

卒業研究報告書

題目

ヒット曲の特徴量分析

指導教員

石水 隆 講師

報告者

21-1-037-0003

宇賀 由人

近畿大学工学部情報学科

令和7年2月2日提出

概要

近年、様々な分野で AI が急速に発展しており、AI が用いられている分野の一つに音楽がある。近年の日本で流される音楽シーンには様々なジャンルの曲があり、人気の高いアーティストもたくさんいる。そんな中でも人気がある曲、人気がない曲が存在する。あくまで人気がない曲が悪いわけではないが、どのような曲が大衆受けするのかを知っておくことは音楽活動をする人たちや音楽を楽しむ消費者にとって、とても重要なテーマである。

本研究では、2008 年から 2023 年までのビルボード年間 Top5 の楽曲と比較対象として YOUTUBE での再生回数 1000 万以下の楽曲、100 万以下の楽曲を Python で作成したプログラムを使用して分析を行なった。用いた特徴量は、音の強さやエネルギーを示す RMS、楽曲の 1 分間あたりの拍数を表す数値である BPM、音波のゼロラインを越える回数で、曲の粗さや滑らかさを表すゼロクロス率、音の明るさやシャープさを示す指標として機能するスペクトル重心、曲の時間である。

楽曲に対する本研究の分析結果として、ヒットする要因は RMS の時間による変動が少ないことと、スペクトル重心値が楽曲を通して安定していることが関係していると判明した。

目次

1.	序論	5
1.1	本研究の背景.....	5
1.2	本研究の目的.....	5
1.3	本報告書の構成.....	5
2	楽曲の特徴量分析	5
2.1	特徴量分析に使用する楽曲.....	5
2.2	特徴量抽出の手法.....	6
2.3	特徴量の説明.....	8
-2.3.1	RMS (Root Mean Square Energy)	8
-2.3.2	BPM (Beats Per Minute)	8
-2.3.3	ゼロクロス率 (ZCR)	9
-2.3.4	スペクトル重心.....	9
-2.3.5	曲の時間	10
3	数値分析結果	10
-3.1	BPM の分析結果	10
-3.2	ゼロクロス率の分析結果	10
-3.3	スペクトル重心の分析結果.....	11
-3.4	楽曲の長さの分析結果.....	11

4	グラフ分析結果	11
•4.1	RMS の分析結果	11
•4.2	スペクトル重心の分析結果	13
■	平坦型	14
■	山なり型	14
■	不規則型	15
5	考察・今後の課題	16
•5.1	考察	16
•5.1.1	BPM	16
•5.1.2	ゼロクロス率	16
•5.1.3	スペクトル重心	16
•5.1.4	楽曲の長さ	19
•5.1.5	RMS	20
•5.1.6	RMS とスペクトル重心の相関性	21
•5.2	今後の展望	21
6	まとめ	21
	謝辞	23
	参考文献	24

1. 序論

1.1 本研究の背景

近年、様々な分野で AI が急速に発展してきており、AI を用いたビッグデータを分析や新製品の開発など、我々の生活に AI は大きく関わるようになった。AI が用いられている分野の一つに音楽がある。

近年の日本で流される音楽シーンには様々なジャンルの曲があり、人気の高いアーティストもたくさんいる。そんな中でも人気がある曲、人気がない曲が存在する。あくまで人気がない曲が悪いわけではないが、どのような曲が大衆受けするのかを知っておくことも音楽活動をする人たちや音楽を楽しむ消費者にとって、とても重要なテーマである。

これまでの研究では、横山らの結果より曲のコード進行を分析することで年代によってヒットしている曲のメロディとコード進行の関係性が違ったり、分かり易いダイアトニックコードや 7th コード以外はあまり使用されていないなかったりすることなどが判明している[1]。また、吉田らの研究では歌詞とボーカルの性質に関して、ヒット曲のボーカルの声の明るさや歌詞の特徴をある程度分析はできているが、その結果がどのようにヒットする要因になったのかは結果として出ていない[2]。このように、既知の結果では曲のコード進行、歌詞、ボーカルの声質などの特徴を元にヒットした要因を分析している。

1.2 本研究の目的

既知の結果では、歌に関しては歌声がとても重要視される。また、曲に関してはコード進行を中心に分析しているため楽器がどのようにヒットする要因に関係しているかがコード進行でしか判断することが出来ない。勿論楽曲においてコード進行や歌声、歌詞はいずれも重要な要素であり、ヒットする要因を研究する上では無視することはできない。しかし、他の特徴量も分析することで楽曲がヒットする要因をより詳しく、新たな視点で見つけ出せるのではないかと考えられる。

そこで本研究では、上記に挙げた特徴以外に、他 5 つの特徴量を分析することで楽曲がヒットする要因を探す。

1.3 本報告書の構成

本報告書の構成としては 2 章で研究に使用した楽曲の説明、その楽曲からの特徴量抽出手法、特徴量の説明を行い、第 3, 4 章で結果を図・表と共に記し、第 5 章で考察と今後の課題を述べた後、第 6 章で結論・まとめを記す。

2 楽曲の特徴量分析

本研究では、ヒット楽曲の特徴量分析を行い、楽曲がヒットした要因を分析する。

2.1 特徴量分析に使用する楽曲

本研究では特徴量を抽出する楽曲として、ビルボードチャートの日本公式サイト[4]より、2008 年から 2023 年までの Top5 に入っている全 80 曲を対象とした。また、比較対象としてビルボードチャートに 2008 年から 2023 年までの Top5 に入っているアーティストの曲かつ、YOUTUBE に楽曲が投稿されていて再生数が低いものを 26 曲分析対象とした。“再生数が低い”の基準としては、1000 万回再生以下のものとする。1000 万回再生もされていれば充分ヒットしていると考えられるが、ビルボードチャート Top5 に入るほどの有名アーティストでは、1000 万回再生を超えていない楽曲を見つける方が困難を極めたため、同じアーティストの中で相対的に再生数が低いものを比較対象として使用することに決めた。また、ビルボードチャートの Top5 には載っていないとも 100 万回再生以上 1000 万回再生未満の曲も 10 曲追加し、分析を行った。そして、再生数が 100 万回以下の楽曲も 40 曲分析し、比較対象とした。

まとめると、ヒットしている楽曲としてビルボードチャートの 2008 年から 2023 年までの Top5 に入っているも

のをを用い、ヒットしていない楽曲として 100 万回再生以上 1000 万回再生以下の楽曲、100 万回再生以下の楽曲を用いた。表 1 に使用した楽曲の一覧を表す。

表 1 使用した楽曲

種別	曲数
ビルボードチャート Top5	80
YOUTUBE 再生回数 100 万回以上 1000 万回以下	40
YOUTUBE 再生回数 100 万回以下	40

2.2 特徴量抽出の手法

特徴量の抽出方法について、本研究では Google Drive に楽曲を保存し、Python の librosa ライブラリ [7] を用いて楽曲を読み込んだ後特徴量を抽出する。そして抽出したデータの数値を平均としてまとめ、ビルボードチャートの楽曲とそうでない楽曲で比較する。また、グラフも用いて視覚的にも比較していく。図 1 に本研究で作成したプログラムを示す。また、以下にプログラムについて説明する。

2 行目で librosa ライブラリをインポートし、5 行目で Google Drive に保存した楽曲を分析できるように Google Drive をマウントする。引数 audio_path に音声ファイルのパスを代入し、11 行目で音声ファイルの読み込みを行う(y には音声データ, sr にはサンプリング周波数が入る)。本研究ではサンプリング周波数はデフォルトの 22050Hz とする。13 行目では分析したデータをグラフにするために Matplotlib ライブラリをインポートしている。17 行目でテンポの推定、22 行目で RMS を計算している。25 行目から 33 行目で RMS をグラフにし、表示している。36 行目ではゼロクロス率を計算している。46 行目でスペクトル重心の計算をし、48 行目で時間軸を作成した後に 52 行目から 58 行目でスペクトル重心をグラフにし、表示している。

```
1 from google.colab import drive
2 import librosa
3
4 # Google Drive をマウント
5 drive.mount('/content/drive')
6
7 # Google Drive 内の音声ファイルのパスを指定
8 audio_path = '/content/drive/MyDrive/夜に.m4a'
9
10 # 音声ファイルの読み込み
11 y, sr = librosa.load(audio_path)
12
13 import matplotlib.pyplot as plt
14 print(audio_path)
15
16 # テンポの推定
```

```

17 tempo, _ = librosa.beat.beat_track(y=y, sr=sr)
18
19 print(f"テンポ (BPM): {tempo}")
20
21 # RMS エネルギーを計算
22 rms = librosa.feature.rms(y=y)
23 print(f"RMS: {rms.mean()}")
24 # RMS エネルギーをプロット
25 plt.figure(figsize=(10, 4))
26 librosa.display.waveshow(y, sr=sr, alpha=0.4)
27 plt.plot(librosa.times_like(rms), rms[0], label='RMS Energy',
28 color='r')
29 plt.title('RMS Energy')
30 plt.xlabel('Time (s)')
31 plt.ylabel('Energy')
32 plt.legend()
33 plt.show()
34
35 # ゼロクロス率の計算
36 zero_crossings = librosa.feature.zero_crossing_rate(y)
37
38 print(f"ゼロクロス率: {zero_crossings.mean()}")
39
40 y, sr = librosa.load(audio_path) # sr=None で元のサンプリング周波数を保
41 持
42
43 print(f"サンプリング周波数: {sr} Hz")
44
45 # スペクトル重心の計算
46 spectral_centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
47 print(f"スペクトル重心: {spectral_centroid.mean()}")
48 # 時間軸を作成
49 frames = range(len(spectral_centroid[0]))
50 t = librosa.frames_to_time(frames, sr=sr)
51 # スペクトル重心のプロット
52 plt.figure(figsize=(10, 4))
53 librosa.display.waveshow(y, sr=sr, alpha=0.4)
54 plt.plot(t, spectral_centroid[0], color='r')

```

```
55 plt.title('Spectral Centroid')
56 plt.xlabel('Time (s)')
57 plt.ylabel('Frequency (Hz)')
58 plt.show()
```

図 1 特徴量の抽出コード

2.3 特徴量の説明

本節では、本研究で用いた特徴量について説明する。

-2.3.1 RMS(Root Mean Square Energy)

RMS は音の振幅の変化を表す特徴量である。音の強さやエネルギーを示し、グラフ化することで、時間領域の信号の変動を捉えることができる。このエネルギーの値(y 軸の値)は絶対的な基準が無いため、曲中の値の変動に着目する。これを分析することで、曲中の盛り上がりがどこにくるのか、落ちサビ(最後のサビ前に挿入される楽器の音量を極端に落としているサビ)があるのか、などを確認することができる。よって、ヒットしている楽曲には何かしらの共通点があるのでは無いかと考えた。図 2 に RMS のグラフの例を示す。赤の線が音の強さを表しており、ラインが高い箇所はエネルギーが高く音量が大きい部分で、ラインが低い箇所はエネルギーが低く静かな部分や無音の部分である。背景の青い部分は元の音声信号を表している。

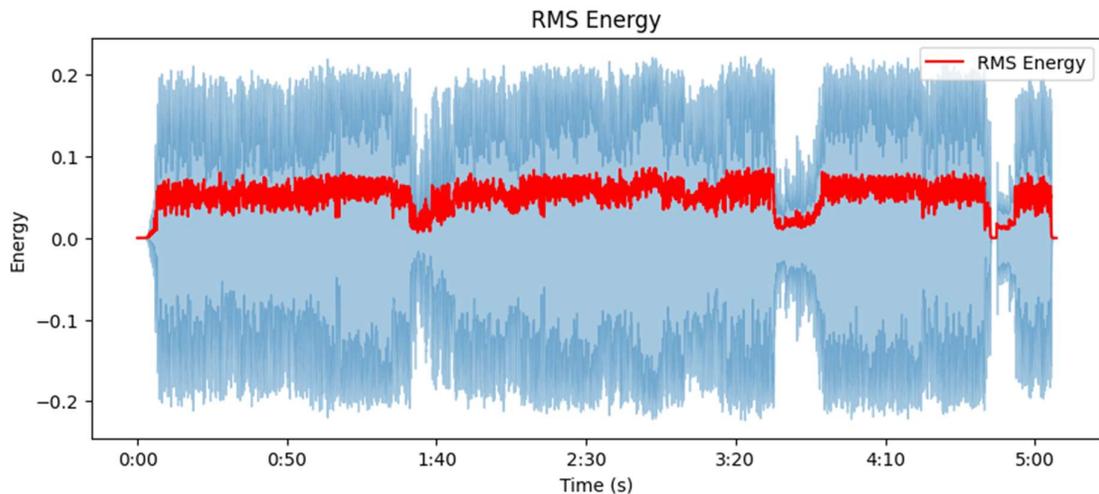


図 2 RMS のグラフ例

-2.3.2 BPM(Beats Per Minute)

BPM は楽曲の 1 分間あたりの拍数を表す数値であり、曲の速さを捉えることができる。ヒットしている楽曲の BPM とヒットしていない楽曲の BPM で違いが出るかもしれないと考えられるため、本研究では BPM を特徴量として用いる。

・2.3.3 ゼロクロス率(ZCR)

波形データの振幅が「0」の値をとる水平線をゼロラインといい、信号処理ではゼロラインを基準として波形が上下に振動する。ゼロクロス率は音波のゼロラインを越える回数で、曲の粗さや滑らかさを表す。例えば、高いゼロクロス率の場合はシンバルやハイハットの音が多く使用されており、低いゼロクロス率の場合はピアノの低音やチェロなどが多く使用されている楽曲ということになる。この特徴量を分析することにより、ヒットしている曲とヒットしていない曲でゆったりな曲調なのか激しめの曲調なのかなどの違いが明らかになるかもしれないと考えられる。

また、他の特徴量は楽曲の音の大きさや明るさをそのままグラフにしたり時間をそのまま抽出したりしたのに対し、ゼロクロス率はやや複雑な計算をしたものを結果として抽出しているため、ゼロクロス率を求める式についても説明をする。まず、ゼロクロス率を求める式は

$$ZCR = \frac{1}{N} \sum_{t=1}^N 1[y[t-1] \cdot y[t] < 0]$$

となる。Nはフレーム内のサンプル数であり、本研究ではlibrosaライブラリ[7]のデフォルト値(2048)を使用している。y[t]はフレーム内のt番目のサンプルであり、t番目のサンプルの値とt-1番目の値の積の符号が0以下になった場合、音波がゼロラインを超えたと判定して分子に1を加算する。これをサンプルの数行い、1フレーム内のゼロクロス率を求めることが出来る。そして、図1のゼロクロス率の計算で使用している。本研究で作成したプログラムでは、上記の方法で求めた1フレーム内のゼロクロス率を信号が分割されたフレームの合計数を加算し、全てのフレームで割ることで全フレームのゼロクロス率の平均値を計算している。

$$ZCRの平均値 = \frac{1}{M} \sum_{i=1}^M ZCR[i]$$

Mはフレームの総数であり、ZCR[i]はi番目のゼロクロス率である。また、上記のゼロクロス率を求める計算で用いている1フレーム(2048サンプル)についての説明も行う。まず2048サンプルの内の1サンプルの長さは

$$\frac{1}{22050} \approx 0.00004535 \text{ 秒}$$

である。これを基にし、1フレーム(2048サンプル)の長さは

$$2048 \times 0.00004535 \approx 0.09288 \text{ 秒}$$

である。まとめると、1フレーム(0.09288秒)の中で何回ゼロラインを超えたかの平均を計算し、その値を分割された数のフレームでさらに平均をとったものを、ゼロクロス率として求めている。

・2.3.4 スペクトル重心

スペクトル重心は音の明るさやシャープさを示す指標として機能し、スペクトル重心の値が高いほど音が「明るい」または「鋭い」傾向がある。逆に低い場合は音が「暗い」または「こもった」感じである。本研究では、楽曲全体の平均として抽出されるスペクトル重心の値に加え、グラフも作成することでどのタイミングで音が明るくなっているのかや暗くなっているのかも確認できるようにする。このようにして得られた結果で、ヒットしている曲とヒットしていない曲での曲の明るさの違いやスペクトル重心の値の増減がどのように変化しているのかが分かるのでは無いかと考えた。図3にスペクトル重心のグラフの例を示す。

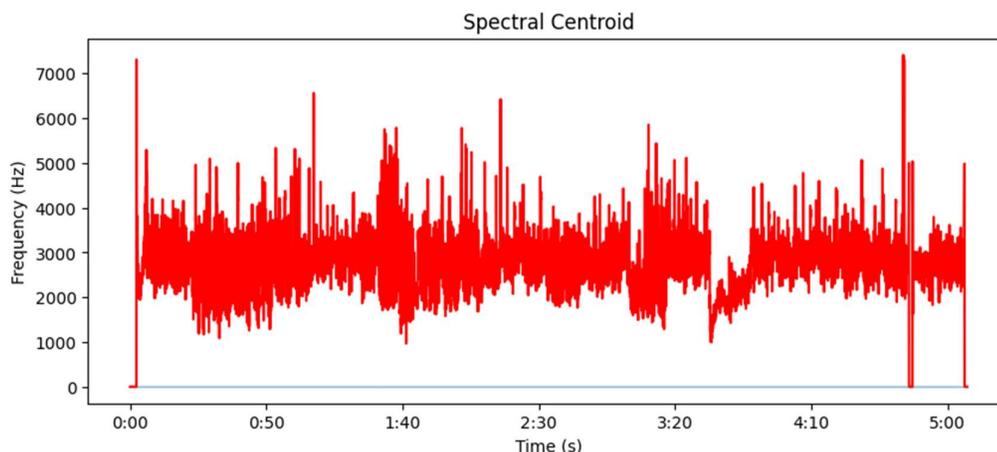


図 3 スペクトル重心のグラフ例

-2.3.5 曲の時間

楽曲には長いものも短いものもある。ヒットする要因として、楽曲の長さも関係しているのではないかと考えられるため、本研究では楽曲の長さも特徴量として扱う。

3 数値分析結果

本章では、各特徴量の分析結果を記す。表 2 に各年代の楽曲が持つ各特徴量分析結果を示す。

表 2 各年代の楽曲の各特徴量

ビルボード										
年代	2008	2009	2010	2011	2012	2013	2014	2015		
BPM	107.8	122.6	135.4	125.4	146.4	142	126.4	132.2		
ゼロクロス率	0.1	0.12	0.12	0.14	0.16	0.12	0.12	0.12		
スペクトル重心	2392	2582	2476	2844	2863	2415	2452	2454		
長さ(m)	4.4	4.2	4.4	4.4	4.4	4.4	4.8	4.6		
									平均	
年代	2016	2017	2018	2019	2020	2021	2022	2023		
BPM	141.5	126.2	128.6	93.4	107	127.4	125.4	139.4	129.775	
ゼロクロス率	0.13	0.11	0.12	0.09	0.1	0.1	0.09	0.11	12%	
スペクトル重心	2600	2484	2549	2153	2337	2268	2208	2450	2559.75	
長さ(m)	5.2	4.6	4.2	4.4	4.4	3.6	3.6	4	4.45	
1000万以下	平均	100万以下	平均							
BPM	118.21194		111.025							
ゼロクロス率	10%		10%							
スペクトル重心	2359.883		2313.35							
長さ(m)	4.1		3.4							

-3.1 BPM の分析結果

表 2 に示される通り、BPM についての分析結果は、100 万再生以下、1000 万再生以下、ビルボードチャートの楽曲と、人気になるにつれて高くなっていった。また、ビルボードチャートの各年代の変化に規則性はなかった。

-3.2 ゼロクロス率の分析結果

表 2 に示される通り、ゼロクロス率についての分析結果は、100 万回再生以下の楽曲、1000 万回再生以下の楽曲

共に 10%であり、ビルボードチャート年間 Top5 の楽曲は 12%となっている。

・3.3 スペクトル重心の分析結果

表 2 に示される通り、スペクトル重心についての分析結果は、100 万再生以下、1000 万再生以下、ビルボードチャートの楽曲と、人気になるにつれて高くなっていった。

・3.4 楽曲の長さの分析結果

表 2 に示される通り、楽曲の長さについての分析結果は、100 万再生以下、1000 万再生以下、ビルボードチャートの楽曲と、人気になるにつれて長くなっていった。

4 グラフ分析結果

本章では、グラフとして抽出した RMS とスペクトル重心についての分析結果を記載する。

・4.1 RMS の分析結果

第 2 章で述べた通り、RMS とは音の振幅の変化を表す特徴量であり、グラフにすることで楽曲の信号の変動を時間領域で捉えることができる。本研究では全 160 曲を分析し、その結果として信号の変動を 3 種類のパターンに分けることができた。

まずは、楽曲全体を通して信号の変動があまり変化しないパターン(本報告書では平坦型と呼ぶ)を図 4 に示す。このような楽曲はイントロ、A メロ、B メロ、サビなどであまり音量に変化がない。赤い線は音の強さの変化を表しており、青の部分は元の音声信号を表している。

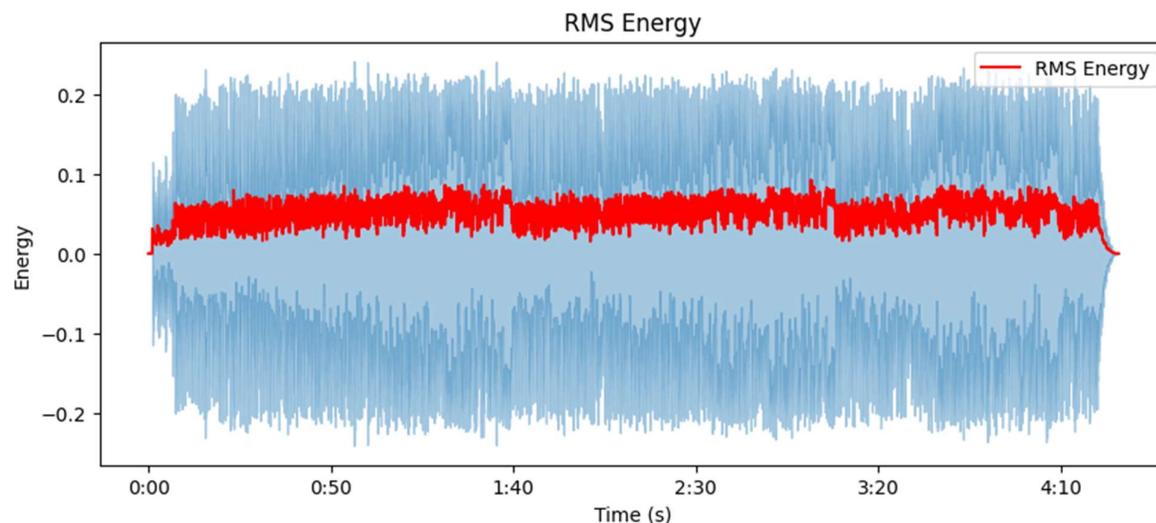


図 4 平坦型の例(I AM YOUR SINGER/サザンオールスターズ)

次に、楽曲全体を通して信号が大きい時間と信号が小さい時間が山なりのようにある程度規則的にあるパターン(本報告書では山なり型と呼ぶ)を図 5 に示す。このような楽曲は A メロや B メロは落ち着いた雰囲気、サビで盛り上がるものが当てはまる。

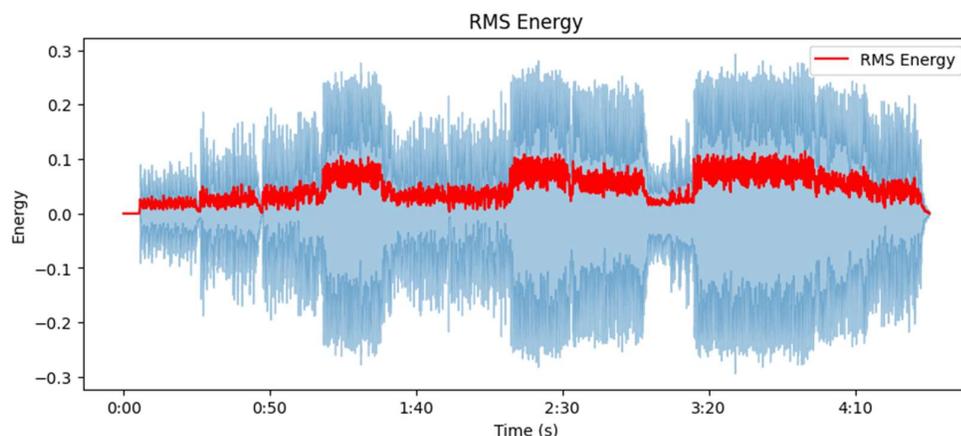


図 5 山なり型の例(キセキ/GReeeeN)

最後に、楽曲全体を通して不規則的に信号の大きさが変動するパターン(本報告書では不規則型と呼ぶ)を図 6 に示す。このような楽曲は不規則的に音の大きさが変動し、急に盛り上がり急に静かになったり、忙しい楽曲になっていることが多い。

楽曲の種別と RMS のパターンの関係を表 3 に示す。

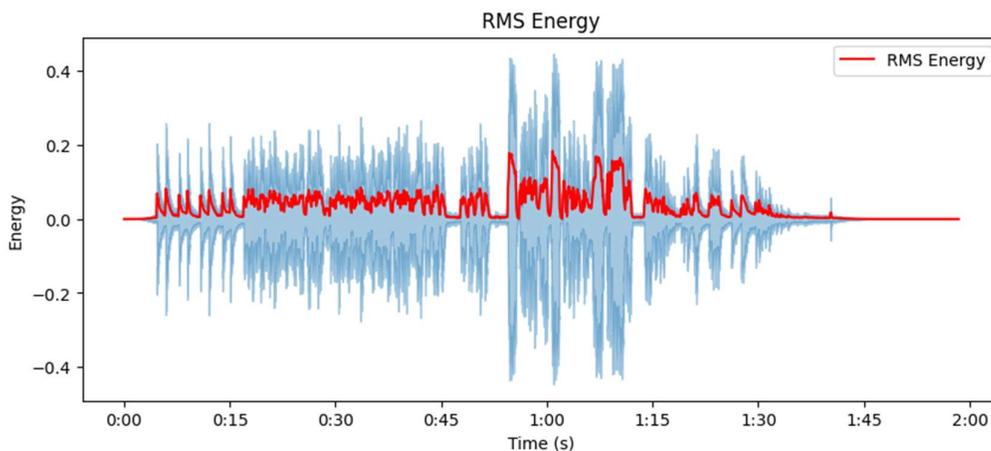


図 6 不規則型の例(Travelers/Official髭男dism)

表 3 パターン内訳

	平坦型	山なり型	不規則型
ビルボード	41	33	6
1000 万以下	18	15	7
100 万以下	25	6	9

ビルボードチャートの楽曲では平坦型が約 51%、山なり型が約 41%、不規則型が約 8%となっている。1000 万再

生以下の楽曲では平坦型が約 45%，山なり型が約 37%，不規則型が約 18%となっている。100 万再生以下の楽曲では平坦型が約 62%，山なり型が約 15%，不規則型が約 23%となっている。

ビルボードチャートの年間 TOP5 の楽曲と、そうでない楽曲とを比較すると、ビルボードチャートの年間 TOP5 でない楽曲の方が不規則型の割合が高くなっていることが分かった。また、100 万再生以下の楽曲は山なり型の割合が極端に低下した。

そして、ビルボードチャートの楽曲の中でも、昔の年代では平坦型が多かったのが、最近の年代に近づくにつれて山なり型のものが増えてき始めた。

・4.2 スペクトル重心の分析結果

第 2 章で述べた通り、スペクトル重心は音の明るさを示す特徴量である。音の大きさを捉える RMS に加えて音の明るさも分析することで楽曲の雰囲気をもっと捉えることができる。スペクトル重心の平均数値については第 3 章で述べた通り、100 万再生以下、1000 万再生以下、ビルボードチャートの楽曲と、人気になるにつれて高くなっていった。ここでは、抽出したグラフを用いてスペクトル重心の安定具合を分析する。図 7 では 1500Hz から 3000Hz にスペクトル重心が集中しており、1500Hz の幅がある。この幅を記録していくことにより、スペクトル重心の安定具合を比較することができる。また、スペクトル重心も RMS と同様に平坦型、山なり型、不規則型の 3 パターンに分けることができた。

まずはスペクトル重心の幅について 1000Hz 以下のもの、1000Hz~1500Hz、1500Hz~2000Hz、2000Hz~2500Hz、2500Hz 以上のものをまとめた結果を表 4 に示す。

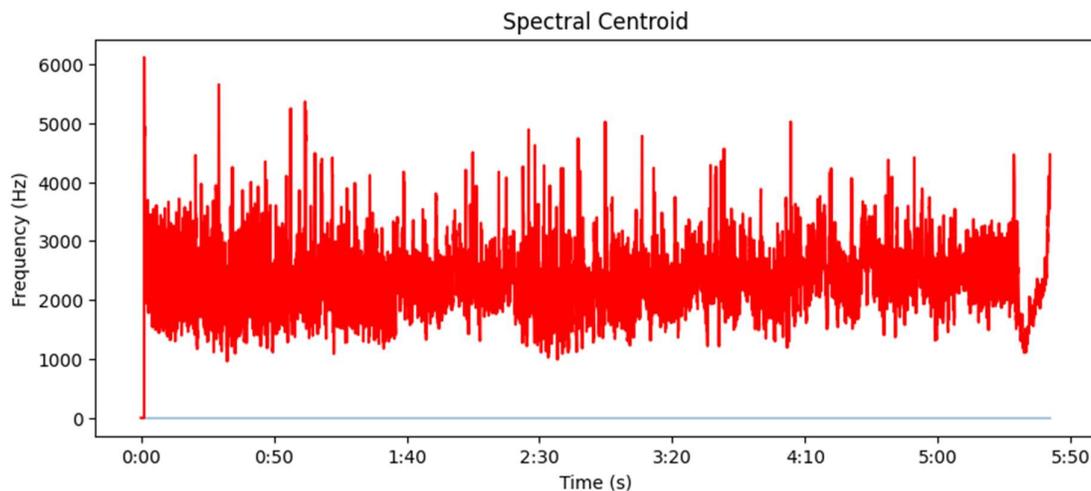


図 7 スペクトル重心の例(HANABI/Mr.Children)

表 4 スペクトル重心の幅まとめ

	1000Hz 以下	1000Hz~1500Hz	1500Hz~2000Hz	2000Hz~2500Hz	2500Hz 以上
ビルボード	14	19	23	11	13
1000 万以下	1	4	15	12	8
100 万以下	3	7	18	4	8

ビルボードチャートの楽曲は 1000Hz 以下が 17%、1000Hz~1500Hz が 24%、1500Hz~2000Hz が 29%、2000Hz~2500Hz が 14%、2500Hz 以上が 16%となった。1000 万再生以下の楽曲は 1000Hz 以下が 2%、1000Hz~1500Hz が 10%、1500Hz~2000Hz が 38%、2000Hz~2500Hz が 30%、2500Hz 以上が 20%となった。100 万再生以下の楽曲は 1000Hz 以下が 7%、1000Hz~1500Hz が 18%、1500Hz~2000Hz が 45%、2000Hz~2500Hz が 10%、2500Hz 以上が 20%となった。

次に、スペクトル重心の時間経過による変動を平坦型、山なり型、不規則型の 3 パターンの特徴を説明した後、3 パターンに分けた結果を示す。

■ 4.1.1 平坦型

スペクトル重心における平坦型(図 8)は楽曲全体を通して、明るさがあまり変化しないことを意味する。つまり、使用される楽器が途中で変化したり追加されたりすることが少なく、最初から最後まで似たような雰囲気楽曲である。

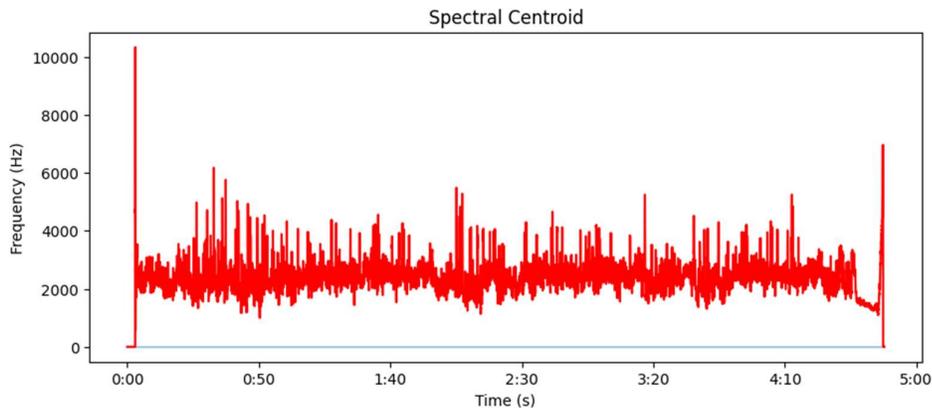


図 8 スペクトル重心における平坦型の例(マイガール/嵐)

■ 4.1.2 山なり型

スペクトル重心における山なり型(図 9)は盛り上がる場所と落ち着いている場所でスペクトル重心の値が山なりのように変化している。これは、サビなどの盛り上がる場所で楽曲の雰囲気が明るくなったり落ちサビで暗くなったりしていることを意味する。

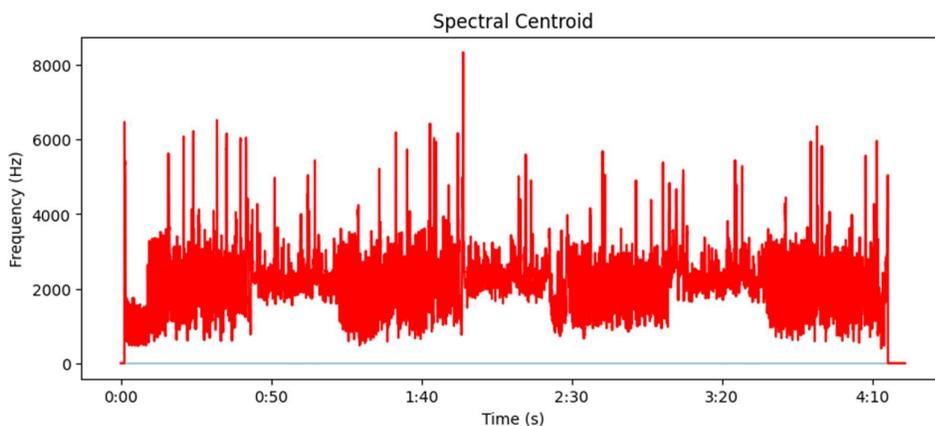


図 9 スペクトル重心における山なり型の例(老人と海/ヨルシカ)

■ 4.1.3 不規則型

スペクトル重心における不規則型(図 10)は楽曲中でスペクトル重心の値が安定せず不規則な形で変動しているものである。このような楽曲は A メロ、B メロがきてサビがくるという JPOP の王道な形ではない楽曲や、歌声を聞かせる目的でイントロや間奏以外で楽器の数を減らしている楽曲が多い。

表 5 にスペクトル重心のパターン内訳を示す。

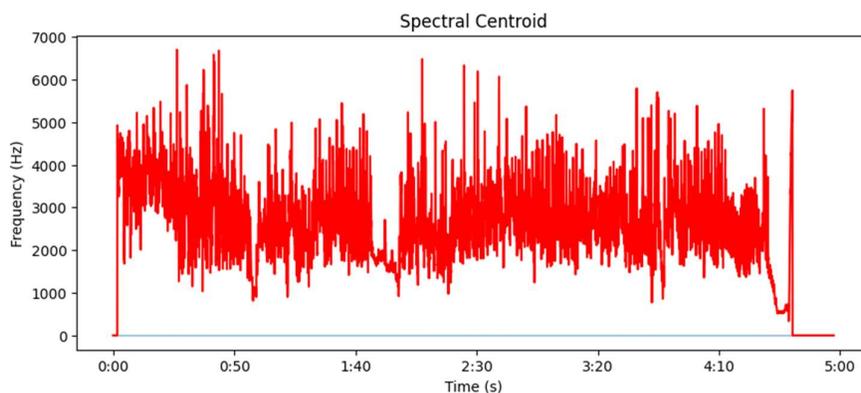


図 10 スペクトル重心における不規則型の例(I LOVE.../Official 髭男 dism)

表 5 スペクトル重心のパターン内訳

	平坦型	山なり型	不規則型
ビルボード	41	30	9
1000 万以下	15	21	4
100 万以下	19	15	6

ビルボードチャートの楽曲では平坦型が約 51%, 山なり型が約 38%, 不規則型が約 11%となっている。1000 万再生以下の楽曲では平坦型が約 37%, 山なり型が約 53%, 不規則型が約 10%となっている。100 万再生以下の楽曲では平坦型が約 47%, 山なり型が約 38%, 不規則型が約 15%となっている。

表 3 に示した RMS のパターン内訳と似たような内訳になった。これは、楽曲中の盛り上がる場所や落ち着いた

ているところで音量と音の明るさが比例の関係にあるからだと考えられる。違いとしては、山なり型の割合が増えていることである。これは、サビなどで音の強さは変わらないが、音の明るさが増している楽曲があるためだと考えられる。

5 考察・今後の課題

・5.1 考察

本章では、第3章、第4章の結果を踏まえてヒット曲の要因となる特徴量について考察する。

・5.1.1 BPM

BPMについては第3章で述べた通り、人気になるにつれて高くなっていき、その差は18であった。結果から見ると、ヒットする要因としてはBPMを約130あたりにすることが分かった。しかし、ビルボードチャート年間TOP5における年代ごとのBPMの平均は表2に示す通り、18以上の差が開いている年代がある。例えば、2011年と2012年ではBPMに20もの差がある。さらに2008年から2023年の間にBPMの平均値と年代には相関性が見られない。このことから、ビルボードチャートの楽曲のBPMと再生回数100万回以下の楽曲の平均値に差が18あったとしても、ビルボードチャートの年代によってBPMの平均値に18以上の差があることからヒットする要因として挙げるには差が小さいと考えられる。

・5.1.2 ゼロクロス率

ゼロクロス率については第3章で述べた通り、100万回再生以下の楽曲、1000万回再生以下の楽曲共に10%であり、ビルボードチャート年間Top5の楽曲は12%となっている。結果から見ると、ヒットする要因としては、ゼロクロス率を約12%にすると良いということが分かる。しかし、開いている差はわずか2%であり、表2に示す通り、2019年から2023年は平均するとゼロクロス率が約10%である。このことから、ゼロクロス率も、ヒットする要因としてはあまり大きな役割は果たせていないと考えられる。

・5.1.3 スペクトル重心

スペクトル重心については第3章で述べた通り、100万再生以下、1000万再生以下、ビルボードチャートの楽曲と、人気になるにつれてスペクトル重心の平均値は高くなっていった。このことから、ヒットする要因として楽曲の明るさが高い方が良いということが考えられる。しかし、100万再生以下の楽曲の平均値とビルボードチャートの平均値には約250の差があり、ビルボードチャートの年代によっては平均値から250以上離れている2011年代、2012年代、2019年代、2021年代、2022年代の5つの年代があるため、平均値の結果だけではまだヒットする要因としてスペクトル重心の明るさが関係していると言うにはまだ弱い。そこで、第4章のグラフの分析結果についても考察する。表4を基に円グラフを図11、図12、図13として示す。これらのグラフから読み取れるのはビルボードチャートの楽曲は、他の楽曲に比べてスペクトル重心の振れ幅が小さいということである。また、スペクトル重心のグラフパターンの内訳を示した図14、図15、図16から分かる通り、ビルボードチャートの楽曲は平坦型の内訳が他の楽曲よりも多い。よって、スペクトル重心の値が大きいかつスペクトル重心の値が楽曲中を通して安定していることが、ヒットする要因の1つだと考えられる。また、ビルボードチャートの楽曲、1000万再生以下の楽曲に比べて100万再生以下の楽曲では不規則型の割合が増えていることからスペクトル重心の安定性がヒットする要因に関係していることが考えられる。

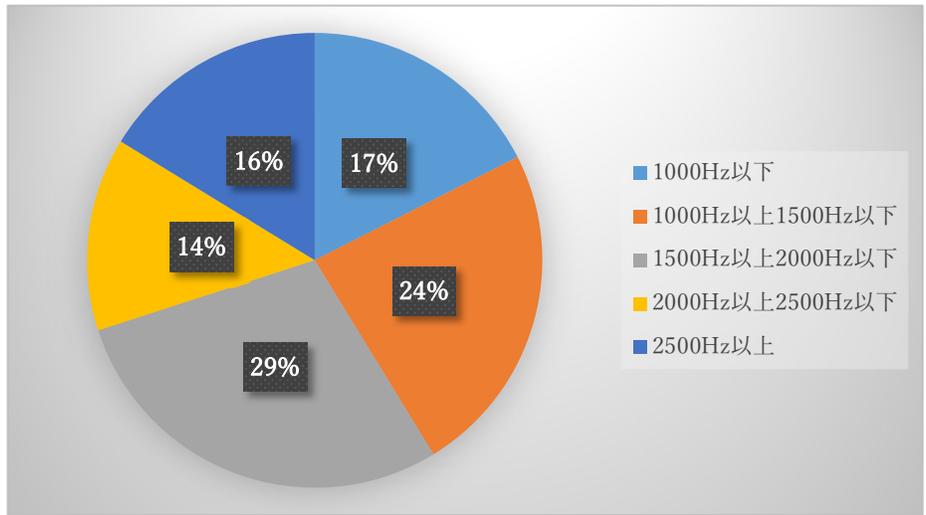


図 11 ビルボードチャートのスペクトル重心振れ幅

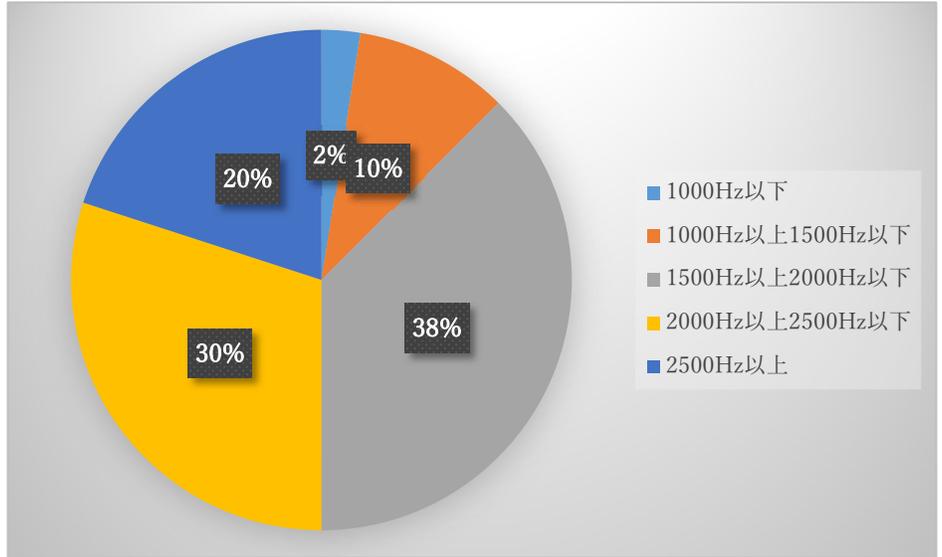


図 12 1000 万再生以下のスペクトル重心振れ幅

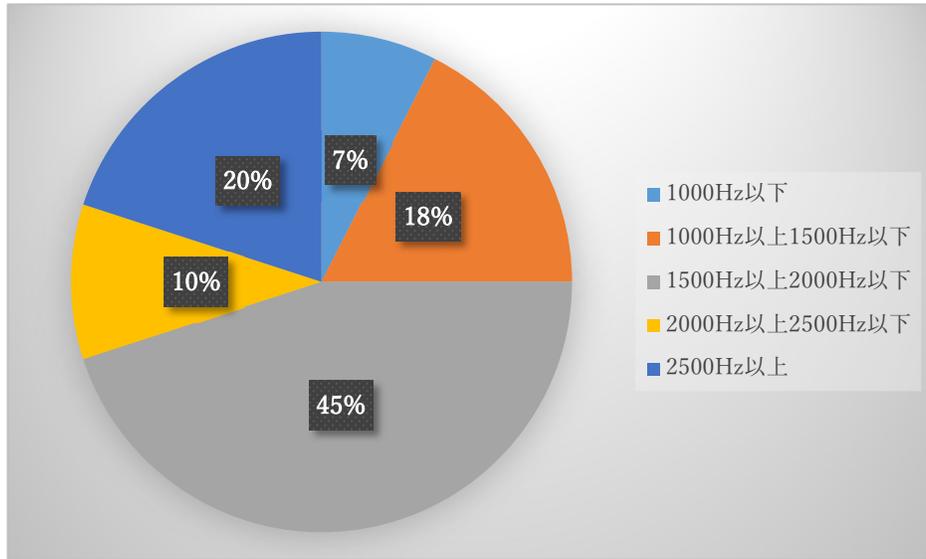


図 13 100万再生以下のスペクトル重心振れ幅

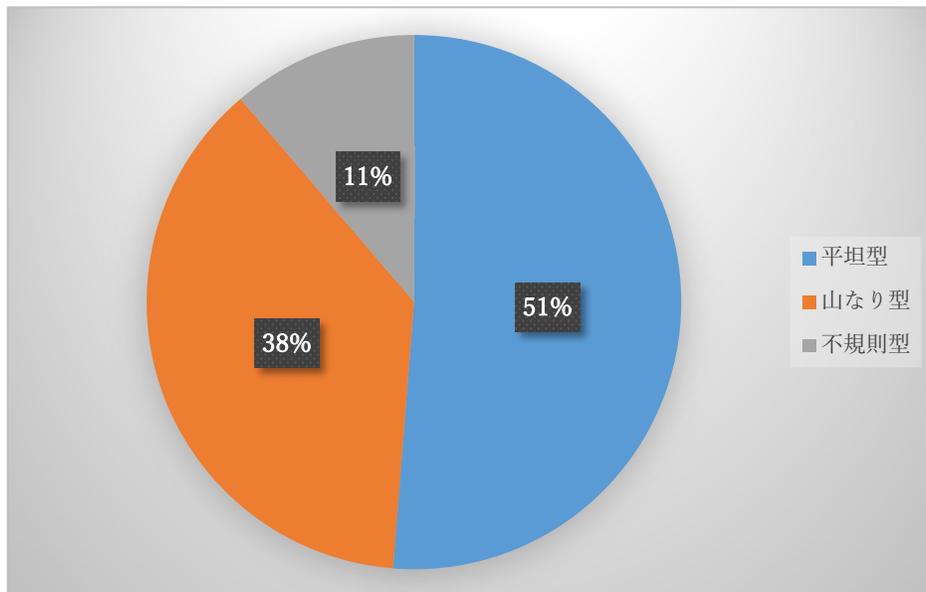


図 14 スペクトル重心のグラフパターン(ビルボード)

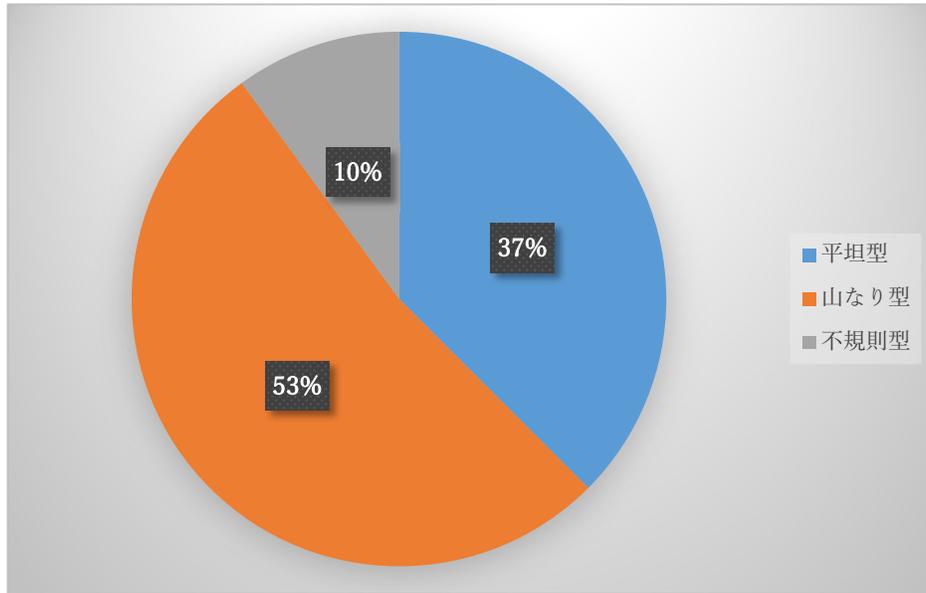


図 15 スペクトル重心のグラフパターン(1000 万再生以下)

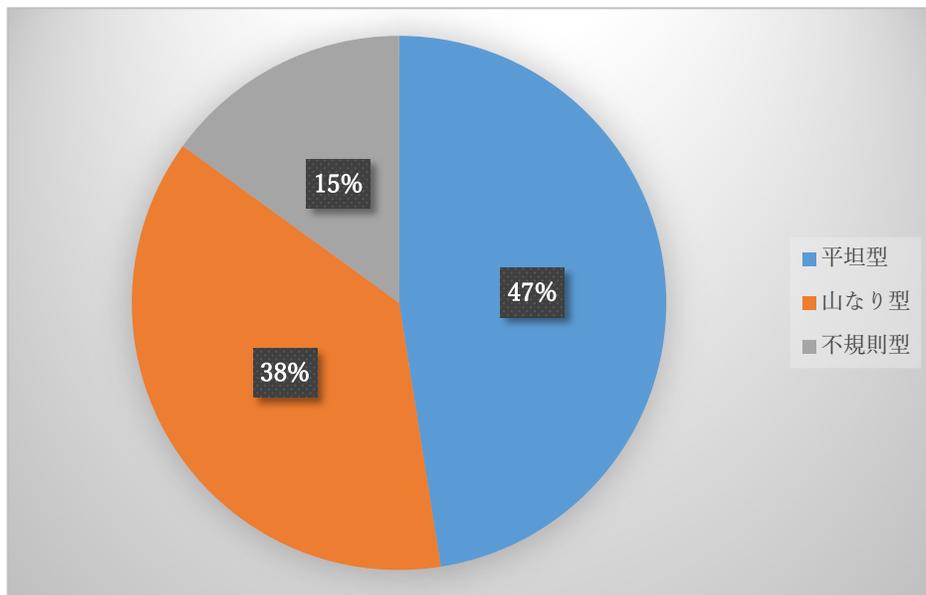


図 16 スペクトル重心のグラフパターン(100 万再生以下)

・5.1.4 楽曲の長さ

表 2 に示される通り、楽曲の長さについての分析結果は、100 万再生以下、1000 万再生以下、ビルボードチャートの楽曲と、人気になるにつれて長くなっていった。長さの平均値だけで判断すると、長い楽曲の方がヒットしやすいと言える。しかし、表 2 に示されているように、2016 年代、2021 年代、2022 年代を除く年代は全て平均が 4 分代であるため長ければ良いというものでもない。さらに、2021 年代、2022 年代はそれぞれ平均値が 3.6 分であり、100 万再生以下の楽曲の平均値は 3.4 分であることから楽曲の長さは極端に短かったり長かったりしない限りヒットする要因としては考えにくい。

・5.1.5 RMS

RMS は第 4 章で示した通り、楽曲中の音の強さが時間によってどのように変動しているのかを平坦型、山なり型、不規則型の 3 パターンに分けて分析した。RMS のグラフパターンを示した図 17、図 18、図 19 から分かる通り、ビルボードチャートの楽曲は不規則型の割合が 10~15%程少なくなっている。このことからヒットする要因として音の信号を、楽曲を通して不規則に変動させない方が良いということが考えられる。

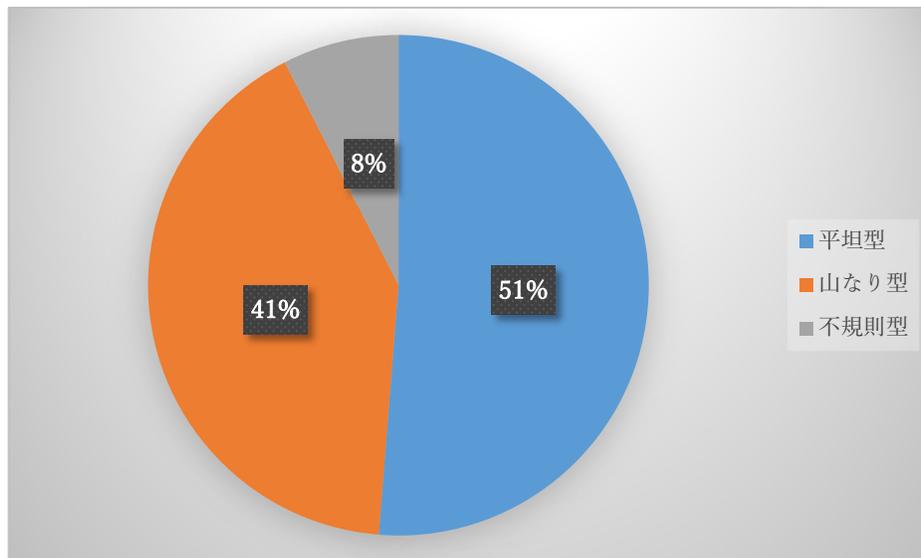


図 17 RMS のグラフパターン(ビルボード)

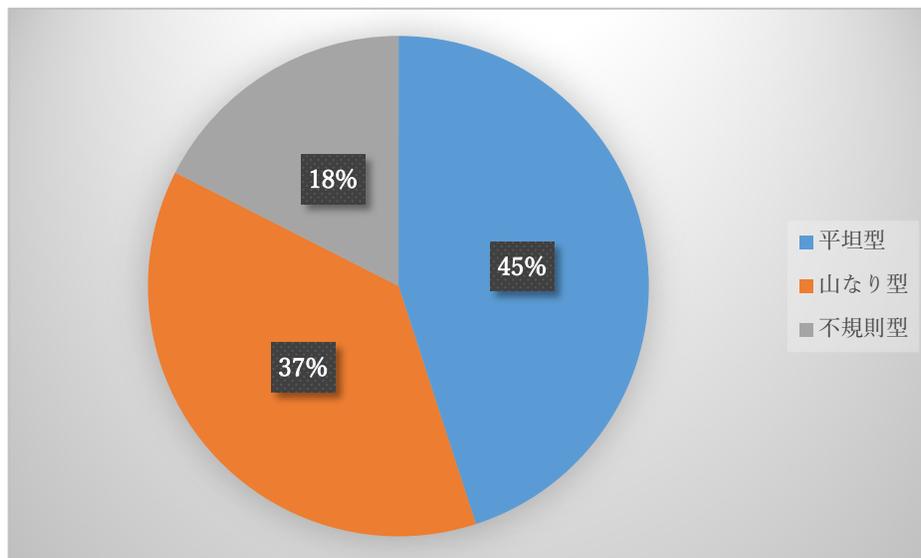


図 18 RMS のグラフパターン(1000 万再生以下)

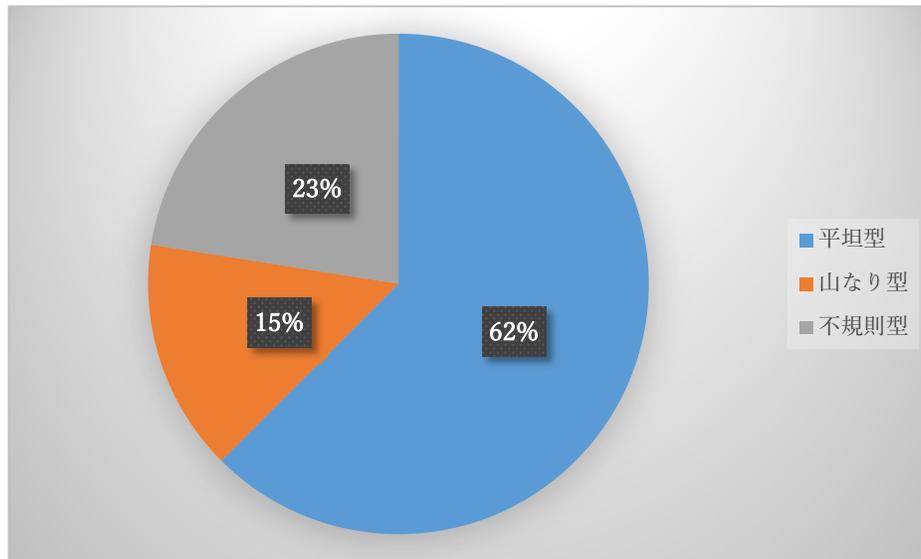


図 19 RMS のグラフパターン(100 万再生以下)

・5.1.6 RMS とスペクトル重心の相関性

RMS は信号の大きさ、スペクトル重心は音の明るさを示しているものであり・5.1.3 節、・5.1.5 節で示した通り、本研究で使用した特徴量の中でヒットする要因に大きく関係していると考えられる特徴量の 2 つである。この 2 つの特徴量は作成したグラフの時間による変動を平坦型、山なり型、不規則型の 3 パターンに分けて分析した。ビルボードチャートの楽曲では 3 パターンの内訳は図 14 と図 17 に示されているように、ほとんど変化が無い。それに比べて 1000 万再生以下の楽曲では図 15 と図 18 に示されているように、RMS に比べてスペクトル重心の方が山なり型の割合が大きくなり、平坦型と不規則型の割合が小さくなっている。これは、楽曲を通して信号の大きさにあまり変化が無くても音の明るさは明るくなったり暗くなったりしていたり、信号の大きさが不規則に変化していても、音の明るさは規則的に明るくなったり暗くなったりしているということである。100 万再生以下の楽曲では図 16 と図 19 に示されているように、RMS に比べてスペクトル重心の方が平坦型の割合が 15%も減少している。これは、信号の大きさにあまり変化が無くても音の明るさは明るくなったり暗くなったりしているということである。

これらのことから分かるように、RMS とスペクトル重心の時間による変動のパターンが同じであることもヒットする要因だと考えられる。

・5.2 今後の展望

本研究では、5 つの特徴量を使用して楽曲の分析を行い RMS とスペクトル重心がヒットする要因になるという結果が出たが、データの数をもっと増やして分析してみると結果に変化が出る可能性がある。

また、表 2 に示されているようにヒットする楽曲というのは年代によって全く違ったりする。本研究のように特徴量を分析することによってヒットする要因を見つける以外にも、各年代でどのような特徴を持った楽曲がヒットしていたのかを調査することにより、これからはどのような楽曲が流行しそうなかを予想することもできるのではないだろうかと考えられる。

6 まとめ

本報告書では、楽曲がヒットする要因として考えられる特徴量について述べ、5 つの特徴量を使用して楽曲の分析

を行った。結果としては **RMS** とスペクトル重心の時間による変動がヒットする要因に深く関係していることが分かった。さらに、**RMS** とスペクトル重心の時間による変動は同じ形になっているほどヒットしていることが分かった。要するに、音の信号を不規則に大きくしたり小さくしたりせず、**RMS** が楽曲を通して安定しており、**RMS** とスペクトル重心の時間による変動が同じ形になっている楽曲がヒットする要因だと考えられる。

謝辞

本研究を行うにあたって，終始石水隆講師からの熱心な指導を行って頂いたことに感謝の意を表します．

参考文献

- [1] 横山真男, 斎藤勇也: ヒットチャートランキング上位に入る楽曲の特徴分析, 情報処理学会研究報告音楽情報科学 (MUS) Vol.2015-MUS-106, No.22, pp.1-6 (2015).
<http://id.nii.ac.jp/1001/00113335/>
- [2] 吉田健太, 大槻明: 声道の共鳴特性・共振点, 曲のテンポ及び歌詞の特徴量を機械学習に応用した“ヒット曲”の特徴分析, 情報処理学会第86回全国大会講演論文集 Vol.2024, No.1, pp.369-370(2024).
<http://id.nii.ac.jp/1001/00235907/>
- [3] 戸上真人: Python で学ぶ音源分離, インプレス (2020) .
- [4] billboard JAPAN, 株式会社阪神コンテンツリンク, <https://www.billboard-japan.com/>
- [5] 伊藤彰則:環境音から異常を検知する統計的手法, 日本音響学会誌 75 巻 9 号, pp.538-543, (2019)
https://www.jstage.jst.go.jp/article/jasj/75/9/75_538/_pdf
- [6] 三好真人, 柘植覚, Choge Kipsang Hillary, 尾山匡浩, 伊藤桃代, 福見稔:音楽検索のための楽曲印象値の自動付与手法, 情報処理学会研究報告 Vol.2011-MUS-89 No23, (2011)
<http://id.nii.ac.jp/1001/00072704/>
- [7] Librosa, audio and music processing in Python, <https://librosa.org/doc/latest/index.html>

プログラムのソース

以下に本研究で作成した Python プログラムのソースを示す。

```
from google.colab import drive
import librosa

# Google Drive をマウント
drive.mount('/content/drive')

# Google Drive 内の音声ファイルのパスを指定
audio_path = '/content/drive/MyDrive/夜に.m4a'

# 音声ファイルの読み込み
y, sr = librosa.load(audio_path)

import matplotlib.pyplot as plt
print(audio_path)

# テンポの推定
tempo, _ = librosa.beat.beat_track(y=y, sr=sr)

print(f"テンポ (BPM): {tempo}")

# RMS エネルギーを計算
rms = librosa.feature.rms(y=y)
print(f"RMS: {rms.mean()}")

# RMS エネルギーをプロット
plt.figure(figsize=(10, 4))
librosa.display.waveshow(y, sr=sr, alpha=0.4)
plt.plot(librosa.times_like(rms), rms[0], label='RMS Energy', color='r')
plt.title('RMS Energy')
plt.xlabel('Time (s)')
plt.ylabel('Energy')
plt.legend()
plt.show()

# ゼロクロス率の計算
zero_crossings = librosa.feature.zero_crossing_rate(y)
```

```

print(f"ゼロクロス率: {zero_crossings.mean()}")

y, sr = librosa.load(audio_path) # sr=None で元のサンプリング周波数を保持

print(f"サンプリング周波数: {sr} Hz")

# スペクトル重心の計算
spectral_centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
print(f"スペクトル重心: {spectral_centroid.mean()}")
# 時間軸を作成
frames = range(len(spectral_centroid[0]))
t = librosa.frames_to_time(frames, sr=sr)
# スペクトル重心のプロット
plt.figure(figsize=(10, 4))
librosa.display.waveshow(y, sr=sr, alpha=0.4)
plt.plot(t, spectral_centroid[0], color='r')
plt.title('Spectral Centroid')
plt.xlabel('Time (s)')
plt.ylabel('Frequency (Hz)')
plt.show()

```