쉬운 멜로디와 리듬을 가진 곡이다. 이와 같은 성격이 explicit 에서 0 이 1 보다 특히 많이 나타난 것으로 보인다.

반면, Hiphop 장르는 미국 빈민가에서 시작된 거친 장르이므로 다른 장르와 비교해서 유일하게 explicit 에서 1 이 0 보다 많았다. 또한 HIP HOP 장르에서도 주류가 되는 랩은 박자에 맞춰 많은 가사를 내뱉는 특징이 있어 speechiness 의 수치가 높게 나타난 것으로 보인다. Hiphop 음악과 함께하는 Street Dance 문화가 발달함에 따라 danceability 또한 0 일 때와 비교해서 그 수치가 높게 나타났다.

R&B 는 리듬 앤 블루스의 약칭으로 비교적 우울한 분위기의 조용한 노래가 많다. 이러한 특징이 energy, loudness, tempo 에서 낮은 수치로 반영된 것으로 보인다. 또한 R&B 장르는 현대에 들어서 전성기가 지난 모습을 보이고 있는데, year 에서 시간이 지남에 따라 그 수가 적어지는 추세를 보인다.

Dance/Electronic 장르는 춤을 추기 좋은 음악과 현대의 클럽, 페스티벌, 파티에서 사용되는 전자음악을 통칭한다. 이러한 특징이 danceability, energy, loudness 에서 높은 수치로 반영된 것으로 보인다. 또한 전자 음악인 부류인 EDM 이 현대에 들어 발전하고 유행하고 있어 R&B 장르와는 반대로 year 에서 시간이 지남에 따라 그 수가 많아지는 추세를 보인다.

2-B-다. 이상치 제거

상자그림을 봤을 때 변수별로 이상치로 의심되는 데이터들이 확인되었다. IQR 을 사용하여 이러한 데이터들을 제거했을 때 전체 데이터셋에서 940 개의 데이터가 제거되어 1060 개의 데이터만 남는 결과가 도출되었다. 남은 데이터로만 분석을 진행한다면 분석이 무의미해질 우려가 있다. 또한 분석에서는 이상치의 영향이 적은 트리를 기반으로 한 모델들을 사용했다.

음악은 저작권으로 보호받는 대표적인 저작물 중의 하나이다. 음악의 제작은 표절로 인해서 제한되기 때문에 각기 다른 특성을 가진 다양한 음악들이 세상에 나오게 되므로 이러한 자연스러운 결과들이 각 변수에서 이상치처럼 보이게 나타난 것으로 해석된다. 따라서 이상치로 의심되는 데이터들은 제거하지 않았다.

2-C. 변수 선택

종속변수를 각각 <Pop>, <Hiphop>, <R&B>, <Dance/Electronic>로 하는 4 개의 데이터를 생성하였다. 각 데이터들에 맞는 4 개의 이진 분류기를 설계할 계획이며, 각 장르마다 영향을 주는 변수들이 다를 것으로 판단하였기에 변수 선택 과정 또한 각 데이터마다 독립적으로 진행하였다.

변수선택 기법은 트리 기반의 모델에서 유용하게 사용되는 Select From Model**[11]**, Permutation Importance**[12]**, MDI importance**[13]**를 사용하였으며, 팀원들이 가지고 있는 도메인 지식을 활용하여 최종적으로 변수를 선택하였다. 데이터의 변수들이 음악에 대한 전문지식을 크게 요구하지 않고, 설명이 충분히 되어있기 때문에 도메인 지식에 근거하여 변수를 선택하기에 충분하다고 판단했다.

본 프로젝트는 가수 및 제목을 제외한 순수 음악적 요소들에 따라 장르를 구분하는 분류기 모델 설계 및 분석이 목적이므로 artist 와 song 은 삭제하고, 각 장르를 종속변수로 둔다.

1. Select From Model**[11]**

   Select From Model은 Feature의 Importance가 지정한 임계치보다 큰 모든 특성을 선택하는 방법으로 관련 매개변수를 Threshold라고 하고, Default는 Mean으로 한다. 모델 자체에서 제공하는 지표에 따라 변수를 선택한다. 모델이 Corresponding Coefficients나 Important Features를 갖는다면 해당 변수선택기법을 사용할 수 있다. 해당 값들이 사전에 설정된 임계값보다 낮다면 그 변수는 중요하지 않은 것으로 간주되어 제거된다.

   특히 트리 기반 모델에서 Select From Model은 의미 있는 효과를 가져온다. Decision Tree, Random Forest 등의 모델은 각 Feature의 중요도가 담겨있는 Important Features를 제공하며 이를 활용하여 한 번에 모든 특성을 고려하므로 Feature 간의 Corresponding Coefficients를 반영할 수 있는 것이다.

2. Permutation Importance**[12]** (순열 중요도)

Permutation Importance는 "특정 변수가 랜덤으로 분포된다면, 얼마나 성능이 떨어지는가?" 는 가정을 가지고 있다. 여러 변수 중에 특정 변수에만 의존했을 경우 무작위로 섞었을 때 모델의 오차가 증가하게 된다면 해당 변수는 중요하다고 판단되며, 오차의 변화가 없거나 감소한다면 그 변수는 불필요하다고 판단된다.

트리 모델에서 기본적으로 제공되는 Feature Importance의 경우 부정적인 영향을 주는 Feature까지는 알 수 없다. 즉, 기존의 변수 중요도는 불필요한 Feature을 탈락시키는 과정에 있어서 큰 도움을 받을 수 없는 반면, Permutation Importance의 경우 이러한 문제점이 보완된 방법론이라고 할 수 있다. 해당 변수선택법은 계산이 빠르고 직관적이며, 사용범위가 넓어 이해하기 쉽다. 또한 일관된 Feature의 중요도를 측정할 수 있는 장점을 가지고 있다.

Permutation Importance 를 수행한 결과는 부록에서 확인할 수 있다.

3. MDI(Mean Decrease in Impurity) Importance**[13]**

트리 기반의 모델에서는 각 노드가 적절한 지니 불순도(Gini impurity) 지표를 기준으로 자식 노드를 생성한다. 불순도가 낮을수록 잘 분류된 것이며 모델은 불순도를 최소화하는 방향으로 분류를 진행한다.

MDI Importance는 각 노드가 자식 노드를 생성할 때 지니 불순도 감소분의 평균을 노드 중요도로 정의한다. 각 노드의 관측치 개수를 고려하여 불순도 감소분이 계산되며, 값이 클수록 중요도가 높다. 트리 모형을 활용하여 이진분류를 진행하기에, 노드 불순도를 활용한 변수선택방법 역시 유의미한 결과를 도출할 수 있을 것이라는 판단을 하였다.

MDI Importance 를 수행한 결과 역시 부록에서 확인할 수 있다.

4. 최종 변수 선택

앞서 살펴본 3가지 변수선택법(MDI, Permutation Importance, Select From Model)을 적용하여 도출된 결과를 취합하였다. 그러나 도메인 지식 또한 변수선택에 있어서 고려해야 할 필요가 있다.

가령, 취합한 결과에 따르면 Dance/Electronic의 경우 danceability가 제거된다. 하지만 해당 장르의 특징을 고려하였을 때, Dance/Electronic에서 danceability는 장르 분류에 있어서 충분히 유의미한 변수이다. 또한 주어진 데이터 내 변수들의 경우 전문지식이 크게 요구되지 않기에, 도메인적 측면을 고려하여 장르별 최종 변수를 선택하였다.

최종 변수 선택에 대한 결과는 **[표 3, 4, 5, 6]**에서 확인 가능하다.

변수선택법		변수
MDI	**선택**	popularity, energy, duration_ms, danceability, year, speechiness, acousticness, instrumentalness, tempo
	제거	loudness, liveness, explicit, valence, key, mode
Select From Model	**선택**	duration_ms, year, popularity, danceability, energy, loudness, speechiness, acousticness, instrumentalness, tempo
	제거	explicit, key, mode, liveness, valence
Permutation Importance	**선택**	popularity, acousticness, duration_ms
	제거	danceability, energy, explicit, key, liveness, loudness, mode, speechiness, tempo, valence, year, instrumentalness,
종합	**선택**	duration_ms, explicit, year, popularity, danceability, energy, speechiness, acousticness, instrumentalness, pop

[표 3] Pop 장르 변수 선택 결과

변수선택법		변수
MDI	선택	Explicit, speechiness, danceability, tempo, duration_ms, instrumentalness, acousticness, energy, loudness
	제거	Liveness, popularity, year, valenc, key, mode
Select From Model	선택	explicit, danceability, speechiness
	제거	duration_ms, year, popularity, energy, key, loudness, mode, acousticness, instrumentalness, liveness, valence, tempo
Permutation Importance	선택	acousticness, duration_ms, instrumentalness, year, explicit, speechiness, danceability, tempo, loudness, energy, valence, key, popularity
	제거	liveness, mode
종합	선택	duration_ms, explicit, year, popularity, danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, tempo, hip hop

[표 4] Hiphop 장르 변수 선택 결과

변수선택법		변수
MDI	선택	year, energy ,duration_ms, acousticness, tempo, speechiness, liveness, loudness, danceability, popularity
	제거	explicit, mode, key, instrumentalness, valence
Select From Model	선택	duration_ms, year, danceability, energy, speechiness, acousticness, liveness, tempo
	제거	explicit, popularity, key, loudness, mode, instrumentalness, valence
Permutation Importance	선택	danceability, duration_ms, speechiness, loudness, tempo, valence, popularity, acousticness, explicit, instrumentalness, year, energy, key, liveness, mode,
	제거	X
종합	선택	duration_ms, year, popularity, danceability ,energy, loudness, speechiness, acousticness, tempo, R&B

[표 5] R&B 장르 변수 선택 결과

변수선택법		변수
MDI	**선택**	duration_ms, explicit, year, popularity, energy, loudness, speechiness instrumentalness, tempo
	제거	acousticness, valence, liveness, danceability, key, mode
Select From Model	**선택**	duration_ms, year, popularity, energy, loudness, instrumentalness, tempo
	제거	explicit, danceability, key, mode, speechiness, acousticness, liveness, valence
Permutation Importance	**선택**	instrumentalness, energy, duration_ms, tempo, loudness, popularity, explicit, speechiness, year, acousticness, danceability, key, liveness, valence
	제거	mode
종합	**선택**	duration_ms, explicit, year, popularity, danceability, energy, loudness, speechiness, instrumentalness, tempo, Dance/Electronic

[표 6] Dance/Electronic 장르 변수 선택 결과

3. 모델 선정

3-A. 오버샘플링

변수 선택 후, Random Forest 모델을 사용하여 분류한 결과는 **[표 7]**에서 확인할 수 있다.

장르	Accuracy	Precision	Recall	F1
Pop	0.8550	0.8658	0.9758	0.9175
Hiphop	0.8833	0.8899	0.8178	0.8523
R&B	0.8583	0.8415	0.4894	0.6188
Dance/Electronic	0.8583	0.8000	0.2330	0.3609

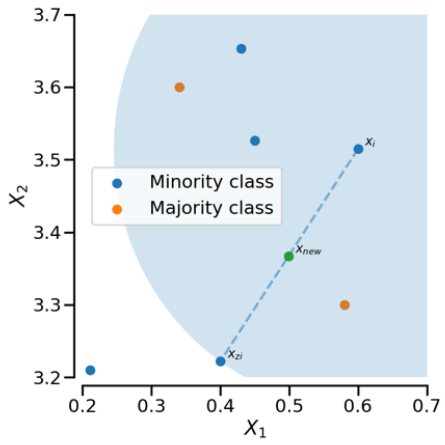
[표 7]

Pop 과 Hiphop 장르에 비해 R&B 와 Dance/Electronic 장르에서 Recall 이 크게 떨어져 F1 점수가 낮아진 것을 확인할 수 있었다. R&B 와 Dance/Electronic 장르는 Pop 과 Hiphop 에 비해 사례집단의 비율이 매우 낮기 때문에 이런 결과로 이어진 것으로 보인다.

이러한 불균형한 데이터를 보완하는 방법으로는 언더샘플링과 오버샘플링 방법이 있다. 하지만 Spotify 데이터는 총 2,000 개의 데이터로, 데이터의 수가 많지 않기 때문에 언더샘플링 기법을 사용할 경우 데이터가 크게 줄어들 수 있다. 따라서 오버샘플링 기법을 사용하여 데이터를 보완해 모델의 성능을 높였다. 데이터에 적용한 오버샘플링 기법은 다음과 같다.

1. SMOTE [2]

임의의 소수 클래스 데이터로부터 인근 소수 클래스 사이에 새로운 데이터를 생성하는 방법이다. 임의의 소수 클래스에 해당하는 관측치 X를 선정하고, 그 X로부터 가장 가까운 K개의 이웃 이웃을 찾는다. 그리고 K개의 이웃과 X 사이에 임의의 새로운 데이터를 생성한다.



[그림 2] [5]

$$x_{new} = x_i + \lambda(x_{zi} - x_i)$$

2. Borderline SMOTE (B. SMOTE) [3]

임의의 소수 클래스 데이터로부터 가장 근접한 K 개의 데이터를 찾아 K 개의 클래스 수에 따라 Danger, Safe, Noise 3 개로 구분하고, SMOTE 를 실시한다.

(1) X에 가장 근접한 K개의 클래스가 전부 다수 클래스인 경우 : Noise

→ SMOTE를 적용하지 않는다.

(2) X에 가장 근접한 K개 중 적어도 절반은 소수 클래스인 경우 : Danger

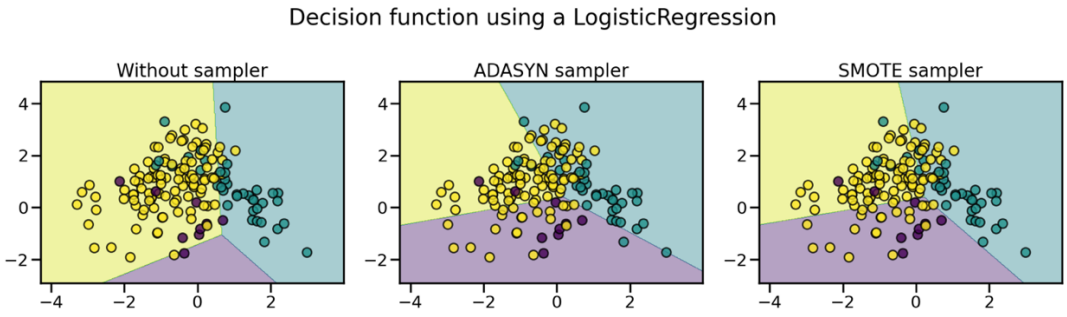
→ SMOTE를 적용한다.

(3) X에 가장 근접한 K개의 클래스가 전부 소수 클래스인 경우 : Safe

→ SMOTE 를 적용하지 않는다. 이미 X 주변에는 소수 클래스의 데이터가 적당히 분포한다고 보기 때문이다.

3. ADASYN [4]

임의의 소수 클래스 데이터로부터 가장 근접한 K개의 데이터를 찾아 K개의 클래스 수에 따라 가중치를 적용해 SMOTE를 실시한다.



[그림 3] [5]

[1]에 따르면 사례집단과 대조집단의 불균형 비율이 1:5 를 넘어서는 경우 오버샘플링 방법을 사용하는 것이 효과적이고, 오버샘플링의 비율은 1:2 와 1:3 의 사이에서 가장 효과적이었다고 한다.

장르별 사례집단과 대조집단의 비율은 **[표 8]**에서 확인할 수 있다.

장르	사례집단 수 : 대조집단 수	비율
Pop	1633 : 367	4.45 : 1
Hiphop	778 : 1222	1 : 1.57
R&B	452 : 1548	1 : 3.42
Dance/Electronic	390 : 1610	1 : 4.13

[표 8] 장르별 사례집단 비율

예측할 장르 중 사례집단과 대조집단의 비율이 1:5 를 넘어서는 경우는 없지만, R&B 와 Dance/Electronic 장르에서의 Random Forest 모델 결과가 데이터 불균형으로 인해 Random Forest 모델의 성능이. 따라서 R&B 및 Dance/Electronic 장르의 모델들은 오버샘플링을 적용했다. 오버샘플링의 비율은 되도록이면 논문에 따르되, Train set 와 Test set 의 결과를 비교해 너무 과대적합되지 않는 선에서 비율을 조정해보면서 Test set 에서의 성능이 가장 좋았던 비율로 선정했다.

오버샘플링 비율을 정확하게 선정하기 위해 Random Forest, LGBM 및 XGBoost 모델 하이퍼 파라미터 튜닝 시 비율 및 오버샘플링 기법도 튜닝을 통해 최적화했다. 장르마다 각 분류 모델에 모든 오버샘플링 기법을 적용해 보았고, 각 오버샘플링의 최적의 하이퍼 파라미터 및 오버샘플링 비율을 찾아보았다. 각 모델의 Test 결과는 **[표 9, 10, 11]**에서 확인할 수 있다.

3-B. 모델 소개 및 선정 이유

1. Random Forest Classifier (Bagging 기반) **[6]**

a. 개요

여러 개의 트리 구조를 Bagging 기법으로 앙상블 한 모형이다. 여기서 Decision Tree 는 계층 구조로 이루어진 노드들과 Edge 들의 집합으로 이루어져 있다. 노드는 시작 노드를 제외하고 Internal Node 와 Terminal Node(마지막 노드)로 이루어져 있는 구조이다. 이런 Decision Tree 들을 모아놓은 것이 Random Forest 이며, Random Forest 의 가장 큰 특징은 랜덤성에 의해 트리들이 서로 조금씩 다른 특성을 갖는다. 이 특성들은 각 트리들의 예측들이 비상환화 되게 하며, 결과적으로 일반화 성능을 향상시킨다. 또한 랜덤화는 Forest 가 Noise 가 포함된 데이터에서도 많은 영향을 받지 않게 만든다.

b. 선정 이유

우리의 목표는 장르를 분류할 수 있는 모델을 만드는 것이다. 가장 간단한 분류 모델은 Decision Tree 이지만, Decision Tree 는 이상치에 영향을 많이 받는 모델이며 Decision Tree 에 비해 일반화 할 수 있는 결과값을 볼 수 있기 때문에 앙상블 기반 가장 많이 사용되는 분류 모델인 Random Forest 를 사용하게 되었다.

2. Gradient Boosting Machine (Boosting 기반) **[7]**

a. 개요

Boosting 알고리즘은 여러 개의 약한 학습기를 순차적으로 학습/예측하며 잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가면서 학습하는 방식이다.

b. Light GBM(LGBM) **[8]**

i) 개요

일반적인 GBM 계열의 트리 분할 방식은 Level-wise 방식을 이용한다. 이 방법을 이용하면 최대한 균형 잡힌 트리를 생성하면서도 트리의 깊이를 최소화할 수 있다는 장점이 있다. 이런 방식으로 트리를 생성할 경우 과대적합 문제에 더 강한 구조를 갖게 되지만 균형을 맞추기 위한 시간이 필요하다는 단점이 있다.

그러나 LGBM 은 트리의 균형을 맞추지 않고 최대 손실 값을 가지는 Terminal Node 를 지속적으로 분할하면서 트리가 깊어지고 비대칭적 트리를 만든다. 이런 방식으로 트리를 계속 분할하여 결국 균형 트리 분할 방식보다 예측 오류 손실을 최소화한다는 방법을 이용한 것이다. 또한 Category 형 변수도 그대로 학습이 되어 One-hot 으로 Encoding 한 경우보다 성능이 더 좋아진다.

ii) 선정 이유

Boosting 기반의 트리 모델 중 실행 속도가 빠르고, XGBoost 와 유사한 성능을 보이기 때문에 선정했다.

c. XGBoost **[9]**

i) 개요

Extreme Gradient Boosting 의 약자로, 극한 변화도 Boosting 기법이다. Boosting 기법을 이용하여 구현한 알고리즘은 Gradient Boost 가 대표적이며 이 알고리즘을 병렬학습이 지원되도록 구현한 라이브러리가 XGBoost 이다. LGBM 과 달리 Level-wise 방식을 사용하여 균형 잡힌 트리를 만들 수 있다.

ii) 선정 이유

병렬 처리로 학습, 분류 속도가 빠르며 과적합을 규제할 수 있다. CART 앙상블 모델을 사용하여 분류와 회귀 영역에서 뛰어난 예측 성능을 발휘하며 결측치를 내부적으로 처리할 수 있어 선정하였다.

3. 하이퍼파라미터 튜닝

a. GridSearchCV **[10]**

모델 하이퍼파라미터에 넣을 수 있는 값들을 순차적으로 입력한 뒤에 교차 검증을 통해 가장 높은 성능을 보이는 하이퍼파라미터를 찾는 탐색방법이다. 모든 모델의 하이퍼파라미터 튜닝에 Grid search 를 이용하여 모델의 성능을 높여보았다.

3-C. 모델 성능 비교

모든 모델 실험 시, Test set 의 비율은 전체 데이터의 20%로 설정했다.

오버샘플링 결과

장르	오버샘플링	비율	Accuracy	Precision	Recall	F1
R&B	-	1 : 3.42	0.8750	0.8194	0.6146	0.7024
	SMOTE	1 : 2.5	0.8750	0.7649	0.6458	0.7126
	B. SMOTE	1 : 2.5	0.8825	0.8025	0.6771	0.7245
	ADASYN	1 : 2.5	0.8750	0.8026	0.6354	0.7093
Dance/ Electronic	-	1 : 4.13	0.8825	0.7255	0.5286	0.6116
	SMOTE	1 : 1.43	0.8725	0.6144	0.7285	0.6666
	B. SMOTE	1 : 1.67	0.8725	0.6266	0.6714	0.6482
	ADASYN	1 : 2	0.8775	0.6479	0.6571	0.6525

[표 9] Random Forest 모델의 Test set 에 대한 평가지표

장르	오버샘플링	비율	Accuracy	Precision	Recall	F1
R&B	-	1 : 3.42	0.8550	0.7317	0.6250	0.6742
	SMOTE	1 : 2.5	0.8725	0.7508	0.6979	0.7243
	B. SMOTE	1 : 2.5	0.8575	0.7092	0.6875	0.6984
	ADASYN	1 : 2.5	0.8600	0.7326	0.6563	0.6923
Dance/ Electronic	-	1 : 4.13	0.8650	0.6290	0.5571	0.5909
	SMOTE	1 : 2.5	0.8725	0.6203	0.7000	0.6577
	B. SMOTE	1 : 2.5	0.8925	0.6849	0.7142	0.6993
	ADASYN	1 : 2.5	0.8650	0.6052	0.6571	0.6301

[표 10] LGBM 모델의 Test set 에 대한 평가지표

장르	오버샘플링	비율	Accuracy	Precision	Recall	F1
R&B	-	1 : 3.42	0.8675	0.7867	0.6146	0.6901
	SMOTE	1 : 2.5	0.8675	0.7416	0.6876	0.7135
	B. SMOTE	1 : 2.5	0.8725	0.7528	0.6979	0.7243
	ADASYN	1 : 2.5	0.8500	0.6800	0.7083	0.6939
Dance/ Electronic	-	1 : 4.13	0.8700	0.6290	0.5571	0.5909
	SMOTE	1 : 3.33	0.8675	0.6164	0.6429	0.6294
	B. SMOTE	1 : 3.33	0.8750	0.6389	0.6571	0.6479
	ADASYN	1 : 2.5	0.8725	0.6301	0.6571	0.6434

[표 11] XGBoost 모델의 Test set 에 대한 평가지표

[표 9]의 Random Forest 의 결과를 보면 오버샘플링 후 정확도는 큰 차이를 보이지 않았지만, F1 점수들이 증가한 것을 확인할 수 있다. R&B 장르에서는 Borderline SMOTE 를 적용한 모델이 가장 큰 F1 점수를 보였으며, 특히 Trade-off 관계를 갖는 Precision 과 Recall 에 대해 ADASYN 과 거의 비슷한 Precision 을 보이면서 Recall 은 더 큰 것을 볼 수 있다. Dance/Electronic

장르에서는 SMOTE 가 가장 큰 F1 점수를 보였지만, Precision 과 Recall 이 가장 유사했던 방법은 ADASYN 이었다.

[표 10]의 LGBM 결과에서는 R&B 장르에서 SMOTE 오버샘플링을 통해 모든 지표를 크게 증가시킬 수 있었다. 정확도를 높인 것은 물론, Precision 과 Recall 을 모두 크게 증가시키면서 유일하게 0.7 대의 F1 점수를 보였다. Dance/Electronic 장르에서는 Precision 은 많이 줄지 않았지만, Recall 을 크게 증가시키면서 성능을 높였다. 특히 Borderline SMOTE 가 좋은 성능을 보였다.

마지막으로 **[표 11]**의 XGBoost 결과에서는 R&B 장르에서 오버샘플링을 통해 Recall 을 크게 증가시킬 수 있었다. 가장 좋았던 F1 점수를 보인 방법은 Borderline SMOTE 였다. Dance/Electronic 에서는 Borderline SMOTE 가 모든 지표에서 가장 좋은 모습을 보였고, ADASYN 역시 비슷한 점수를 보였다.

장르 예측 모델의 경우 Recall 이 높아야 하는 암환자 분류, Precision 이 높아야 하는 스팸 메일 분류와 달리 Recall 과 Precision 이 모두 중요한 경우라고 볼 수 있다. 따라서 최적의 하이퍼 파라미터 선택 시 F1 을 중점으로 선택했다. 또한 최종 앙상블 모델 선택 시, F1 점수를 통해 앙상블에 적용할 모델들을 선택했다. 표에서 색칠된 모델들이 앙상블에 사용된 모델들이며, 최종 모델에 대한 자세한 설명은 다음 섹션에서 설명하겠다.

4. 결론

4-A. 최종 선택 모델

최종적으로 각 장르별로 최적 모델들의 앙상블을 진행했다. 앙상블 방법으로는 Voting 을 사용했으며, Hard Voting 과 Soft Voting 두 가지 방법을 모두 사용했다.

Voting	비율 [RF, LGBM, XGB]	Accuracy	Precision	Recall	F1
Hard	[0.5, 0.25, 0.25]	0.8950	0.9114	0.9667	0.9382
Soft	[0.1, 0.5, 0.4]	0.8975	0.9140	0.9667	0.9396

[표 12] Pop 앙상블 결과

Pop 장르에서는 앙상블을 통해 성능이 향상되었다. 그리고 Hard Voting 보단 Soft Voting 의 결과가 더 향상된 것을 확인할 수 있었다.

Voting	비율 [RF, LGBM, XGB]	Accuracy	Precision	Recall	F1
Hard	[0.4, 0.2, 0.4]	0.9025	0.8765	0.8820	0.8793
Soft	[0.2, 0.3, 0.5]	0.9025	0.8765	0.8820	0.8793

[표 13] Hiphop 앙상블 결과

Hiphop 장르에서도 앙상블을 통해 성능이 향상되었다. 하지만 Hard Voting 과 Soft Voting 의 성능 차이는 없었다.

Voting	비율 [RF, LGBM, XGB]	Accuracy	Precision	Recall	F1
Hard	[0.6, 0.2, 0.2]	0.8875	0.8072	0.6979	0.7486
Soft	[0.9, 0.05, 0.05]	0.8825	0.7952	0.6875	0.7374

[표 14] R&B 앙상블 결과

R&B 장르에서는 앙상블을 통한 성능 향상을 볼 수 없었다. Hard Voting 수행 시 가장 성능이 잘 나온 Random Forest 와 같은 결과가 나왔고, Soft Voting 은 더 낮은 결과를 보여주었다.

Voting	비율 [RF, LGBM, XGB]	Accuracy	Precision	Recall	F1
Hard	[0.4, 0.5, 0.1]	0.8850	0.6500	0.7429	0.6933
Soft	[0.05, 0.9, 0.05]	0.8825	0.6533	0.7000	0.6759

[표 15] Dance/Electronic 앙상블 결과

Dance/Electronic 장르에서도 앙상블을 통해 성능 향상을 볼 수 없었다. Hard Voting 과 Soft Voting 모두 성능이 가장 잘 나온 LGBM 보다 낮은 성능을 보였다.

4-B. 분석 결론 및 의의

최종적으로 Spotify 데이터에 대한 4 개의 장르 예측 모델을 생성했다.

먼저 각 데이터가 여러 장르를 가진 경우가 많았기 때문에 하나의 장르만을 예측하는 모델보다는 각 장르를 Binary 로 따로 예측하면서 여러 장르를 한 번에 예측할 수 있도록 했다. 이때 선정한 4 개의 장르는 섹션 1 에서 말했듯이 데이터의 불균형이 1:5 를 넘어서지 않는 장르들로 선정했다. Spotify 의 데이터는 데이터의 수가 적어 과대적합의 위험이 있고, 데이터 불균형 문제로 학습이 잘 안될 수 있기 때문에 위와 같이 선정했다.

각 장르를 One-hot 으로 Encoding 하여 장르별 데이터셋을 따로 만들었고, Random Forest 모델을 기준으로 변수 선택을 진행했다. 변수 선택 기법으로는 Permutation Importance, MDI, Select From Model 을 사용했고, 추가적으로 도메인적 지식에 기반하여 결과를 조합해 장르별로 최종 변수들을 선정했다.

이렇게 선정한 4 개의 장르 데이터 중, R&B 와 Dance/Electronic 장르는 데이터 불균형으로 인해 매우 낮은 F1 점수를 보여 오버샘플링 기법을 사용하여 이를 보완했다. 오버샘플링 기법으로는 SMOTE, Borderline SMOTE, ADASYN 을 이용했다. 각 장르의 오버샘플링 비율은 하이퍼파라미터 튜닝 시에 Test set 의 지표와 [1]의 기준으로 선정했다. 오버샘플링을 통해 위 두 장르에 대한 F1 점수를 높일 수 있었다.

각 장르에 대한 분류 예측 모델은 트리 기반의 Random Forest, LGBM, XGB 를 사용했다. 트리 기반 모델은 이상치와 데이터 불균형에 비교적 영향을 덜 받기 때문에 트리 기반 모델들로 선정했다. 각 모델의 하이퍼파라미터 튜닝은 GridSearchCV 를 통해 선정했고, 각 모델들을 앙상블하여 최종 모델을 만들었다.

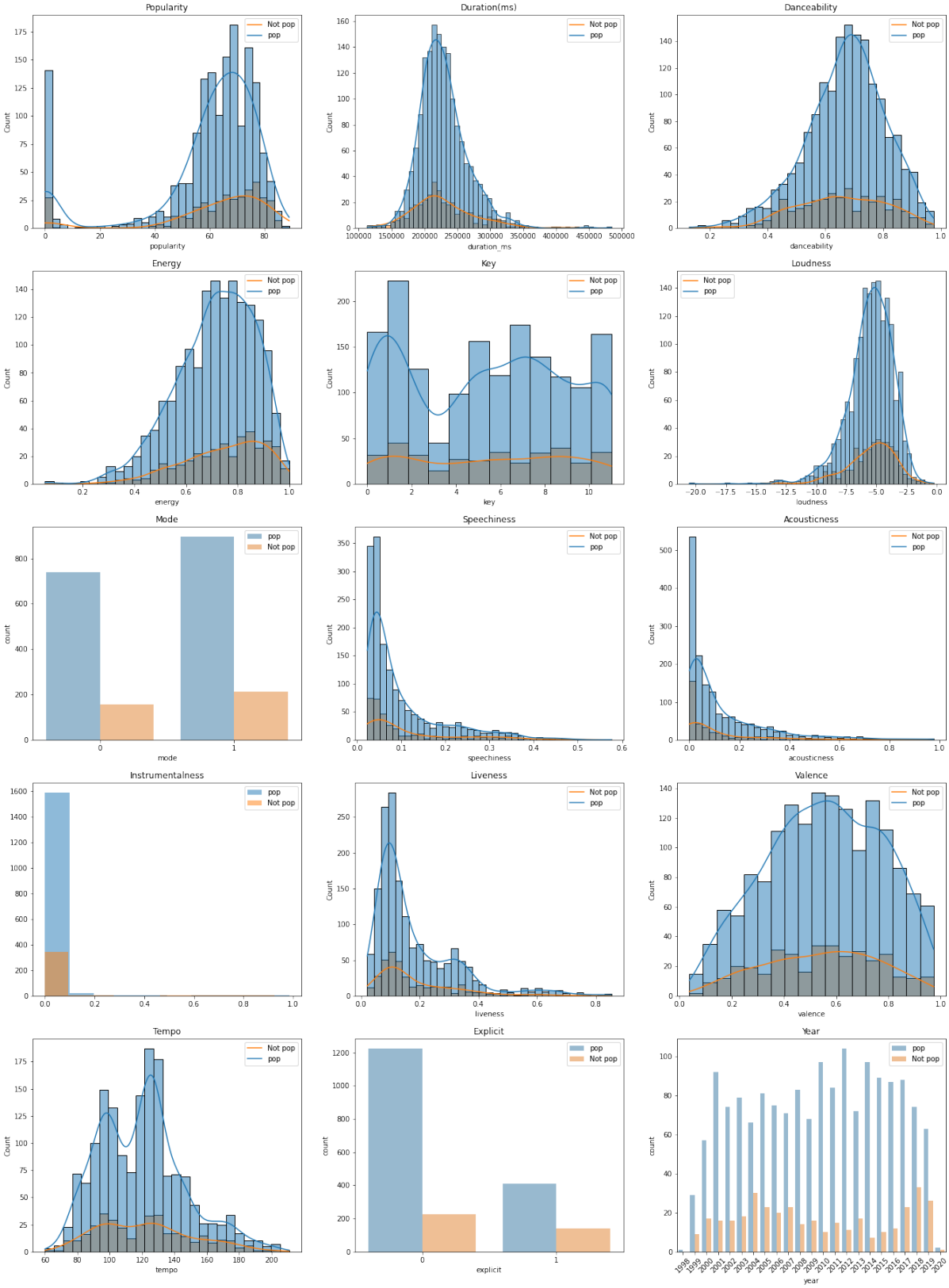
이번 실험을 통해서 각 모델에 대한 변수 선택을 진행하여 성능 향상을 이끌었고, 오버샘플링을 통해 R&B 와 Dance/Electronic 장르에 대한 성능 지표를 많이 끌어올릴 수 있었다. 또한 Grid Search 를 통해 각 모델의 최적의 성능을 이끌어 낼 수 있었다.

4-C. 한계점

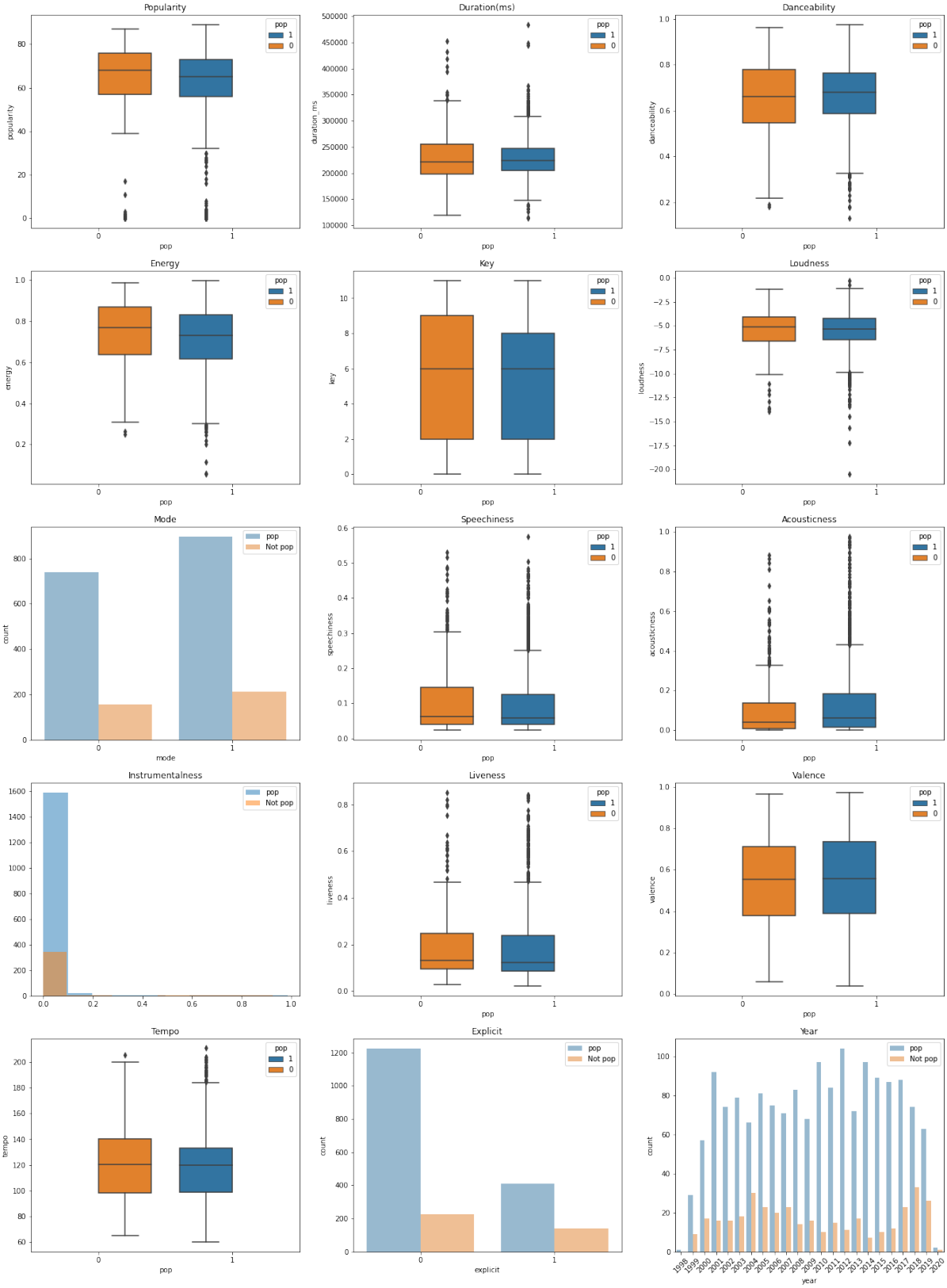
우선 데이터 수가 2,000 개로 적었고, 각 장르에 대한 데이터 불균형이 심해 분류 예측 생성에 어려움이 있었다. 실제로 R&B 와 Dance/Electronic 장르는 오버샘플링을 통해 이를 보완했지만, 데이터 수가 많았던 Pop 와 Hiphop 장르에 비하면 떨어지는 성능을 보였고, 나머지 장르들은 사례집단의 수가 매우 적어 분류 예측을 수행하지 못 했다.

그리고 데이터에는 실제 오디오 파일과 가사가 없기 때문에 음원 자체를 분석하지 못하고 주어진 특성만으로 분류 예측을 진행할 수 밖에 없었다. 그리고 artist 나 song 특성은 One-hot 으로 Encoding 할 경우 데이터가 매우 Sparse 해진다. 만약 단어를 Encoding 하여 유사도나 감성분석 등을 이용했다면, 더 좋은 결과를 기대할 수 있을 것 같다.

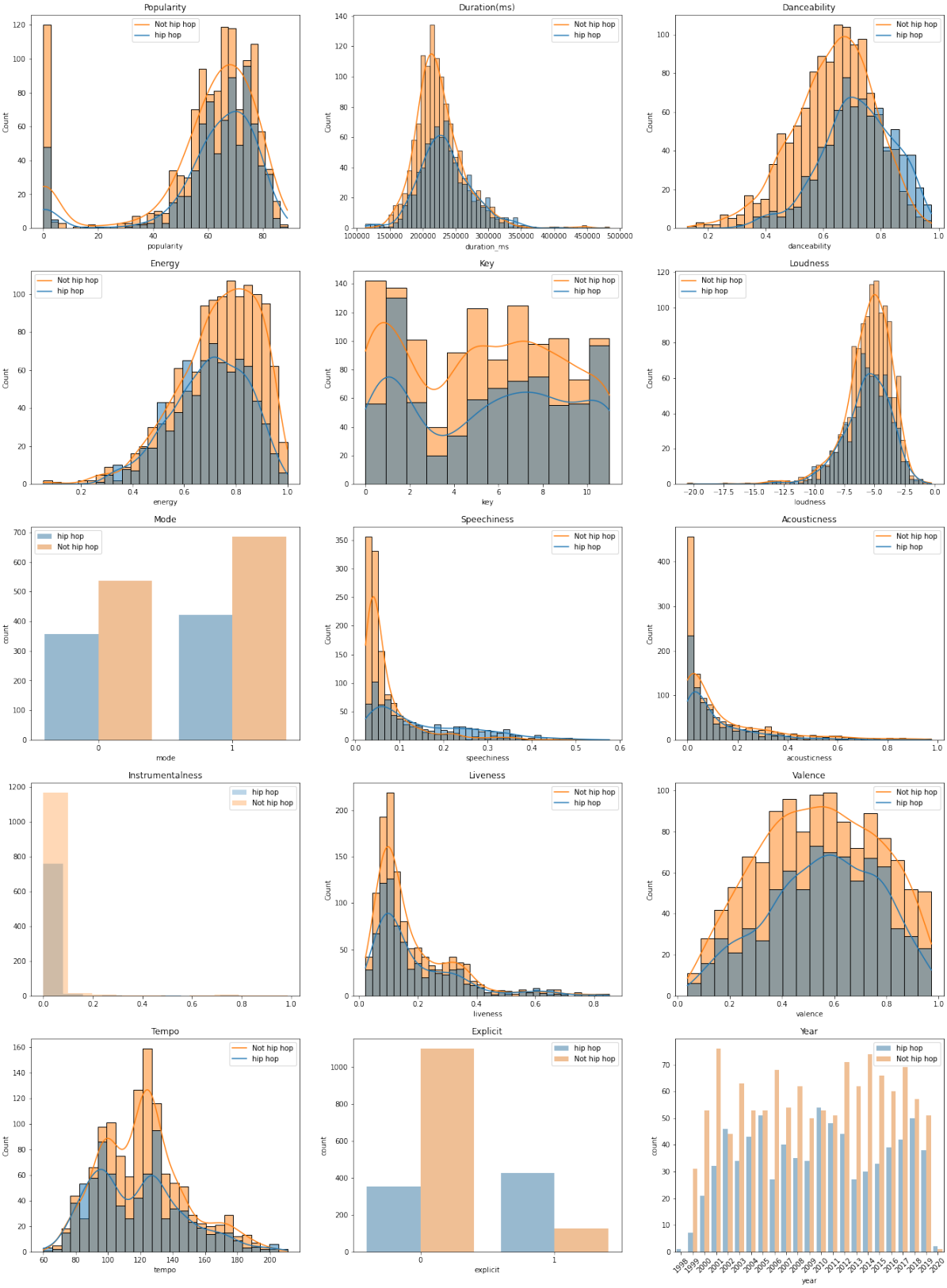
5. 부록



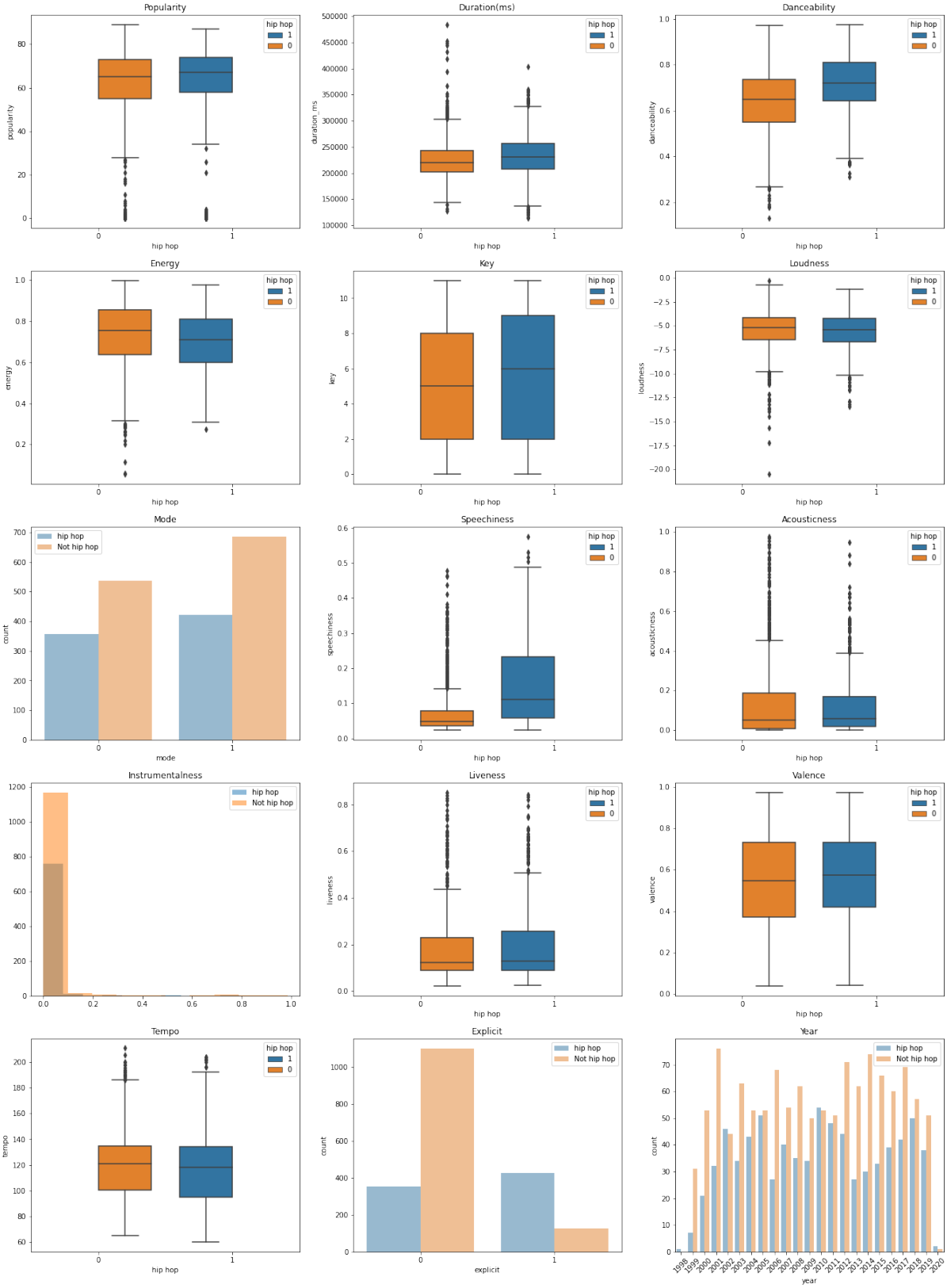
[그림 3] Pop 히스토그램


Popularity
Duration(ms)
Danceability
Energy
Key
Loudness
Mode
Speechiness
Acousticness
Instrumentalness
Liveness
Valence
Tempo
Explicit
Year
pop
Not pop
popularity
duration_ms
danceability
energy
key
loudness
mode
speechiness
acousticness
liveness
valence
tempo
explicit
count
year

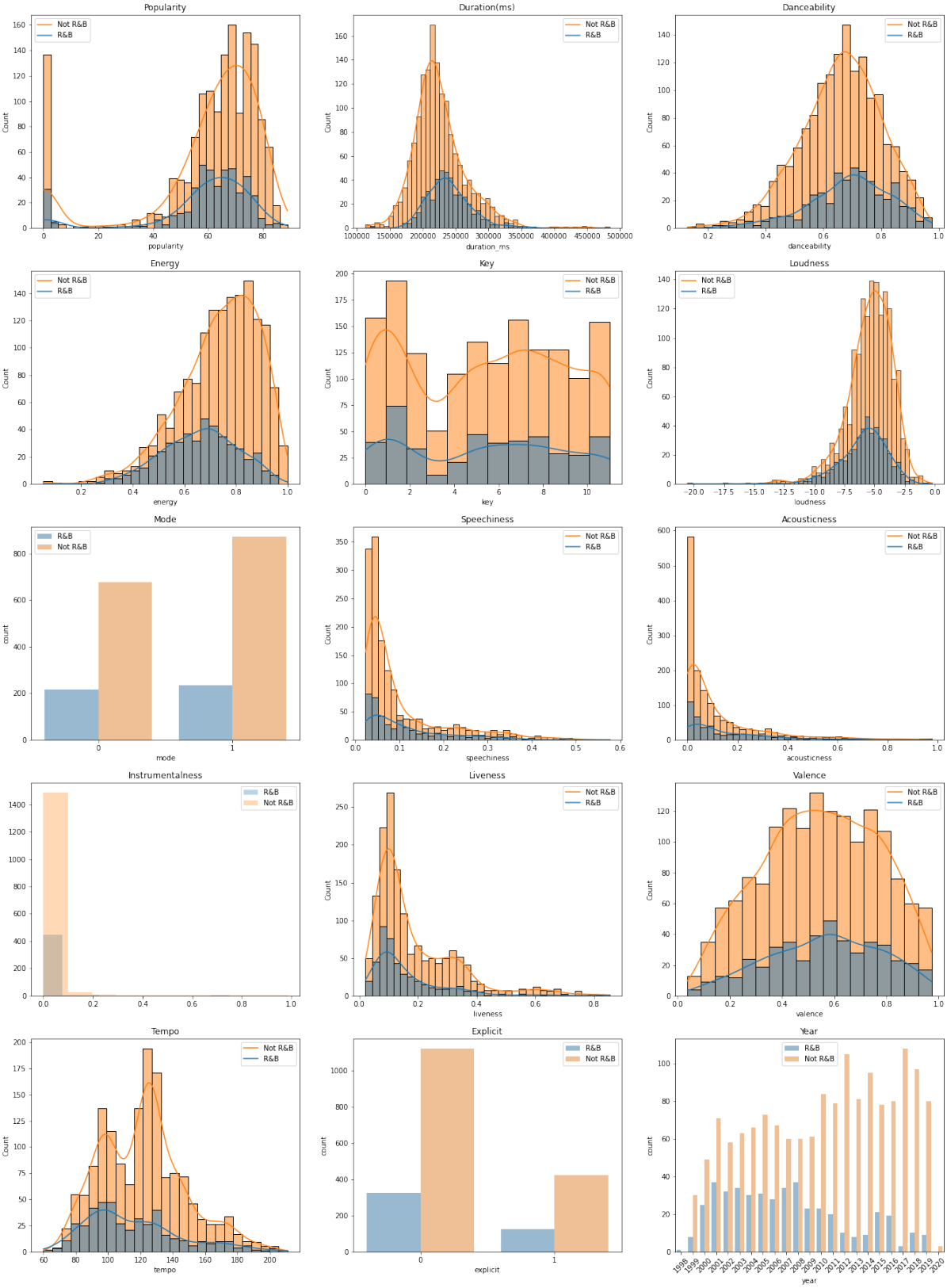
[그림 4] Pop 상자그림



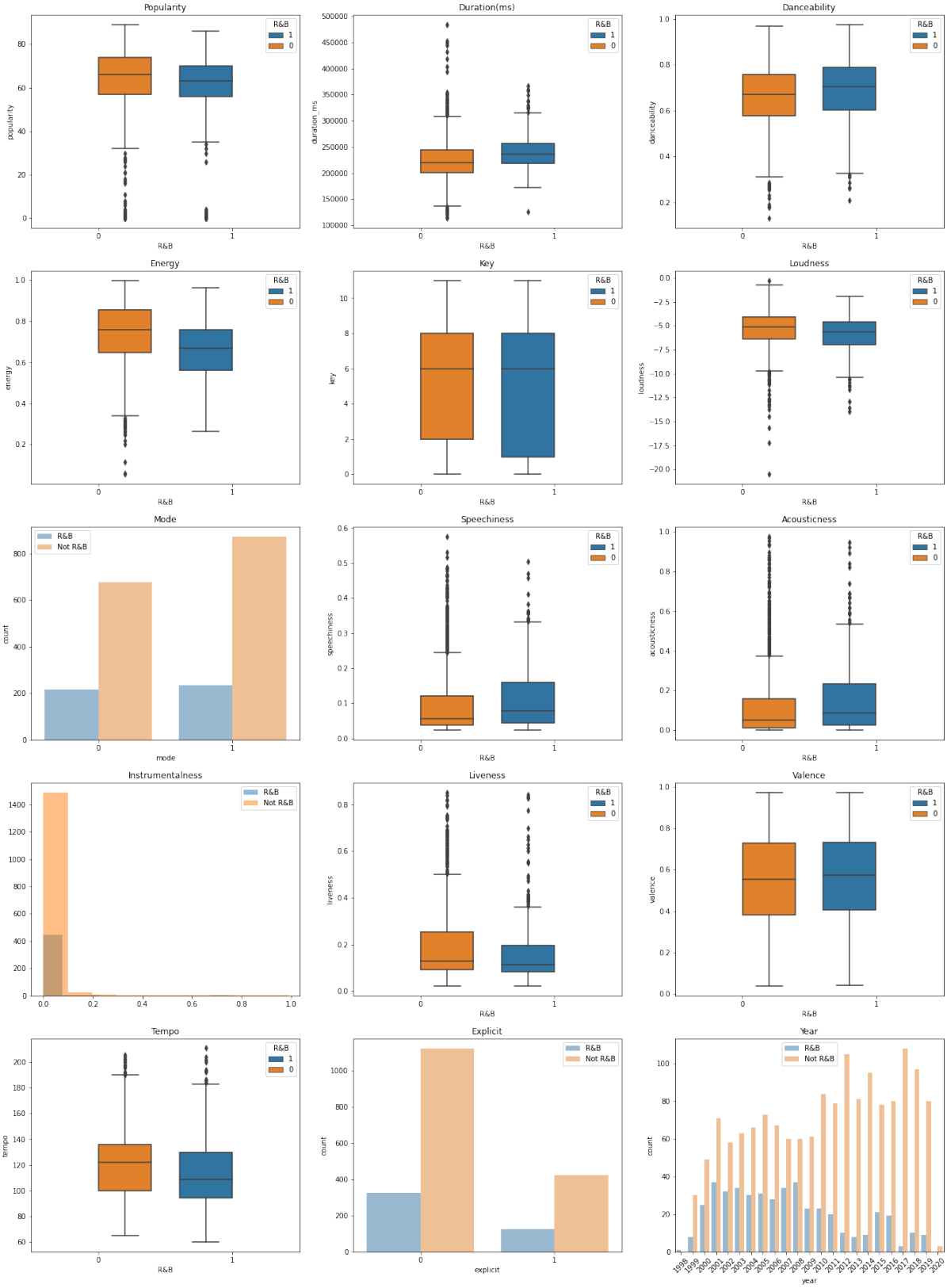
[그림 5] Hiphop 히스토그램



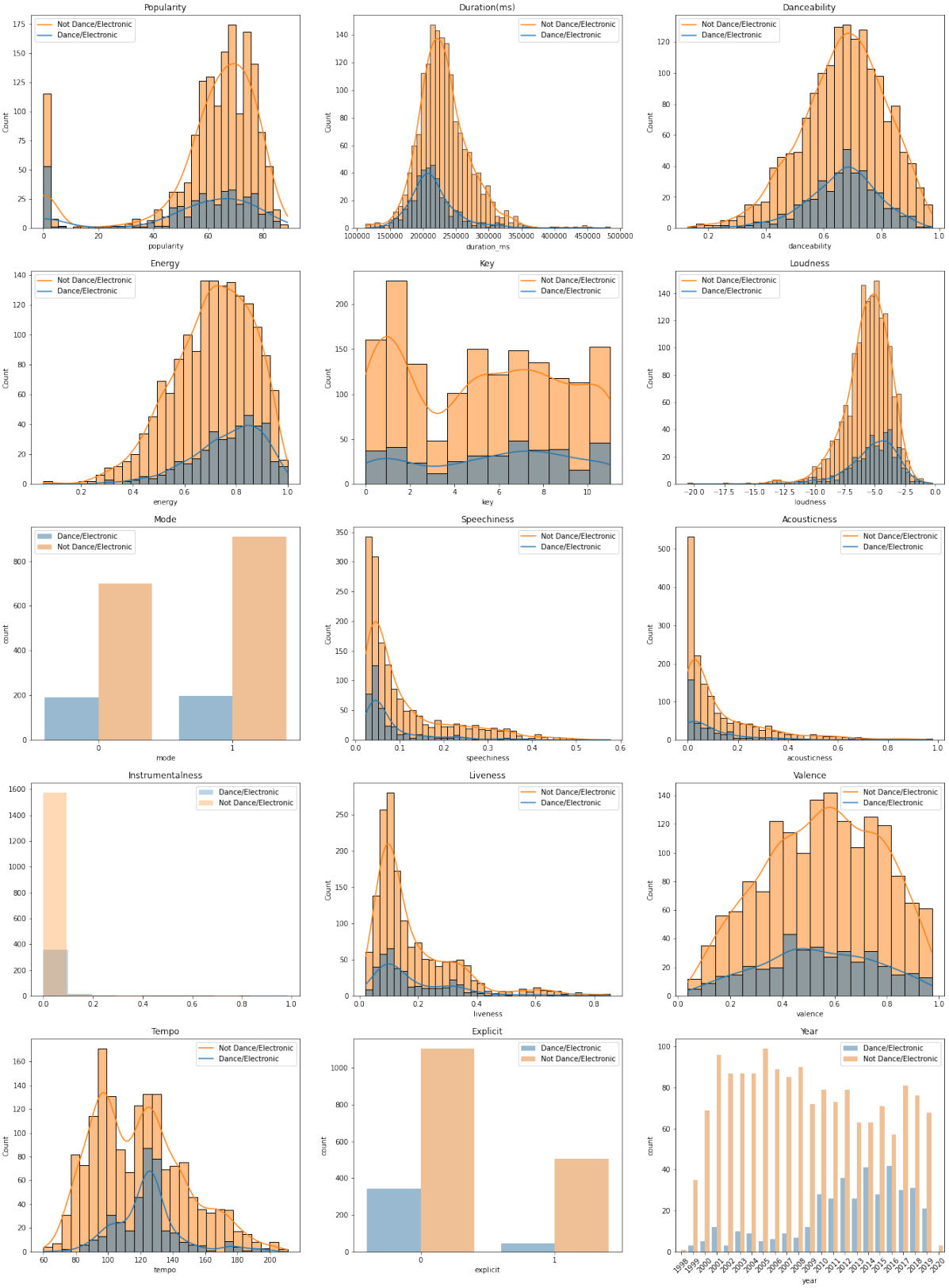
[그림 6] Hiphop 상자그림


Popularity
Duration(ms)
Danceability
Energy
Key
Loudness
Mode
Speechiness
Acousticness
Instrumentalness
Liveness
Valence
Tempo
Explicit
Year
Not R&B
R&B
popularity
duration_ms
danceability
energy
key
loudness
mode
speechiness
acousticness
liveness
valence
tempo
explicit
year
Count
count

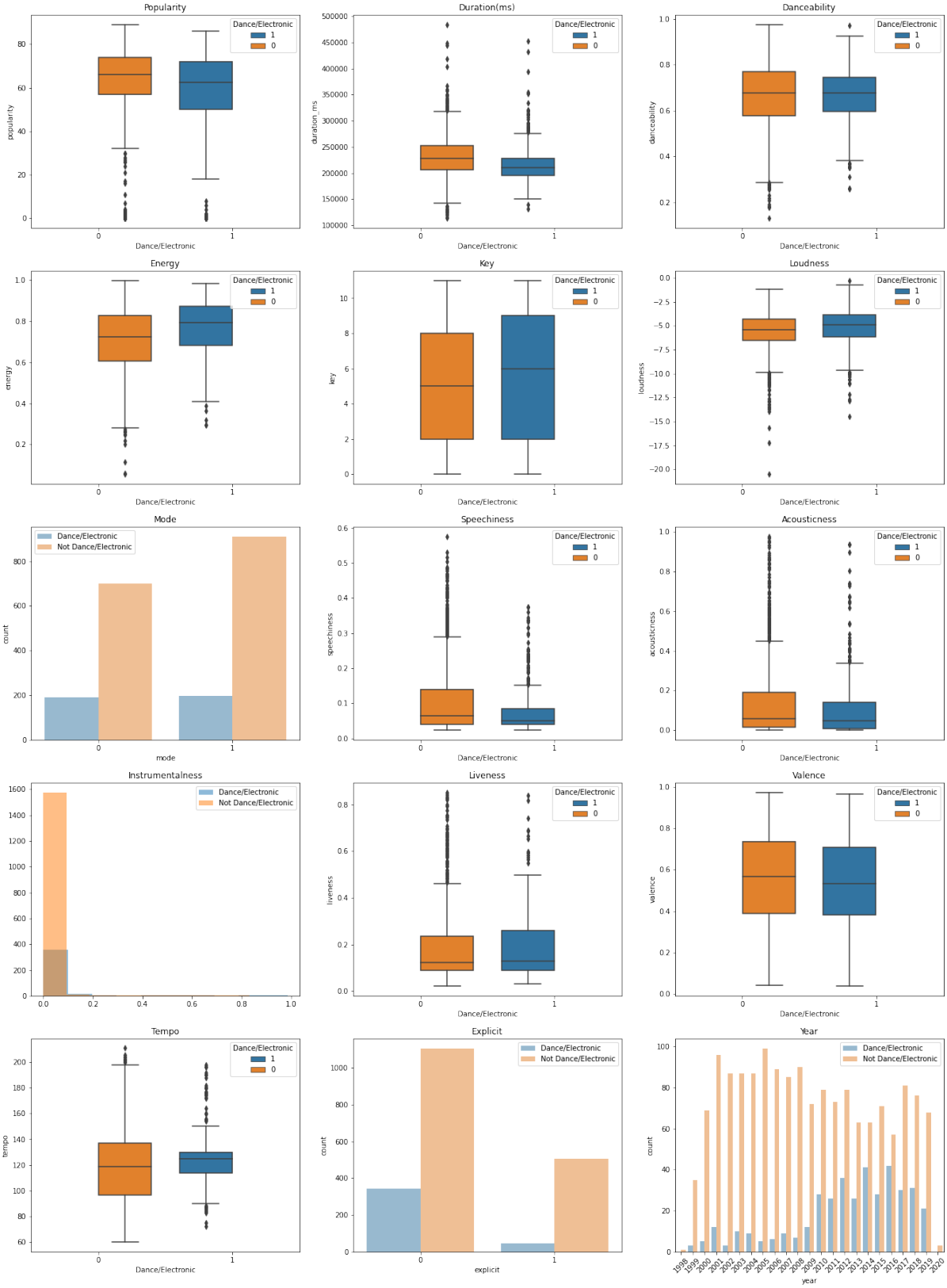
[그림 7] R&B 히스토그램


Popularity
Duration(ms)
Danceability
Energy
Key
Loudness
Mode
Speechiness
Acousticness
Instrumentalness
Liveness
Valence
Tempo
Explicit
Year
R&B
Not R&B
1
0
popularity
duration_ms
danceability
energy
key
loudness
mode
speechiness
acousticness
instrumentalness
liveness
valence
tempo
explicit
count
year

[그림 8] R&B 상자그림



[그림 9] Dance/Electronic 히스토그램



[그림 10] Dance/Electronic 상자그림

```
[Accuracy] Train: 0.8293 | Test: 0.8267
[Precision] Train: 0.8263 | Test: 0.8278
[Recall] Train: 1.0 | Test: 0.998
[F1 Score] Train: 0.9049 | Test: 0.9049
```

Weight	Feature
0.0002 ± 0.0018	popularity
0.0001 ± 0.0012	acousticness
-0.0001 ± 0.0011	duration_ms
-0.0001 ± 0.0007	energy
-0.0002 ± 0.0008	liveness
-0.0003 ± 0.0009	tempo
-0.0003 ± 0.0008	mode
-0.0004 ± 0.0015	explicit
-0.0005 ± 0.0015	speechiness
-0.0005 ± 0.0011	instrumentalness
-0.0007 ± 0.0011	year
-0.0007 ± 0.0009	danceability
-0.0008 ± 0.0012	valence
-0.0010 ± 0.0015	key
-0.0011 ± 0.0008	loudness

[그림 11] Pop Permutation Importance

```
[Accuracy] Train: 0.8286 | Test: 0.775
[Precision] Train: 0.8943 | Test: 0.8333
[Recall] Train: 0.6215 | Test: 0.5668
[F1 Score] Train: 0.7333 | Test: 0.6747
```

Weight	Feature
0.2037 ± 0.0422	explicit
0.0646 ± 0.0288	speechiness
0.0303 ± 0.0194	danceability
0.0214 ± 0.0152	instrumentalness
0.0141 ± 0.0138	tempo
0.0108 ± 0.0092	loudness
0.0088 ± 0.0084	acousticness
0.0087 ± 0.0161	duration_ms
0.0073 ± 0.0085	year
0.0065 ± 0.0110	energy
0.0055 ± 0.0096	popularity
0.0053 ± 0.0097	valence
0.0035 ± 0.0070	key
0 ± 0.0000	mode
-0.0004 ± 0.0071	liveness

[그림 12] Hiphop Permutation Importance

```
[Accuracy] Train: 0.8079 | Test: 0.7733
[Precision] Train: 0.92 | Test: 0.6471
[Recall] Train: 0.1479 | Test: 0.078
[F1 Score] Train: 0.2548 | Test: 0.1392
```

Weight	Feature
0.0741 ± 0.0430	year
0.0576 ± 0.0485	energy
0.0426 ± 0.0423	popularity
0.0404 ± 0.0439	acousticness
0.0383 ± 0.0429	duration_ms
0.0340 ± 0.0425	tempo
0.0234 ± 0.0409	speechiness
0.0187 ± 0.0288	valence
0.0181 ± 0.0398	mode
0.0179 ± 0.0302	instrumentalness
0.0124 ± 0.0331	danceability
0.0096 ± 0.0421	liveness
0.0063 ± 0.0310	loudness
0.0063 ± 0.0148	explicit
0.0045 ± 0.0232	key

[그림 13] R&B Permutation Importance

```
[Accuracy] Train: 0.8357 | Test: 0.8433
[Precision] Train: 1.0 | Test: 0.8
[Recall] Train: 0.1986 | Test: 0.1165
[F1 Score] Train: 0.3314 | Test: 0.2034
```

Weight	Feature
0.0964 ± 0.0528	instrumentalness
0.0943 ± 0.0483	tempo
0.0863 ± 0.0563	year
0.0744 ± 0.0447	duration_ms
0.0712 ± 0.0453	energy
0.0624 ± 0.0471	loudness
0.0546 ± 0.0412	speechiness
0.0527 ± 0.0500	popularity
0.0324 ± 0.0381	explicit
0.0203 ± 0.0235	acousticness
0.0166 ± 0.0249	key
0.0145 ± 0.0274	liveness
0.0144 ± 0.0267	danceability
0 ± 0.0000	mode
-0.0125 ± 0.0245	valence

[그림 14] Dance/Electronic Permutation Importance

```
[Accuracy] Train: 0.83 | Test: 0.8233
[Precision] Train: 0.8269 | Test: 0.8261
[Recall] Train: 1.0 | Test: 0.996
[F1 Score] Train: 0.9053 | Test: 0.9031

 popularity          0.110157
energy               0.105477
duration_ms          0.093679
danceability         0.089827
year                 0.085667
speechiness          0.082579
acousticness         0.081397
instrumentalness     0.073421
tempo                0.071171
loudness             0.060540
liveness             0.047771
explicit             0.038115
valence              0.037274
key                  0.018308
mode                 0.004616
```

[그림 15] Pop MDI

```
[Accuracy] Train: 0.8279 | Test: 0.7633
[Precision] Train: 0.8984 | Test: 0.8182
[Recall] Train: 0.6158 | Test: 0.5466
[F1 Score] Train: 0.7307 | Test: 0.6553

 explicit            0.320609
speechiness          0.222023
danceability         0.095523
tempo                0.054428
duration_ms          0.047996
instrumentalness     0.041430
acousticness         0.037789
energy               0.037198
loudness             0.028405
liveness             0.025508
popularity           0.025106
year                 0.024432
valence              0.023442
key                  0.012246
mode                 0.003866
```

[그림 16] Hiphop MDI

```
[Accuracy] Train: 0.8079 | Test: 0.7717
[Precision] Train: 0.9565 | Test: 0.75
[Recall] Train: 0.1415 | Test: 0.0426
[F1 Score] Train: 0.2465 | Test: 0.0805

 year                0.171117
energy               0.159630
duration_ms          0.109397
acousticness         0.099304
tempo                0.071708
speechiness          0.069998
liveness             0.053991
loudness             0.050024
danceability         0.049785
popularity           0.047470
valence              0.040502
instrumentalness     0.036902
key                  0.020742
mode                 0.011417
explicit             0.008014
```

[그림 17] R&B MDI

```
[Accuracy] Train: 0.8336 | Test: 0.8383
[Precision] Train: 0.9821 | Test: 0.75
[Recall] Train: 0.1916 | Test: 0.0874
[F1 Score] Train: 0.3207 | Test: 0.1565

 year                0.179740
tempo                0.132375
instrumentalness     0.123867
energy               0.081109
loudness             0.079788
duration_ms          0.071393
popularity           0.062330
speechiness          0.051089
explicit             0.050595
acousticness         0.040277
valence              0.039766
liveness             0.036719
danceability         0.029758
key                  0.017782
mode                 0.003412
```

[그림 18] Dance/Electronic MDI

6. 참고문헌

[1] 정현승, 강창완, 김규곤 (2008). 불균형 데이터에 대한 오버샘플링 효과 연구, Journal of the Korean Data Analysis Society, Vol. 10, No. 4 (B), August 2008, pp. 2089-2098.

[2] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer (2002). SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16 (2002), pp. 321-357.

[3] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, ICIC 2005, Part I, LNCS 3644, pp. 878 – 887.

[4] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, 2008 International Joint Conference on Neural Networks (IJCNN 2008) , pp. 1322-1328.

[5] Imbalanced scikit lean - Over-sampling
https://imbalanced-learn.org/stable/over_sampling.html#mathematical-formulation

[6] Random Forest
https://eunsukimme.github.io/ml/2019/11/26/Random-Forest/
https://ko.wikipedia.org/wiki/%EB%9E%9C%EB%8D%A4_%ED%8F%AC%EB%A0%88%EC%8A%A4%ED%8A%B8

[7] Gradient Boosting Machine
https://velog.io/@sset2323/04-05.-GBMGradient-Boosting-Machine

[8] Light GBM
https://kimdingko-world.tistory.com/184
https://herjh0405.tistory.com/40

[9] XGBoost
https://seethefuture.tistory.com/91

[10] Grid Search
https://huidea.tistory.com/32

[11] Huljanah, Mia, et al (2019). Feature selection using random forest classifier for predicting prostate cancer, Vol. 546. No. 5. IOP Publishing, 2019.

[12] Permutation Importance
https://eli5.readthedocs.io/en/latest/blackbox/permutation_importance.html

[13] Scornet, Erwan (2020). Trees, forests, and impurity-based variable importance, arXiv preprint arXiv:2001.04295, 2020.