**Faculdade de Tecnologia do Estado de São Paulo**

**Fatec carapicuíba**

**Gustavo Cardoso Santos da Silva**

**Gustavo de Deus Conceição**

**José Antônio Ramos Barreto Gomes**

**Kevin Ryan Gomes dos Santos**

**Faixas de Música: Analisar a evolução e tendencia dos gêneros musicais**

Trabalho apresentado para a disciplina de Inteligência Artificial e Aprendizado de Máquina, A.M.S., 2° Ano, 2024.

Professora: Magali Andreia Rossi

**Carapicuíba**

**2024**

sumário

[LISTA DE FIGURAS 3](#_Toc178171918)

[LISTA DE TABELAS 4](#_Toc178171919)

[1. INTRODUÇÃO 1](#_Toc178171920)

[1.1. FUNDAMENTAÇÃO CONTEXTUAL 1](#_Toc178171921)

[1.2. CONSUMO E TENDÊNCIAS DA INDUSTRIA 1](#_Toc178171922)

[1.3. OBJETIVOS E HIPÓTESE 2](#_Toc178171923)

[2. MÉTODOLOGIA 2](#_Toc178171924)

[2.1. KDD 3](#_Toc178171925)

[2.1.1. Seleção 3](#_Toc178171926)

[2.1.2. Pré-processamento 5](#_Toc178171927)

[2.1.3. Transformação 5](#_Toc178171928)

[2.1.4. Regressão linear 6](#_Toc178171929)

[2.2. AVALIAÇÃO DO MODELO DE REGRESSÃO 7](#_Toc178171930)

[2.3. INTERPRETAÇÃO DOS COEFICIENTES DA REGRESSÃO 8](#_Toc178171931)

[3. RESULTADOS E DISCUSSÃO 9](#_Toc178171932)

[4. CONSIDERAÇÕES FINAIS 17](#_Toc178171933)

[5. REFERÊNCIAS 18](#_Toc178171934)

[apêndice – Código de requisição 19](#_Toc178171935)

# LISTA DE FIGURAS

[Figura 1: Ano de Lançamento vs Popularidade: Real vs Estimada (Fonte: Os Autores, 2024) 6](#_Toc178103305)

[Figura 2: Popularidade Média por Categoria de Gênero Musical (Fonte: Os Autores, 2024) 9](#_Toc178103306)

[Figura 3: Popularidade Média por Categoria de Gênero Musical (Fonte: Os Autores, 2024) 10](#_Toc178103307)

[Figura 4: Popularidade dos Gêneros Musicais ao Longo dos Anos (Fonte: Os Autores, 2024) 11](#_Toc178103308)

[Figura 5: Distribuição da Popularidade por Década de Lançamento (Fonte: Os Autores, 2024) 12](#_Toc178103309)

[Figura 6: Comparação das Métricas da Categoria Rock (Os Autores, 2024) 13](#_Toc178103310)

[Figura 7: Comparação das Métricas da Categoria Rock (Fonte: Os Autores, 2024) 14](#_Toc178103311)

# LISTA DE TABELAS

[Tabela 1: Descrição dos recurso de áudio. (Fonte: Os Autores, 2024) 3](#_Toc178102505)

# INTRODUÇÃO

Desde o surgimento, o rock se destacou por não ser apenas uma derivação de outros gêneros musicais, mas por representar uma ruptura inovadora, marcada pela rebeldia e insubordinação às normas estabelecidas. Esse movimento contracultural teve origem na década de 1940, inspirado por gêneros como o blues, jazz e a música clássica, mas redefiniu a forma como a música era consumida e interpretada. Com sua popularização, o rock não só transformou o cenário musical, mas também impactou profundamente a cultura e as gerações, culminando em marcos históricos como o festival de Woodstock em 1969. A partir daí, bandas icônicas como The Beatles, Queen, Guns N' Roses e The Rolling Stones emergiram, não apenas como artistas musicais, mas como símbolos culturais que moldaram a identidade de diferentes épocas e influenciaram a sociedade globalmente.

## FUNDAMENTAÇÃO CONTEXTUAL

Nós realizamos uma depuração e análise sobre todos os dados coletados entre 1950 e 2020, destacando qual foi o gênero musical predominante entre a população mundial e como os gostos e preferências mudaram ao longo dessas sete décadas.

Além disso, esta análise busca entender a relação entre a evolução do rock e as mudanças nas preferências da audiência, bem como a forma como fatores socioculturais, como a era digital e o acesso à música por meio de plataformas de streaming, têm impactado o consumo musical. Ao longo deste estudo, pretendemos destacar essas transformações e proporcionar uma visão abrangente sobre a relevância do rock no cenário musical atual.

## CONSUMO E TENDÊNCIAS DA INDUSTRIA

A maneira como consumimos música passou por uma revolução significativa com a popularização da compressão acústica e a distribuição digital. O formato MP3, graças às redes P2P (peer-to-peer) como o Napster, facilitou a troca de músicas sem a necessidade de mídia física, transformando a forma de distribuição musical (ALLEN-ROBERTSON, 2013; WIKSTRÖM, 2013; SCHWARZ, 2014). Esse cenário abriu caminho para plataformas lícitas, como o iTunes, e, posteriormente, o Spotify, que consolidou seu domínio no mercado de streaming.

Com a ascensão da internet, novas formas de consumo de música surgiram, tanto legais quanto ilegais, e mudaram drasticamente a indústria fonográfica. O Spotify, que é hoje a maior plataforma de streaming, foi um dos maiores facilitadores dessa transformação. Segundo a Recording Industry Association of America (RIAA), em 2022, os serviços de streaming representaram 84% da receita da música gravada, enquanto as mídias físicas caíram para 11% e os downloads digitais, que já dominaram o mercado, representaram apenas 3% e continuam em declínio (HERRMAN, 2023).

Para os artistas, o Spotify atua não apenas como uma plataforma de distribuição, mas também como uma espécie de rede social idiossincrática. Nesse ambiente, os músicos precisam entender e adaptar-se não só aos gostos dos ouvintes, mas também ao algoritmo e às preferências da própria plataforma para maximizar sua visibilidade.

## OBJETIVOS E HIPÓTESE

A principal motivação para essa análise foi observar como os gêneros musicais têm se transformado, especialmente em resposta às novas tendências culturais e modas. O rock, gênero em foco nesta pesquisa, já passou por diversas mudanças ao longo das décadas e, atualmente, parece estar mais consolidado, influenciado por diferentes espectros de consumo e novas influências musicais. Ao contrário de décadas passadas, quando sua popularidade era mais prevalente e frequente, o rock agora se apresenta como um gênero que se adapta e se reinventa, misturando-se com outros estilos e refletindo as preferências do público contemporâneo.

# MÉTODOLOGIA

Para realizar nossa análise, começamos com a coleta de dados de gêneros musicais de uma base abrangente que incluía registros de 1950 a 2020 do dataset do spotify. A primeira etapa foi realizar requisições por gênero musical, utilizando o DataFrame df\_all. Em seguida, verificamos os valores nulos e tipos de dados, e transformamos a tabela, removendo registros com dados ausentes ou considerados irrelevantes para a análise.

Posteriormente, foi realizada uma análise exploratória dos dados, onde identificamos as categorias musicais predominantes ao longo das décadas. Nosso foco principal foi no gênero rock, devido à sua influência histórica e atual no cenário musical. Para isso, calculamos sua popularidade média, categorizamos por década e analisamos a evolução do gênero.

As ferramentas utilizadas foram Python, como linguagem de programação, pandas, para manipulação e estruturação de dados, matplotlib e seaborn, para visualização dos resultados, numPy, para operações numéricas e Scikit-learn, para aplicar modelos preditivos e regressão linear.

## KDD

### *Seleção*

No início do desenvolvimento, uma das primeiras etapas envolve a identificação e leitura do arquivo CSV selecionado para o projeto, utilizando a biblioteca Pandas para carregar os dados. Esse processo nos permite analisar os dados disponíveis em nossa base. Entre os atributos mais relevantes para nossa pesquisa, destacam-se as informações relacionadas aos recursos de áudio, popularidade, bem como os artistas e bandas responsáveis pelas composições.

A popularidade, um dos aspectos centrais do nosso estudo, é representada por uma coluna que desempenha um papel fundamental na análise de tendências musicais. Esse valor varia entre 0 e 100, sendo 100 o indicador de maior popularidade. A métrica de popularidade é calculada por um algoritmo que leva em consideração, principalmente, o número total de reproduções da faixa e a recência dessas reproduções, refletindo a relevância atual das músicas.

Os dados completos de todas as faixas, sem qualquer filtro de categoria, estão armazenados na variável df\_all. Essa variável contém todas as músicas disponíveis para análise inicial, permitindo que possamos aplicar filtros específicos posteriormente, conforme necessário.

A seguir, na Tabela 1, são descritos os recursos de áudio, que incluem uma variedade de informações técnicas e qualitativas sobre as faixas. Esses recursos desempenham um papel essencial na construção de insights mais aprofundados sobre as características musicais e seu impacto na popularidade das músicas.

Tabela 1: Descrição dos recurso de áudio. (Fonte: Os Autores, 2024)

|  |  |
| --- | --- |
| Recursos de Áudio | Descrição |
| Instrumentalness | Esse valor representa a quantidade de vocais na música. Quanto mais perto de 1.0, mais instrumental é a música. |
| Acousticness | Este valor descreve como uma música acústica é. Uma pontuação de 1,0 significa que a música é mais provável de ser acústica. |
| Liveness | Este valor descreve a probabilidade de que a música foi gravada com um público ao vivo. De acordo com a documentação oficial, “um valor acima de 0,8 fornece forte probabilidade de que a faixa esteja ao vivo”. |
| Speechiness | “A falabilidade detecta a presença de palavras faladas em uma faixa”. Se a discursidade de uma música estiver acima de 0,66, provavelmente é feita de palavras faladas, uma pontuação entre 0,33 e 0,66 é uma música que pode conter música e palavras, e uma pontuação abaixo de 0,33 significa que a música não tem qualquer fala. |
| Danceability | A dança descreve o quão adequado é uma faixa para dançar com base em uma combinação de elementos musicais, incluindo tempo, estabilidade do ritmo, força de batida e regularidade geral. Um valor de 0,0 é menos dançável e 1.0 é mais dançante. |
| Valence | Uma medida de 0,0 a 1,0 descrevendo o positivo musical transmitido por uma faixa. Faixas com alta valência soam mais positivas (por exemplo, feliz, alegre, eufórica), enquanto as faixas com baixa valência soam mais negativas (por exemplo, tristes, deprimidas, zangadas). |

### *2.1.2. Pré-processamento*

Com o intuito de preservar a qualidade dos dados, realizou-se uma análise de dados nulos na tabela, constatando que não há indícios de dados nulos,

### *2.1.3. Transformação*

Durante a fase de transformação dos dados, aplicamos um filtro fundamental para segmentar nossa análise, concentrando-nos no gênero musical "rock". A criação desse filtro é essencial para organizar os dados e facilitar o desenvolvimento da pesquisa, já que ela será a base para a busca de novos dados através da API do Spotify, como o ano de produção das faixas.

A seguir, o código utilizado para essa categorização

def categorize\_genre(genre):

    categories = {

        'Pop': ['pop', 'k-pop', 'j-pop', 'dance pop', 'power-pop', 'indie-pop', 'synth-pop', 'mandopop', 'cantopop'],

        'Rock': ['rock', 'alternative', 'punk', 'metal', 'punk-rock', 'psych-rock', 'j-rock', 'rockabilly', 'rock-n-roll', 'alt-rock', 'hard-rock', 'grunge', 'emo', 'indie'],

        'Eletrônica e Hip Hop': ['hip hop', 'hip-hop', 'rap', 'trap','electronic', 'edm', 'house', 'techno', 'progressive-house', 'minimal-techno', 'trance', 'detroit-techno', 'deep-house', 'chicago-house', 'breakbeat', 'drum-and-bass', 'dubstep', 'electro', 'idm'],

        'Clássica': ['classical', 'opera'],

        'Dance': ['dance', 'disco', 'party', 'club'],

        'MPB': ['mpb', 'sertanejo', 'pagode'],

        'Música para Relaxar': ['chill', 'sleep', 'study', 'new-age'],

        'Funk': ['funk', 'groove'],

        'Outros': []

    }

    genre\_lower = genre.lower()

    for category, genres in categories.items():

        if any(g in genre\_lower for g in genres):

            return category

    return 'Outros'

df\_all['category'] = df\_all['track\_genre'].apply(categorize\_genre)

Este código cria uma coluna chamada `category`, que classifica as músicas com base no gênero atribuído a cada faixa. A função `categorize\_genre` agrupa os gêneros musicais em categorias pré-definidas, como Pop, Rock, Eletrônica e Hip Hop, clássica, entre outras. O gênero de cada faixa é comparado com essas categorias, e a faixa é atribuída à sua respectiva categoria.

Essa categorização desempenha um papel essencial na estruturação dos dados, permitindo a aplicação de filtros mais específicos. Ao organizar as músicas por categorias, podemos conduzir análises direcionadas para determinados estilos musicais, como o rock, que foi o escolhido para o estudo. Além disso, essa segmentação facilitará a busca por novos dados na API do Spotify, como o ano de lançamento das faixas, aprimorando ainda mais a profundidade da análise.

### *2.1.4. Regressão linear*

Durante a fase de mineração de dados do nosso projeto, aplicamos a técnica de **regressão linear** para extrair insights valiosos sobre a popularidade do gênero musical **rock** ao longo dos anos. Nosso objetivo foi identificar uma relação entre o ano de lançamento das faixas e sua popularidade média, utilizando como base dados históricos de músicas categorizadas como rock.

Figura 1: Ano de Lançamento vs Popularidade: Real vs Estimada (Fonte: Os Autores, 2024)

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

A regressão linear revelou uma **correlação negativa** entre o ano de lançamento e a popularidade média do gênero, indicando uma tendência de queda na popularidade do rock com o passar do tempo no spotify. O gráfico gerado (Figura 1) ilustra essa tendência, onde a linha de tendência em vermelho aponta para um declínio contínuo na popularidade média desde a década de 1960. Esse comportamento reforça uma hipótese de que o rock, ao longo dos anos, foi gradualmente perdendo espaço no cenário musical global de streaming.

## AVALIAÇÃO DO MODELO DE REGRESSÃO

Para validar a qualidade do modelo de regressão linear, calculamos algumas métricas estatísticas que nos permitiram avaliar sua precisão:

* **Erro Médio Absoluto (MAE)**: mede a magnitude média dos erros entre os valores previstos e os valores reais, fornecendo uma noção de quanto, em média, o modelo "erra".
* **Erro Médio Quadrático (MSE)**: enfatiza a penalização de erros maiores ao elevar as diferenças ao quadrado, sendo útil para detectar grandes discrepâncias entre os valores preditos e os observados.
* **Coeficiente de Determinação (R²)**: reflete a proporção da variância total dos dados explicada pelo modelo. Quanto mais próximo de 1, maior a capacidade preditiva do modelo.

Os valores dessas métricas para nosso modelo foram calculados da seguinte forma:

mae = mean\_absolute\_error(y\_popularity, y\_pred\_year)

mse = mean\_squared\_error(y\_popularity, y\_pred\_year)

r2 = r2\_score(y\_popularity, y\_pred\_year)

print(f'Erro Médio Absoluto (MAE): {mae:.2f}')

print(f'Erro Médio Quadrático (MSE): {mse:.2f}')

print(f'Coeficiente de Determinação (R²): {r2:.2f}')

Essas métricas nos permitem avaliar a precisão da regressão e indicam que o modelo captura de forma razoável a tendência de declínio da popularidade do gênero. Embora o modelo apresente uma certa dispersão, ele oferece uma base quantitativa sólida para entendermos a dinâmica de popularidade do rock ao longo dos anos.

## INTERPRETAÇÃO DOS COEFICIENTES DA REGRESSÃO

Para além das métricas de erro, os **coeficientes da regressão** fornecem informações cruciais sobre a relação entre as variáveis estudadas. Dois parâmetros importantes são:

* **Coeficiente de inclinação** (coef\_year): representa o declínio na popularidade média a cada ano. Ou seja, quanto menor o valor, maior a velocidade com que o rock perde popularidade ao longo do tempo.

Esses coeficientes foram extraídos do modelo com o seguinte código:

coef\_year = reg\_model\_year.coef\_[0]

print(f'Coeficiente (declínio por ano): {coef\_year:.2f}')

A análise desses valores confirma uma **tendência decrescente** na popularidade do rock. O coeficiente de inclinação negativo indica que, a cada ano, a popularidade média do rock diminui de forma constante, reforçando a ideia de que o gênero está em declínio. O intercepto nos dá uma ideia da popularidade média do rock em seus anos iniciais, antes que a queda começasse a se acentuar.

# RESULTADOS E DISCUSSÃO

Entre 1950 e 2020, a popularidade dos gêneros musicais passou por profundas mudanças, e o rock, que uma vez foi o centro das atenções, viu sua relevância diminuir ao longo das décadas. O rock surgiu como um movimento revolucionário na década de 1950, associado à quebra de normas sociais e políticas. Durante as décadas de 1960 e 1970, esse gênero desempenhou um papel crucial no movimento de contracultura, sendo uma plataforma para protestos contra a guerra, lutas pelos direitos civis e a juventude rebelde da época (OURMUSICWORLD, 2024). Festivais como Woodstock e Monterey Pop foram ícones dessa era, trazendo artistas como The Beatles, The Rolling Stones e Bob Dylan, que representaram a rebelião juvenil e o espírito de contestação social (SINUSOIDAL MUSIC, 2023).

O declínio da popularidade do rock pode ser compreendido ao observar a fragmentação do movimento de contracultura e o surgimento de novos gêneros musicais que mais adequadamente refletiam as preocupações das gerações seguintes, como o pop e o hip hop. Conforme o mundo se distanciava dos protestos anti-guerra e das revoluções sociais da década de 1960, a juventude passou a se identificar com sons mais urbanos e contemporâneos, levando à ascensão desses novos gêneros (SINUSOIDAL MUSIC, 2023; OURNUSICWORLD, 2024).

Como reafirmado na figura 1 apresenta um gráfico que exibe a popularidade média de diferentes gêneros musicais. A MPB (Música Popular Brasileira) é o gênero mais popular, com uma média de 44,3, seguida de perto pelo Pop, que tem 43,2. Em terceiro lugar, temos o Rock, com uma popularidade média de 36,1. Outros gêneros como Música para Relaxar (36,0) e Funk (35,1) também têm uma posição considerável entre as preferências do público.

Eletrônica e Hip Hop apresentam uma média de popularidade de 32,4, enquanto a categoria outros tem 31,0. Gêneros como dance (28,4) têm uma popularidade menor. Por fim, a Música Clássica, com uma média de apenas 18,8, é o gênero menos popular.

Figura 2: Popularidade Média por Categoria de Gênero Musical (Fonte: Os Autores, 2024)

Gráfico, Gráfico de barras

Descrição gerada automaticamente

O rock como um todo também houve alguns insights interessantes como demostrado na figura 2, onde ela apresenta a popularidade média de subgêneros de **rock** e **metal**, destacando algumas tendências interessantes. O **grunge** lidera, com uma popularidade média de 49,6, seguido de perto pelo **metal** (43,7) e **metalcore** (43,5). Esses gêneros têm mantido um apelo forte ao longo dos anos, provavelmente devido à sua capacidade de explorar temas profundos e de rebeldia, que continuam a ressoar com o público jovem.

Gêneros como **psych-rock** (42,8) e **hard-rock** (40,7) também mostram força, sugerindo que, embora o "rock" em termos gerais tenha visto um declínio, subgêneros mais específicos ainda têm seguidores dedicados. O **J-Rock** (38,3) e o **punk** (36,2) mantêm uma relevância moderada, provavelmente devido às suas origens culturais e à subcultura associada a eles, que continua a atrair fãs em várias partes do mundo.

É interessante observar que, enquanto o rockabilly e gêneros mais antigos como o **punk rock** (34,4) ainda mantêm uma presença significativa, os estilos mais extremos, como **heavy metal** (22,4) e **black metal** (19,0), estão entre os menos populares no gráfico. O **rock** tradicional, com 19,0 de popularidade média, é o que mais sofreu um declínio, o que reforça a ideia de que o gênero como um todo perdeu força, mas seus subgêneros evoluíram e se fragmentaram em nichos com grande apelo para públicos específicos.

Figura 3: Popularidade Média por Categoria de Gênero Musical (Fonte: Os Autores, 2024)

Gráfico, Gráfico de barras

Descrição gerada automaticamente

O gráfico apresentado na **Figura 3** fornece uma visão detalhada da popularidade dos subgêneros de **rock**, ao longo das décadas de 1960 até 2020, confirmando as mudanças discutidas anteriormente sobre a evolução e o declínio de certos gêneros musicais.

Figura 4: Popularidade dos Gêneros Musicais ao Longo dos Anos (Fonte: Os Autores, 2024)

Gráfico

Descrição gerada automaticamente

O **grunge** teve seu auge nos anos 1990, com bandas como *Nirvana* e *Pearl Jam* definindo o som de uma geração, marcado por uma mistura de alienação e autenticidade emocional. No entanto, após a década de 2000, sua popularidade começou a declinar, com o surgimento de novos gêneros mais voltados ao mercado contemporâneo. O **hard-rock**, que dominou entre as décadas de 1970 e 1980 com bandas como *Led Zeppelin* e *AC/DC*, também apresentou declínio após essa era, refletindo a transição dos gostos musicais ao longo das décadas.

O **metal**, que também teve seu apogeu nas mesmas décadas, manteve uma presença forte durante os anos 1980 e 1990, sustentado por bandas como *Metallica* e *Iron Maiden*. Entretanto, à medida que novos gêneros começaram a emergir e se conectar mais com o público moderno, o metal viu sua popularidade gradualmente diminuir.

A figura 3 apresentado oferece uma visão detalhada da evolução da popularidade das músicas de rock e seus subgêneros ao longo das décadas, de 1950 até 2020. A popularidade é representada no eixo vertical, variando de 0 a 100, enquanto o eixo horizontal marca as décadas de lançamento. Cada "violino" no gráfico reflete a distribuição da popularidade em uma determinada década, mostrando como as músicas foram recebidas ao longo do tempo.

diminuiu. Esse declínio se intensifica ainda mais na década de 2020, onde o gráfico exibe uma concentração de músicas em faixas de popularidade mais baixas, indicando que o gênero como um todo perdeu relevância no cenário musical atual.

Figura 5: Distribuição da Popularidade por Década de Lançamento (Fonte: Os Autores, 2024)

Uma imagem contendo Gráfico

Descrição gerada automaticamente

As décadas de 1960, 1970 e 1980 apresentam uma distribuição ampla, o que indica que essas foram épocas de grande diversidade dentro do rock, com muitas músicas atingindo diferentes níveis de popularidade. Isso se alinha ao auge de subgêneros como hard rock, metal e rock psicodélico, que dominaram o cenário musical nesses períodos. A década de 1990 também se destaca, especialmente pela ascensão do grunge, impulsionado por bandas como Nirvana e Pearl Jam, bem como pela continuidade da força do metal, com Metallica e Iron Maiden mantendo sua popularidade.

Contudo, a partir dos anos 2000, o gráfico mostra uma clara redução na amplitude e na popularidade, sugerindo que o rock começou a perder força no mainstream musical. O surgimento de novos gêneros mais voltados ao público contemporâneo contribuiu para essa mudança, e a quantidade de músicas de rock que atingiram altos níveis de popularidade

A figura 6 apresentados comparam métricas de áudio de subgêneros do rock, como alt-rock, hard rock, punk e rock, mostrando características como danceability, energy, speechiness e outras. Observa-se que gêneros como hard rock e punk apresentam alta energia e presença de vocais, refletindo sua intensidade e estilo direto, enquanto o alt-rock destaca-se por uma maior danceability, indicando um caráter mais experimental. O punk tem baixa valence, sugerindo uma sonoridade mais negativa ou crítica, em contraste com o rock tradicional, que equilibra positividade e intensidade. Esses padrões sonoros se alinham às tendências de popularidade discutidas, onde subgêneros como hard rock e punk tiveram forte apelo no passado, mas hoje enfrentam um declínio, enquanto o alt-rock mantém certo apelo contemporâneo. Essas métricas ajudam a entender como as qualidades sonoras moldam o impacto e a relevância de cada estilo ao longo do tempo.

Figura 6: Comparação das Métricas da Categoria Rock (Os Autores, 2024)

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Por fim a figura 7 ilustra as métricas de áudio dos gêneros **dance**, **hip-hop**, **latin** e **pop**, analisando características como **danceability**, **energy**, **speechiness**, **instrumentalness**, **liveness**, **valence** e **acousticness**. Ao comparar com os subgêneros do rock mencionados no texto anterior, como alt-rock, hard rock, punk e rock, podemos identificar semelhanças e diferenças que ajudam a entender como as qualidades sonoras moldam o apelo de cada estilo musical.

Primeiramente, gêneros como **dance** e **latin** se destacam pela alta **danceability** e **energy**, o que indica um forte foco em ritmos cativantes e acessíveis para o público. Isso lembra o **alt-rock**, que também exibe uma alta **danceability**, refletindo seu caráter mais experimental e dançante. Esses gêneros tendem a ter maior aceitação em ambientes sociais e pistas de dança, o que contribui para seu apelo contemporâneo.

Por outro lado, o **hip-hop** apresenta um alto nível de **speechiness**, similar ao que se observa em gêneros como **punk** e **hard rock**, onde os vocais e as letras diretas têm um papel predominante. No entanto, o **hip-hop** mostra uma **instrumentalness** muito baixa, sugerindo que, ao contrário de algumas vertentes mais instrumentais do rock, o foco está quase inteiramente nos vocais. Isso contrasta, por exemplo, com o punk, que, apesar de sua ênfase em letras críticas, pode incluir faixas mais instrumentais.

O **pop**, como o **rock** tradicional, equilibra **energy** e **valence**, sugerindo um estilo mais otimista e comercial, que tem grande apelo entre o público. A baixa **acousticness** no **pop** sugere uma produção mais eletrônica, assim como em certos subgêneros do rock contemporâneo, como o **alt-rock**, que também incorpora elementos modernos em suas sonoridades.

Os valores de **instrumentalness** são baixos em todos os gêneros analisados, o que reflete a predominância de faixas centradas nos vocais, um fator importante para a conexão emocional com os ouvintes, tanto em gêneros como **pop** quanto em subgêneros do rock, como **punk** e **hard rock**.

Em resumo, os gêneros mais populares da atualidade, como **dance**, **pop** e **latin**, se destacam por atributos como alta **danceability** e **valence**, enquanto gêneros como **punk** e **hard rock**, com suas sonoridades mais intensas e diretas, hoje têm menor apelo comercial. A análise das métricas de áudio destaca como esses elementos sonoros contribuem para a popularidade e a relevância de cada estilo musical ao longo do tempo.

Figura 7: Comparação das Métricas da Categoria Rock (Fonte: Os Autores, 2024)

Gráfico, Gráfico de barras

Descrição gerada automaticamente

# 4. CONSIDERAÇÕES FINAIS

A análise desenvolvida ao longo deste estudo permitiu observar a evolução dos gêneros musicais entre 1950 e 2020, com foco específico no gênero rock. Embora o rock tenha passado por uma fase de intensa popularidade, especialmente nas décadas de 1970 e 1990, notamos uma queda gradual na sua prevalência a partir dos anos 2000, em parte devido à ascensão de gêneros como pop, rap e eletrônica.

No entanto, o rock demonstrou uma grande capacidade de adaptação e reinvenção, mesclando-se com outros estilos musicais e criando subgêneros híbridos. Essa flexibilidade assegurou sua relevância em um cenário musical cada vez mais dominado por novas formas de consumo e produção, principalmente impulsionadas pela era digital e pelas plataformas de streaming. O público contemporâneo, ao ter acesso a uma vasta diversidade de músicas, também contribuiu para a formação de gostos mais diversos e ecléticos, favorecendo a integração do rock com outros gêneros.

Ao longo dos anos, o rock tornou-se um reflexo das mudanças socioculturais, mantendo uma audiência fiel e adaptando-se às novas realidades do mercado musical. A análise dos dados de 2010 a 2020 revela que, apesar da diminuição em termos absolutos de popularidade, o rock continua a influenciar a música global, seja diretamente ou através de sua fusão com outros estilos.

# 5. REFERÊNCIAS

1. VINT, Ethan. How user research at Spotify is getting ahead of shifting trends in global culture. Medium. Disponível em: <https://medium.com/spotify-insights/how-user-research-at-spotify-is-getting-ahead-of-shifting-trends-in-global-culture-303c85cfee66>. Acesso em: 20/09/2024.

2. ALLEN-ROBERTSON, James. Digital culture industry: a history of digital distribution. Londres: Palgrave Macmillan, 2013.

3. WIKSTRÖM, Patrik. The music industry: music in the cloud. 2 ed. Cambridge: Polity Press, 2013.

4. SCHWARZ, Jonas A. Online file sharing: innovations in media consumption. Nova York: Routledge, 2014.

5. HERRMAN, John. Spotify Is Eating the Entire Music Business. New Work Magazine, 2023.

6. CHO, Kevin. Spotify Wrapped: Data Visualization and Machine Learning on Your Top Songs. Towards Data Science. Disponível em: <https://towardsdatascience.com/spotify-wrapped-data-visualization-and-machine-learning-on-your-top-songs-1d3f837a9b27>. Acesso em: 20/09/2024.

7. OURMUSICWORLD. How rock music transformed society in the 1950s. Our Music World, 2024. Disponível em: <https://www.ourmusicworld.com/how-rock-music-transformed-society>. Acesso em: 24 set. 2024.

8. SINUSOIDAL MUSIC. Revolutionary rhythms: The 1960s -70s counter-culture, music and impact. Sinusoidal Music, 2023. Disponível em: <https://sinusoidalmusic.com/revolutionary-rhythms>. Acesso em: 24 set. 2024.

# apêndice – Código de requisição

As nossas requisições foram feitas desta forma:

import requests

import pandas as pd

import time

import os

# Função para autenticação na API do Spotify

def get\_spotify\_token(client\_id, client\_secret):

    auth\_url = 'https://accounts.spotify.com/api/token'

    auth\_response = requests.post(auth\_url, {

        'grant\_type': 'client\_credentials',

        'client\_id': client\_id,

        'client\_secret': client\_secret,

    })

    if auth\_response.status\_code != 200:

        raise Exception(f"Failed to authenticate. Status code: {auth\_response.status\_code}. Reason: {auth\_response.text}")

    auth\_response\_data = auth\_response.json()

    return auth\_response\_data['access\_token']

# Função para buscar informações de uma faixa pelo track\_id

def get\_track\_info(track\_id, token):

    url = f"https://api.spotify.com/v1/tracks/{track\_id}"

    headers = {

        "Authorization": f"Bearer {token}"

    }

    try:

        response = requests.get(url, headers=headers)

    except requests.exceptions.ConnectionError as e:

        print(f"Connection error: {e}. Retrying in 10 seconds...")

        time.sleep(10)

        return get\_track\_info(track\_id, token)

    if response.status\_code == 429:

        retry\_after = int(response.headers.get('Retry-After', 1))

        print(f"Rate limit exceeded. Retrying after {retry\_after} seconds...")

        time.sleep(retry\_after)

        return get\_track\_info(track\_id, token)

    if response.status\_code != 200:

        print(f"Failed to fetch track info for {track\_id}. Status code: {response.status\_code}")

        return None

    return response.json()

# Função principal

def update\_csv\_with\_release\_date(csv\_path, client\_id, client\_secret, save\_every=10):

    # Autenticar na API do Spotify

    token = get\_spotify\_token(client\_id, client\_secret)

    # Verificar se já existe um CSV atualizado

    if os.path.exists('updated\_tracks.csv'):

        df\_updated = pd.read\_csv('updated\_tracks.csv')

        print("Carregando progresso anterior...")

    else:

        df\_updated = pd.read\_csv(csv\_path)

        df\_updated['release\_date'] = None

    # Obter índices das músicas que ainda não têm data de lançamento

    df\_to\_update = df\_updated[df\_updated['release\_date'].isnull()]

    # Iterar sobre cada linha restante para buscar a data de lançamento

    for index, row in df\_to\_update.iterrows():

        track\_id = row['track\_id']

        print(f"Fetching data for track ID: {track\_id}...")

        # Buscar informações da faixa pela API

        track\_info = get\_track\_info(track\_id, token)

        if track\_info and 'album' in track\_info and 'release\_date' in track\_info['album']:

            release\_date = track\_info['album']['release\_date']

            df\_updated.at[index, 'release\_date'] = release\_date

            print(f"Track ID: {track\_id}, Release Date: {release\_date}")

        else:

            print(f"No release date found for track ID: {track\_id}.")

        # Salvar progresso a cada N músicas processadas

        if (index + 1) % save\_every == 0:

            df\_updated.to\_csv('updated\_tracks.csv', index=False)

            print(f"Progress saved after {index + 1} tracks.")

    # Salvar o CSV final atualizado

    df\_updated.to\_csv('updated\_tracks.csv', index=False)

    print("Arquivo atualizado salvo como 'updated\_tracks.csv'.")

# Exemplo de uso

if \_\_name\_\_ == "\_\_main\_\_":

    # Substitua pelos seus valores de client\_id e client\_secret

    client\_id = '2b3fb6bb05fa4d50a85f83d0d3bf2400'

    client\_secret = '00fd6a333e2f4eea9bd0ca506464d564'

    # Caminho para o arquivo CSV com a coluna 'track\_id'

    csv\_path = 'rock\_dataset.csv'

    # Atualizar o CSV com as datas de lançamento

    update\_csv\_with\_release\_date(csv\_path, client\_id, client\_secret)

Esse código realiza a autenticação na API do Spotify e atualiza um arquivo CSV com a data de lançamento de faixas musicais, usando os seus **track IDs**.

**Descrição Técnica:**

1. **Bibliotecas Importadas**:
   * requests: Faz requisições HTTP, necessário para interagir com a API do Spotify.
   * pandas: Manipula o CSV, carregando e salvando dados.
   * time: Lida com atrasos de tempo, especialmente útil para gerenciar limitações de requisições da API.
   * os: Verifica a existência de arquivos locais (neste caso, para verificar se o CSV já foi atualizado).

**Funções**

**1. get\_spotify\_token(client\_id, client\_secret)**

* **Propósito**: Autentica o cliente na API do Spotify usando suas credenciais (client ID e client secret) e retorna um **token** de acesso.
* **Processo**:
  + Faz uma requisição POST para a URL de autenticação do Spotify, passando os parâmetros client\_id e client\_secret.
  + Se a resposta for bem-sucedida (status 200), o token de acesso é extraído e retornado.
  + Caso contrário, uma exceção é lançada.

**2. get\_track\_info(track\_id, token)**

* **Propósito**: Busca informações de uma faixa específica no Spotify usando seu **track ID**.
* **Processo**:
  + Faz uma requisição GET para a API de faixas do Spotify, enviando o token de acesso no cabeçalho de autorização.
  + Se houver um erro de conexão, a função tenta novamente após uma pausa de 10 segundos.
  + Se a API retornar status 429 (rate limit), aguarda o tempo especificado no cabeçalho Retry-After antes de tentar novamente.
  + Se a resposta for bem-sucedida, retorna os dados da faixa em formato JSON.
  + Se a resposta não for 200 (sucesso), retorna None.

**3. update\_csv\_with\_release\_date(csv\_path, client\_id, client\_secret, save\_every=10)**

* **Propósito**: Atualiza um arquivo CSV, adicionando a data de lançamento das faixas musicais associadas aos **track IDs** no Spotify.
* **Processo**:
  + Primeiramente, autentica na API do Spotify chamando get\_spotify\_token().
  + Verifica se já existe um arquivo CSV parcialmente atualizado chamado updated\_tracks.csv. Se sim, carrega esse arquivo, caso contrário, carrega o CSV original e adiciona uma nova coluna release\_date.
  + Filtra as linhas cujas datas de lançamento ainda não foram preenchidas (release\_date vazia).
  + Para cada faixa sem data de lançamento:
    1. Busca as informações da faixa na API usando o track\_id e get\_track\_info().
    2. Se os dados retornarem corretamente, a data de lançamento do álbum da faixa é extraída e salva na coluna release\_date.
    3. Salva o progresso a cada N faixas (configurado por save\_every), evitando perder dados no meio do processo.
  + Quando todas as faixas são processadas, o CSV final é salvo com o nome updated\_tracks.csv.

**Fluxo de Execução:**

1. O script começa no bloco if \_\_name\_\_ == "\_\_main\_\_", onde o **client ID** e o **client secret** do Spotify são fornecidos.
2. O caminho para o arquivo CSV (contendo os IDs das faixas) é especificado.
3. O script chama a função update\_csv\_with\_release\_date(), que se encarrega de atualizar o CSV, autenticando-se na API do Spotify, buscando a data de lançamento de cada faixa, e salvando as mudanças progressivamente.

**Recursos Importantes:**

* **Taxa de Requisições (Rate Limit)**: A função lida com as limitações de requisições da API (status 429) e espera o tempo necessário antes de continuar.
* **Autenticação**: O token de autenticação expira, mas a função usa o token enquanto ele é válido.
* **Persistência dos Dados**: O progresso é salvo no arquivo updated\_tracks.csv periodicamente, evitando perda de dados durante a execução.

Este código é eficiente para lidar com grandes volumes de dados, gerenciando tanto a autenticidade da requisição quanto a limitação de taxa da API.

Este código é um script em Python que interage com a API do Spotify para buscar informações sobre faixas musicais e atualizar um arquivo CSV com as datas de lançamento dessas faixas. O script está estruturado em várias funções, cada uma com uma responsabilidade específica.

**1. Importação de Bibliotecas**

import requests

import pandas as pd

import time

import os

* **requests**: Biblioteca utilizada para fazer requisições HTTP. Essencial para a comunicação com a API do Spotify.
* **pandas**: Usada para manipulação de dados em formato de tabela (DataFrames), facilitando operações de leitura e escrita de arquivos CSV.
* **time**: Permite a inclusão de delays no script, principalmente para tratar situações de rate limiting e erros de conexão.
* **os**: Fornece funcionalidades para interagir com o sistema de arquivos, como verificar a existência de arquivos.

**2. Função de Autenticação**

def get\_spotify\_token(client\_id, client\_secret):

    auth\_url = 'https://accounts.spotify.com/api/token'

    auth\_response = requests.post(auth\_url, {

        'grant\_type': 'client\_credentials',

        'client\_id': client\_id,

        'client\_secret': client\_secret,

    })

    if auth\_response.status\_code != 200:

        raise Exception(f"Failed to authenticate. Status code: {auth\_response.status\_code}. Reason: {auth\_response.text}")

    auth\_response\_data = auth\_response.json()

    return auth\_response\_data['access\_token']

* **Objetivo**: Obter um token de acesso da API do Spotify.
* **Fluxo**:
  + Uma requisição POST é enviada ao endpoint de autenticação do Spotify (/api/token), passando o client\_id, client\_secret, e o grant\_type como parâmetros.
  + Se a autenticação falhar (código de status diferente de 200), uma exceção é levantada.
  + Se bem-sucedida, a função retorna o access\_token, que será utilizado para autenticar as requisições subsequentes à API.

**3. Função de Busca de Informações da Faixa**

def get\_track\_info(track\_id, token):

    url = f"https://api.spotify.com/v1/tracks/{track\_id}"

    headers = {

        "Authorization": f"Bearer {token}"

    }

* **Objetivo**: Recuperar informações sobre uma faixa específica, utilizando seu ID.
* **Fluxo**:
  + A URL da requisição é montada usando o track\_id fornecido.
  + Um cabeçalho de autorização é preparado, incluindo o access\_token.

    try:

        response = requests.get(url, headers=headers)

    except requests.exceptions.ConnectionError as e:

        print(f"Connection error: {e}. Retrying in 10 seconds...")

        time.sleep(10)

        return get\_track\_info(track\_id, token)

A requisição GET é feita para a URL construída. Se houver uma falha de conexão, a função espera 10 segundos e tenta novamente.

if response.status\_code == 429:

        retry\_after = int(response.headers.get('Retry-After', 1))

        print(f"Rate limit exceeded. Retrying after {retry\_after} seconds...")

        time.sleep(retry\_after)

        return get\_track\_info(track\_id, token)

Se a API retornar um código de status **429**, indicando que o limite de requisições foi excedido, a função espera o tempo indicado no cabeçalho Retry-After antes de tentar novamente.

    if response.status\_code != 200:

        print(f"Failed to fetch track info for {track\_id}. Status code: {response.status\_code}")

        return None

    return response.json()

* Se a resposta for bem-sucedida (código 200), os dados da faixa são retornados em formato JSON. Se houver um erro, a função retorna None.

**4. Função Principal para Atualização do CSV**

def update\_csv\_with\_release\_date(csv\_path, client\_id, client\_secret, save\_every=10):

    token = get\_spotify\_token(client\_id, client\_secret)

    if os.path.exists('updated\_tracks.csv'):

        df\_updated = pd.read\_csv('updated\_tracks.csv')

        print("Carregando progresso anterior...")

    else:

        df\_updated = pd.read\_csv(csv\_path)

        df\_updated['release\_date'] = None

* **Objetivo**: Atualizar um arquivo CSV com as datas de lançamento das faixas.
* **Fluxo**:
  + O script obtém o token de acesso chamando a função de autenticação.
  + Verifica se já existe um CSV atualizado (updated\_tracks.csv). Se existir, carrega esse arquivo; caso contrário, carrega o CSV original (csv\_path) e adiciona uma coluna release\_date, inicializada com None.

    df\_to\_update = df\_updated[df\_updated['release\_date'].isnull()]

* Filtra o DataFrame para obter apenas as faixas que ainda não possuem uma data de lançamento.

**5. Iteração para Atualizar as Datas de Lançamento**

    for index, row in df\_to\_update.iterrows():

        track\_id = row['track\_id']

        print(f"Fetching data for track ID: {track\_id}...")

Para cada faixa que precisa de atualização, o script imprime o track\_id e busca as informações da faixa.

        track\_info = get\_track\_info(track\_id, token)

        if track\_info and 'album' in track\_info and 'release\_date' in track\_info['album']:

            release\_date = track\_info['album']['release\_date']

            df\_updated.at[index, 'release\_date'] = release\_date

            print(f"Track ID: {track\_id}, Release Date: {release\_date}")

        else:

            print(f"No release date found for track ID: {track\_id}.")

* Se as informações forem obtidas com sucesso e contiverem uma data de lançamento, essa data é armazenada no DataFrame na linha correspondente. Se não, uma mensagem de aviso é exibida.

        if (index + 1) % save\_every == 0:

            df\_updated.to\_csv('updated\_tracks.csv', index=False)

            print(f"Progress saved after {index + 1} tracks.")

* A cada N faixas processadas (definido por save\_every), o DataFrame atualizado é salvo em um arquivo CSV para evitar perda de dados em caso de falhas.

    df\_updated.to\_csv('updated\_tracks.csv', index=False)

    print("Arquivo atualizado salvo como 'updated\_tracks.csv'.")

* Ao final do processo, o DataFrame atualizado é salvo no arquivo updated\_tracks.csv.

**6. Execução do Script**

if \_\_name\_\_ == "\_\_main\_\_":

    client\_id = '2b3fb6bb05fa4d50a85f83d0d3bf2400'

    client\_secret = '00fd6a333e2f4eea9bd0ca506464d564'

    csv\_path = 'rock\_dataset.csv'

    update\_csv\_with\_release\_date(csv\_path, client\_id, client\_secret)

* O bloco if \_\_name\_\_ == "\_\_main\_\_": é o ponto de entrada do script, onde os valores de client\_id, client\_secret e o caminho do arquivo CSV são definidos. A função principal é então chamada para iniciar o processo.

**Resumo do Fluxo Geral**

1. O script autentica na API do Spotify e obtém um token de acesso.
2. Ele carrega um CSV existente ou cria um novo para armazenar as datas de lançamento.
3. O script filtra as faixas que não têm datas de lançamento e itera sobre elas.
4. Para cada faixa, busca informações da API do Spotify, atualiza o DataFrame e salva o progresso a cada N faixas processadas.
5. No final, o arquivo CSV atualizado é salvo.

Este script é uma implementação eficaz para automatizar a busca de informações de faixas musicais e a atualização de arquivos CSV, integrando várias técnicas de programação, como manipulação de arquivos, requisições HTTP e tratamento de erros.