

BTS 다음 타이틀 어떻게 할 것인가?

글로벌 음원 데이터 분석을 통한 비즈니스 전략:
10년간의 차트 트렌드와 음악적 요소 연구

프로젝트 개요

문제 정의

데이터 분석

데이터 수집



데이터 가공



인사이트
& 결론

모델링



EDA
& 시각화



문제 정의

목표

빠르게 변화하는 글로벌 음원 트렌드에 **Data-Driven**한 비즈니스 의사결정 방식을 제안

기존 방법

A&R 방식) 주로 **경험**과 **직관**, 그리고 음악 전문가들의 **창의적 판단**에 의존

제안 방향

데이터 기반 의사결정: **데이터 분석**과 **통계적 인사이트**에 기반한 의사결정 방식

개선 방법

다차원의 음원 관련 데이터 구축 및 인사이트 도출, 빌보드 차트 기준 top20을 target으로 한 **분류 모델 구축**

기대 효과

- 글로벌 음원 시장의 트렌드를 파악하여 이에 맞춘 전략을 수립, 그에 따른 **시장 적응력 상승**
- 데이터에 기반한 예측으로 직관에 의존하는 것보다 높은 정확도 기대, 잘못된 결정으로 인한 **리스크 감소**

데이터 수집

칼럼	수집 방법	출처	데이터타입	칼럼	수집 방법	출처	데이터 타입
Year	crawling	billboard	INT	Duration_sec	API	spotify	FLOAT
Month	crawling	billboard	INT	Genre	crawling	melon	STRING
Week	crawling	billboard	INT	Album cover	API	spotify	STRING
Title	crawling	billboard	STRING	BPM	API	spotify	FLOAT
Artist	crawling	billboard	STRING	Lyrics	crawling & API	1차) lyrics/ovh 2차) Genie 3차) Google	STRING

데이터 가공: Genre, Duration, Featuring, BPM

Genre

- 복수 장르 시 **대표값 1개만** 지정(, 이후 값 삭제)
- 부분 중복 데이터 하나로 병합
(예, 랩/힙합 + 힙합 -> 랩/힙합)
- 장르 갯수가 가장 큰 2개 Outlier 제거 (POP, 랩/힙합)

Duration

- ms(밀리세컨)단위 **sec(초) 단위**로 변환

Featuring

- **Featuring(1), Non-Featuring(0)**으로 구분

BPM

- **소수점 3자리까지만** 표기
- 연 - 월 통합 (Year + Month -> Year-Month)

데이터 가공: Lyrics

[1] 기본 전처리

- 공백 행 제거
- 가사 안내(chorus, pre-chorus, intro, outro 등) 제거
- 감탄사('yeah', 'woah', 'oh', 'uh', 'ah', 'na', 'la' 등) 제거

[2] Word Tokenization (토큰화)

: 말뭉치로부터 토큰을 분리하는 작업

*토큰: 문법적으로 더 이상 나눌 수 없는 언어 요소

[3] Cleaning (정제)

: 노이즈, 불용어 제거 (영어, 스페인어)

원본 데이터

I'm friends with the monster\nThat's under my bed
head\nOh well, guess beggars You're trying to save me
think I'm crazy\nYeah, you think I'm crazy (crazy)\n



['I', '', 'm', 'friends', 'with', 'the', 'monster', 'That', '',
's', 'under', 'my', 'bed', 'well', 'guess', 'beggars', 'Get', 'along',
'with', 'the', 'voices', 'inside', 'of', 'my', 'head', 'You', '', 're',
'trying', 'to',



['friends', 'monster', 'bed', 'get', 'along', 'voices', 'inside',
'head', 'trying', 'save', 'stop', 'holding', 'breath', 'think',
'crazy', 'think', 'crazy', 'wanted', 'fame', 'cover',
'newsweek', 'well', 'guess', 'beggars',

■: 제거 ■: 토큰화

데이터 가공: Lyrics

[4] POS Tagging (품사 태깅)

: '(토큰, 품사)'의 형태로 태깅

['friends', 'NNS'], ('monster', 'RBR'), ('bed', 'NN'),
('get', 'VB'), ('along', 'IN'), ('voices', 'NNS'),
('inside', 'IN'), ('head', 'NN'), ('trying', 'VBG'),
('save', 'JJ'), ('stop', 'VBN'), ('holding', 'VBG'),



[5] Normalization (정규화)

: Lemmatization (표제어 추출)

(예) starts -> start

started -> start

['friend', 'monst', 'bed', 'get', 'along', 'voice',
'inside', 'head', 'try',
'save', 'stop', 'hold', 'break', 'think', 'crazy',
'think', 'crazy', 'crazy', 'want', 'fame', 'cov',

데이터 가공: Lyrics

감성 분석

- 사용 라이브러리: **nltk.sentiment vader_lexicon**
- 구성) 부정 점수 (negative)
중립 점수 (neutral)
긍정 점수 (positive)
종합 점수 (compound)

TF-IDF

- 사용 라이브러리
: **sklearn.feature_extraction.text TfidfVectorizer**
- 범위: 1달, 곡당 상위 3개 키워드

KeyBERT

- 사용 라이브러리: **KeyBERT**
- 범위: 1년, 상위 30개 키워드

데이터 가공: Album

[1] 기본 데이터

- spotify API를 활용한 앨범 커버 링크
- 앨범 커버 이미지 임시 저장

<https://i.scdn.co/image/ab67616d0000b27342859c496ab80d9867ef21b6>



[2] COLOR THIEF

: 대표 색상을 뽑아 내는 라이브러리,

이미지의 모든 색상 데이터를 RGB 중앙값 기준으로 분할 후 클러스터링 진행



[3] HSL기반 범주화

: RGB 색상을 HSL 모델로 변환하여 색조(Hue), 채도(Saturation),

명도(Lightness) 기준으로 분류

숫자의 RGB 값을 더 직관적으로 범주화한 데이터 생성

(64, 181, 219)

(6, 13, 16)

(215, 219, 220)

(60, 94, 97)



Medium Medium Cyan

Very Dark Medium Cyan

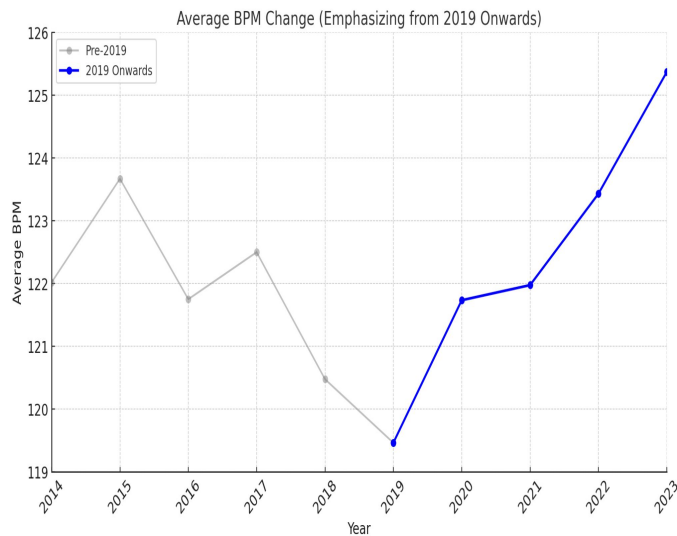
Very Light Gray Cyan

Dark Muted Cyan

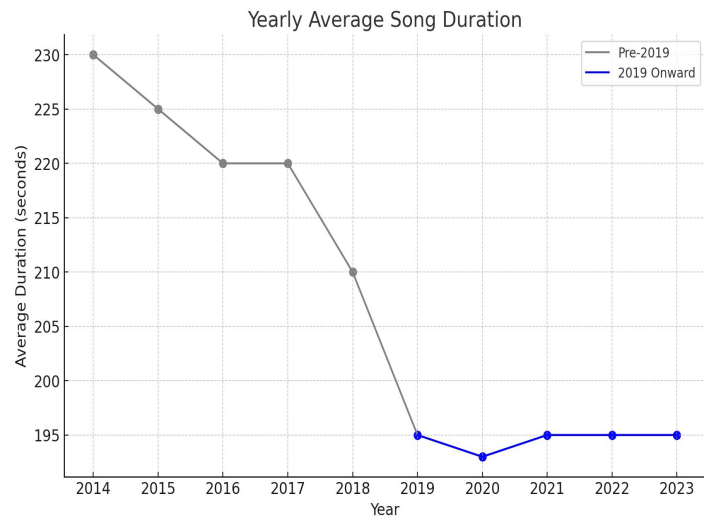
EDA 및 시각화

1. 팬데믹 이후 트렌드 (BPM, Duration, Genre)

BPM의 변화 (2014 ~ 2023)



Duration 변화 (2014 ~ 2023)



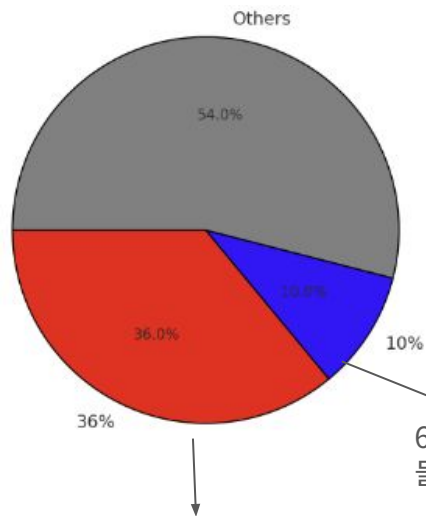
2019년 팬데믹 이후 BPM은 급속도로 증가하였고, 노래 길이는 급격히 감소하였다.

EDA 및 시각화

1. 펜데믹 이후 트렌드 (BPM, Duration, Genre)

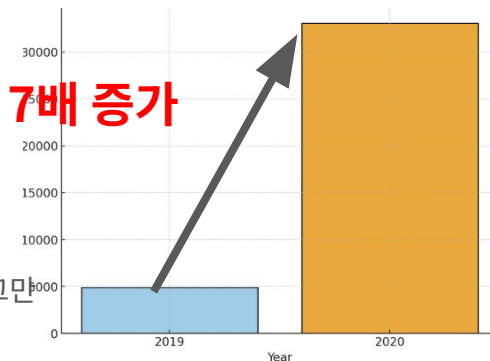
BPM은 증가하고 노래 길이는 감소한 이유

2019 모건 스탠리 유럽 직장인 대상 콘텐츠
플랫폼 이용경험 조사



콘텐츠 제작 플랫폼 부수입 경험

2019 한국전파 진흥협회, 크리에이터 소득
신고 조사



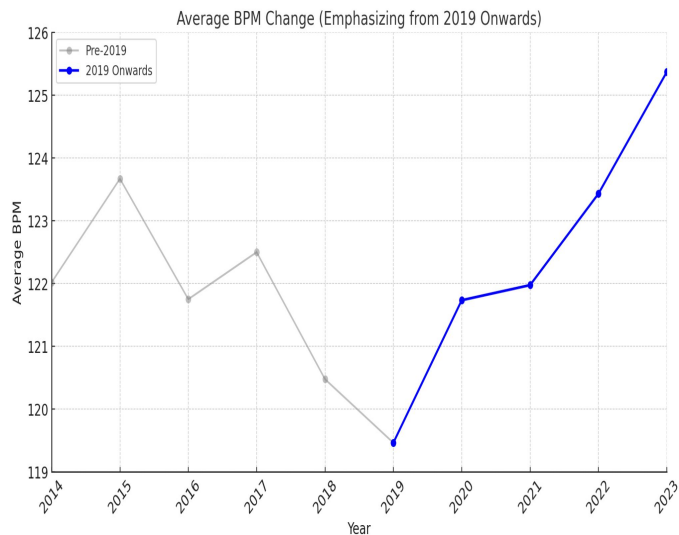
6개월 내 본업을 그만
둘 것을 고려 중

2019년 코로나 이후 유튜브나
틱톡과 같은 콘텐츠 제작의
비중이 높아졌다. 쇼츠형
음악들이 대중화되면서 BPM은
증가하였고 노래 길이는 급격히
감소하였다.

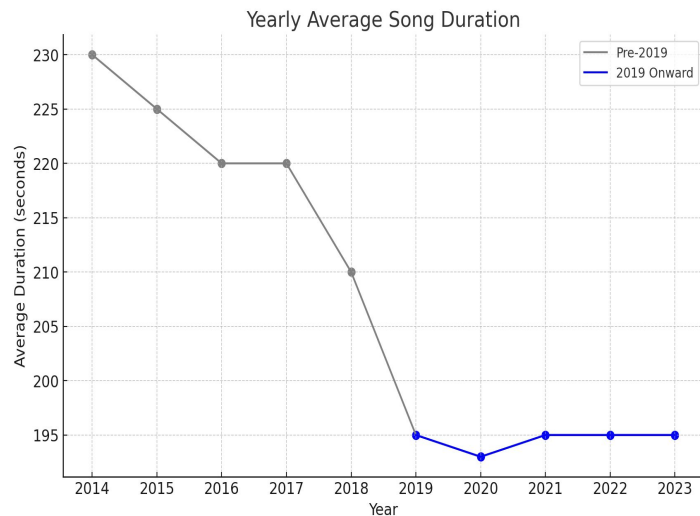
EDA 및 시각화

1. 팬데믹 이후 트렌드 (BPM, Duration, Genre)

BPM의 변화 (2014 ~ 2023)



Duration 변화 (2014 ~ 2023)



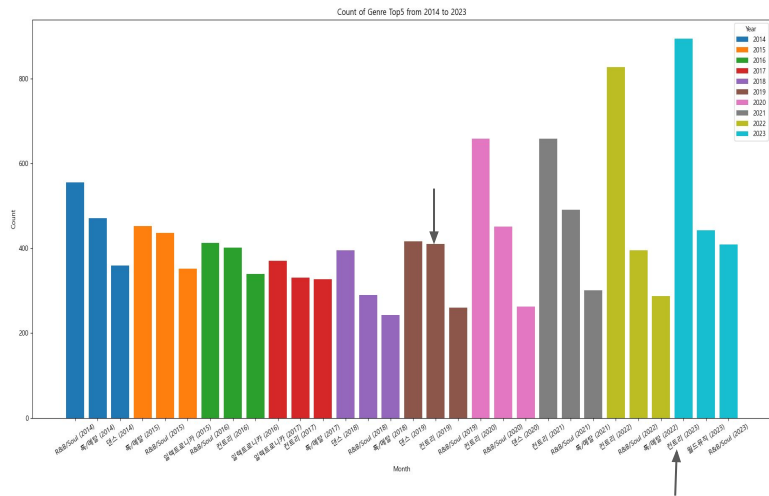
팬데믹 이후 쇼츠에 중독된 현대인들은 돌아오지 않았다..!

EDA 및 시각화

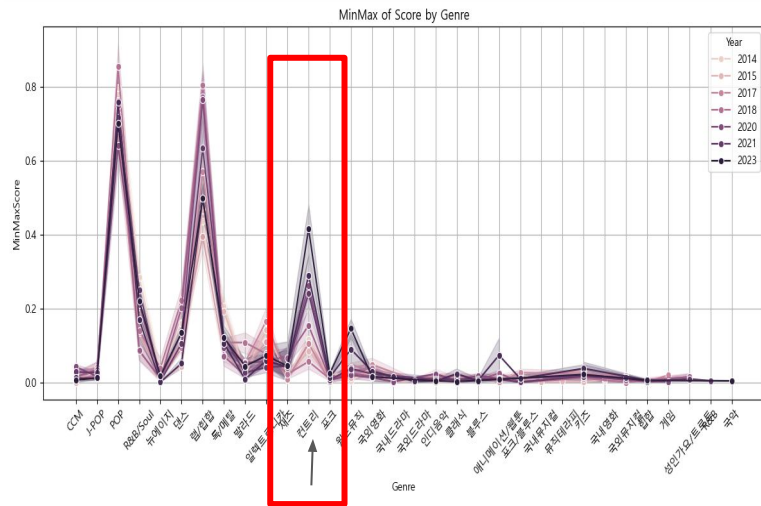
1. 펜데믹 이후 트렌드 (BPM, Duration, Genre)

장르의 변화 (2014 ~ 2023)

항상 1,2등인 POP, 랩/힙합 제거한 장르 TOP 3



MinMaxScaler를 이용한 장르 EDA



2019년 이후 다른 장르는 변함이 거의 없지만 컨트리 장르가 급속도로 증가하였음을 알 수 있다.

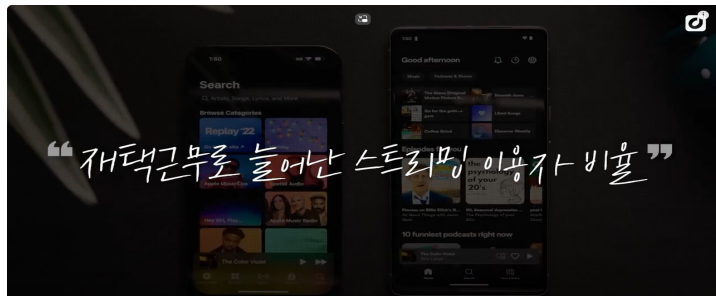
EDA 및 시각화

1. 펜데믹 이후 트렌드 (BPM, Duration, Genre)

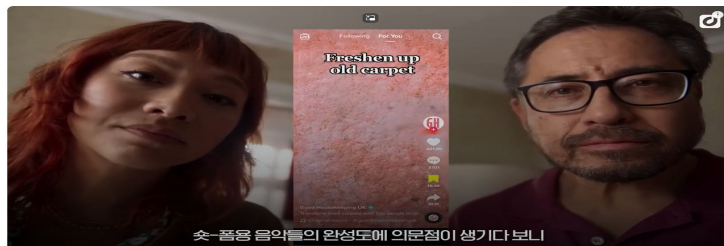
컨트리 빈도수가 급증한 이유

미국 전통의 음악 장르, '컨트리 뮤직'의 향유층이 백인 중장년층에만 한정된다는 말도 이제 옛말이다. 2019년 혜성같이 등장한 틱톡 스타 '릴 나스 엑스'의 메가 히트곡 'Old Town Road'는 컨트리 요소를 접목한 힙합 곡이었고, 2023년 빌보드 연말 차트 1위의 주인공은 컨트리 뮤지션 '모건 월렌(Morgan Wallen)'의 'Last Night'이 차지했다. 최근에는 레전드 디바 바운세마저 이 장르에 매료되었을 정도로 미국 음악 시장에서 컨트리의 영향력은 나날이 커지고 있다. 마니아층이 확실한 장르에서 메인스트림으로 거듭난 이상, 컨트리 뮤직을 모르고 지나칠 수는 없다. 아직 컨트리 팝이 생소한 당신을 위해 알아두면 반드시 쓸모 있는 컨트리 히트곡 플레이리스트를 마련해 보았다.

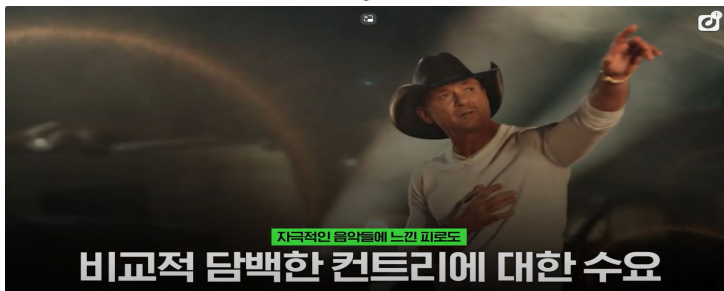
2019년 틱톡 스타 컨트리 음악 1등



이유 1 : 2019년 펜데믹 이후 재택근무 증가



이유 2 : 숏-폼용 음악의 완성도 증가

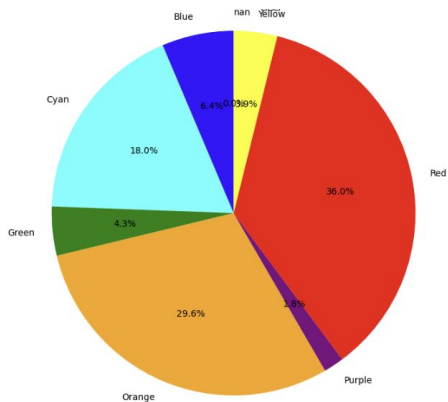
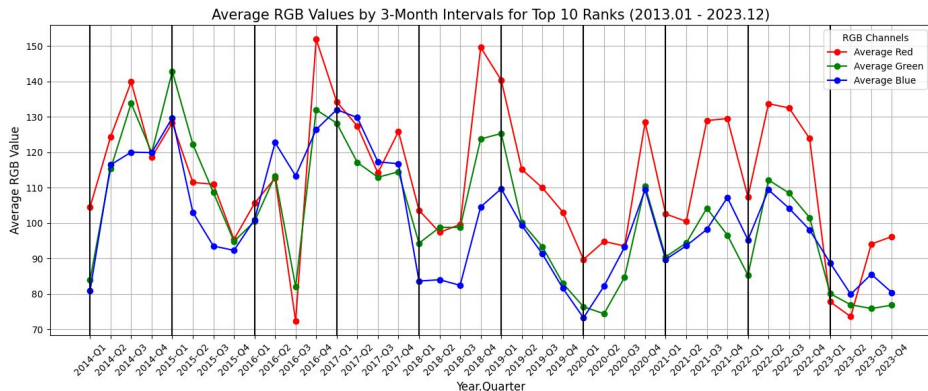


2019년 ~ 2022년까지 컨트리 음악 수요 58% 증가

출처 : <https://brunch.co.kr/@assp/42>
<https://www.youtube.com/watch?v=ed2MCEkJPZw>

EDA 및 시각화

2. 앨범 커버 트렌드 변화



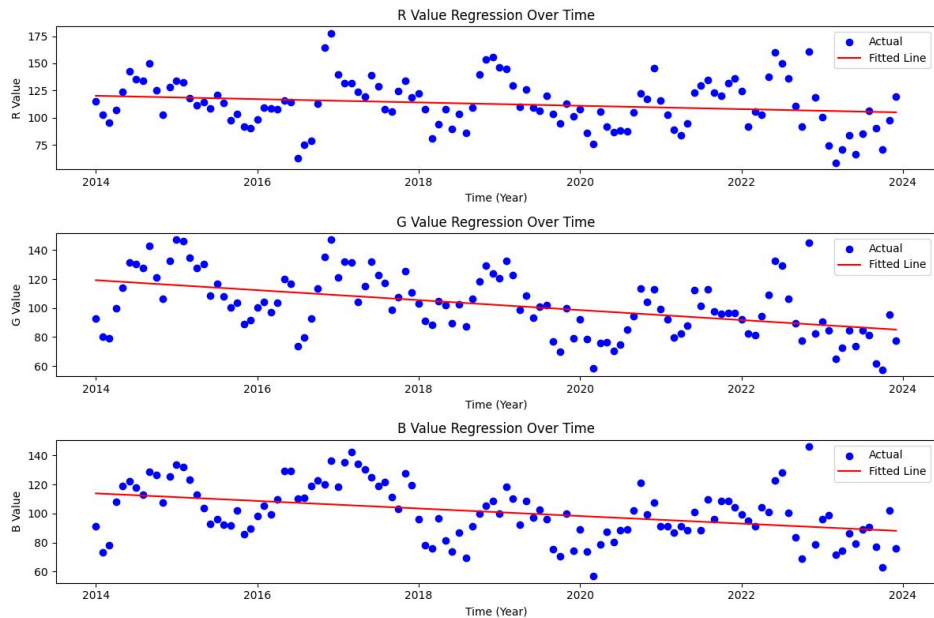
2년 주기로 밝아지는 컬러 트렌드 & 레드의 강세

10년 간(2014~2023)의 빌보드 HOT100에 속한 앨범의 RGB컬러의 추이를 그래프로 살펴보면 **짝수에서 홀수로 넘어가는 겨울에 RGB값 모두 치솟는 현상**을 볼 수 있다.

또한 전반적으로 **RED**가 상위에 위치해있으며, HSL기반으로한 텍스트 데이터를 단순 집계한 결과 또한 **‘RED’의 비율이 약 36%**를 차지한 것을 알 수 있다.

EDA 및 시각화

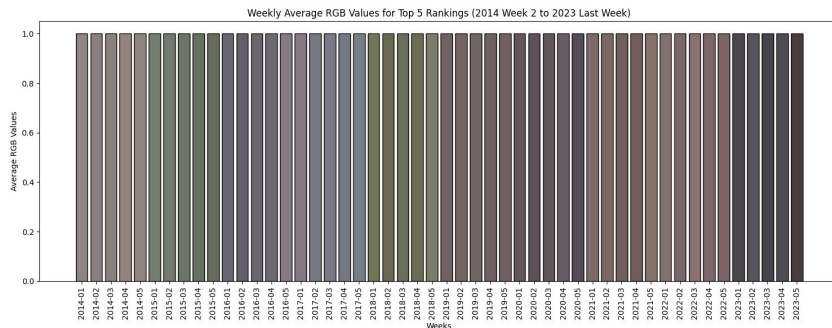
2. 앨범 커버 트렌드 변화



점점 진해지고 있는 컬러 트렌드

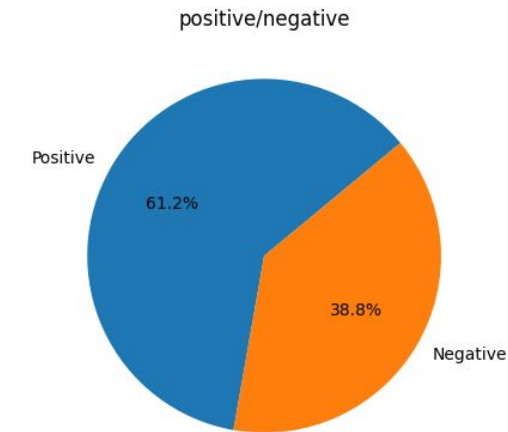
지난 10년간의 RGB값은 시간을 기준으로 기울기는 작다고 볼 수 있으나 **모두 꾸준히 음의 상관관계**를 갖고 있다.

이는 직접 평균 색상을 출력한 아래의 표에서도 확인 할 수 있듯이 **해가 갈 수록 진해지고 있는 것**을 확인 할 수 있다.



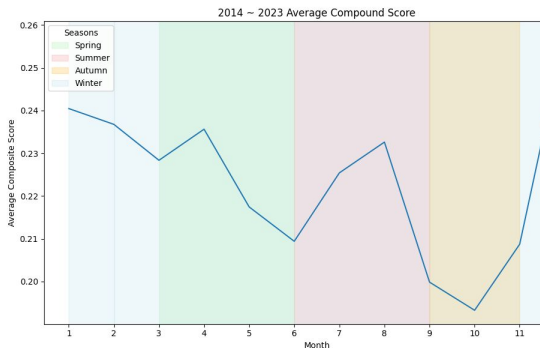
EDA 및 시각화

3. 가사 감성 분석



10년 간 빌보드 HOT100의 긍정/부정 비율

10년 간(2014~2023)의 빌보드 HOT100 가사를 감성 분석 한 결과, 종합점수 기준 양수인 **긍정 비율은 61.2%**, 음수인 **부정 비율은 38.8%**로 긍정적인 노래가 많은 것으로 나타났다.

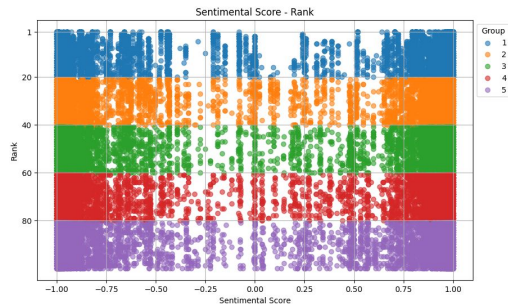


월별 감성 종합점수 평균

10년 간(2014~2023)의 빌보드 HOT100 가사를 감성 분석 한 결과, 종합점수 기준 새해를 맞는 **겨울(12월)에 평균 감성 점수가 올라가는 것을 확인**할 수 있다. 추가적으로, **가을인 10월에는 평균 감성 점수가 크게 하락해 부정적인 노래의 비중이 올라간 것을 확인**할 수 있다.

EDA 및 시각화

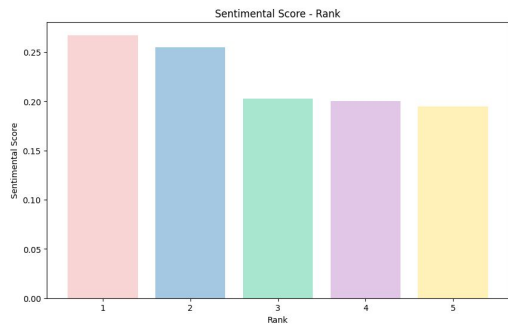
3. 가사 감성 분석



순위 그룹별 감성 종합 점수 분포

극단적일 수록 인기가 많다!

10년 간(2014~2023)의 빌보드 HOT100 곡들의 순위를 5 그룹(1~20, 21~40, ...)으로 클러스터링 한 뒤, 순위 그룹 간 감성 종합 점수 분포를 확인하였다. 그 결과, **상위 그룹일수록 중립 점수의 분포가 적은 것**을 확인할 수 있었다.



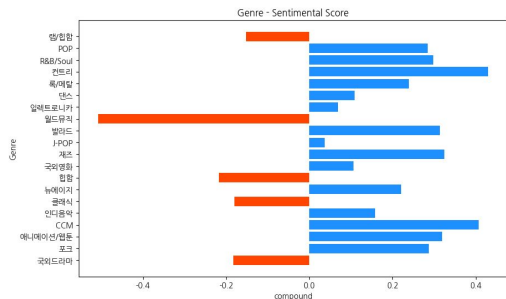
순위 그룹별 평균 감성 점수

긍정의 힘!

10년 간(2014~2023)의 빌보드 HOT100 곡들의 순위를 5 그룹(1~20, 21~40, ...)으로 클러스터링 한 뒤, 순위 그룹별 평균 감성 점수를 확인하였다. 그 결과, **상위 그룹일수록 평균 감성 점수가 높은 것**으로 나타나 긍정적인 노래가 상위 그룹에 위치해있다고 예상해볼 수 있다.

EDA 및 시각화

3. 가사 감성 분석



상위 20개 장르의 평균 감성 점수 비교

장르별로 분류한 10년 간(2014~2023)의 빌보드 HOT100곡의 평균 감성 점수 비교 결과, **선호**
장르와 평균 감성 점수의 관계는 확인할 수 없었다.

EDA 및 시각화

5. 요인 분석: PCA



변수 목록

- Year/Month/Week
- Rank
- Title/Artist
- Genre
- BPM
- Duration_sec
- neu
- pos
- compound
- R, G, B
- isTop20

Numeric 변수 추출

최종 변수 목록

- Year/Month/Week
- Rank
- BPM
- Duration_sec
- neu
- pos
- compound
- R, G, B
- isTop20

EDA 및 시각화

5. 요인 분석: PCA

Principle Component #1

[-0.08605039, 0.02073556, -0.01279347,
0.04624918, 0.00520649, -0.07698718
,-0.21190866, -0.0716421, **0.27226938**,
0.27624031, **0.88249325**, **0.94958366**,
0.91261207]

Component #1 값이 높은 항목들은 **감성점수, 색과 관련된** 수치가 높다.



PC1: 감정적이고 긍정적인 노래를 구별하는 데 유용

Principle Component #2

[-0.12489807, -0.01937185, -0.00334287,
0.09246588, **0.08516646**, -0.06961546,
-0.07576165, **0.00693347**, **0.65726282**,
0.88940346, -0.808978, -0.21865672,
-0.20072749]

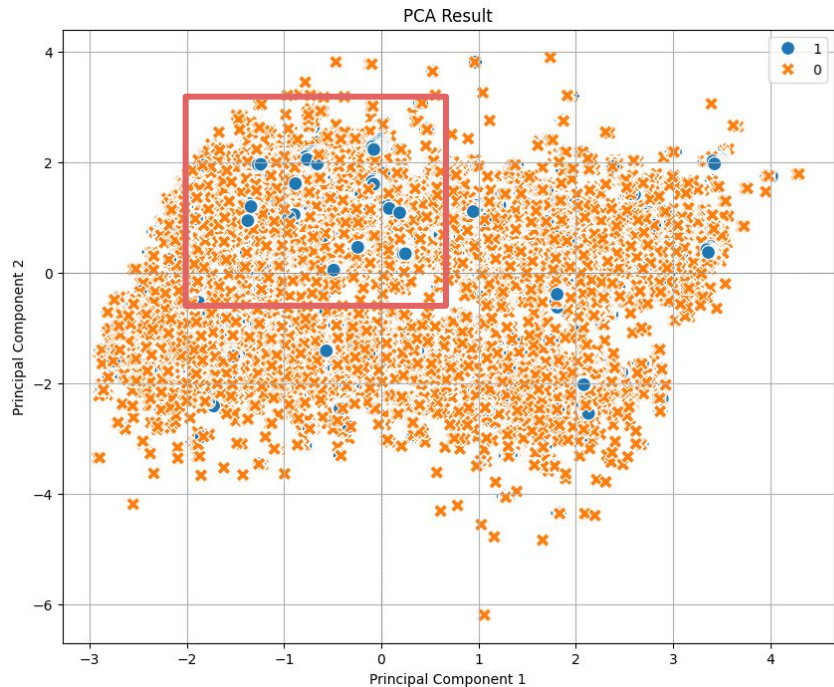
Component #2 값이 높은 항목들은 **노래의 템포** 등과 관련된 수치가 높다.



PC2: 리듬, 템포 등 음악적요소를 반영

EDA 및 시각화

5. 요인 분석: PCA



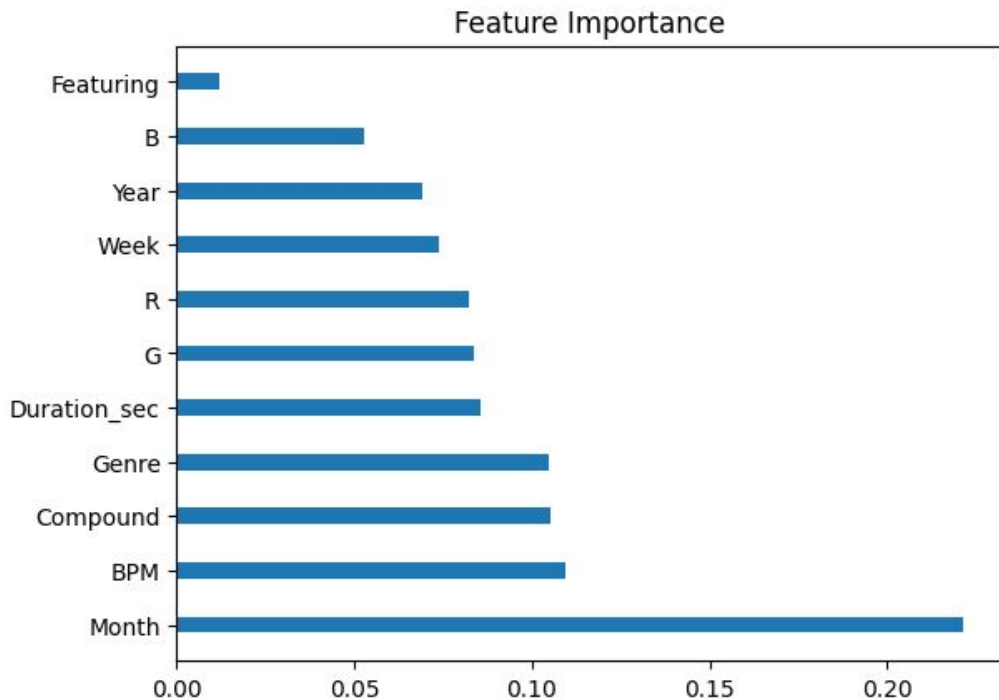
타겟은 1위 ~ 20위로 지정하고, 타겟의 결과로만 PCA 분포를 확인한 결과 **주성분 1에서는 음수, 주성분 2에서는 양수인 경우가 다른 경우에 비해 많은 것**을 알 수 있다.

부정적인 감정이나 특정한 주제를 포함하고, 리듬, 템포 등과 같은 음악적 요소를 이용한다면 충분히 TOP20안에 들 것이다

EDA 및 시각화

5. 요인 분석: Feature Importance

feature	importance
Month	0.2214
BPM	0.1093
compound	0.1054
Genre	0.1046
Duration_sec	0.0854
G	0.0838
R	0.0821
Week	0.0736
Year	0.0691
B	0.0529
Featuring	0.0120



모델링

목표

빌보드 차트 기준 Top20을 Target으로 한 **분류 모델 구축**

사용 데이터

학습 데이터

독립변수

Year, Month, Week, BPM,
Duration_sec, R, G, B,
compound, Genre, Featuring

종속변수

isTop20
(0:X, 1:O)

모델링

사용 Feature

사용

- Year
- Month
- Week
- Genre
- BPM
- Duration_sec
- R
- G
- B
- Featuring
- compound
- isTop20

제외

- Rank: 'isTop20'으로 대체
- Title
- Artist
- Lyrics
- color1, color2, color3
: 'R', 'G', 'B'로 대체
- neg, neu, pos: compound로 대체

모델링

	Year	Month	Week	Genre	BPM	Duration_sec	R	G	B	Featuring	compound	isTop20
0	2014	1	2	랩/힙합	110.009	251.246	186	97	156	1.0	0.9904	1
1	2014	1	2	POP	129.992	204.160	18	22	36	1.0	0.9958	1
2	2014	1	2	랩/힙합	122.013	257.840	82	68	48	0.0	0.9867	1
3	2014	1	2	POP	84.876	190.185	151	123	112	0.0	0.9887	1
4	2014	1	2	발라드	141.284	229.400	33	39	67	0.0	0.9771	1
...
52095	2023	12	5	재즈	77.810	1037.907	200	215	236	0.0	0.8405	0
52096	2023	12	5	POP	129.918	143.940	31	30	30	0.0	0.9676	0
52097	2023	12	5	월드뮤직	125.012	189.426	161	163	191	0.0	0.7319	0
52098	2023	12	5	POP	156.975	146.752	54	32	13	0.0	0.9849	0
52099	2023	12	5	R&B/Soul	84.828	244.685	71	83	80	0.0	0.9946	0

52100 rows × 12 columns

전처리가 끝난 데이터

모델링

[1] 데이터 준비

1) X, y로 데이터 나누기

2) 수치형 변수 스케일링

Standard Scaling: 평균을 0, 표준편차를 1로 하는 데이터로 변환

MinMax Scaling: 최대값은 1, 최소값은 0으로 하여 0~1 사이의 값으로 변환

Robust Scaling: 중앙값, IQR 사용

3) 범주형 변수 인코딩

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

고정

4) 2), 3) 데이터 병합

5) train, test 데이터 나누기

모델링

[2] 모델 정의

경사하강법 기반: Logistic Regression

확률 기반: Naive Bayes

거리 기반: SVM

트리 기반: 기본) Decision Tree

앙상블) Random Forest

[3] 모델 학습 & 예측

```
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

[4] 성능 평가 지표: F1-Score

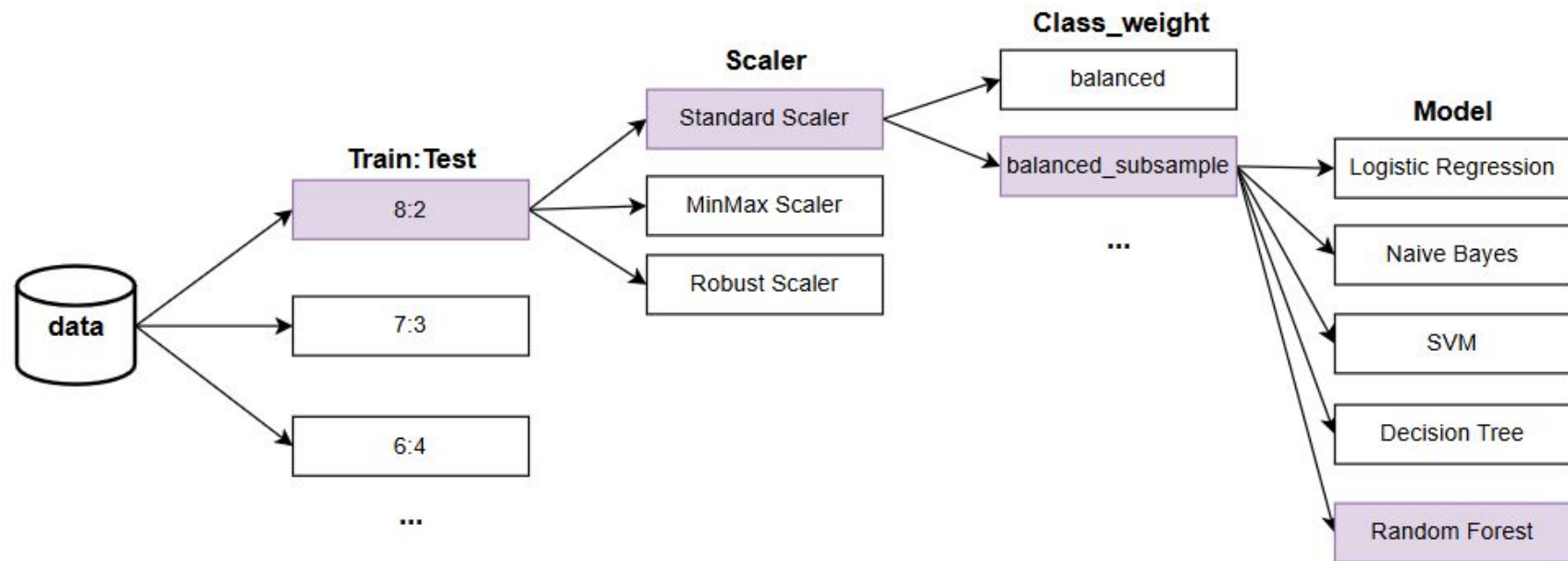
$$2 * \frac{Precision * Recall}{Precision + Recall}$$

- **정밀도(Precision)**
: 예측값이 얼마나 정확한가?
- **재현율(Recall)**
: 실제 정답을 얼마나 맞췄는가?

모델링

testID	features	train:test	scaling	class_weight	model	accuracy	F1-Score	추가사항	비고
6 전체		7:3	Robust	balanced	SVM				
6 전체		7:3	Robust	balanced	Decision Tree	0.9204	0.8		
6 전체		7:3	Robust	balanced	Random Forest	0.9345	0.8303		
7 전체		7:3	Standard	balanced	Logistic Regression	0.5956	0.3279	SMOTE	
7 전체		7:3	Standard	balanced	Naive Bayes	0.5476	0.3302	SMOTE	
7 전체		7:3	Standard	balanced	SVM	0.521	0.3484	SMOTE	
7 전체		7:3	Standard	balanced	Decision Tree	0.8001	0.536	SMOTE	
7 전체		7:3	Standard	balanced	Random Forest	0.9149	0.7833	SMOTE	
8 전체		7:3	Standard	balanced	SVM	0.5902	0.3264	kernel = 'linear'	
8 전체		7:3	Standard	balanced	SVM	0.521	0.3484	kernel = 'rbf'	
8 전체		7:3	Standard	balanced	SVM	0.401	0.3583	kernel = 'polynomial'	
8 전체		7:3	Standard	balanced	SVM	0.5206	0.3144	kernel = 'sigmoid'	
9 전체		7:3	Standard	balanced	BalancedRandomForestClassifier	0.8712	0.7399		
10 전체		7:3	Standard	scale_pos_weight = y_train.value_counts()[0] / y_train.value_counts()[1]	XGBoost	0.8559	0.7093		
11 전체		7:3	Standard	X	Decision Tree	0.9232	0.8086	y_train에 compute_sample_weight 적용	
11 전체		7:3	Standard	X	Random Forest	0.9366	0.8376	y_train에 compute_sample_weight 적용	
12 전체		7:3	Standard	balanced	Random Forest	0.938	0.8394	y_train에 compute_sample_weight 적용	
13 전체		8:2	Standard	balanced_subsample	Random Forest	0.9386	0.8413	y_train에 compute_sample_weight 적용	
14 전체, abs(compound)		8:2	Standard	balanced_subsample	Random Forest	0.9387	0.8415	y_train에 compute_sample_weight 적용	
15 전체, abs(compound)		8:2	Standard	balanced_subsample	Random Forest	0.9387	0.8415	y_train에 compute_sample_weight 적용, scaling에서 compound 제외	가장 최적화된 모형
16 전체, abs(compound)		8:2	Standard	balanced_subsample	Random Forest	0.9374	0.8379	y_train에 compute_sample_weight 적용, scaling에서 compound 제외, genre top10 제외 나머지 기타로	
17 전체, abs(compound)		8:2	Standard	balanced_subsample	Random Forest	0.9377	0.839	y_train에 compute_sample_weight 적용, scaling에서 compound 제외, genre top7 제외 나머지 기타로	
18 전체, abs(compound)		8:2	Standard	balanced_subsample	Random Forest	0.9383	0.8407	y_train에 compute_sample_weight 적용, scaling에서 compound 제외, genre top5 제외 나머지 기타로	

모델링



최종 모델

*encoder는 label encoder로 고정

베이스라인

- features) 전체
- train:test) 7:3
- scaler) X
- class_weight) X



[Best Model]

Random Forest: 0.8290

최종 모델

- features) 전체, abs(compound)
- train:test) 8:2
- scaler) Standard
- class_weight) balanced_subsample



[Best Model]

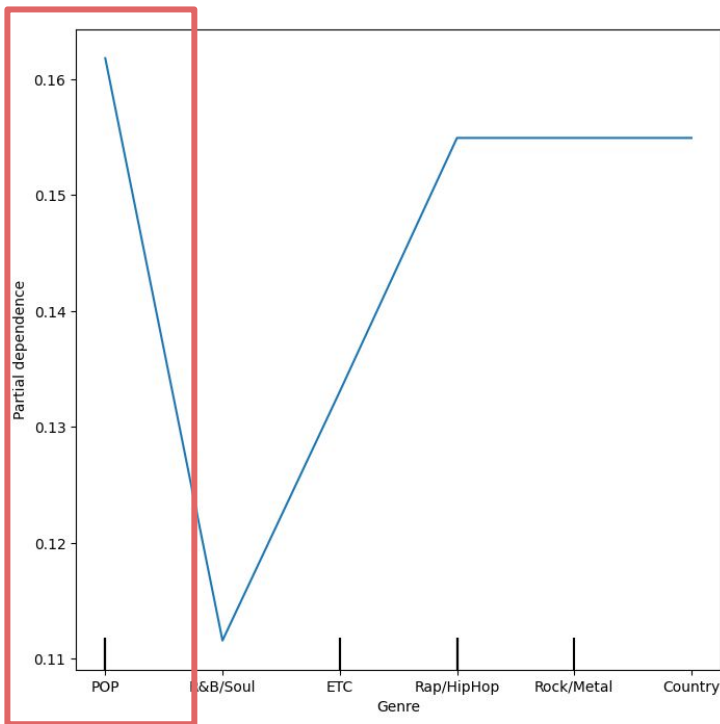
Random Forest: 0.8415

최종 모델: 해석

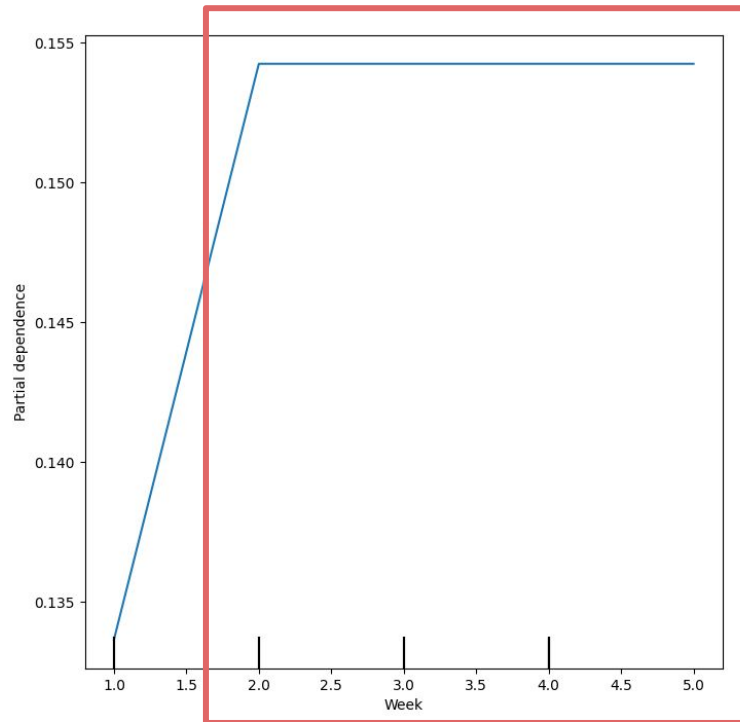
x축: feature, y축: 변수 의존도(확률)

- PDP(Partial Dependence Plot): 부분 의존도

[1] Genre



[2] Week

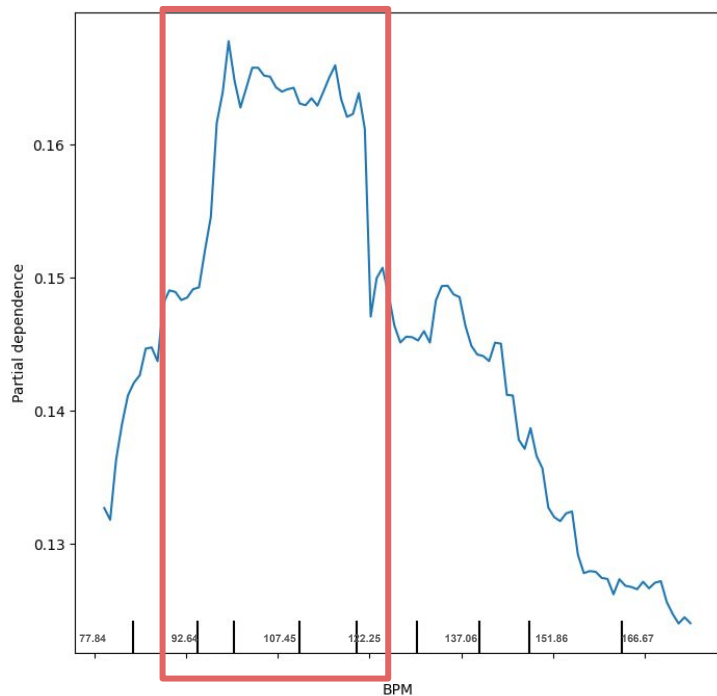


최종 모델: 해석

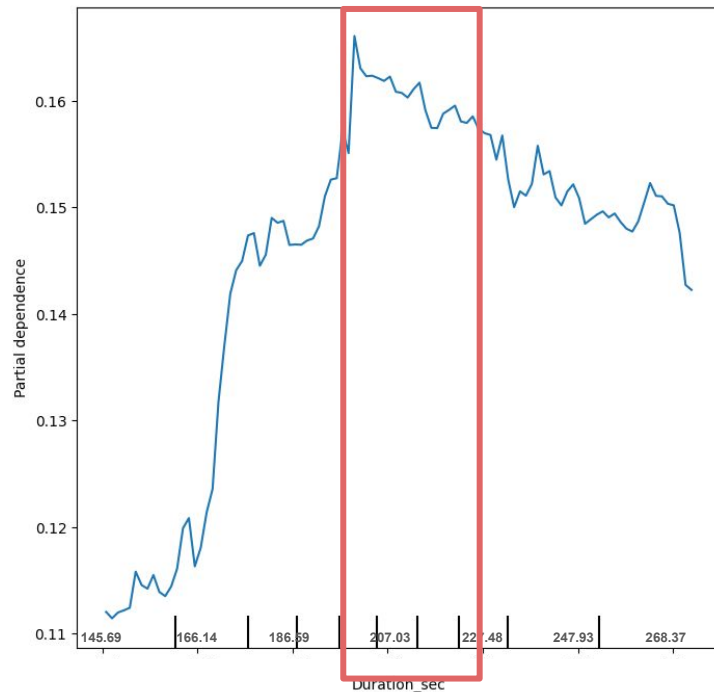
x축: feature, y축: 변수 의존도(확률)

- PDP(Partial Dependence Plot): 부분 의존도

[3] BPM



[4] Duration

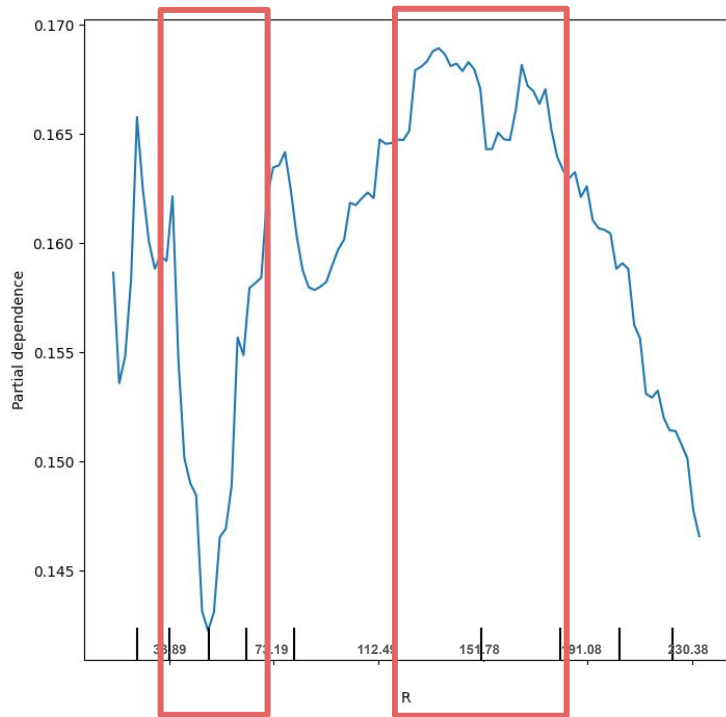


최종 모델: 해석

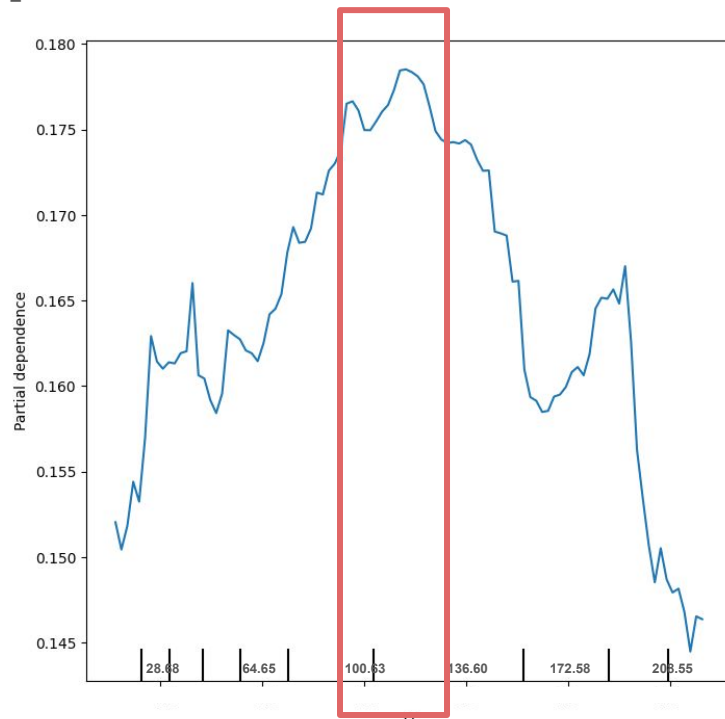
x축: feature, y축: 변수 의존도(확률)

- PDP(Partial Dependence Plot): 부분 의존도

[5] Color: Red



[6] Color: Green

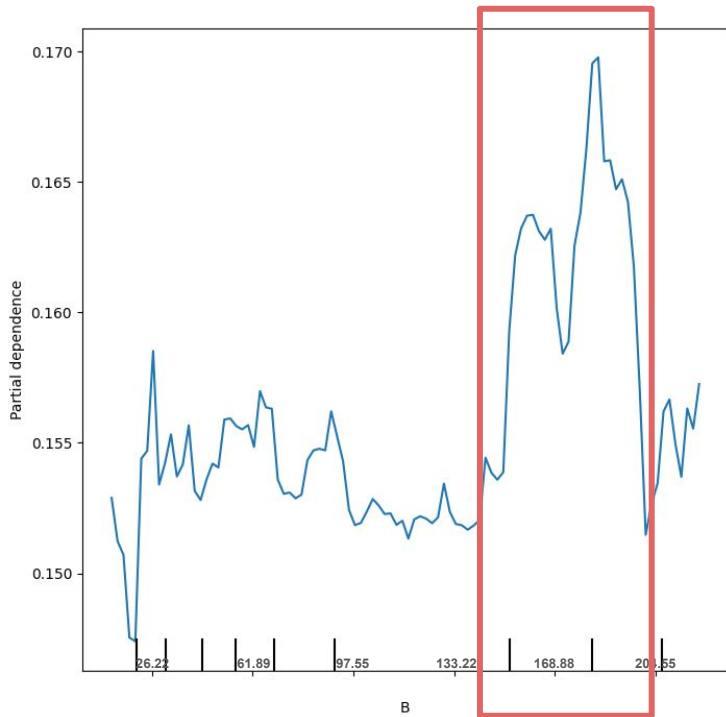


최종 모델: 해석

x축: feature, y축: 변수 의존도(확률)

- PDP(Partial Dependence Plot): 부분 의존도

[5] Color: Blue

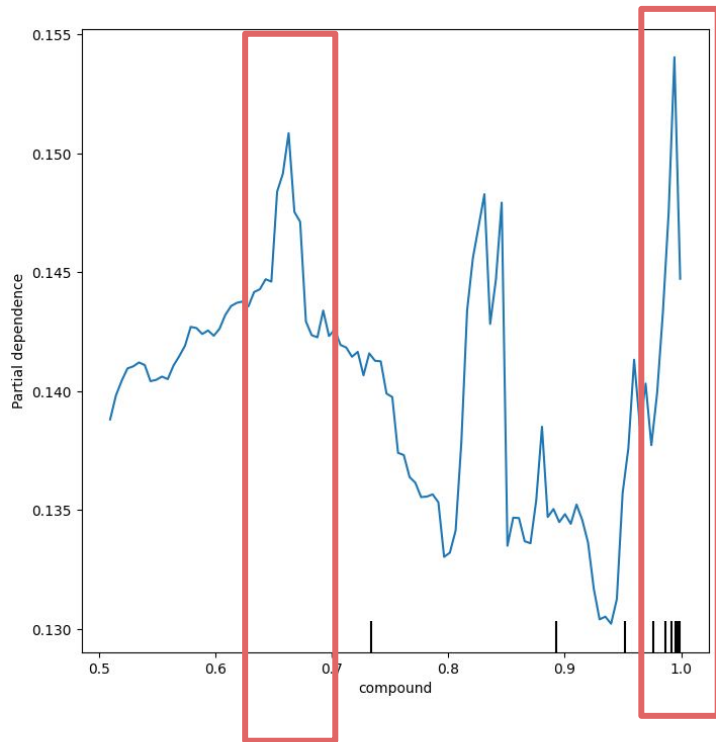


최종 모델: 해석

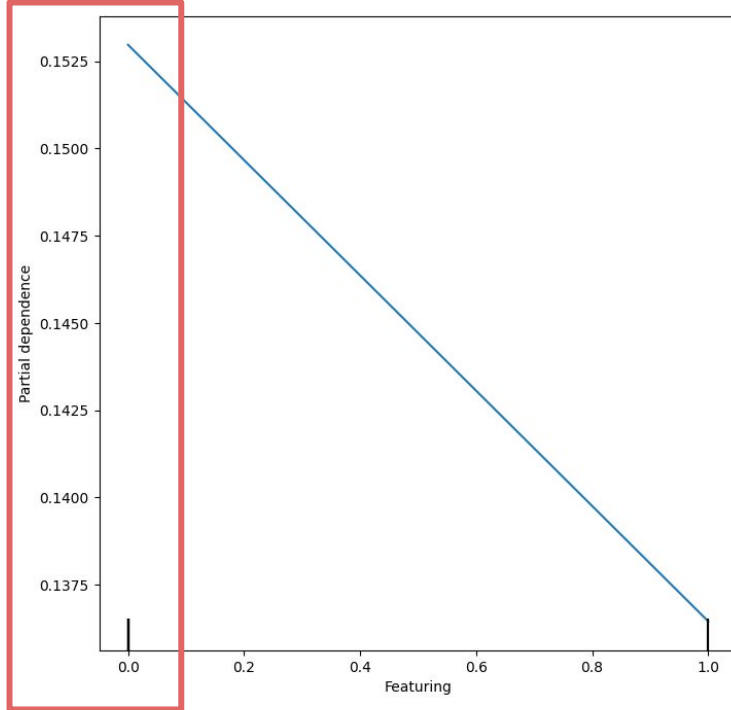
x축: feature, y축: 변수 의존도(확률)

- PDP(Partial Dependence Plot): 부분 의존도

[7] compound



[8] Featuring



지난 10년 간의 음악 트렌드

- 2019년 이후 변화: **BPM 증가**, **Duration 감소**, Genre는 **컨트리**
- 앨범 커버 컬러 트렌드: **2년 주기**로 밝아짐, **레드** 강세, 전반적으로 **어두워지는 경향**
- 가사 감성: **긍정** 강세, **계절성**(가을 - 감성 점수 하락, 겨울 - 감성 점수 증가)
- 가사 키워드: 비슷한 단어를 **반복적**으로 쓰는 경향

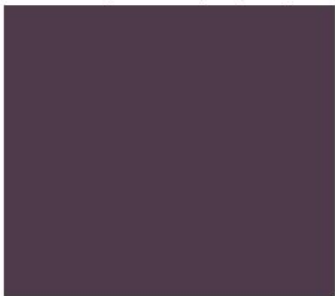
분류 모델에서 얻은 인사이트

빌보드 Top20에 들기 위해서는 **BPM, Compound, Genre**가 크게 작용한다.

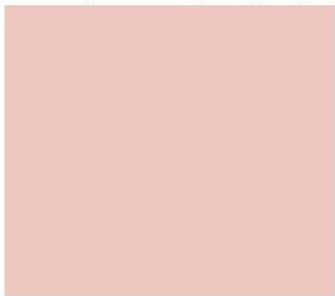
기존 BTS 스타일 분석

장르별 앨범 커버 컬러

POP (RGB: 79, 58, 75)



댄스 (RGB: 238, 199, 192)



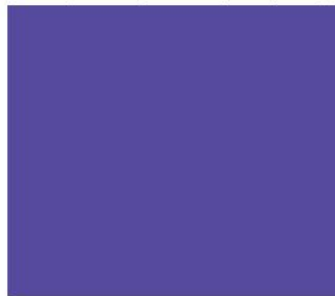
랩/힙합 (RGB: 100, 95, 108)



록/메탈 (RGB: 249, 218, 7)

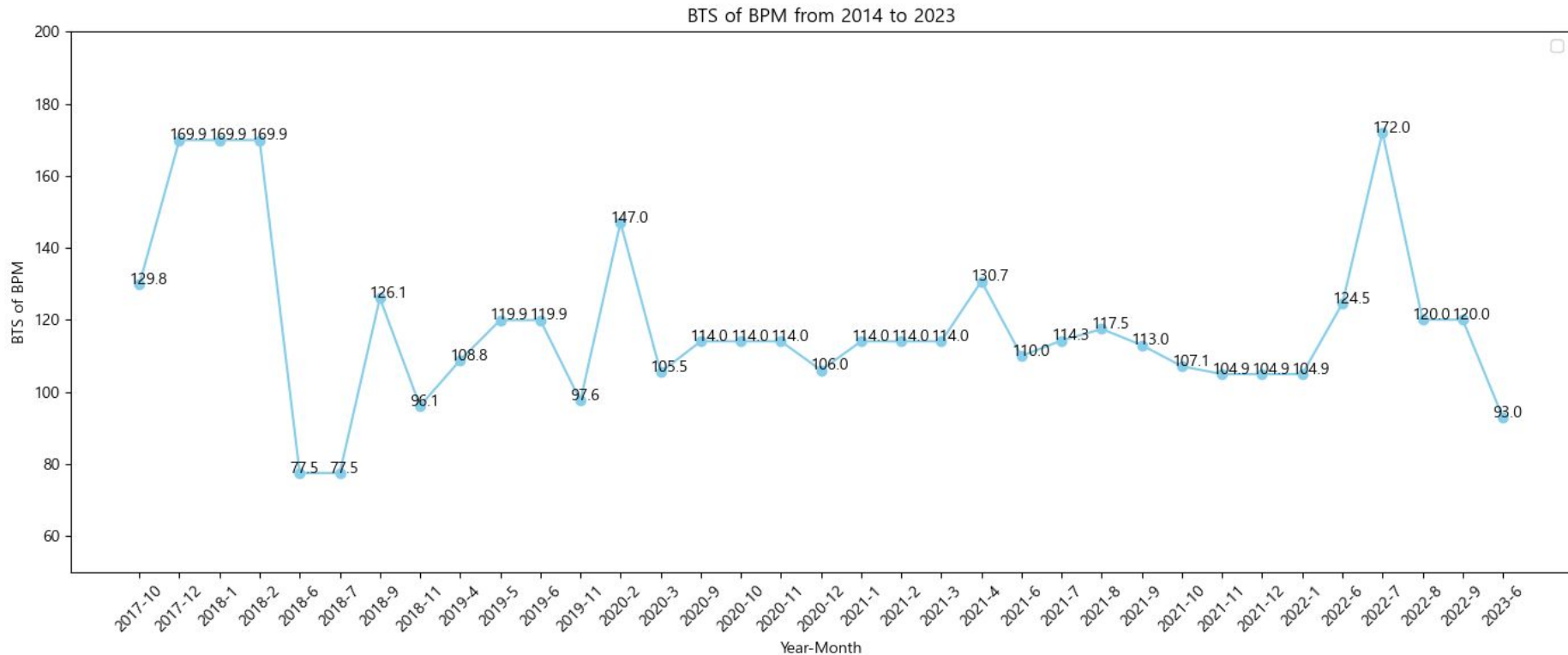


R&B/Soul (RGB: 85, 74, 157)



기존 BTS 스타일 분석

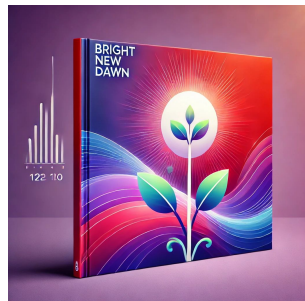
2014~2023 BTS 노래의 BPM 변화



결론

제안 타이틀 곡 컨셉

- 장르: **컨트리풍의 POP**
- 발매 시기: **연초, 첫 번째 주**
- **Butter**와 같은 BPM, **110**
- 앨범 표지는 **빨간색 + 보라색** (120,120,168)
- 곡의 길이, **3분 30초**
- 감성은 **매우 긍정적인**
- 피쳐링은 **없이**



앨범 커버 예시

보완점, 발전 방향

- **추가 항목 수집:** spotify API에 있는 노래 분위기 데이터를 추가로 수집하여 관련 분석을 진행한다면, 더욱 다차원적인 분석이 가능할 것으로 기대
- **다장르 노래 분석:** 장르의 경우, 해당 프로젝트에서는 복잡성을 줄이기 위해 한곡 당 하나의 장르만 파악했지만 여러 개의 장르를 추출하여 분석한다면 명확한 분석이 될 것으로 기대.
(현대곡의 경우, 단일 장르가 아닌 경우가 대부분)
- **팬덤 데이터 적용:** 가수의 영향력을 반영한 분석, 해당 팬층의 당시 반응이나 sns 전략등을 분석한다면 더욱 공교한 비즈니스 전략이 될 것으로 기대