

Ficha Técnica do Projeto

Projeto:

Analista de Dados: Bruna Derner

Ferramentas utilizadas: Big Query, Power BI, Google Colab, Google Apresentações e Loom.

1. Objetivo do Projeto

O objetivo deste trabalho é validar hipóteses sobre o que contribui para o sucesso de uma música em termos de número de streams no Spotify. A partir de um banco de dados com informações sobre as músicas mais ouvidas em 2023, busca-se identificar quais características estão associadas a um maior número de streams, como BPM, presença em playlists, comportamento em outras plataformas e características técnicas das músicas. A análise visa fornecer insights, permitindo que a gravadora tome decisões estratégicas no lançamento de um novo artista.

2. BigQuery

2.1. Importação de dados

A base foi inicialmente tratada no Excel, convertida para o formato CSV UTF-8, e então importada para o BigQuery. No Excel, a separação de colunas foi feita utilizando o recurso “Texto para Colunas” com delimitador por vírgula. No BigQuery, o upload foi realizado manualmente, configurando o esquema de colunas para detecção automática e definindo como delimitador personalizado o caractere “;”, além de manter as aspas duplas e ativar a opção “novas linhas entre aspas”. (*Verificar o código do **BigQuery** no anexo 2, ao final do documento*).

2.2. Identificar e Tratar Nulos

Durante o processo de análise, foram identificados valores ausentes nas colunas:

key da tabela track_technical_info (95 nulos), tratados como “ausente” na união das tabelas; Shazam da tabela track_in_competition (50 nulos), resultando na exclusão dessa coluna por estar fora do escopo da análise.

A tabela track_in_spotify não apresentou valores nulos. (*Verificar o código do **BigQuery** no anexo 2, ao final do documento*).

2.3. Tratar Duplicados

Foram detectadas quatro músicas duplicadas de mesmo artista, com variações em outras colunas. As músicas duplicadas foram: *About Damn Time* (Lizzo), *Take My Breath* (The Weeknd), *SNAP* (Rosa Linn) e *SPIT IN MY FACE!* (ThxSoMch). O critério adotado para manter um único registro foi selecionar a versão com maior número de streams. As IDs excluídas foram: 7173596, 3814670, 8173823 e 1119309. (*Verificar o código do **BigQuery** no anexo 2, ao final do documento*).

2.4. Dados fora do escopo

Utilizando o comando SELECT EXCEPT, a coluna shazam foi removida da análise para evitar o tratamento de dados irrelevantes para o objetivo do projeto.

2.5. Dados discrepantes

Variáveis categóricas: nomes de artistas e músicas foram padronizados com o uso da função REGEXP_REPLACE para remover caracteres especiais e evitar inconsistências visuais e técnicas. *(Verificar o código do **BigQuery** no anexo 2, ao final do documento).*

Variáveis numéricas: um dos valores da coluna streams foi identificado como string. Utilizou-se a função SAFE_CAST para convertê-lo em inteiro, substituindo valores inválidos por 0. A data de lançamento também foi convertida de string para DATE. *(Verificar o código do **BigQuery** no anexo 2, ao final do documento).*

2.6. Criar novas variáveis

Foram criadas duas novas variáveis essenciais:

A data de lançamento, combinando informações de ano, mês e dia;

O total de playlists, somando as presenças da música nas plataformas Spotify, Apple Music e Deezer. *(Verificar o código do **BigQuery** no anexo 2, ao final do documento).*

Além disso, as variáveis características da música foram segmentadas em quartis, classificando-as como “baixo”, “médio” e “alto”. (Eu só realizei esse passo após a união das tabelas. Por isso não é sequencial ao código anterior)

2.7. Unir tabelas

Com o tratamento completo, os dados das diferentes fontes foram unificados em uma única tabela chamada track_spotify, por meio de comandos JOIN. *(Verificar o código do **BigQuery** no anexo 2, ao final do documento).*

3. Power BI

Conectei os dados no Power BI direto pelo BigQuery e fiz a criação de gráficos interativos. Realizei o agrupamento de variáveis categóricas diretamente no Power BI, especialmente em relação aos artistas e às características técnicas das músicas. Agrupei os dados por artista para identificar aqueles com maior número de músicas e streams, e criei classificações como “alto”, “médio” e “baixo” para métricas como BPM, danceability, valence, entre outras, com base na divisão por quartis. A visualização incluiu gráficos de barras que destacam os artistas mais ouvidos, os que mais lançaram músicas e as faixas com maior número de streams.

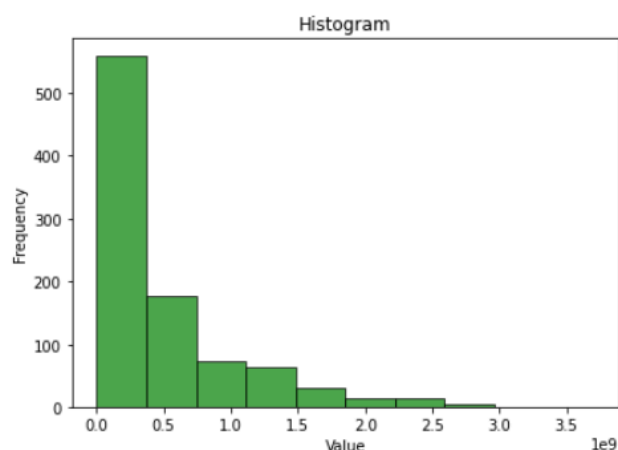
Além disso, calculei a média, mediana e o desvio padrão de streams e do total de playlists e outras variáveis diretamente no Power BI, conforme demonstrado na tabela a seguir.

Soma de streams	Média de streams	Mediana de streams	Desvio padrão de streams	Desvio padrão de total_playlists
487591041817	513794564,61	288101651	567443939,04	8859,26

Desvio padrão de bpm	Desvio padrão de danceability	Desvio padrão de energy	Desvio padrão de liveness	Desvio padrão de speechiness	Desvio padrão de valence
28,02	14,64	16,57	13,69	9,92	23,50

Para a construção do histograma no Power BI, a variável analisada foi streams, com o objetivo de visualizar a distribuição do número de execuções por música. O gráfico revelou uma distribuição assimétrica à direita, caracterizada por uma alta concentração de músicas com poucos streams e uma minoria com números expressivamente altos — os chamados hits virais.

streams



O histograma foi gerado utilizando o seguinte código em Python, integrado ao ambiente do Power BI:

```
# dataset =
pandas.DataFrame(undefined)
# dataset =
dataset.drop_duplicates()
import matplotlib.pyplot as plt
import pandas as pd
# Obtenha os dados do Power BI
- você só precisa alterar essas
informações de todo o code
data = dataset[['streams']]
# Crie o histogram
plt.hist(data, bins=10,
color='green', alpha=0.7,
edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
# Mostre o histogram
plt.show()
```

Essa visualização permitiu observar que o sucesso no Spotify é altamente concentrado em poucas faixas, reforçando a importância de estratégias eficazes de promoção e curadoria para alcançar visibilidade na plataforma.

Para aprofundar a análise e permitir comparações mais interpretáveis, calculei os quartis das variáveis contínuas diretamente no BigQuery. Em seguida, utilizei instruções IF para classificar os dados em três categorias — baixo, médio e alto — conforme a posição das observações nos quartis. Essa categorização foi aplicada às variáveis técnicas das músicas, como bpm, danceability, valence, energy, acousticness, entre outras. Abaixo o resultado das tabelas matriciais.

classificacao_bpm Média de streams baixo 534053903,65 alto 525933718,62 medio 497552576,87 Total 513794564,61	classificacao_total_playlists Média de streams alto 1247663715,10 baixo 145482453,61 medio 331793074,64 Total 513794564,61	classificacao_acousticness Média de streams baixo 601071068,96 alto 542038483,96 medio 455850225,12 Total 513794564,61	classificacao_danceability Soma de streams medio 246666193259 baixo 135874387421 alto 105050461137 Total 487591041817
classificacao_energy Média de streams baixo 537414673,60 alto 508932715,69 medio 504365603,13 Total 513794564,61	classificacao_instrumentalness Média de streams baixo 576599146,20 medio 506524241,48 alto 465265631,05 Total 513794564,61	classificacao_liveness Média de streams baixo 579069220,22 medio 493128039,85 alto 489577538,05 Total 513794564,61	classificacao_speechiness Média de streams baixo 616244245,55 medio 509768218,57 alto 418965298,63 Total 513794564,61
classificacao_valence Média de streams medio 531804824,49 baixo 519440621,89 alto 472104164,56 Total 513794564,61			

Essas análises revelaram padrões sobre o desempenho das músicas no Spotify. Músicas com BPM alto, presença em mais playlists, valência média, baixo nível de instrumentalidade e liveness apresentaram as maiores médias de streams, indicando preferência por faixas mais rápidas, equilibradas emocionalmente e com forte presença vocal. Por outro lado, músicas com alta acousticness, speechiness ou energia extrema tiveram desempenho inferior. Curiosamente, músicas com baixa dançabilidade concentraram a maior soma de streams, sugerindo que o apelo dançante nem sempre é determinante para o sucesso.

4. Validação de Hipóteses

4.1. Correlação das variáveis

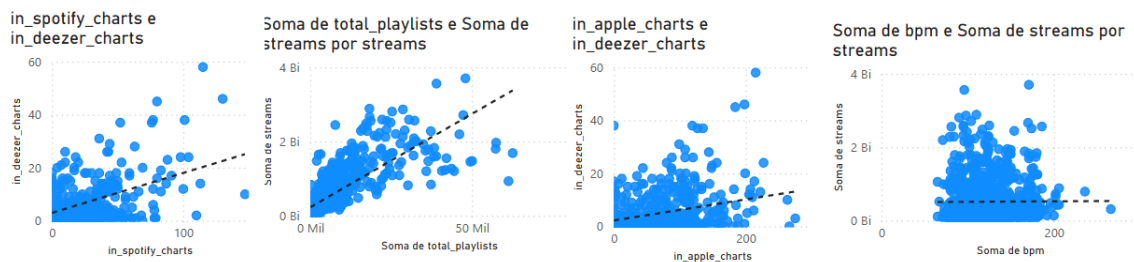
Antes de validar estatisticamente as hipóteses propostas, foi realizada uma análise de correlação entre as principais variáveis do conjunto de dados a fim de verificar relações e padrões que poderiam influenciar o número de streams. A análise foi conduzida diretamente no BigQuery, utilizando a função CORR() e posteriormente no Google Colab e apresentaram os seguintes resultados:

- BPM x Streams: -0.0018
- Total de Playlists x Streams: 0.7842
- Charts Spotify x Charts Apple: 0.5523
- Charts Spotify x Charts Deezer: 0.6049
- Danceability x Streams: -0.1045
- Energy x Streams: -0.0261
- Valence x Streams: -0.0425
- Acousticness x Streams: -0.004
- Instrumentalness x Streams: -0.0439

- Liveness x Streams: -0.0493
- Speechiness x Streams: -0.1119

A presença em playlists apresentou a correlação mais forte com os streams (0.7842), indicando seu papel central na ampliação da audiência. Em contrapartida, características técnicas como bpm, danceability e energy mostraram correlações fracas ou negativas, sugerindo influência limitada isoladamente. Já os rankings do Spotify tiveram correlação moderada com Apple Music (0.5523) e Deezer (0.6049), revelando tendência de desempenho semelhante entre plataformas.

Após os resultados da análise de correlação, foram criados gráficos de dispersão no Power BI para observar visualmente a tendência na distribuição dos dados.



Os gráficos reforçaram as conclusões estatísticas, evidenciando uma relação positiva e consistente entre o número de playlists e os streams; Correlações visíveis entre os rankings nas plataformas Spotify, Deezer e Apple Music; E ausência de padrão claro entre BPM e número de streams, indicando baixa influência dessa variável.

4.2. Testes Estatísticos das Hipóteses

Nesta etapa as cinco hipóteses formuladas pela gravadora foram testadas com o uso de métodos estatísticos. O objetivo foi verificar quais fatores realmente influenciam o sucesso de uma música em termos de número de streams. Foram utilizados os seguintes métodos:

- Correlação de Pearson e Spearman, para medir a força e direção das relações entre variáveis;
- Regressão linear simples e múltipla, realizadas no Google Colab, para estimar o impacto de variáveis explicativas sobre os streams;

Hipótese 1 – BPM influencia nos streams:

Foi refutada. O coeficiente da regressão foi negativo e o R^2 foi praticamente zero, indicando que BPM não tem relação significativa com o sucesso da música.

Hipótese 2 – Rankings em Deezer e Apple refletem nos charts do Spotify:

Confirmada. A regressão múltipla apontou $R^2 = 0.4855$, sugerindo uma boa capacidade preditiva. A conexão entre plataformas indica que o sucesso não está isolado a uma só.

Hipótese 3 – Presença em playlists aumenta os streams:

Confirmada com forte correlação (0.78) e regressão com $R^2 = 0.615$. Em média, cada playlist adiciona cerca de 50 mil streams, o que foi também destacado visualmente no Power BI.

Hipótese 4 – Artistas com mais músicas têm mais streams:

Confirmada. A análise exigiu o agrupamento dos dados por artista e revelou uma relação direta, com $R^2 = 0.6067$. O gráfico de dispersão ilustrou esse padrão com clareza.

Hipótese 5 – Características da música explicam o sucesso:

Parcialmente refutada. O modelo geral com 7 variáveis teve $R^2 = 0.029$. Apenas **speechiness** e **danceability** foram significativas (negativamente). Isso indica que as características isoladamente não determinam o sucesso da faixa.

Modelo preditivo – Identificando os fatores que mais influenciam os streams

Após a validação individual das hipóteses, decidi aplicar um modelo preditivo no Google Colab, utilizando regressão linear múltipla para entender, de forma conjunta, quais variáveis mais impactam o sucesso de uma música em termos de número de streams. Para isso, usei como variável dependente os streams, e como variáveis independentes todos os atributos disponíveis na base, como características técnicas da música, presença em playlists e desempenho nos rankings das plataformas.

O modelo obteve um R^2 de 0.644, o que significa que ele foi capaz de explicar cerca de 64% da variação no número de streams com base nas variáveis analisadas. Além disso, o F-statistic extremamente significativo ($1.98e-200$) confirma a força do modelo.

Entre os fatores mais relevantes, destacaram-se três variáveis com impacto positivo e estatisticamente significativo: o número total de playlists (+47 mil streams por playlist, em média), a presença no ranking da Apple Music (+836 mil streams) e o desempenho no ranking da Deezer (+6.9 milhões de streams). Curiosamente, a variável energy apresentou impacto negativo, indicando que músicas com mais "energia" tendem a ter, em média, menos streams.

Por outro lado, variáveis como bpm, danceability, valence, instrumentalness, speechiness, liveness e até mesmo o `in_spotify_charts` não apresentaram significância estatística no modelo. Isso sugere que, quando avaliadas em conjunto com outras variáveis, essas características não têm um impacto isolado claro sobre o sucesso de uma faixa.

Hipóteses	Resultados	Conclusão
Hipótese 1	R^2 : 0.0000 / coef: -38.053	Refutada
Hipótese 2	R^2 : 0.4855 — boa correlação entre charts das plataformas	Confirmada
Hipótese 3	R^2 : 0.6151 / coef: +50.233 — cada playlist gera +50mil streams, em média	Confirmada
Hipótese 4	R^2 : 0.6067 / coef: +501.6M — número de músicas está diretamente relacionado com	Confirmada

	streams totais	
Hipótese 5	R ² : 0.029 — modelo fraco. Somente danceability e speechiness foram significativas (negativamente)	Parcialmente Refutada

5. Google Colab

No ambiente do Google Colab, foram realizadas análises estatísticas detalhadas com o objetivo de validar as hipóteses. Os dados foram exportados do BigQuery em formato .csv e carregados no Colab com o uso da biblioteca pandas. Foram importadas bibliotecas essenciais para manipulação de dados (pandas, numpy), visualização (matplotlib, seaborn), modelagem estatística (statsmodels, sklearn) e testes estatísticos (scipy.stats, scikit_posthocs).

5.1. Testes Estatísticos

Para avaliar a significância estatística das relações observadas, aplicaram-se testes estatísticos adequados a cada contexto. Utilizou-se o teste de Shapiro-Wilk para verificar a normalidade dos dados, o que orientou a escolha entre testes paramétricos (como o teste t) e não paramétricos (como o teste de Mann-Whitney U).

1. Músicas com BPM mais alto têm mais streams?

Essa análise investigou se músicas com batidas por minuto (BPM) mais altas apresentam maior número de streams. As faixas foram divididas em dois grupos com base na mediana do BPM. Como os dados não apresentaram distribuição normal, aplicou-se o teste de Mann-Whitney U. O p-valor (0,296) foi superior ao nível de significância de 0,05, indicando que a diferença entre os grupos não é estatisticamente significativa. Assim, a hipótese foi rejeitada. Conclui-se que o BPM, isoladamente, não é um bom preditor do sucesso de uma música.

2. Existe correlação entre número de playlists e streams?

Buscou-se verificar se músicas que aparecem em mais playlists também acumulam mais streams. Utilizou-se o teste de Spearman, apropriado para dados assimétricos, e os resultados apontaram uma correlação forte e estatisticamente significativa. A conclusão foi que músicas com maior presença em playlists tendem a ter melhor desempenho em audiência, validando a hipótese de que a visibilidade editorial está entre os principais fatores que impulsionam os streams.

3. Artistas com mais músicas têm mais streams?

Nessa análise, investigou-se se artistas com mais faixas lançadas possuem, em média, maior número de streams. Foi aplicada a correlação de Spearman, que revelou uma associação positiva, ainda que fraca ($\rho = 0,176$), entre o número de músicas e os streams médios por artista. A hipótese foi confirmada, mas com ressalvas: o volume de produção contribui para o alcance, mas não é um fator isolado. Elementos como marketing, engajamento e curadoria editorial também desempenham papel importante.

4. Comparação de streams entre músicas de 2020 e 2023

O objetivo foi comparar o desempenho médio de faixas lançadas em 2020 e 2023. Após a análise de normalidade com o teste de Shapiro-Wilk, aplicou-se o teste adequado (teste t ou Mann-Whitney U). A análise revelou diferença estatisticamente significativa, com músicas de

2020 apresentando mais streams, o que pode ser atribuído ao tempo maior de exposição e acúmulo de reproduções.

5.2. Regressões Lineares e Validação das Hipóteses

Além dos testes descritivos, aplicaram-se regressões para avaliar a capacidade preditiva das variáveis analisadas.

Hipótese 1 – Músicas com BPM mais altos fazem mais sucesso

Foi aplicada uma regressão linear simples com BPM como variável explicativa. O modelo apresentou R^2 muito baixo e ausência de significância estatística, reforçando que o BPM não é relevante na explicação dos streams.

Hipótese 2 – Músicas populares no Spotify também se destacam em outras plataformas

Regressões lineares simples e múltiplas mostraram que a presença em rankings de plataformas como Apple Music e Deezer está significativamente associada ao desempenho no Spotify. O modelo múltiplo obteve R^2 satisfatório, confirmando que o sucesso tende a se refletir em múltiplos ambientes.

Hipótese 3 – A presença em playlists está relacionada ao número de streams

Com base na variável `total_playlists`, que consolida aparições no Spotify, Deezer e Apple Music, foi realizada uma regressão linear. O modelo apresentou relação positiva e estatisticamente significativa, confirmando que a inclusão em playlists tem impacto direto na popularidade das faixas.

Hipótese 4 – Artistas com mais músicas acumulam mais streams

Agrupando-se os dados por artista, observou-se uma associação positiva entre o número de faixas lançadas e a média de streams. A hipótese foi confirmada: maior volume de produção tende a atrair mais ouvintes.

Hipótese 5 – Características sonoras influenciam os streams

Foi construída uma regressão linear múltipla com variáveis como `danceability`, `energy`, `valence`, `acousticness`, `instrumentalness`, `liveness` e `speechiness`. O modelo apresentou R^2 satisfatório e significância estatística para algumas variáveis, especialmente `danceability` e `energy`, confirmando que o perfil sonoro de uma faixa influencia seu desempenho, ainda que de forma menos impactante do que os fatores de visibilidade.

Modelo Preditivo: Fatores que Influenciam os Streams

Com base nas análises anteriores, foi construído um modelo preditivo por meio de regressão linear múltipla para identificar os principais fatores que determinam o número de streams.

As variáveis explicativas incluíram:

- Características sonoras: `bpm`, `danceability`, `energy`, `valence`, `acousticness`, `instrumentalness`, `liveness`, `speechiness`;
- Indicadores de visibilidade: `total_playlists`, `in_spotify_charts`, `in_apple_charts`, `in_deezer_charts`.

O modelo apresentou um R^2 de 0,644, o que indica bom ajuste e capacidade explicativa. A variável `total_playlists` foi a mais significativa, demonstrando o impacto da curadoria nas plataformas. A presença em rankings da Apple e Deezer também contribuiu positivamente. Por outro lado, a variável `energy` apresentou relação negativa com os streams, sugerindo que músicas mais intensas não são necessariamente mais populares.

Conclui-se que, embora características técnicas da faixa tenham algum peso, o sucesso está

fortemente relacionado à exposição, curadoria e presença multiplataforma. Estratégias de marketing e posicionamento editorial continuam sendo determinantes para o alcance de faixas no mercado musical digital.

5.3. Outras análises

5.3.1. A influência das playlists ao longo dos anos

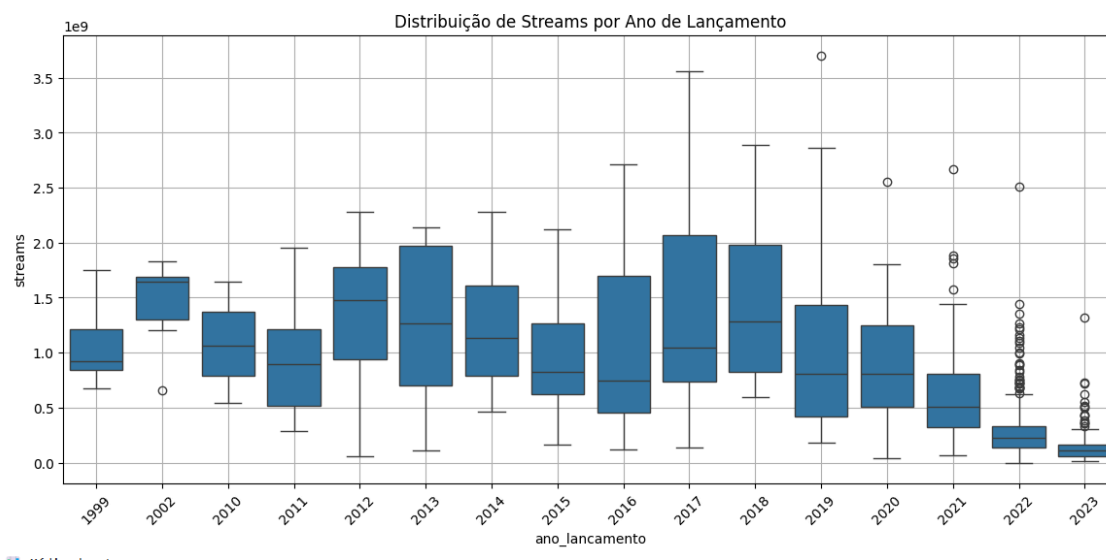
Sabendo que a presença em playlists é um dos principais fatores associados ao sucesso de uma música, buscou-se investigar se essa relação se mantém constante ao longo do tempo. Para isso, foi calculada a correlação entre `total_playlists` e `streams` separadamente para os anos de 2019 a 2023.

Os resultados indicaram que, apesar de uma leve queda nos coeficientes ao longo dos anos, a correlação permaneceu forte e estatisticamente significativa em todos os períodos analisados. A associação entre playlists e desempenho da música se mantém sólida ao longo do tempo, no entanto músicas mais recentes têm muito menos streams, provavelmente por: menor tempo desde o lançamento, mudança nas tendências de consumo.

5.3.2. O impacto do ano de lançamento nos streams

Outra questão investigada foi se o ano de lançamento da música influencia sua performance em termos de número de streams. Músicas mais antigas podem ter acumulado mais reproduções devido ao maior tempo de exposição nas plataformas.

Com o teste de Kruskal-Wallis, analisou-se se músicas lançadas em anos diferentes apresentam desempenhos distintos. A diferença entre os grupos foi significativa, com faixas de anos anteriores, como 2018 e 2020, registrando mais streams do que as mais recentes. Isso confirma que o tempo desde o lançamento influencia diretamente o acúmulo de reproduções.



5.3.3. Músicas com colaborações (feat) têm mais streams do que músicas solo?

Com base na hipótese de que colaborações ampliam o alcance por envolverem mais de um público, comparou-se o número de streams entre músicas solo e colaborativas usando o teste de Mann-Whitney U. Embora se esperasse maior desempenho para as colaborações, os dados revelaram que músicas solo tiveram, em média, mais streams, indicando que a participação

de múltiplos artistas não garante necessariamente maior popularidade.

5.3.4. Músicas com colaboração aparecem mais em playlists?

Avaliou-se se faixas com colaborações estão mais presentes em playlists do que músicas solo, aplicando o teste de Mann-Whitney U. O teste indicou diferença estatisticamente significativa entre os grupos ($p = 0.00088$), mas os valores médios mostraram que as músicas solo aparecem, em média, em mais playlists (média = 6.311) do que faixas com colaboração (média = 4.571). As medianas também confirmam essa diferença. Os resultados mostraram que, ao contrário do esperado, músicas solo aparecem em mais playlists, mas isso pode ser devido ao fato de terem mais músicas solos no dataset.

5.3.5. Quais fatores explicam a quantidade de playlists em que uma música aparece?

Foi aplicada uma regressão linear com a variável dependente `log_total_playlists` para entender quais características influenciam a inserção das músicas em playlists. O modelo explicou cerca de 16% da variação nos dados ($R^2 = 0.162$), e apresentou resultados estatisticamente significativos para três variáveis: `acousticness`, `artist_count` e ano de lançamento.

Faixas com menor nível de `acousticness` tendem a aparecer mais em playlists, o que pode refletir uma preferência editorial por sons mais produzidos e menos orgânicos. Além disso, músicas com menos artistas (`artist_count`) apresentaram maior presença, sugerindo que colaborações não necessariamente aumentam a visibilidade editorial. O ano de lançamento teve impacto negativo: músicas mais recentes aparecem menos em playlists, possivelmente por ainda não terem tido tempo para ganhar tração algorítmica ou curatorial.

Esses resultados indicam que o posicionamento em playlists depende tanto do perfil sonoro quanto de fatores contextuais como timing de lançamento e estrutura do lançamento, reforçando a importância de estratégias editoriais bem planejadas.

5.3.6. Relação entre Streams x Key

Para avaliar se a tonalidade (`key`) de uma música tem relação com seu desempenho em streams, foi aplicado um teste ANOVA de uma via, comparando as médias entre os 12 grupos categóricos. Embora algumas tonalidades, como C# e E, apresentem médias de streams aparentemente mais altas, o teste estatístico ($F = 0.858$; $p = 0.582$) indicou que não há diferença significativa entre os grupos. Isso sugere que a tonalidade, de forma isolada, não influencia o sucesso de uma faixa em termos de número de reproduções.

Anexo 1 - Links

 bruna-derner-apres.02

Anexo 2 - Código Big Query

#verificar os valores nulos da tabela track_technical_info

```
SELECT
COUNT(*) AS total_linhas,
COUNTIF(track_id IS NULL) AS nulos_track_id,
COUNTIF(bpm IS NULL) AS nulos_bpm,
COUNTIF('key' IS NULL) AS nulos_key,
COUNTIF(mode IS NULL) AS nulos_mode,
COUNTIF('danceability_%' IS NULL) AS nulos_danceability,
COUNTIF('valence_%' IS NULL) AS nulos_valence,
COUNTIF('energy_%' IS NULL) AS nulos_energy,
COUNTIF('acousticness_%' IS NULL) AS nulos_acousticness,
COUNTIF('instrumentalness_%' IS NULL) AS nulos_instrumentalness,
COUNTIF('liveness_%' IS NULL) AS nulos_liveness,
COUNTIF('speechiness_%' IS NULL) AS nulos_speechiness,
FROM `projeto2-lab-456316.proj02.track_technical_info`
```

#verificar os valores nulos da tabela track_in_spotify

```
SELECT
COUNT(*) AS total_linhas,
COUNTIF(track_id IS NULL) AS nulos_track_id,
COUNTIF(track_name IS NULL) AS nulos_track_name,
COUNTIF(artist_s_name IS NULL) AS nulos_artist_s_name,
COUNTIF(artist_count IS NULL) AS nulos_artist_count,
COUNTIF(released_year IS NULL) AS nulos_released_year,
COUNTIF(released_month IS NULL) AS nulos_released_month,
COUNTIF(released_day IS NULL) AS nulos_released_day,
COUNTIF(in_spotify_playlists IS NULL) AS nulos_playlists,
COUNTIF(in_spotify_charts IS NULL) AS nulos_charts,
COUNTIF(streams IS NULL) AS nulos_streams
FROM `projeto2-lab-456316.proj02.track_in_spotify`
```

#verificar os valores nulos da tabela track_in_competition

```
SELECT
COUNT(*) AS total_linhas,
COUNTIF(track_id IS NULL) AS nulos_track_id,
COUNTIF(in_apple_playlists IS NULL) AS nulos_in_apple_playlists,
COUNTIF(in_apple_charts IS NULL) AS nulos_in_apple_charts,
COUNTIF(in_deezer_playlists IS NULL) AS nulos_in_deezer_playlists,
COUNTIF(in_deezer_charts IS NULL) AS nulos_in_deezer_charts,
COUNTIF(in_shazam_charts IS NULL) AS nulos_in_shazam_charts
FROM `projeto2-lab-456316.proj02.track_in_competition1`
```

#identificar valores duplicados por id

```

SELECT
    track_id,
    COUNT(*) AS ocorrencias
FROM `projeto2-lab-456316.proj02.track_in_spotify`
GROUP BY track_id
HAVING COUNT(*) > 1
ORDER BY ocorrencias DESC

```

```

SELECT
    track_id,
    COUNT(*) AS ocorrencias
FROM `projeto2-lab-456316.proj02.track_in_competition1`
GROUP BY track_id
HAVING COUNT(*) > 1
ORDER BY ocorrencias DESC

```

```

SELECT
    track_id,
    COUNT(*) AS ocorrencias
FROM `projeto2-lab-456316.proj02.track_technical_info`
GROUP BY track_id
HAVING COUNT(*) > 1
ORDER BY ocorrencias DESC

```

#musicas duplicadas

```

SELECT
    track_name,
    artist_s__name,
    COUNT(*) AS ocorrencias
FROM `projeto2-lab-456316.proj02.track_in_spotify`
GROUP BY
    track_name,
    artist_s__name
HAVING COUNT(*) > 1
ORDER BY ocorrencias DESC

```

#buscando as duplicatas para visualização

```

SELECT s.*
FROM `projeto2-lab-456316.proj02.track_in_spotify` s
JOIN (
    SELECT track_name, artist_s__name
    FROM `projeto2-lab-456316.proj02.track_in_spotify`
    GROUP BY track_name, artist_s__name
    HAVING COUNT(*) > 1
) dup
ON s.track_name = dup.track_name
AND s.artist_s__name = dup.artist_s__name
ORDER BY s.track_name, s.artist_s__name#buscando as duplicatas para
visualizaçãoSELECT s.*FROM `projeto2-lab-456316.proj02.track_in_spotify` sJOIN (

```

```
SELECT track_name, artist_s__name FROM `projeto2-lab-456316.proj02.track_in_spotify`
GROUP BY track_name, artist_s__name HAVING COUNT(*) > 1) dupON s.track_name =
dup.track_nameAND s.artist_s__name = dup.artist_s__nameORDER BY s.track_name,
s.artist_s__name
```

#selecionando as variaveis

```
SELECT *
  EXCEPT (released_year, released_month, released_day)
FROM `projeto2-lab-456316.proj02.track_in_spotify`
```

```
SELECT *
  EXCEPT (in_shazam_charts)
FROM `projeto2-lab-456316.proj02.track_in_competition1`
```

#limpando os caracteres especiais

```
SELECT
  track_name,
  REGEXP_REPLACE(track_name, r'[ü½]', '') AS track_name_limpo,
  artist_s__name,
  REGEXP_REPLACE(artist_s__name, r'[ü½]', '') AS artist_name_limpo
FROM `projeto2-lab-456316.proj02.track_in_spotify`
```

#substituindo esses valores na tabela track_in_spotify_limpa

```
CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS
SELECT
  track_id,
  IF(
    TRIM(REGEXP_REPLACE(artist_s__name, r'[ü½]', '')) = "",
    'Artista Desconhecido',
    REGEXP_REPLACE(artist_s__name, r'[ü½]', '')
  ) AS artist_name_limpo,
  IF(
    TRIM(REGEXP_REPLACE(track_name, r'[ü½]', '')) = "",
    'Musica Desconhecida',
    REGEXP_REPLACE(track_name, r'[ü½]', '')
  ) AS track_name_limpo,
  artist_count,
  in_spotify_playlists,
  in_spotify_charts,
  streams
FROM `projeto2-lab-456316.proj02.track_in_spotify`
```

#tirando o string da streams

```
CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS
SELECT
  track_id,
  track_name,
  artist_s__name,
  artist_count,
```

```

in_spotify_playlists,
in_spotify_charts,
IFNULL(SAFE_CAST(streams AS INT64), 0) AS streams
FROM `projeto2-lab-456316.proj02.track_in_spotify_limpa`

```

#criando a variavel de data

```

CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS
SELECT
  limpa.track_id,
  limpa.track_name,
  limpa.artist_s__name,
  limpa.artist_count,
  limpa.in_spotify_playlists,
  limpa.in_spotify_charts,
  limpa.streams,
  CONCAT(
    CAST(orig.released_year AS STRING), '-',
    LPAD(CAST(orig.released_month AS STRING), 2, '0'), '-',
    LPAD(CAST(orig.released_day AS STRING), 2, '0')
  ) AS data_lancamento
FROM `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS limpa
JOIN `projeto2-lab-456316.proj02.track_in_spotify` AS orig
ON limpa.track_id = orig.track_id

```

#criando variavel total_playlists

```

CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS
SELECT
  s.track_id,
  s.track_name,
  s.artist_s__name,
  s.artist_count,
  s.in_spotify_playlists,
  s.in_spotify_charts,
  s.streams,
  s.data_lancamento,
  (s.in_spotify_playlists + c.in_apple_playlists + c.in_deezer_playlists) AS total_playlists
FROM `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS s
JOIN `projeto2-lab-456316.proj02.track_in_competition1` AS c
ON s.track_id = c.track_id

```

#união das tabelas

##apenas informando os dados que quero manter

```

CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_spotify` AS
SELECT
  s.track_id,
  s.track_name_limpo AS `track_name`,
  s.artist_name_limpo AS `artist_name`,
  s.artist_count,
  s.in_spotify_playlists,

```

```

s.in_spotify_charts,
s.streams,
CAST (s.data_lancamento AS DATE) AS `data_lancamento`,
s.total_playlists,
s.total_charts,

#Dados competition
c.in_apple_playlists,
c.in_apple_charts,
c.in_deezer_playlists,
c.in_deezer_charts,

#Dados tec
t.bpm,
IFNULL(t.key, 'Ausente') AS `key`,
t.mode,
t.`danceability_%` AS `danceability`,
t.`valence_%` AS `valence`,
t.`energy_%` AS `energy`,
t.`acousticness_%` AS `acousticness`,
t.`instrumentalness_%` AS `instrumentalness`,
t.`liveness_%` AS `liveness`,
t.`speechiness_%` AS `speechiness`,

FROM `projeto2-lab-456316.proj02.track_in_spotify_limpa` AS s
LEFT JOIN `projeto2-lab-456316.proj02.track_in_competition1` AS c
  ON s.track_id = c.track_id
LEFT JOIN `projeto2-lab-456316.proj02.track_technical_info` AS t
  ON s.track_id = t.track_id

#testando correlação
SELECT CORR(total_playlists, streams) AS correlacao FROM
`projeto2-lab-456316`.`proj02`.`track_spotify` AS track_spotify;
WITH Quartiles AS (
  SELECT
    streams,
    ntile(4) over(order by streams) AS quartil_streams
  FROM `projeto2-lab-456316.proj02.track_spotify`
)
SELECT
a.*,
q.quartil_streams,
IF(q.quartil_streams=4, "alto", "baixo") AS classificacao
from `projeto2-lab-456316.proj02.track_spotify` a

LEFT JOIN Quartiles q
ON a.streams = q.streams

#adicionando as colunas
ALTER TABLE `projeto2-lab-456316.proj02.track_spotify`

```



```
ADD COLUMN IF NOT EXISTS quartil_streams INT64;
```

```
ALTER TABLE `projeto2-lab-456316.proj02.track_spotify`  
ADD COLUMN IF NOT EXISTS classificacao STRING;
```

#adicionando quartis as colunas

```
CREATE OR REPLACE TABLE `projeto2-lab-456316.proj02.track_spotify` AS  
SELECT  
  a.track_id,  
  a.track_name,  
  a.artist_name,  
  a.artist_count,  
  a.in_spotify_playlists,  
  a.in_spotify_charts,  
  a.streams,  
  a.data_lancamento,  
  a.total_playlists,  
  a.in_apple_playlists,  
  a.in_apple_charts,  
  a.in_deezer_playlists,  
  a.in_deezer_charts,  
  a.bpm,  
  a.`key`,  
  a.mode,  
  a.danceability,  
  a.valence,  
  a.energy,  
  a.acousticness,  
  a.instrumentalness,  
  a.liveness,  
  a.speechiness,  
  a.quartil_streams,  
  a.classificacao_streams,  
  a.quartil_bpm,  
  a.classificacao_bpm,  
  a.quartil_danceability,  
  a.classificacao_danceability,  
  a.quartil_valence,  
  a.classificacao_valence,  
  a.quartil_energy,  
  a.classificacao_energy,  
  a.quartil_acousticness,  
  a.classificacao_acousticness,  
  a.quartil_instrumentalness,  
  a.classificacao_instrumentalness,  
  a.quartil_liveness,  
  a.classificacao_liveness,  
  q_speechiness.quartil_speechiness,  
  q_total_playlists.quartil_total_playlists,
```

```

CASE
  WHEN q_speechiness.quartil_speechiness = 4 THEN "alto"
  WHEN q_speechiness.quartil_speechiness = 3 THEN "medio"
  WHEN q_speechiness.quartil_speechiness = 2 THEN "medio"
  WHEN q_speechiness.quartil_speechiness = 1 THEN "baixo"
  ELSE NULL
END AS classificacao_speechiness,

```

```

CASE
  WHEN q_total_playlists.quartil_total_playlists = 4 THEN "alto"
  WHEN q_total_playlists.quartil_total_playlists = 3 THEN "medio"
  WHEN q_total_playlists.quartil_total_playlists = 2 THEN "medio"
  WHEN q_total_playlists.quartil_total_playlists = 1 THEN "baixo"
  ELSE NULL
END AS classificacao_total_playlists

```

```

FROM `projeto2-lab-456316.proj02.track_spotify` a
LEFT JOIN (
  SELECT track_id, ntile(4) OVER (ORDER BY speechiness) AS quartil_speechiness
  FROM `projeto2-lab-456316.proj02.track_spotify`
) q_speechiness ON a.track_id = q_speechiness.track_id
LEFT JOIN (
  SELECT track_id, ntile(4) OVER (ORDER BY total_playlists) AS quartil_total_playlists
  FROM `projeto2-lab-456316.proj02.track_spotify`
) q_total_playlists ON a.track_id = q_total_playlists.track_id;

```

#fiz esse processo de 2 em 2 até terminar todas as variaveis que eu queria
trabalhar estivessem listadas. essas são as duas ultimas.

```

#salvando arquivo em csv
select
*
from `projeto2-lab-456316.proj02.track_spotify`

```