

Henrique Hermann de Figueiredo

**Análise Quantitativa da
Repetitividade em letras de música no período
de 1959 a 2019**

Belo Horizonte

2019

Henrique Hermann de Figueiredo

**Análise Quantitativa da
Repetitividade em letras de música no período de 1959 a
2019**

Monografia apresentada durante o Seminário dos Trabalhos de Conclusão do Curso de Graduação em Engenharia Elétrica da UFMG, como parte dos requisitos necessários à obtenção do título de Engenheiro Eletricista.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Curso de Graduação em Engenharia Elétrica

Orientador: Prof. Hani Camille Yehia

Belo Horizonte

2019

*Este trabalho é dedicado às pessoas que reclamam do estado atual da música pop. Sem
você eu não teria me inspirado a fazer este trabalho.*

Agradecimentos

Agradeço todos meus amigos e companheiros do G3E por tornarem meus muitos anos na faculdade o mais agradáveis o possível. Agradeço os bons professores por terem me ensinado pelo menos uma parte do que tenho que aprender e aos outros professores por me ensinar que não há nada que não possa ser aprendido sozinho. Agradeço ao meu orientador Hani por aceitar orientar este trabalho, mesmo tendo um tema tão alternativo. E agradeço minha família por ter me dado todo o suporte o possível, para que eu pudesse estudar e me formar no meu tempo.

“Without music, life would be a mistake“
(Friedrich Nietzsche)

Resumo

Este trabalho visa analisar a repetitividade das músicas mais populares do período de 1959 a 2019. Web scrapers são usados para se pegar os rankings semanais da Billboard de todas as semanas neste período, e para encontrar a letra das músicas destes rankings. As letras de música são comprimidas e o índice de compressão usado como nível de repetitividade. Gráficos são traçados com a repetitividade média de cada ano, e a popularidade média para cada repetitividade.

Palavras-chaves: Música, letra, repetição, complexidade.

Abstract

The goal of this paper is to study how repetitive the lyrics of pop songs are, now and in the past 60 years, and to look for any correlation between repetitiveness and popularity.

Key-words: lyrics, popularity, repetitiveness

Lista de ilustrações

Figura 1 – billboard logo	21
Figura 2 – fragmento do site da billboard	24
Figura 3 – fragmento do HTML do site da billboard	24
Figura 4 – diagrama do banco de dados usados nesta etapa	26
Figura 5 – Exemplo de construção de um apontador	35
Figura 6 – Exemplos de palavras com a contagem de sílabas reais e por agrupamentos de vogais	38
Figura 7 – Porcentagem de rankings da billboard e MaisTocadas coletados por ano	40
Figura 8 – Porcentagem de letras das músicas da billboard e MaisTocadas coletadas por ano	40
Figura 9 – Repetitividade média das músicas da Billboard e MaisTocadas por ano	41
Figura 10 – Repetitividade média das músicas da Billboard por ano	41
Figura 11 – Popularidade média e quantidade de músicas por repetitividade	42
Figura 12 – Popularidade média por repetitividade para níveis de repetitividade com mais de 200 músicas	42
Figura 13 – Tamanho efetivo médio por ano	43
Figura 14 – Popularidade e quantidade de músicas por tamanho efetivo.	43
Figura 15 – Popularidade por tamanho efetivo	44
Figura 16 – Média de palavras difíceis por ano	44
Figura 17 – Popularidade por quantidade de palavras difíceis	45
Figura 18 – Popularidade por quantidade de palavras difíceis	45
Figura 19 – Média de palavras pouco usadas por ano	46
Figura 20 – Popularidade média por palavras pouco usadas	46
Figura 21 – Popularidade média por palavras pouco usadas	47
Figura 22 – Complexidade média por ano	48
Figura 23 – Complexidade média por ano	49
Figura 24 – Complexidade média por ano	49

Lista de tabelas

Tabela 1 – Trecho de "Around the World", do album "Homework" de 1997, Daft Punk	27
Tabela 2 – Trecho original e original reduzido de "Around the World", Daft Punk .	28
Tabela 3 – Trecho alterado e alterado reduzido de "Around the World", Daft Punk	28
Tabela 4 – Função lyricCompress	30
Tabela 5 – Trecho original e original reduzido de "Around the World", Daft Punk .	31
Tabela 6 – Trecho alterado e alterado reduzido de "Around the World", Daft Punk	31
Tabela 7 – Visualização do significado do apontador mostrado na tabela 5	32
Tabela 8 – Visualização do significado dos apontadores mostrados na tabela 6 . .	33
Tabela 9 – Músicas mais populares	53
Tabela 10 – Músicas mais vezes na primeira posição	54
Tabela 11 – Músicas mais presentes nos rankings	54
Tabela 12 – Artistas mais populares	55
Tabela 13 – Artistas mais vezes na primeira posição	55
Tabela 14 – Artistas mais presentes nos rankings	56
Tabela 15 – Artistas em mais posições simultâneas	56

Sumário

1	INTRODUÇÃO	19
1.1	Motivação	19
1.2	Abordagem	19
2	COLETA DE DADOS	21
2.1	Escolha da Fonte dos Dados	21
2.2	Como coletar os dados	22
2.3	Web Scraper	22
2.3.1	Geração das URLs	22
2.3.2	Tratamento dos Dados	23
2.4	Armazenando os dados	25
3	PROCESSAMENTO DE DADOS	27
3.1	Repetitividade	27
3.1.1	Mensurando Repetitividade	27
3.1.2	Compressão	29
3.2	Popularidade	35
3.3	Tamanho Efetivo	36
3.4	Palavras difíceis	37
3.5	Palavras pouco usadas	38
4	INTERPRETAÇÃO DOS DADOS	39
4.1	Qualidade da Coleta de dados	39
4.2	Análise de Repetitividade	40
4.3	Análise de tamanho Efetivo	42
4.4	Análise de Palavras difíceis	44
4.5	Análise de Palavras pouco usadas	46
4.6	Complexidade	47
5	CONCLUSÕES	51
6	EXTRA	53
	Bibliografia	57

1 Introdução

Parte-se do pressuposto que o leitor saiba o que é uma música [1], o que é a letra de uma música e o que um ranking [17]. No decorrer do trabalho também espera-se que o leitor tenha conhecimentos mesmo que rudimentares de programação, saiba o que é MySQL [2] e o que é uma query.

1.1 Motivação

Músicas são uma parte integral da cultura humana desde antes da escrita. Estão muito presentes no nosso dia a dia e muitas vezes definem culturas e gerações. Algo que costuma-se ouvir com alguma frequência, principalmente de pessoas de gerações passadas, é que "não fazem mais músicas como antigamente", que "músicas estão ficando cada vez mais repetitivas" ou que "músicas pop são muito repetitivas".

Este trabalho tem como objetivo descobrir se estas afirmações são verdadeiras usando métodos empíricos e replicáveis.

1.2 Abordagem

Este trabalho trata principalmente com a repetitividade em músicas e com sua popularidade relativa. Por isso é preciso definir uma forma de se medir esta repetitividade e popularidade, e uma forma de escolher as músicas que participarão da análise.

Para a repetitividade pode-se usar o áudio da música, mas identificar repetitividade em arquivos de áudio, principalmente em uma grande volume, consumiria uma quantidade considerável de recursos computacionais [6], mais do que temos disponível. Além disso a obtenção de arquivos de áudio de uma quantidade suficientemente alta de músicas específicas é um problema complexo que foge do escopo do trabalho.

Outra possibilidade é utilizar a partitura da música. Encontrar padrões de repetitividade em partituras é bem mais simples do que em arquivos de áudio. Porém encontrar partituras de músicas específicas é difícil.

Uma terceira possibilidade é analisar a letra da música. Encontrar letras de músicas específicas é fácil, há vários sites dedicados a isso, com letras de praticamente qualquer música. E encontrar repetitividade em letras de música (que são simplesmente arquivos de texto) é mais fácil que em arquivos de áudio. Por estes motivos foi definido que a repetitividade nas músicas será analisada com base em suas letras.

Como foi definido que será usada a letra das músicas, decidiu-se tentar analisá-las também com outras métricas que permitam uma análise mesmo que rudimentar da complexidade das mesmas. Neste trabalho complexidade se refere a dificuldade de se memorizar a letra de uma música. Uma letra de maior complexidade é mais difícil de ser memorizada por completo. Estas métricas escolhidas foram:

- palavras difíceis;
- palavras pouco usadas;
- tamanho efetivo;

O que exatamente são estas métricas e como elas podem ser calculadas será discutido posteriormente.

Para a popularidade pode-se usar um ranking de popularidade, que nos dá uma lista de músicas ordenadas por sua popularidade relativa.

Os rankings de popularidade também podem ser usados para se definir quais músicas devem participar da análise. Qualquer música que apareça no ranking de popularidade vai entrar no estudo, se sua letra puder ser obtida.

O trabalho será então dividido em 4 partes:

- obtenção de rankings semanais de popularidade de músicas dos últimos 60 anos;
- obtenção das letras das músicas presentes nos rankings;
- criação das métricas para mensurar a repetitividade das letras e outras para tentar mensurar sua complexidade, e as aplicar a todas as músicas obtidas;
- análise dos dados obtidos;

As primeiras duas partes, ambas de coleta de dados, serão explicados no capítulo 2, a terceira parte, de processamento de dados, no capítulo 3, e a última parte, de análise de dados, nos capítulos 4 e 5.

2 Coleta de Dados

2.1 Escolha da Fonte dos Dados

Para a escolha das fonte dos dados usados no trabalho alguns requisitos foram considerados:

- a) os dados devem ser precisos e consistentes, pois só assim os resultados obtidos podem ser considerados válidos;
- b) é necessária uma abundância de dados, pois assim os resultados serão mais precisos;
- c) os dados devem ser apresentados de uma forma parametrizada, para que sua obtenção possa ser automatizada;

Com estes pré requisitos em mente pode-se escolher a fonte dos rankings de popularidade:



Figura 1 – billboard logo

A billboard é uma revista americana fundada em 1894, inicialmente com um foco em propaganda, mas posteriormente em música [14].

Desde a década de 1950 ela tem publicado rankings semanais com as músicas mais populares dos EUA, inicialmente somente 20 músicas por ranking, depois 50 e, a partir de 1959, 100 músicas por ranking através do seu famoso "Hot 100". Inicialmente os rankings eram compilados usando-se listas de músicas tocadas nas rádios, providos pelas próprias rádios, e dados de vendas de álbuns, providos pelas gravadoras. Hoje os rankings também consideram a venda de músicas individuais, em plataformas como o Itunes, e streaming das músicas em plataformas como o youtube e o spotify.

Quanto às letras das músicas, foram escolhidos sites de letras com arquivos grandes e variados, além de URLs determinísticas (o motivo da necessidade destas URLs determinísticas será explicado posteriormente). Os sites escolhidos foram "metrolyrics.com" e "songlyrics.com".

Foi escolhido ainda uma segunda fonte de rankings, para rankings voltados para o mercado brasileiro. Para este foi usado o site "maistocadas.mus.br". Ele também tem rankings de 100 músicas mais populares, mas nele os rankings são anuais, e não semanais.

Para as letras das músicas brasileiras foi usado o site "vagalume.com.br".

2.2 Como coletar os dados

A coleta dos dados pode ser feita de diversas maneiras. As mais comuns são através de uma API (Application Programming Interface, ou interface de aplicação de programação [13]) ou usando o HTML (código fonte) dos sites fontes.

No geral é preferível usar APIs pois as empresas que as utilizam e tornam públicas o fazem para melhor administrar requisições pesadas, torná-las mais rápidas e eficientes e evitar tráfego desnecessário em seus sites. O problema em usar APIs é que nem todo site tem uma API pública, e os que o tem muitas vezes cobram para que desenvolvedores e pesquisadores possam utiliza-las. As APIs gratuitas costumam limitar o número de utilizações diárias em 100, o que seria muito pouco para este trabalho.

Como usar as APIs dos sites não é uma opção, foi definido que os dados seriam coletados direto do HTML dos sites.

2.3 Web Scraper

Um web scraper [12] é, em termos simples, um programa que coleta dados da internet.

No caso deste trabalho os web scrapers usados fazem um loop onde, a cada iteração novo endereço URL é gerado, este é usado para se fazer uma requisição (análogo a digitar o endereço de um site em um browser e o acessar), o HTML do site acessado com a URL gerada é copiado, procura-se e encontra-se (se houver) dados relevantes no HTML e estes são então salvos em um banco de dados previamente criado.

2.3.1 Geração das URLs

A importância das URLs serem determinísticas é que só assim pode-se "adivinhar" as URLs necessárias. Pode-se prever a URL necessária para se acessar uma página nunca visitada com base nas URLs já conhecidas.

Oos rankings da billboard por exemplo podem ser acessados pelo site da billboard, usando URLs no formato "https://www.billboard.com/charts/hot-100/YYYY-MM-DD", onde YYYY-MM-DD é a data da qual o ranking é desejado. Já as letras de música no Metrolyrics podem ser acessadas por URLs no formato "http://www.metrolyrics.com/nome-

da-musica-lyrics-nome-do-artista.html", onde "nome-da-musica" e "nome-do-artista" são o nome da música e do artista em questão, com quaisquer espaços em branco substituídos por um "-".

2.3.2 Tratamento dos Dados

Uma vez que a URL desejada é gerada, usa-se a função "urlopen" da biblioteca "urllib" [5] do python [11] para se obter o HTML da página em questão. Porém, como este método gasta a mesma banda que o acesso de um usuário real, mas pode ser feito muito mais rapidamente (e portanto em uma frequência maior), ele pode apresentar um risco à integridade dos sites acessados. Muitos programas acessando um mesmo site dessa forma simultaneamente podem congestionar os servidores do site com grande facilidade. Por este motivo alguns sites, como o da billboard por exemplo, restringem este tipo de acesso. Mas o acesso à Billboard por este método é necessário, o que torna importante burlar este sistema. Para isso foi feita uma função chamada "get_url_data" que cria um cabeçalho falso e o insere na função "urlopen" junto da URL, garantindo assim que o site acessado trate o acesso como um acesso de usuário comum. Para evitar problemas nos sites acessados foi ainda inserido um delay entre acessos (assim a frequência de requisições não fica muito alta), garantiu-se que os web scrapers funcionassem nas horas de menor tráfego, ou seja, dentre meia noite e 7 da manhã para os sites brasileiros e entre 3 e 10 da manhã para os americanos.

Após pegar o HTML do site referente à URL em questão pode-se usar a função "soup" da biblioteca "BeautifulSoup" [9] do python para se indexar os vários elementos do HTML, e então o método ".findAll" para se encontrar elementos específicos no HTML indexado. É importante já saber de antemão como os dados serão recebidos, para que se possa customizar o método ".findAll". No caso do ranking da billboard por exemplo, um fragmento da página com o ranking da semana de 09/03/2019 pode ser visto na figura 2, e um fragmento do HTML referente ao mesmo na figura 3:

Pode-se observar que cada posição no ranking tem, no HTML, um elemento <div> individual, da classe "chart-list-item", e que cada um destes <div> tem 3 elementos chamados "data-rank", "data-artist" e "data-title", contendo informação sobre o ranking, artista e título da música em questão. Assim, pode-se usar o método ".findAll" com os parâmetros ("div", {"class": "chart-list-item"}) para se criar uma lista com todos os elementos <div> com classe "chart-list-item", onde cada elemento dessa lista terá 3 parâmetros indexados com os índices "data-rank", "data-artist" e "data-title" contendo as informações desejadas.

Um processo semelhante ao descrito é feito para todos os sites usados como fontes de dados.

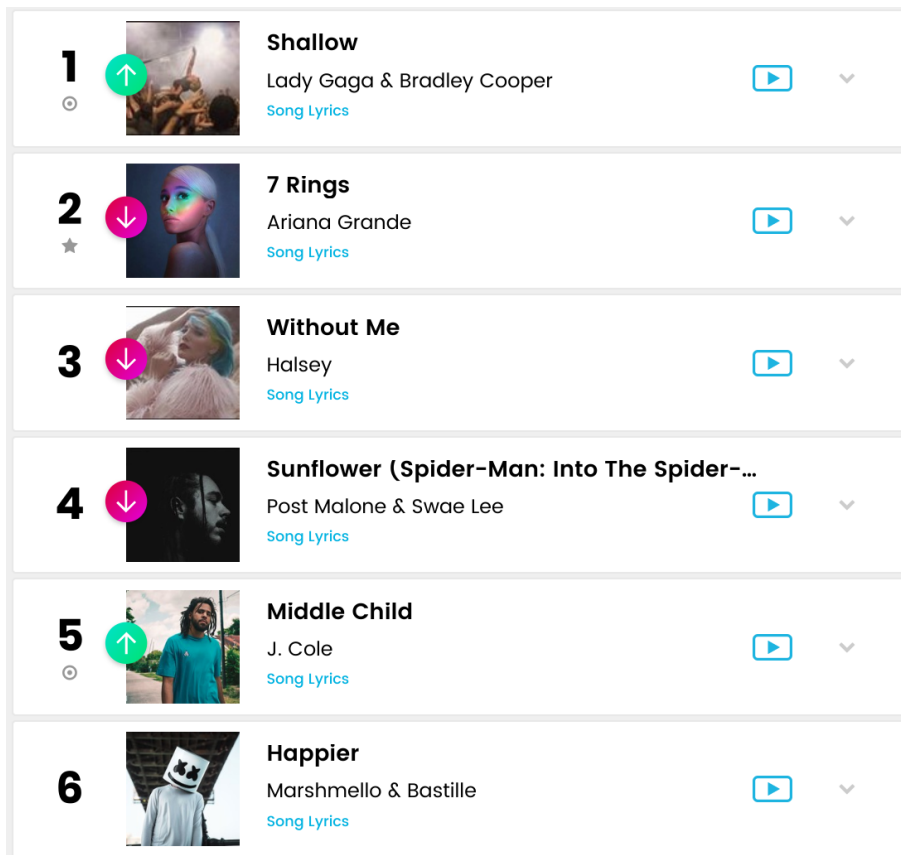


Figura 2 – fragmento do site da billboard

```

<div class="chart-list-item" data-rank="1" data-artist="Lady Gaga & Bradley Cooper" data-title="Shallow" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="2" data-artist="Ariana Grande" data-title="7 Rings" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="3" data-artist="Halsey" data-title="Without Me" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="4" data-artist="Post Malone & Swae Lee" data-title="Sunflower (Spider-Man: Into The Spider-Verse)" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="5" data-artist="J. Cole" data-title="Middle Child" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="6" data-artist="Marshmello & Bastille" data-title="Happier" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="7" data-artist="Ariana Grande" data-title="Thank U, Next" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="8" data-artist="Post Malone" data-title="Wow." data-has-content="true">...</div>
<div class="chart-list-item" data-rank="9" data-artist="Blueface" data-title="Thotiana" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="10" data-artist="Travis Scott" data-title="Sicko Mode" data-has-content="true">...</div>
<div class="ad-holder ad-holder--m" style="display:none">...</div>
<div class="chart-list-item" data-rank="11" data-artist="Panic! At The Disco" data-title="High Hopes" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="12" data-artist="benny blanco, Halsey & Khalid" data-title="Eastside" data-has-content="true">...</div>
<div class="chart-list-item" data-rank="13" data-artist="Ariana Grande" data-title="Break Up With Your Girlfriend, I'm Bored" data-has-content="true">...

```

Figura 3 – fragmento do HTML do site da billboard

2.4 Armazenando os dados

Os dados coletados pelos web scrapers devem ser, idealmente, armazenados de forma eficiente e evitando redundância, permitindo que eles sejam posteriormente usados mais facilmente.

Foi decidido que o ideal é criar um banco de dados em MySQL para armazenar estes dados. No banco criado nesta primeira etapa existem 3 tabelas, uma para os rankings da Billboard, uma para os rankings da MaisTocadas, e uma para as músicas presentes nos rankings. Um esquemático disso pode ser visto na Figura 4. Nela podemos ver que:

- os elementos de cada tabela tem um ID único (sua primary key);
- as músicas da tabela Songss tem uma "unique key" composta por artista e música (não podem existir dois elementos com o mesmo artista e a mesma música na tabela Songss);
- as tabelas de rankings, Billboard e MaisTocadas, tem uma "unique key" composta pela data e pelo ranking (não podem existir duas posições iguais numa mesma data);
- as tabelas de ranking tem uma "foreign key" que referencia o id de alguma música da tabela Songss;

Com isso as informações como nome do artista e nome da música só precisam ser escritas uma vez, pois os rankings simplesmente os referenciam.

Para se armazenar as informações no banco de dados criado foi usada uma biblioteca chamada "mysql.connector"[8], também de python. Com ela pode-se executar queries no banco de dados, então o processo de inserir dados é simplesmente criar uma query contendo os dados a ser inseridos, e então executá-la.

Porém é preciso garantir que cada música seja inserida somente uma vez na tabela Songss. Para isso usamos das "unique keys" definidas na banco de dados, e do comando "ignore" nas queries usadas. Com esse comando, caso a query não possa ser executada o banco de dados não gera um erro, ele só prossegue com a execução da próxima query, como se a query problemática não tivesse existido. Assim, com as "unique keys", caso haja tentativa de inserir uma música já inserida não haverá um erro, e a música não será inserida. Isso torna o processo de inserção mais eficiente, pois não é necessário checar se os elementos podem ou não ser inseridos.



Figura 4 – diagrama do banco de dados usados nesta etapa

3 Processamento de Dados

Com os dados relevantes (ranking, artista, título das músicas e letras das músicas) já coletados e devidamente armazenados deve-se agora processá-los para se encontrar métricas que possam ser analisadas. Neste capítulo serão definidas quais métricas são desejadas e como consegui-las.

3.1 Repetitividade

Todos sabemos intuitivamente o que é repetitividade. É fácil identificar uma letra de música repetitiva. Mas quantizar isso é bem complicado. Contar quantas palavras únicas há em uma letra de música e dividir pelo número total de palavras é um método que alguns podem considerar simples e intuitivo. Porém ele não leva em conta a ordem das palavras. Neste método ambos os trechos abaixo teria a mesma repetitividade:

Tabela 1 – Trecho de "Around the World", do album "Homework" de 1997, Daft Punk

Trecho Original	Trecho alterado
Around the World	Around the World
Around the World	The Around World
Around the World	The World Around
Around the World	Around the World
Around the World	The Around World
Around the World	The World Around

É obvio que o segundo trecho é menos repetitivo, então ele tem uma repetitividade menor, porém se considerarmos somente as palavras, e não como estão ordenada, ambos os trechos teriam a mesma repetitividade.

3.1.1 Mensurando Repetitividade

Como procurar palavras repetidas não é o bastante, pode-se então procurar não por palavras repetidas, mas grupos de caracteres repetidos. Assim serão consideradas palavras que se repetem, grupos de palavras que se repetem na mesma ordem e também fragmentos de palavras que se repetem (sílabas por exemplo). Há várias formas diferentes de se fazer isso.

O método escolhido usa "apontadores". Sobre apontadores por hora basta saber que são conjuntos de caracteres facilmente identificáveis que referenciam um outro caractere e uma quantidade. A explicação do método escolhido para este trabalho segue:

Procura-se na letra conjuntos de 4 ou mais caracteres que se repitam. Mantém-se a primeira vez que o conjunto aparece intacto, e substitua o nas vezes consecutivas por por algo que aponte para o conjunto original, ou seja, substitua-o por um apontador. Os conjuntos devem ter no mínimo 4 caracteres pois os apontadores construídos tem sempre 3 caracteres, então eles estariam substituindo algo maior que eles.

O trecho original de Around the World tem o mesmo verso repetido 6 vezes. Dessa forma, com o uso de apontadores o trecho exemplo original poderia ser reduzido para:

Tabela 2 – Trecho original e original reduzido de "Around the World", Daft Punk

Trecho Original	Trecho reduzido
Around the World Around the World Around the World Around the World Around the World Around the World	Around the World %@#

Onde %@# é o apontador (%@# é uma representação, o apontador real será explicado posteriormente).

Já no trecho alterado os conjuntos de caracteres "The " "Around " e "World" no segundo e quinto versos também aparecem no primeiro mas em ordem diferente, e o "The World " e "Around" do terceiro e sexto versos também aparecem no primeiro, enquanto o quarto verso é exatamente igual o primeiro. Portanto o trecho alterado poderia ser reduzido para:

Tabela 3 – Trecho alterado e alterado reduzido de "Around the World", Daft Punk

Trecho Alterado	Trecho reduzido
Around the World The Around World The World Around Around the World The Around World The World Around	Around the World %@#%@#%@#%@#%@#%@#

Onde cada %@# representa um apontador. Pode-se observar que este método faz que um trecho mais repetitivo fique menor que um menos repetitivo. Podemos então usar a razão entre o tamanho original e o tamanho reduzido como uma proporção de originalidade. Analogamente podemos usar 1 menos esta razão como a proporção de repetitividade [16]. De forma geral:

$$R_{ep} = 1 - \frac{S_{red}}{S_{org}}$$

Onde R_{ep} é a repetitividade do texto analisado, S_{red} é o tamanho do texto reduzido e S_{org} é o tamanho do trecho original.

Para os trechos exemplo original (1) e alterado (2) temos:

$$R_{ep1} = 1 - \frac{S_{red1}}{S_{org1}} = \frac{20}{50} = 0.60$$

$$R_{ep2} = 1 - \frac{S_{red2}}{S_{org2}} = \frac{32}{50} = 0.36$$

Assim podemos verificar que a repetitividade do exemplo trecho original é 60%, e do alterado é 36%, o que evidencia que o trecho alterado é menos repetitivo que o original.

Verificamos portanto que a repetitividade de um conjunto qualquer de caracteres pode ser calculada com duas variáveis. A primeira é o tamanho do trecho, e a segunda o tamanho do trecho reduzido (ou comprimido).

É necessário então formalizar um método de compressão a ser usado para se encontrar esse tamanho reduzido. O princípio do método de compressão criado para este trabalho já foi explicado, ele consiste em substituir conjuntos de 4 ou mais caracteres por um apontador que aponte para um conjunto igual ao que foi substituído. Os detalhes de como isso é feito seguem na próxima sessão.

3.1.2 Compressão

Como foi dito anteriormente o método de compressão consiste em se substituir conjuntos de 4 ou mais caracteres no trecho a ser comprimido por um apontador que aponte para um conjunto de caracteres igual ao substituído. Esse processo é feito pela função "lyricCompressor", que pode ser vista na íntegra na tabela 4.

A função de compressão funciona através de um loop que varre o arquivo original caractere por caractere, e a cada iteração o varre da primeira posição até a posição atual. Com isso ele não só encontra os trechos repetidos mas garante que serão encontrados

Tabela 4 – Função lyricCompress

```

1 def lyricCompress(lyric):
2     aux_curr = 0
3     aux_comp = 0
4     com_lyrics = ""
5     while (aux_curr < len(lyric)):
6         cur_rep_size = 0
7         rep_size = 0
8         aux_rep = 0
9         cur_aux_comp = 0
10        for aux_comp in range(0, aux_curr+1):
11            if (lyric[aux_comp] == lyric[aux_curr] and aux_comp < aux_curr)
12        :
13            aux_rep = aux_comp
14            aux_curr2 = aux_curr
15            rep_size = 0
16            while (lyric[aux_rep] == lyric[aux_curr2]):
17                rep_size += 1
18                aux_rep += 1
19                aux_curr2 += 1
20                if (aux_curr2 == len(lyric)):
21                    break
22            if (rep_size > cur_rep_size):
23                cur_aux_comp = aux_comp
24                cur_rep_size = rep_size
25                cur_aux_curr2 = aux_curr2
26            if (cur_rep_size > 3):
27                string = pointerGen(cur_aux_comp, cur_rep_size)
28                com_lyrics = com_lyrics + string
29                aux_curr = cur_aux_curr2
30            else:
31                com_lyrics = com_lyrics + lyric[aux_comp]
32                aux_curr += 1
33    return com_lyrics

```

os maiores trechos repetidos, o que faz que o mínimo de apontadores seja necessário, maximizando a compressão.

Os apontadores são compostos de dois números, o primeiro indica a posição do caractere para o qual ele aponta e o segundo indica quantos caracteres ele representa. Estes dois números são definidos ao se encontrar sequências de caracteres repetidos (processo já mencionado).

O exemplo de compressão do trecho de "Around the World" usado anteriormente é reescrito na tabela 2, mas usando um exemplo de apontador onde seus dois números podem ser facilmente identificados. Neste exemplo o apontador %0-84% está indicando que está substituindo um conjunto de 84 caracteres, e que pode-se encontrar uma cópia do mesmo começando na posição 0. Assim sendo o primeiro caractere do trecho sendo substituído pelo apontador é igual ao caractere 0, o segundo caractere do trecho sendo substituído é igual ao caractere 1 e o último é igual ao 83.

Tabela 5 – Trecho original e original reduzido de "Around the World", Daft Punk

Trecho Original	Trecho reduzido
Around the World Around the World Around the World Around the World Around the World Around the World	Around the World %0-84%

A compressão do trecho alterado pode ser visto na tabela 6, onde %7-4% está substituindo um trecho de 4 caracteres igual a um que começa na posição 7, %0-7% está substituindo um trecho de 7 caracteres igual a um que começa na posição 0 e assim sucessivamente.

Tabela 6 – Trecho alterado e alterado reduzido de "Around the World", Daft Punk

Trecho Alterado	Trecho reduzido
Around the World The Around World The World Around Around the World The Around World The World Around	Around the World %7-4%%0-7%11-6%%7-9%%0-6%%0-50%

Note que no exemplo de compressão do trecho original o apontador indica que deve ser substituído pelos caracteres de 0 a 83, mas o trecho comprimido é composto somente dos caracteres 0-16 e do próprio apontador. Isso pode parecer um problema, afinal como copiar os caracteres além do 16, se eles nem existem? Na realidade isso não é um problema, pois o apontador não referencia o arquivo comprimido, e sim o que está sendo reconstruído. Quando o trecho sendo reconstruído chega no apontador, ele só tem 16 caracteres. Ele então copia o caractere da posição 0 na posição 17, o da posição 1 na posição 18, o da posição 2 na 19 ... o da posição 17 na posição 34, o da posição 18 na posição 35 ... o da posição 83 na posição 100. Quando ele tenta copiar o caractere 17 este já existe, pois já foi inserido anteriormente, copiado do caractere 0.

A relação de qual trecho cada apontador referencia e substitui nas tabelas 5 e 6 pode ser vista nas tabelas 7 e 8.

Tabela 7 – Visualização do significado do apontador mostrado na tabela 5

Apontador	Trecho Substituído	Trecho Referenciado
%0-84%	Around the World <u>Around the World</u> <u>Around the World</u> <u>Around the World</u> <u>Around the World</u> <u>Around the World</u>	<u>Around the World</u> <u>Around the World</u> <u>Around the World</u> <u>Around the World</u> <u>Around the World</u> Around the World

Sabendo interpretar e como definir os números de um apontador, vamos agora descobrir como efetivamente escrever este apontador. Para entender como foi decidido como escrevê-los, é necessário primeiro entender os quatro critérios definidos para sua implementação:

- como o algoritmo foi escrito para comprimir letras de música é intuitivo que os dois números do apontador serão, no máximo, o tamanho da maior letra de música. Das letras coletadas a maior é "Rap God", do rapper "Eminem" com um total de 7852 caracteres incluindo espaços em branco. Portanto espera-se que os apontadores consigam representar todos os números iguais ou menores a 7852;
- Os apontadores reais, diferentemente dos mostrados nas tabelas 7 e 8, não devem ter os números escritos explicitamente pois isso ocuparia um espaço muito grande, precisando de um caractere para indicar o início do apontador, um para indicar a separação dos números, e um para indicar o final do apontador, ocupando assim entre 5 e 11 caracteres;
- Como assume-se que todos os caracteres de um texto são representados pelo mesmo número de bits, e que este é o menor número o possível, então os caracteres do apontador devem ter o menor número de bits o possível;
- Todos os caracteres usados em Português e Inglês podem ser representados com até 8 bits;

Como sabemos que os números do apontador devem representar valores até 7852, então sabemos que eles devem ter até 13 bits (12 bits podem representar números de 0 a 4095, enquanto com 13 bits pode-se representar números de 0 a 8191 [15]).

Tabela 8 – Visualização do significado dos apontadores mostrados na tabela 6

Apontador	Trecho Substituído	Trecho Referenciado
%7-4%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around
%0-7%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around
%11-6%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around
%7-10%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around
%0-7%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around
%0-50%	Around The World The Around World The World Around Around The World The Around World The World Around	Around The World The Around World The World Around Around The World The Around World The World Around

É necessário ainda uma forma de se identificar que um caractere é parte de um apontador. Caso o número desejado fosse simplesmente escrito em binário e convertido para um caractere, o caractere resultante poderia ser um usado em português ou inglês, no caso de o número escrito estar entre 0 e 255, o que impossibilitaria sua identificação como parte de um apontador. É definido então que o valor do primeiro caractere do apontador

na tabela UNICODE [18] deve ser maior que 255, garantindo que ele precise de no mínimo 9 bits, evitando que seja confundido com um caractere normal de português ou inglês..

Adicionar um bit igual a 1 no início dos números (fazendo que eles tenham de 9 a 14 bits) garantiria que eles possam ser facilmente identificados como parte de um apontador. Mas como foi definido que todos os caracteres são representados com o mesmo número de bits isso faria que os caracteres que normalmente tem 8 bits passem a ser representados com até 14, um aumento de 75%, o que não é bom quando o objetivo é reduzir o tamanho total do arquivo.

Por estes motivos foi decidido que os apontadores seriam construídos usando-se 3 caracteres de 9 bits cada, totalizando 27 bits. O primeiro bit é sempre 1. Os 13 bits seguintes representam o primeiro número e os 13 bits finais representam o segundo número. Dessa forma o início dos apontadores é sempre um caractere com valor superior a 255, nunca usado em português ou inglês, o que facilita sua identificação. Todos os apontadores tem o mesmo tamanho, o que facilita a identificação dos outros caracteres do mesmo. E os 3 caracteres do apontador são representados com 9 bits cada, o que garante que o aumento no tamanho dos outros caracteres será o mínimo o possível, de 8 para 9 bits (um aumento de 12,5%).

O processo de construção de um apontador descrito acima pode ser visto na imagem 5. Nele podemos ver como um apontador que representa os números 4964 e 181 (ou seja, aponta para o caractere da posição 4964 e representa 181 caracteres) é construído. Pode ser percebido também que o primeiro bit do primeiro caractere vai ser sempre 1, garantindo que o caractere em questão não seja confundido com um caractere "normal".

A desvantagem deste método construtivo de apontadores é que ele parte do pressuposto que não será necessário comprimir nenhuma letra de música com caracteres que usem mais de 8 bits. Como as letras das músicas a ser comprimidas foram coletadas de diversos sites não se pode ter certeza se ela realmente tem somente caracteres que atendam este requisito. Por este motivo quando as letras das músicas estão sendo salvas todos os caracteres que não atendem este requisito são descartados. A maioria das letras de música não perdem nenhum caractere com este descarte, pois todos atendem este requisito, embora algumas usem caracteres proibidos, normalmente de pontuação, provavelmente por erro de quem colocou a letra em questão em seu site fonte.

Para se calcular a repetitividade das letras de música usa-se então um loop. Em cada iteração do loop uma letra de música é comprimida, a conta $R_{ep} = 1 - \frac{S_{red}}{S_{org}}$ é feita, e o resultado armazenado com 4 casas decimais.

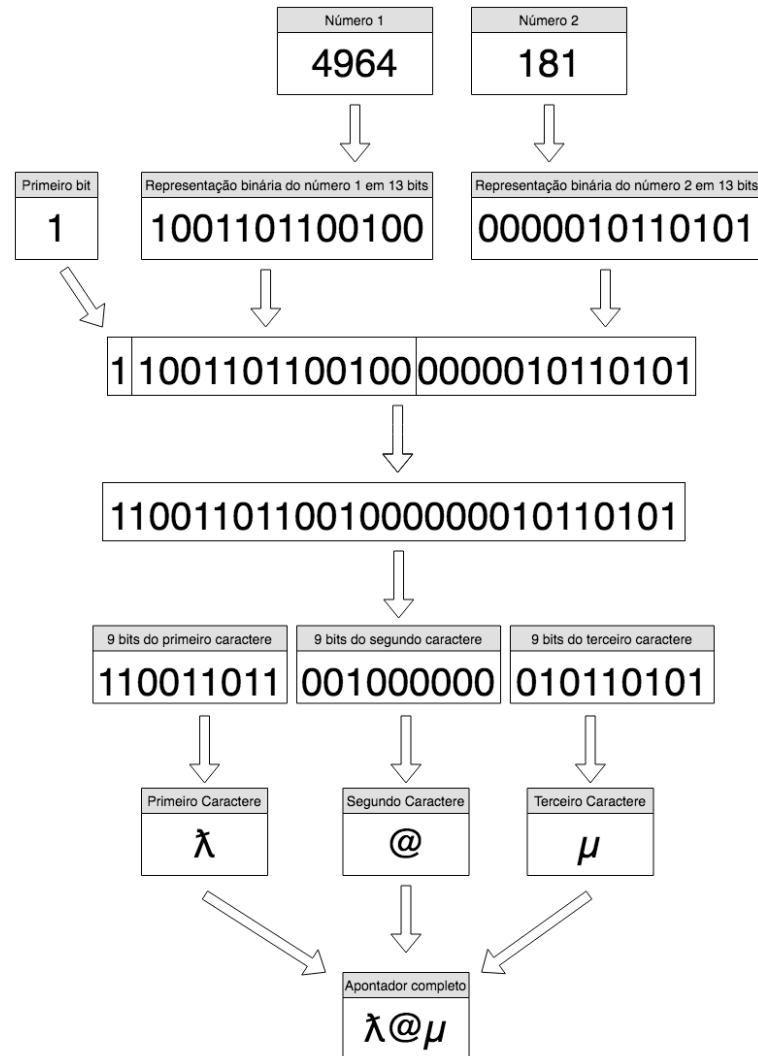


Figura 5 – Exemplo de construção de um apontador

3.2 Popularidade

A análise de popularidade foi feita somente nas músicas que aparecem nos rankings da billboard, e não nos da MaisTocadas, pois os rankings da Billboard são semanais enquanto os da MaisTocadas são anuais, o que significa que a Billboard tem um volume de dados mais de 50 vezes maior que a MaisTocadas.

Os rankings coletados da Billboard são rankings de popularidade. Mas como definir qual a música mais popular entre a primeira colocada do ranking em uma semana de 1959 e a primeira colocada do ranking em uma semana de 2019? Ou como fazer um ranking de popularidade anual, com base nos rankings de popularidade semanais?

Foi decidido que cada vez que uma música aparece em um ranking da Billboard ela ganha "pontos de popularidade". A quantidade de pontos é definida pela posição da música no ranking. A primeira colocada ganha 100 pontos, a segunda colocada 99, a terceira 98... e a última colocada ganha 1 ponto. De forma geral:

$$P_{pop} = 101 - R_{ank}$$

Onde P_{pop} são os pontos de popularidade ganhos pela música, e R_{ank} é a posição do ranking no qual a música apareceu. Assim a popularidade total de uma música é a soma do total de pontos que ela recebeu de todos os rankings. Se uma música apareceu duas vezes em primeiro lugar e uma vez em terceiro lugar ela terá 298 pontos.

3.3 Tamanho Efetivo

Letras de músicas variam em tamanho, algumas tem somente alguns poucos versos enquanto outras passam das 2000 palavras. Mas o tamanho da letra não é uma boa métrica para a complexidade da mesma, uma letra de música muito grande mas também muito repetitiva terá poucos arranjos únicos, o que fará que ela provavelmente seja mais simples. "Around the World" do "Daft Punk" por exemplo tem 144 versos, todos iguais (repetitividade calculada de 98.75%), enquanto "Bohemian Rhapsody" do "Queen" tem 52 versos, poucos repetidos (repetitividade calculada 35%). "Around the world" tem mais versos, mas é claramente menos complexa (como dito no capítulo 1, neste trabalho complexidade se refere ao quão difícil é memorizar a letra de uma música).

Mas usar a repetitividade da letra como análise de complexidade também não é ideal. "Bohemian Rhapsody" como já mencionado tem uma repetitividade de 35% enquanto "Quando o sol bater na janela do teu quarto" da "Legião Urbana" tem uma repetitividade de 33.5%, o que poderia indicar que a música da "Legião Urbana" é a mais complexa das duas, algo que, tendo somente 26 versos e todos bem menores que os de "Bohemian Rhapsody", claramente não é verdade.

Temos então o tamanho efetivo da música, ou seu tamanho após comprimida, também definida por

$$T_{eff} = T_{tot} * (1 - R_{ep})$$

Onde T_{eff} é o tamanho efetivo da letra de música, T_{tot} é o tamanho total da mesma e R_{ep} é a repetitividade da mesma.

Usando esta métrica temos um tamanho efetivo de 1200 para "Bohemian Rhapsody", 31 para "Around the World" e 352 para "Quando o sol bater na janela do teu quarto", o que já aparenta ser uma métrica bem mais útil e próxima da realidade no que diz respeito a medir a complexidade.

3.4 Palavras difíceis

Outra métrica que decidiu-se analisar foi "palavras difíceis", ou mais especificamente quantas palavras difíceis únicas estão presentes em cada música. Como é provável que seja mais difícil memorizar uma palavra difícil que uma palavra comum, faz sentido que palavras difíceis aumentem a complexidade de uma letra de música. A dificuldade de se definir essa métrica é que ela requer uma definição formal de o que é uma palavra difícil, e a mesma deve ser válida em português e inglês.

Como a dificuldade de uma palavra é algo bem subjetivo foi decidido que serão consideradas características comuns a várias palavras normalmente consideradas difíceis. As características consideradas são:

- Muitos caracteres;
- Muitas sílabas;
- Mais consoantes que vogais (vogais neste caso incluem o Y);

Mesmo com essas considerações ainda é necessário decidir o que são "muitos caracteres" e "muitas sílabas". Como não há uma resposta correta para isso foi decidido que muitas sílabas são mais que 3 sílabas e que muitas letras são mais que 12 letras.

Assim fica decidido que uma palavra é considerada difícil se ela tem 4 ou mais sílabas e mais consoantes que vogais e/ou se ela tem mais que 12 caracteres.

Contar o número de caracteres e consoantes em uma palavra é um processo bem simples, mas o mesmo não pode ser dito para sílabas. Sílabas tem quantidades quase arbitrárias de caracteres, vogais e consoantes, e são definidas de forma diferente em português e inglês. Porém foi observado que há uma forte relação entre conjuntos de vogais sequenciais e sílabas tanto em inglês quanto em português. De forma geral, grupos de vogais (A E I O U Y) não separadas por consoantes fazem parte de uma mesma sílaba, então a quantidade de grupos de vogais encontrados em uma palavra tende a ser igual ao número de vogais na mesma palavra. Isso não é verdade para todas as palavras, palavras com hiatos por exemplo são exemplos de exceções a esta regra. A imagem 6 mostra palavras em português e inglês que seguem e que não seguem essa regra. Como palavras que seguem essa regra são bem mais comuns foi decidido que a mesma seria aplicada a todas as palavras.

A métrica "palavras difíceis" é então a quantidade de palavras únicas (palavras são contabilizadas uma vez, independente de quantas vezes elas aparece na letra da música) consideradas difíceis pelo método descrito encontradas em cada letra de música.

Funciona	Inglês	
	Português	
Não Funciona	Inglês	
	Português	

Figura 6 – Exemplos de palavras com a contagem de sílabas reais e por agrupamentos de vogais

3.5 Palavras pouco usadas

Foi decidido ainda contabilizar quantas palavras pouco usadas únicas cada letra de música tem, pois assim como palavras difíceis, estas também devem ser mais dificilmente memorizadas que palavras comuns.

Para que isso seja feito foi inicialmente contabilizado quantas vezes cada palavra aparece em cada letra de música. Tendo uma tabela contendo esta informação é simples descobrir quantas vezes cada palavra aparece no total, em todas as músicas, e em quantas músicas cada palavra aparece.

A definição de "pouco usada" é completamente arbitrária. Foi decidido que pouco usada é a palavra que apareceu em menos de 100 das músicas do banco de dados (músicas com letra coletada da billboard no banco de dados são mais de 20000). Assim a quantidade de vezes que uma uma palavra pouco usada aparece em cada música passa a ser a métrica "palavras pouco usadas".

4 Interpretação dos dados

Com os dados coletados e devidamente processados tem-se um banco de dados com mais de 20000 músicas com várias métricas. Neste capítulo tentara-se encontrar alguma relação entre a popularidade de uma música e as métricas repetitividade, tamanho efetivo, palavras difíceis e palavras pouco usadas, e entre o ano de lançamento da música e estas mesmas métricas.

Todos os gráficos deste capítulo são gerados com respostas de queries inseridas no servidor MySQL.

4.1 Qualidade da Coleta de dados

Os dados coletados vieram de 5 sites diferentes. Não é possível verificar quais dos dados coletados são ou não válidos, mas pode-se comparar quantos dados foram coletados com quantos deveriam ser coletados.

Sabemos que o ranking da billboard é composto de 100 músicas por semana, e o da MaisTocadas é composto de 100 músicas por ano. Algumas músicas ou artistas que apareceram nos rankings poderiam ter um nome contendo caracteres não aceitos pelo servidor MySQL e não previstos na função de coleta (e portanto não tratados), o que acarreta na sua não aquisição. Portanto a quantidade real de dados coletada dos rankings é menor do que a quantidade de dados disponível nas fontes para o período desejado.

A quantidade percentual de dados de rankings coletados por ano pode ser vista no gráfico da imagem 7. Podemos ver que em todos os anos em ambos os ranking foi coletado 98% ou mais dos dados.

No gráfico da imagem 8 temos ainda um gráfico com a porcentagem das letras de música de cada ano que foram encontradas. Podemos ver que temos mais de 70% das letras de todos os anos para as músicas da billboard, e mais de 40% das letras de todas as músicas da MaisTocadas.

No total foram coletadas 99.97% dos rankings da Billboard, 99.96% dos rankings da MaisTocadas, 86.98% das letras de música da Billboard e 61.15% das letras de música da MaisTocadas.

Percebe-se que a porcentagem de letras coletadas das músicas da MaisTocadas é significativamente menor que a da billboard. Portanto é possível que as conclusões feitas com base nas letras da MaisTocadas não sejam tão confiáveis. Porém, como será visto, os resultados obtidos com os dados da Billboard é bem similar com os obtidos dos dados e



Figura 7 – Porcentagem de rankings da billboard e MaisTocadas coletados por ano



Figura 8 – Porcentagem de letras das músicas da billboard e MaisTocadas coletadas por ano

da MaisTocadas.

4.2 Análise de Repetitividade

Foram trassados gráficos de repetitividade média das músicas da Billboard e MaisTocadas por ano (imagem 9), de repetitividade média total e das 10 músicas mais e menos populares por ano da Billboard (imagem 10), e de popularidade média e quantidade de músicas por repetitividade para as músicas da Billboard (imagem 11).

Pelo gráfico da imagem 9 podemos ver que a repetitividade média das músicas tanto da Billboard quanto da MaisTocadas tende a aumentar com o tempo.

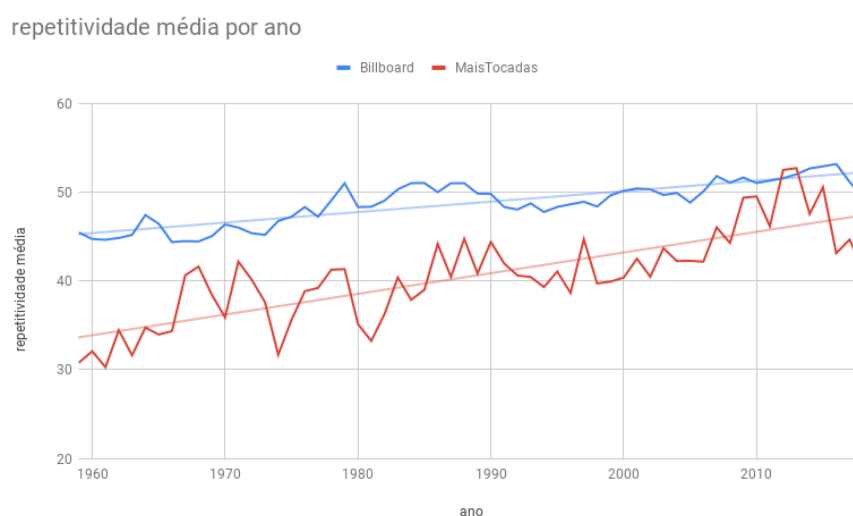


Figura 9 – Repetitividade média das músicas da Billboard e MaisTocadas por ano

Com o gráfico da imagem 10 podemos ver que a repetitividade média das músicas mais populares de cada ano costuma ser mais alta que a média total, e que a repetitividade média das músicas menos populares de cada ano costuma ser menor que a média total.

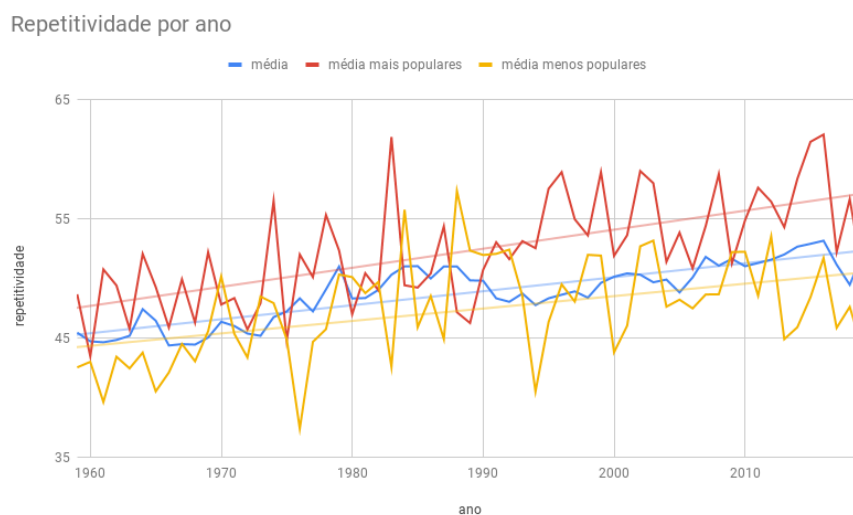


Figura 10 – Repetitividade média das músicas da Billboard por ano

Com o gráfico da imagem 11 podemos ver que o aumento da repetitividade acarreta num aumento da popularidade média. Isso não é verdade para as extremidades do gráfico, mas se for considerado somente a parte do gráfico que tem 200 músicas ou mais por nível de repetitividade, mostrada com destaque na imagem 12, essa relação se torna clara.

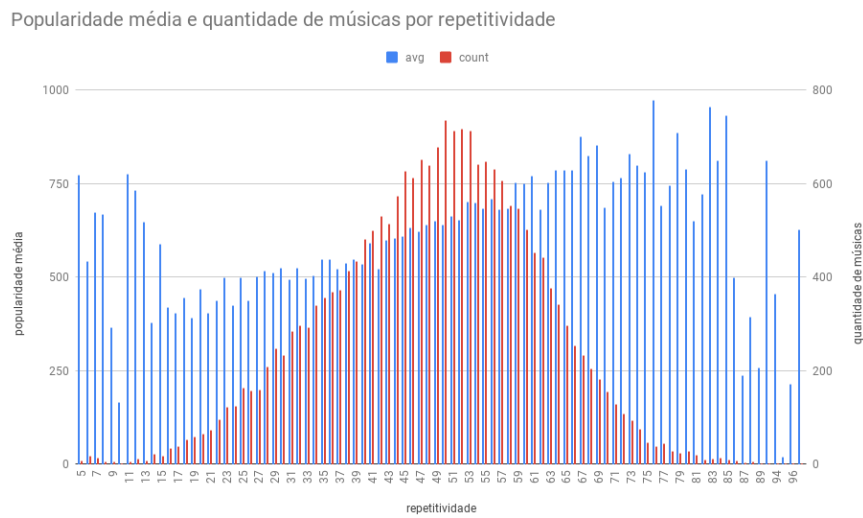


Figura 11 – Popularidade média e quantidade de músicas por repetitividade



Figura 12 – Popularidade média por repetitividade para níveis de repetitividade com mais de 200 músicas

4.3 Análise de tamanho Efetivo

Foram traçados gráficos correlacionando o tamanho efetivo das músicas e seu ano de lançamento (imagem 13) e o tamanho efetivo das músicas e sua popularidade (imagem 14).

Na imagem 13 é fácil perceber o aumento anual médio no tamanho efetivo das letras das músicas. Percebe-se ainda que não há uma diferença considerável o bastante entre a média total e a médias das músicas mais e menos populares.

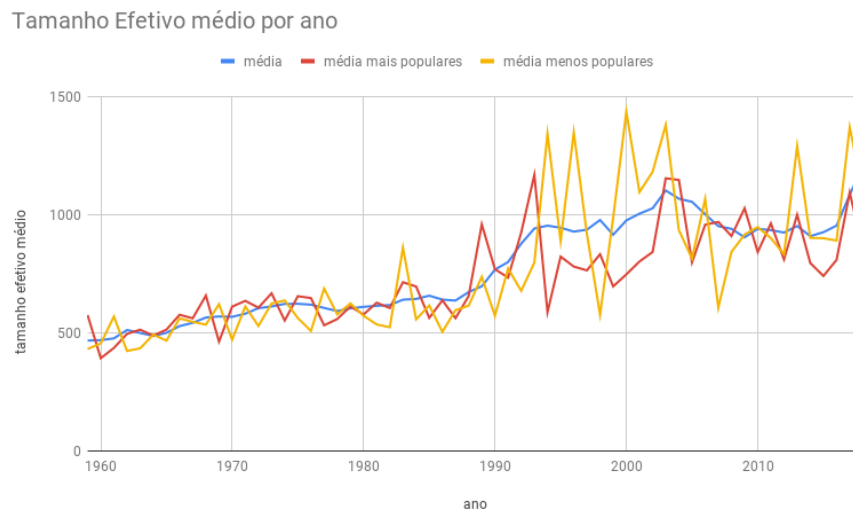


Figura 13 – Tamanho efetivo médio por ano

Na imagem 14 é difícil ver alguma relação entre o tamanho efetivo da letra de uma música e sua popularidade. Mas podemos perceber que quase todas as músicas se concentram em uma faixa de tamanhos efetivos bem restrita, aproximadamente de tamanho efetivo 250 a 900.

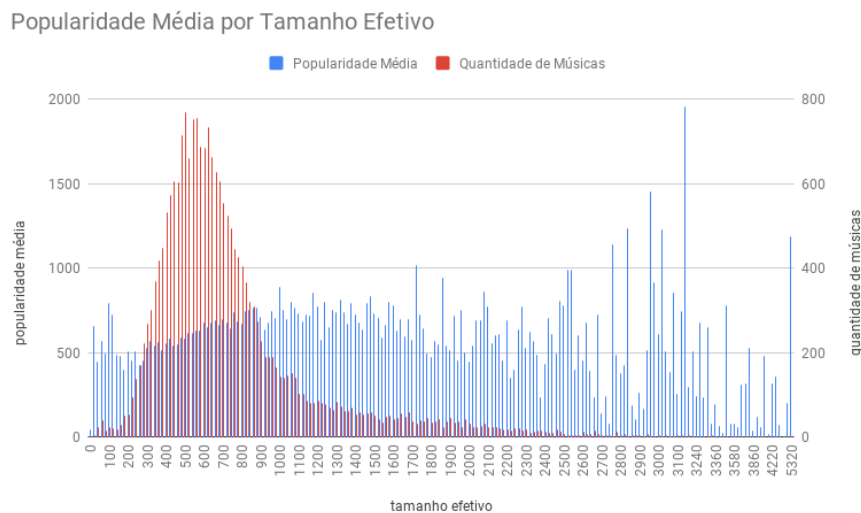


Figura 14 – Popularidade e quantidade de músicas por tamanho efetivo.

Um gráfico considerando somente os tamanhos efetivos com mais de 150 músicas, que pode ser visto na imagem 15, torna mais claro que há sim uma relação entre o tamanho efetivo de uma música e sua popularidade. Nesta faixa de tamanhos efetivos, quanto maior o tamanho efetivo maior a popularidade. Acreditamos que essa tendência não seja constante, e que a relação comece a diminuir para tamanhos efetivos muito altos, mas não há dados o bastante nesta faixa mais alta para se comprovar isso.

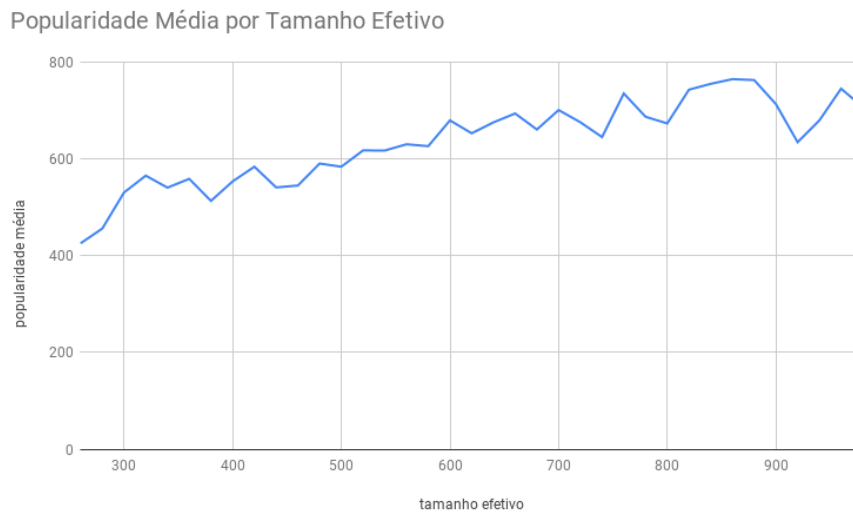


Figura 15 – Popularidade por tamanho efetivo

4.4 Análise de Palavras difíceis

Foram traçados gráficos correlacionando a quantidade de palavras difíceis em uma música e seu ano de lançamento (imagem 16) e a quantidade de palavras difíceis em uma música e sua popularidade (imagem 17).

Na imagem 16 podemos perceber que há um aumento anual médio na quantidade de palavras difíceis por música. Embora este aumento não seja numericamente significativo, crescendo somente duas palavras por música no período de 1959 a 2019, este aumento de 2 palavras representa um aumento de 200%. A quantidade média de palavras difíceis por música aparenta então estar crescendo ano a ano.

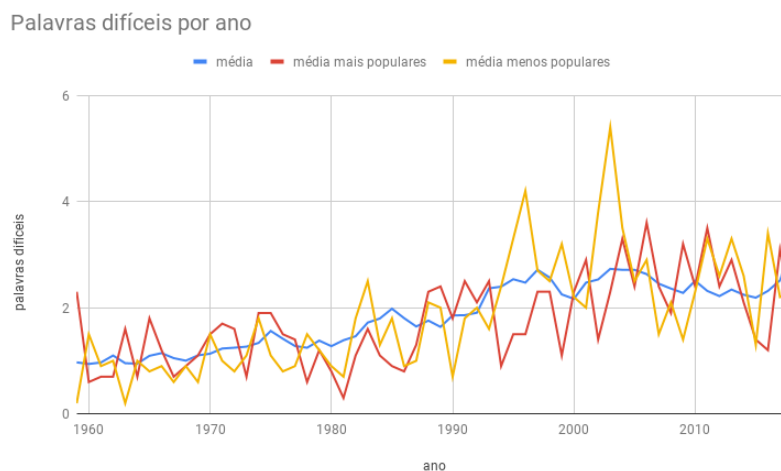


Figura 16 – Média de palavras difíceis por ano

Na imagem 16 não percebe-se alguma ligação entre a popularidade de uma música

e a quantidade de palavras difíceis, as linhas que representam as médias das músicas mais e menos populares não ficam constantemente acima ou abaixo da linha que representa a média total.

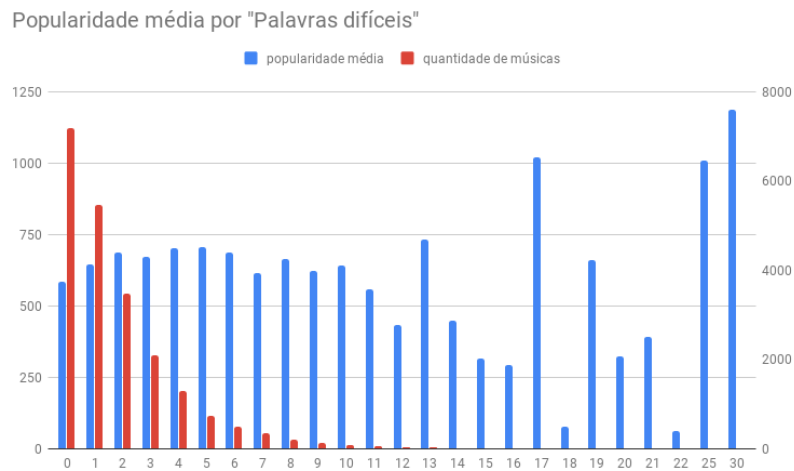


Figura 17 – Popularidade por quantidade de palavras difíceis

Na imagem 17 ainda é difícil perceber alguma relação entre popularidade e palavras difíceis. Mas nesta imagem pode-se perceber que a grande maioria das músicas tem poucas palavras difíceis, 95% do total de músicas tem 5 ou menos palavras difíceis. Um gráfico considerando somente as músicas com 5 ou menos palavras difíceis pode ser visto na imagem 18. Nela já parece existir uma relação entre a quantidade de palavras difíceis e a popularidade de uma música, pois a popularidade média cresce com o aumento da quantidade de palavras difíceis. Mas esta como este é um gráfico feito com somente pontos é possível que esta relação possa ser mera coincidência.

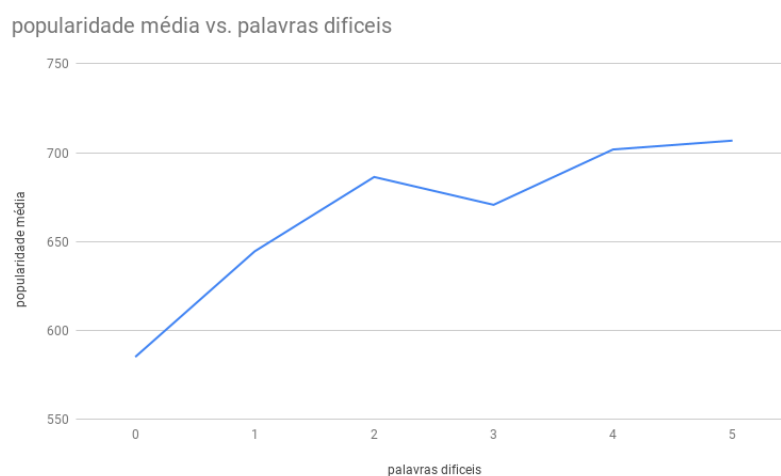


Figura 18 – Popularidade por quantidade de palavras difíceis

4.5 Análise de Palavras pouco usadas

Foram traçados gráficos correlacionando a quantidade de palavras pouco usadas em uma música e seu ano de lançamento (imagem 19) e a quantidade de palavras pouco usadas em uma música e sua popularidade (imagem 20).

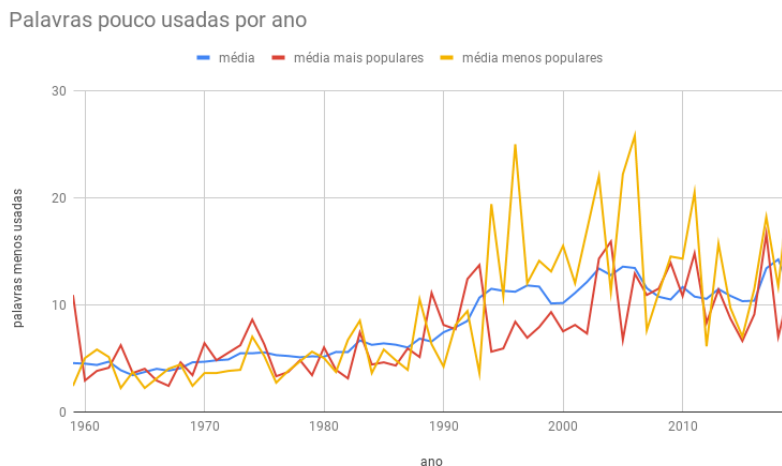


Figura 19 – Média de palavras pouco usadas por ano

Na imagem 19 podemos perceber que há um aumento anual na quantidade de palavras pouco usadas por música, o que é esperado até certo ponto, pois novas palavras surgem todos os anos e a probabilidade de uma palavra ser considerada pouco usada só diminui com o tempo. Esse aumento pode não ser só por isso, mas não pode-se afirmar nada a respeito. Nesta imagem não há nenhuma relação óbvia entre popularidade e palavras pouco usadas, as linhas que representam a média das músicas mais e menos populares não ficam constantemente acima ou abaixo da que representa a média total.

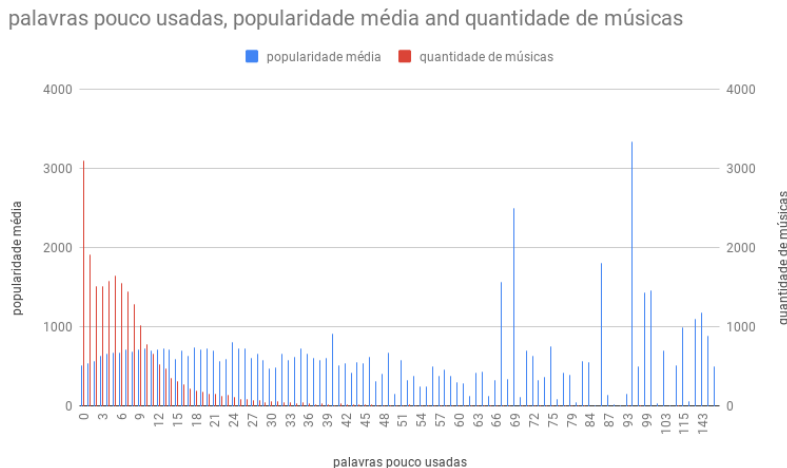


Figura 20 – Popularidade média por palavras pouco usadas

O gráfico da imagem 20, que relaciona popularidade e palavras pouco usadas, também não mostra nenhuma relação óbvia entre as duas coisas.

Porém ele também mostra que a grande maioria das músicas tem poucas palavras pouco usadas, o que é bem lógico. 90% das músicas tem 16 ou menos palavras pouco usadas. Um gráfico que considera somente estes 90% pode se visto na imagem 21.

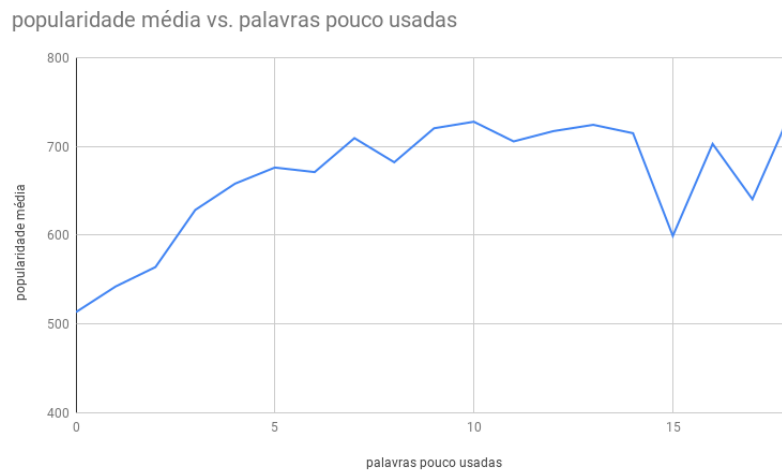


Figura 21 – Popularidade média por palavras pouco usadas

Nela já pode-se perceber uma tendencia no aumento da popularidade média a medida que a quantidade de palavras pouco usadas aumenta.

4.6 Complexidade

As métricas "palavras difíceis", "palavras pouco usadas" e "tamanho efetivo", previamente calculadas, estão relacionadas a complexidade da letra de música analisada. Uma música com mais palavras difíceis ou pouco usadas que a média é, provavelmente, mais complexa que a média.

Foi decidido então criar uma métrica para complexidade, algo que agrupe as 3 métricas já calculadas. O ideal é que essa métrica não favoreça nenhuma das 3 originais, pois não se sabe se alguma delas é ou não mais importante para a complexidade da letra. Como as 3 tem valores bem distintos umas das outras, com tamanho "efetivo variando" de variando 0 a 5000 e "palavras difíceis" variando de 0 a 30, estas métricas devem ser normalizadas, o que pode ser feito dividindo-as pelo seu valor máximo possível. Isso fará que as 3 fiquem limitadas entre 0 e 1.

Assim a complexidade de cada música passa a ser a média quadrática destas 3 métricas já normalizadas:

$$C_{plx} = \sqrt{\frac{T_{ef}^2 + P_{pu}^2 + P_{dif}^2}{3}}$$

Onde C_{plx} é a complexidade, T_{ef} é o "tamanho efetivo" já normalizado, P_{pu} é o "palavras pouco usadas" já normalizado e P_{dif} é o "palavras difíceis" já normalizado.

Com essa nova métrica calculada gráficos de complexidade média por ano e popularidade média por complexidade podem ser traçados.

Na imagem 22 pode-se ver um gráfico de complexidade média por ano. Nele é claro que a complexidade média está crescendo com o tempo. Como os gráficos de "palavras difíceis", "palavras pouco usadas" e "tamanho efetivo" (imagens 16, 19 e 13 respectivamente) também mostram um aumento de suas respectivas métricas com o tempo isso já era esperado. Porém neste gráfico não há nenhuma relação perceptível entre a popularidade de uma música e sua complexidade, as linhas que representam as médias das músicas mais e menos repetitivas não ficam constantemente acima ou abaixo da que representa a média total.

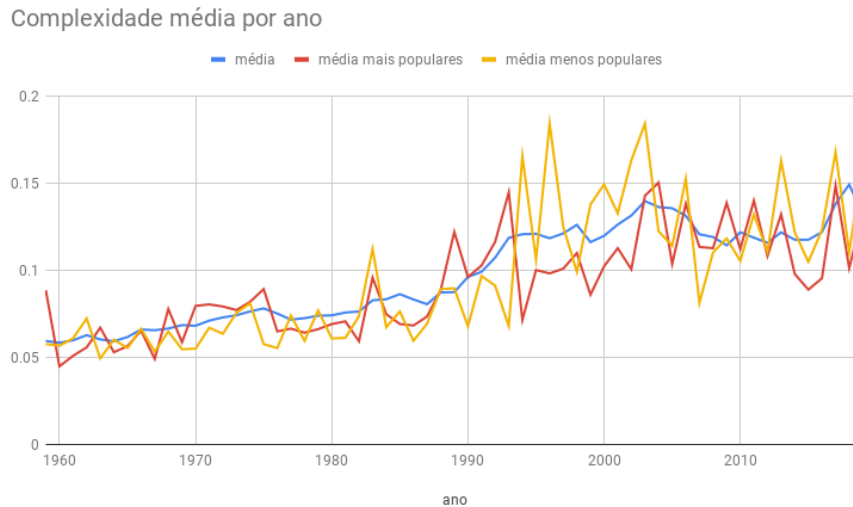


Figura 22 – Complexidade média por ano

Na imagem 23 pode-se ver um gráfico de popularidade média por complexidade. Nele a relação entre complexidade e popularidade ainda não é muito clara.

Mas se ficarmos somente na parte do gráfico com mais de 200 músicas por complexidade, visto na imagem 24, uma relação entre a popularidade de uma música e sua complexidade se torna clara. Como uma relação entre popularidade e "palavras difíceis", "palavras pouco usadas" e "tamanho efetivo" já havia sido observada, e complexidade é composta por estas 3 outras métricas, então este resultado era esperado.

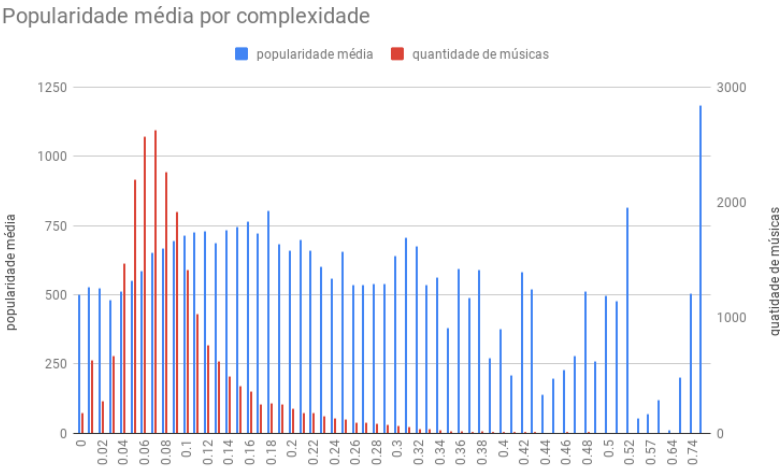


Figura 23 – Complexidade média por ano



Figura 24 – Complexidade média por ano

5 Conclusões

Como dito no início deste artigo, o objetivo do mesmo é investigar frases como "não fazem mais músicas como antigamente", "músicas estão ficando cada vez mais repetitivas" e "músicas pop são muito repetitivas".

Através dos gráficos mostrados no capítulo 4 podemos perceber que as músicas estão sim ficando mais repetitivas, a média de 1960 foi 44,7, de 1970 foi 46,4, de 1980 foi 48,3, de 1990 foi 49,8, de 2000 foi 50,2, de 2010 foi 51,0 e a de 2019 até o momento é de 52,0. Além disso eles também mostram que músicas mais repetitivas costumam ser mais populares, uma música com repetitividade 30 tem uma popularidade média de 524 enquanto que uma música com uma repetitividade de 70 tem uma popularidade média de 850.

Porém os outros gráfico mostram que as músicas não estão somente ficando mais repetitivas. Os gráficos de palavras difíceis, palavras pouco usadas e tamanho efetivo mostram que todas estas métricas estão crescendo com o tempo, a média de palavras difíceis por música passou de 1 em 1959 para 3 em 2019, a de palavras pouco usadas passou de 5 em 1959 para 12 em 2019 e a de tamanho efetivo passou de 500 em 1959 para 1200 em 2019

Além disso os gráficos da métrica complexidade, que agrega as outras 3, mostram a mesma tendência. A complexidade em 1959 é 0,059 enquanto que em 2019 já subiu para 0,132.

Com isso a conclusão é que músicas, além de mais repetitivas, também estão ficando mais complexas, e que o público prefere músicas mais repetitivas e mais complexas.

O aumento anual em complexidade se acelera na década de 1990, o que acreditamos poder ser explicado pela popularização do rap da década de 1990 [10]. É provável que os artistas e produtores tentem fazer músicas que sigam a tendência do que é preferido pelo público, e como o público parece preferir músicas mais repetitivas e complexas estas acabam sendo produzidas com mais frequência.

Isso é análogo à seleção natural [4], ou a um sistema de controle em loop fechado [3]. Artistas e produtores fazem músicas, e então usam o feedback provido pelo público para ajustar suas próximas músicas. Se músicas mais repetitivas e complexas recebem um feedback melhor, mais músicas novas serão complexas e repetitivas. Já foram até criados programas que geram músicas com base no feedback de quem as escuta [7], este programa é uma simulação do que acontece com a produção musical real.

6 Extra

Este capítulo é composto de alguns fatos interessantes, descobertos com os ranking da Billboard.

Usando o cálculo de popularidade explicado no capítulo 3, podemos gerar rankings de músicas e artistas por popularidade. Podemos também gerar rankings com a quantidade de vezes que músicas e artistas aparecem nos rankings, aparecem na primeira posição, entre outros. Seguem os primeiros colocados de alguns destes rankings.

As 10 músicas mais populares estão na tabela 9.

Tabela 9 – Músicas mais populares

Música	Artista	Popularidade
Radioactive	Imagine Dragons	5932
How Do I Live	LeAnn Rimes	5615
I'm Yours	Jason Mraz	5433
Party Rock Anthem	LMFAO	5351
Shape Of You	Ed Sheeran	5197
Counting Stars	OneRepublic	5079
Uptown Funk!	Mark Ronson Featuring Bruno Mars	4963
Rolling In The Deep	Adele	4958
Smooth	Santana Featuring Rob Thomas	4838
Perfect	Ed Sheeran	4718

Curiosamente, a primeira colocada deste ranking nunca ficou em primeiro lugar nos rankings semanais.

As 10 músicas que já apareceram mais vezes em primeiro lugar estão na tabela 10.

Tabela 10 – Músicas mais vezes na primeira posição

Música	Artista	Número de vezes
Despacito	Luis Fonsi and Daddy Yankee	16
One Sweet Day	Mariah Carey and Boyz II Men	16
I Gotta Feeling	The Black Eyed Peas	14
Macarena (Bayside Boys Mix)	Los Del Rio	14
Uptown Funk!	Mark Ronson Feat Bruno Mars	14
We Belong Together	Mariah Carey	14
Something About The Way You Look Tonight	Elton John	14
I Will Always Love You	Whitney Houston	14
I'll Make Love To You	Boyz II Men	14
End Of The Road (From 'Boomerang')	Boyz II Men	13

As 10 músicas que já apareceram mais vezes na Billboard, independente da posição, estão na tabela 11.

Tabela 11 – Músicas mais presentes nos rankings

Música	Artista	Número de vezes
Radioactive	Imagine Dragons	87
Sail	AWOLNATION	79
I'm Yours	Jason Mraz	76
How Do I Live	LeAnn Rimes	69
Counting Stars	OneRepublic	68
Party Rock Anthem	LMFAO	68
Rolling In The Deep	Adele	65
Foolish Games/You Were Meant For Me	Jewel	65
Before He Cheats	Carrie Underwood	64
Ho Hey	The Lumineers	62

Podemos perceber que esta tabela já é bem mais similar que a de popularidade geral, o que faz sentido pois já considera todas as posições.

Já os artista mais populares podem ser vistos na tabela 12.

Tabela 12 – Artistas mais populares

Artista	Popularidade
Madonna	57369
Elton John	55477
Taylor Swift	54640
Mariah Carey	46460
Stevie Wonder	42542
The Beatles	42470
Drake	40004
Michael Jackson	39767
Rihanna	38109
Rod Stewart	37860

Considerando que a diferença entre os três primeiros colocados é bem baixa, e que deles só a terceira colocada ainda aparece no rankings com frequência, acreditamos que a Madonna vá deixar de ser a artista mais popular em breve, perdendo sua posição para Taylor Swift.

Os artistas que já ocuparam a primeira posição mais vezes podem ser vistos na tabela 13.

Tabela 13 – Artistas mais vezes na primeira posição

Artista	Número de vezes
Mariah Carey	60
The Beatles	54
Boyz II Men	34
Madonna	32
Whitney Houston	31
Michael Jackson	30
Drake	29
The Black Eyed Peas	28
Bee Gees	27
Adele	24

E uma tabela com os artistas mais frequentes nos rankings está na tabela 15.

Tabela 14 – Artistas mais presentes nos rankings

Artista	Número de vezes
Elton John	889
Taylor Swift	869
Madonna	857
Tim McGraw	719
Kenny Chesney	709
Drake	697
Stevie Wonder	659
Rod Stewart	657
Keith Urban	638
Michael Jackson	609

Os 10 artistas que já apareceram em mais posições simultâneas no ranking da billboard podem ser vistos na tabela

Tabela 15 – Artistas em mais posições simultâneas

Semana	Artista	Número posições
2018-07-09	Drake	22
2018-08-13	Travis Scott	17
1964-04-06	The Beatles	14
2018-10-08	Lil Wayne	13
2018-05-07	Post Malone	13
2016-12-12	The Weeknd	13
2017-03-20	Ed Sheeran	13
2016-12-26	J. Cole	12
2019-02-18	Ariana Grande	12
2015-11-30	Justin Bieber	12

Bibliografia

- [1] Philip Alperson. *What is music?: an introduction to the philosophy of music*. Penn State Press, 2010.
- [2] Chris J Date e Hugh Darwen. *A Guide to the SQL Standard*. Vol. 3. Addison-Wesley New York, 1987.
- [3] Richard C Dorf e Robert H Bishop. “Modern control systems”. Em: (1998).
- [4] John A Endler e Robert M May. *Natural selection in the wild*. 21. Princeton University Press, 1986.
- [5] Python Software Foundation. *URLlib documentation*. <https://docs.python.org/3/library/urllib.html>. [Online; accessed 17-June-2019]. 2019.
- [6] Anssi Klapuri e Manuel Davy. *Signal processing methods for music transcription*. Springer Science & Business Media, 2007.
- [7] Robert M. MacCallum et al. “Evolution of music by public choice”. Em: *Proceedings of the National Academy of Sciences* 109.30 (jun. de 2012), pp. 12081–12086. DOI: [10.1073/pnas.1203182109](https://doi.org/10.1073/pnas.1203182109). URL: <https://doi.org/10.1073/pnas.1203182109>.
- [8] Oracle. *MySQL.connector documentation*. <https://dev.mysql.com/doc/connector-python/en/>. [Online; accessed 17-June-2019]. 2019.
- [9] Leonard Richardson. “Beautiful soup documentation”. Em: *April* (2007).
- [10] Derek Thompson. “1991: The Most Important Year in Pop-Music History”. Em: *May* (2015).
- [11] Guido Van Rossum e Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- [12] Eloisa Vargiu e Mirko Urru. “Exploiting web scraping in a collaborative filtering-based approach to web advertising”. Em: *Artificial Intelligence Research* 2.1 (nov. de 2012). DOI: [10.5430/air.v2n1p44](https://doi.org/10.5430/air.v2n1p44). URL: <https://doi.org/10.5430/air.v2n1p44>.
- [13] Wikipedia. *Application programming interface* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Application%20programming%20interface&oldid=898631998>. [Online; accessed 17-June-2019]. 2019.
- [14] Wikipedia. *Billboard* — *Wikipedia, The Free Encyclopedia*. [http://en.wikipedia.org/w/index.php?title=Billboard%20\(magazine\)&oldid=902168038](http://en.wikipedia.org/w/index.php?title=Billboard%20(magazine)&oldid=902168038). [Online; accessed 17-June-2019]. 2019.

-
- [15] Wikipedia. *Binary number* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Binary%20number&oldid=902162352>. [Online; accessed 17-June-2019]. 2019.
 - [16] Wikipedia. *Data compression ratio* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Data%20compression%20ratio&oldid=890778783>. [Online; accessed 17-June-2019]. 2019.
 - [17] Wikipedia. *Ranking* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Ranking&oldid=900803492>. [Online; accessed 16-June-2019]. 2019.
 - [18] Wikipedia. *Unicode* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Unicode&oldid=902260944>. [Online; accessed 17-June-2019]. 2019.