Ficha Técnica
Projeto de Análise de Dados

1. Informações Gerais

- **Nome do Projeto:** Validação de hipóteses

2. Objetivo do Projeto

Neste marco, vamos explorar um conjunto de dados para **identificar padrões ou características que possam influenciar a popularidade (número de streams) de uma música em plataformas como Spotify, Apple Music e Deezer**. Apesar das várias hipóteses levantadas pela gravadora, ainda não se sabe ao certo quais delas impactam o sucesso medido pelo número de streams.

O **teste de hipóteses** pode ser usado em uma variedade de contextos e mercados para, por exemplo, planejar estratégias de marketing, estabelecer políticas de preços, avaliar investimentos, entender melhor o comportamento do mercado e consumidores. Ou seja, tomar todos os tipos de decisões estratégicas.

3. Fonte dos Dados

track_in_competition
track_in_spotify
track_technical_info

4. Metodologia

- **Ferramentas Utilizadas:** Google Sheets, BigQuery, SQL, Power BI, Python
- **Técnicas de Análise:** Validação de hipóteses
- **Processo:**

a. Processar e preparar a base de dados

Neste marco, trabalharemos com uma ferramenta chamada BigQuery para realizar a etapa de processamento e preparação de dados. A linguagem SQL, utilizada nesta ferramenta, é amplamente utilizada no ambiente de trabalho e um bom analista de dados deve saber como utilizá-la a seu favor.

Conectar/importar dados para outras ferramentas

- Google sheets -> BigQuery

Identificar e tratar valores nulos

- Comandos SQL COUNT, WHERE e IS NULL

Identificar e tratar valores duplicados

- Comandos SQL COUNT, GROUP BY, HAVING

Identificar e tratar dados fora do escopo de análise

- Manipular variáveis que não são úteis para análise através de comandos SQL SELECT EXCEPT

Identificar e tratar dados discrepantes em variáveis categóricas

- Comandos de manipulação de strings LIKE, REGEXP_REPLACE

Identificar e tratar dados discrepantes em variáveis numéricas

- Comandos MAX, MIN e AVG

Verificar e alterar o tipo de dados

- Comando CAST

Criar novas variáveis

- Comando CAST, CONCAT

Unir tabelas

- LEFT JOIN, ON

Construir tabelas auxiliares

- Comando WITH para criar uma tabela temporária

Limpeza realizada por tabela:

1. Tabela *track_in_competition*

<u>Decisões:</u>

- **Nulos:** 50 nulos na coluna *in_shazam_charts*; decidimos manter os nulos que podem ser relevantes mais a frente na análise
- **Duplicados:** N/A

```
SELECT
  in_apple_charts,
  in_apple_playlists,
  in_deezer_charts,
  in_deezer_playlists,
  in_shazam_charts,
  COUNT (*) as quantidade
FROM `laboratoria-projeto-2.projeto2.competition`
GROUP BY
  in_apple_charts,
  in_apple_playlists,
  in_deezer_charts,
  in_deezer_playlists,
  in_shazam_charts
HAVING COUNT (*) > 1;
```

- **Dados fora do escopo de análise:** nenhum dos dados dessa tabela foram considerados como fora do escopo de análise
- **Dados discrepantes em variáveis categóricas:** nenhum dos dados dessa tabela foram considerados como discrepantes
- **Dados discrepantes em variáveis numéricas:** nenhum dos dados numéricos dessa tabela foram considerados como discrepantes

Nenhuma conversão foi necessária para obtenção dos dados

Variavel	Tipo Inicial	Min	Max	AVG
in_apple_playlists	Int	0	672	62
in_apple_charts	Int	0	275	52
in_deezer_playlists	Int	0	12367	363
in_deezer_charts	Int	0	58	2,7
in_shazam_charts	Int	0	1451	59

```
SELECT
  MIN(in_shazam_charts) AS min_in_shazam_charts,
  MAX(in_shazam_charts) AS max_in_shazam_charts,
  AVG(in_shazam_charts) AS med_in_shazam_charts,
  MIN(in_deezer_charts) AS min_in_deezer_charts,
  MAX(in_deezer_charts) AS max_in_deezer_charts,
  AVG(in_deezer_charts) AS med_in_deezer_charts,
  MIN(in_deezer_playlists) AS min_in_deezer_playlists,
  MAX(in_deezer_playlists) AS max_in_deezer_playlists,
  AVG(in_deezer_playlists) AS med_in_deezer_playlists,
  MIN(in_apple_charts) AS min_in_apple_charts,
  MAX(in_apple_charts) AS max_in_apple_charts,
  AVG(in_apple_charts) AS med_in_apple_charts,
  MIN(in_apple_playlists) AS min_in_apple_playlists,
  MAX(in_apple_playlists) AS max_in_apple_playlists,
  AVG(in_apple_playlists) AS med_in_apple_playlists
 FROM `laboratoria-projeto-2.projeto2.competition`
```

2. Tabela *track_in_spotify*

Decisão:

- **Nulos:** N/A
- **Duplicados:** 4 duplicados, com nome de música e artistas iguais, então, optamos por excluí-los

Row	track_name	artist_name	ocorrencias
1	SNAP	Rosa Linn	2
2	SPIT IN MY FACE!	ThxSoMch	2
3	About Damn Time	Lizzo	2
4	Take My Breath	The Weeknd	2

```
SELECT
  track_name,
  artist_s_name,
  COUNT (*) as quantidade
FROM `laboratoria-projeto-2.projeto2.spotify`
GROUP BY
  track_name,
  artist_s_name
HAVING COUNT (*) > 1;
```

```
SELECT *
FROM (
  SELECT *,
              ROW_NUMBER() OVER (PARTITION BY track_name, artist_s_name
ORDER BY track_name) AS quantidade
  FROM `laboratoria-projeto-2.projeto2.spotify`
)
WHERE quantidade = 1;
```

- **Dados fora do escopo de análise:** nenhum dos dados dessa tabela foram considerados como fora do escopo de análise
- **Dados discrepantes em variáveis categóricas:** encontramos alguns dados nas colunas track_name e artist_s_name que possuíam caracteres desconhecidos, resolvemos removê-los

```
SELECT
  track_name,
  artist_s_name,
  REGEXP_REPLACE(track_name, r'[^a-zA-Z0-9]', " ") AS track_name_limpo,
       REGEXP_REPLACE(artist_s_name,    r'[^a-zA-Z0-9]',    "    ")    AS
artist_s_name_limpo

FROM `laboratoria-projeto-2.projeto2.spotify`
```

- **Dados discrepantes em variáveis numéricas:** Um dado alfanumérico foi encontrado na coluna streams_int, necessitando a conversão do tipo de variável

```
SELECT SAFE_CAST (streams AS INT64) AS streams_int

FROM `laboratoria-projeto-2.projeto2.spotify`
```

Variavel	Tipo Inicial	Min	Max	AVG
artist_count	Int	1	8	1,55
released_year	Int	1930	2023	2018
released_month	Int	1	12	6
released_day	Int	1	31	13
in_spotify_playlists	Int	31	52898	5200
in_spotify_charts	Int	0	147	12
streams	String → Int	2762	3703895074	514137424

```
SELECT
  MIN(streams_int) AS min_streams,
  MAX(streams_int) AS max_streams,
  AVG(streams_int) AS med_streams,
  MIN(in_spotify_charts) AS min_in_spotify_charts,
  MAX(in_spotify_charts) AS max_in_spotify_charts,
  AVG(in_spotify_charts) AS med_in_spotify_charts,
  MIN(in_spotify_playlists) AS min_in_spotify_playlists,
  MAX(in_spotify_playlists) AS max_in_spotify_playlists,
  AVG(in_spotify_playlists) AS med_in_spotify_playlists,
  MIN(released_day) AS min_released_day,
  MAX(released_day) AS max_released_day,
  AVG(released_day) AS med_released_day,
  MIN(released_month) AS min_released_month,
  MAX(released_month) AS max_released_month,
  AVG(released_month) AS med_released_month,
  MIN(released_year) AS min_released_year,
  MAX(released_year) AS max_released_year,
  AVG(released_year) AS med_released_year,
  MIN(artist_count) AS min_artist_count,
  MAX(artist_count) AS max_artist_count,
  AVG(artist_count) AS med_artist_count

FROM (
  SELECT
    *,
    SAFE_CAST(streams AS INT64) AS streams_int
  FROM `laboratoria-projeto-2.projeto2.spotify`
)
```

E 4 datas de música foram identificadas como erradas e foram corrigidas:

```sql
CREATE OR REPLACE TABLE `proj02-lab.proj02_dados.track_in_spotify_streams_corrigida` AS
SELECT
  *,
  CASE
    WHEN track_name = 'Riptide' AND artist_name = 'Vance Joy' THEN '2013'
    WHEN track_name LIKE '%Cupid Twin Ver.%' THEN '2023'
    WHEN track_name = 'Sigue' AND artist_name = 'Ed Sheeran, J Balvin' THEN '2022'
    WHEN track_name = 'Agudo Mgi' THEN '2023'
    ELSE released_year
  END AS released_year_corrigido
FROM `proj02-lab.proj02_dados.track_in_spotify_streams_int`;
```

3. Tabela *track_technical_info*

Decisão:

- **Nulos:** 95 nulos na coluna *key*; decidimos manter os nulos que podem ser relevantes mais a frente na análise
- **Duplicados:** 1 duplicado; decidimos manter pois o id é diferente

```sql
SELECT
      bpm,
      key,
      mode,
      danceability__,
      valence__,
      energy__,
      acousticness__,
      instrumentalness__,
      liveness__,
      speechiness__,
      COUNT (*) AS quantidade
FROM `laboratoria-projeto-2.projeto2.technical_info`
GROUP BY

  bpm,
  key,
  mode,
  danceability__,
  valence__,
  energy__,
  acousticness__,
  instrumentalness__,
  liveness__,
  speechiness__
HAVING COUNT (*) > 1;
```

- **Dados fora do escopo de análise:** optamos por tratar 2 colunas fora do escopo de análise (key, mode), porque elas não parecem relevantes para esse projeto

```sql
 SELECT
  *
  EXCEPT(key, mode)
  FROM `laboratoria-projeto-2.projeto2.technical_info`
```

- **Dados discrepantes em variáveis categóricas:** nenhum dos dados dessa tabela foram considerados como discrepantes
- **Dados discrepantes em variáveis numéricas:** nenhum dos dados numéricos dessa tabela foram considerados como discrepantes

Variavel	Tipo Inicial	Min	Max	AVG
bpm	Int	65	206	122
danceability__	Int	23	96	67
valence__	Int	4	97	51
energy_	Int	9	97	64
acousticness__	Int	0	97	27
instrumentalness__	Int	0	1	1,6
liveness__	Int	3	97	18
speechiness__	Int	2	64	10

```
SELECT
  MIN (bpm) AS min_bpm,
  MAX (bpm) AS max_bpm,
  AVG (bpm) AS med_bpm,
  MIN (danceability__) AS min_danceability,
  MAX (danceability__) AS max_danceability,
  AVG (danceability__) AS med_danceability,
  MIN (valence__) AS min_valence,
  MAX (valence__) AS max_valence,
  AVG (valence__) AS med_valence,
  MIN (energy__) AS min_energy,
  MAX (energy__) AS max_energy,
  AVG (energy__) AS med_energy,
  MIN (acousticness__) AS min_acousticness,
  MAX (acousticness__) AS max_acousticness,
  AVG (acousticness__) AS med_acousticness,
  MIN (instrumentalness__) AS min_instrumentalness,
  MAX (instrumentalness__) AS max_instrumentalness,
  AVG (instrumentalness__) AS med_instrumentalness,
  MIN (liveness__) AS min_liveness,
  MAX (liveness__) AS max_liveness,
  AVG (liveness__) AS med_liveness,
  MIN (speechiness__) AS min_speechiness,
  MAX (speechiness__) AS max_speechiness,
  AVG (speechiness__) AS med_speechiness
FROM `laboratoria-projeto-2.projeto2.technical_info`
```

- Novas Variáveis:
  - Data completa (aaaa-mm-dd):

```
SELECT
  DATE(
    CONCAT(
      CAST(released_year AS STRING), "-",
      LPAD(CAST(released_month AS STRING), 2, "0"), "-",
      LPAD(CAST(released_day AS STRING), 2, "0")
    )
  ) AS data
FROM `laboratoria-projeto-2.projeto2.spotify`
```

- Soma das playlists:

```
SELECT
  s.track_id,
  s.in_spotify_playlists,
  c.in_deezer_playlists,
  c.in_apple_playlists,
   s.in_spotify_playlists + c.in_deezer_playlists + c.in_apple_playlists
AS total_playlists
   FROM `laboratoria-projeto-2.projeto2.spotify` s
JOIN
  `laboratoria-projeto-2.projeto2.competition` c

ON

  s.track_id = c.track_id
```

- Junção das tabelas:

  Para realização da junção das tabelas utilizamos o LEFT OUTER JOIN

```
SELECT
  spotify.*,
  competition.in_apple_charts,
  competition.in_shazam_charts,
  competition.in_deezer_charts,
  technical.bpm,
  technical.danceability__,
  technical.valence__,
  technical.energy__,
  technical.acousticness__,
  technical.instrumentalness__,
  technical.liveness__,
  technical.speechiness__
FROM `laboratoria-projeto-2.projeto2.spotify_tratado` AS spotify
LEFT OUTER JOIN `laboratoria-projeto-2.projeto2.competition_tratado` AS
competition
  ON spotify.track_id = competition.track_id
LEFT OUTER JOIN `laboratoria-projeto-2.projeto2.technical_info_tratado`
AS technical
  ON spotify.track_id = technical.track_id
WHERE spotify.quantidade < 2;
```

Tabela auxiliar utilizando WITH:

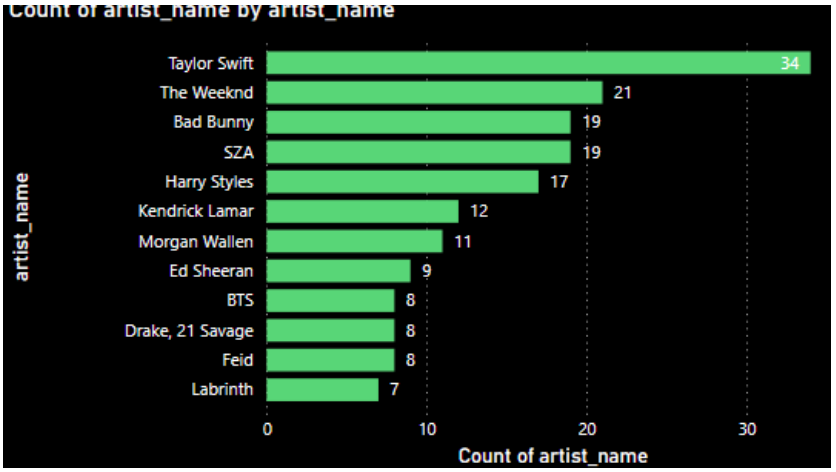
```
SELECT
WITH teste AS (
SELECT
    artist_s_name_limpo,
    COUNT(*) AS total_musicas
  FROM `laboratoria-projeto-2.projeto2.dados_consolidados`
  GROUP BY artist_s_name_limpo
)
SELECT
  dados.*,
  teste.total_musicas
FROM `laboratoria-projeto-2.projeto2.dados_consolidados` AS dados

LEFT JOIN teste
ON dados.artist_s_name_limpo = teste.artist_s_name_limpo
```

b. Fazer uma análise exploratória

A análise exploratória de dados (EDA) é uma fase fundamental na compreensão do conjunto de dados, e ferramentas como Power BI e o BigQuery desempenham um papel crucial neste processo.

Agrupar dados de acordo com variáveis categóricas

- Agrupamos variáveis categóricas para visualizá-las com tabelas matriciais e gráficos de barras para, assim, analisar a relação entre variáveis categóricas e numéricas, como quantidade de músicas por artista, quantidade de músicas lançadas por ano, streams por artista e quantidade de playlists que uma música está presente.

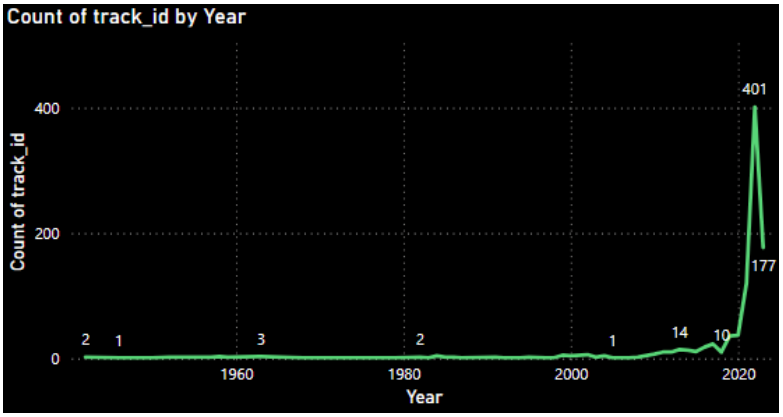


Aplicar medidas de tendência central

- Aplicamos medidas de tendência central nas variáveis *streams* e *total_playlists,* gerando tabelas com min/max valores, média, mediana, e desvio padrão para cada uma das variaveis;

Visualizar o comportamento dos dados ao longo do tempo

- Visualizamos o comportamento dos dados através do tempo por meio de gráficos de linhas, como para analisar a quantidade de músicas lançadas por ano, e a quantidade de streams no spotify por ano



Calcular quartis, decis ou percentis

- Criamos categorias de quartis para variáveis de características no BigQuery

```
WITH Quartiles AS (
  SELECT
  streams,
   NTILE(4) OVER(ORDER BY streams) AS quartile_streams
  FROM
 `proj02-lab.proj02_dados.view_tracks_complete_dataset`
)
SELECT
a.*,
Quartiles.quartile_streams,
  IF
  (Quartiles.quartile_streams = 4, "alto", "baixo")
  AS
  categoria_quartile
  FROM
`proj02-lab.proj02_dados.view_tracks_complete_dataset` a
LEFT JOIN
Quartiles
ON
a.streams=Quartiles.streams
```

Calcular correlação entre variáveis

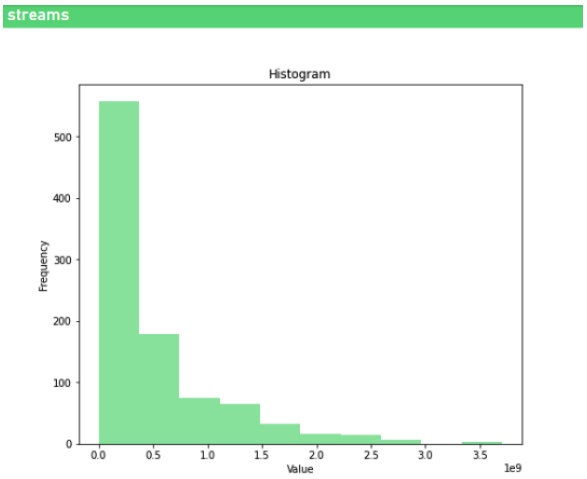
- Calculamos a correlação no BigQuery via comando CORR para analisar a relação entre diferentes variáveis, como bpm e número de streams, danceability da música e o número de streams, participação da música em playlists e o número de streams, entre outros

```
SELECT CORR(in_spotify_charts, in_apple_charts)
 AS correlation_value
FROM `proj02-lab.proj02_dados.tracks_complete_dataset`
```

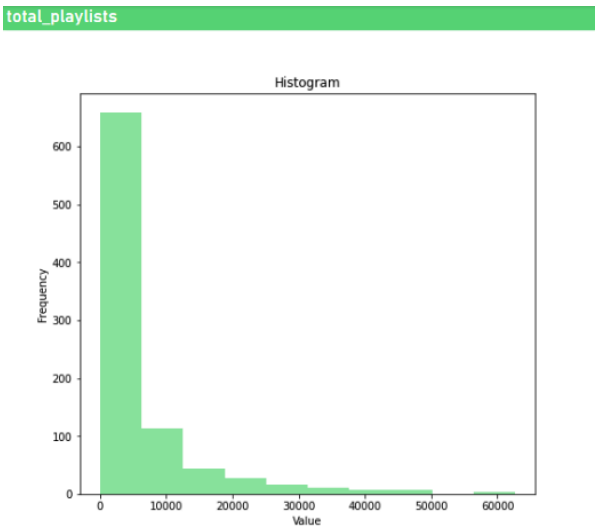
Phyton

- Histogramas gerados a partir do código Python para visualizar a distribuição:

```
import matplotlib.pyplot as plt
import pandas as pd

# Obtenha os dados do Power BI - você só preciso alterar essas informações de todo o code
data = dataset[['streams']]

# Crie o histogram
plt.hist(data, bins=10, color='#6DD87D', alpha=0.7)
plt.xlabel('Value' )
plt.ylabel('Frequency')
plt.title('Histogram')

# Mostre o histogram
plt.show()
```



- Assimetria, distribuição não-normal,
- Poucas músicas com streams extremamente altos (hits/virais)
- Grande maioria com streams modestos ou mínimos
- Assimetria
- Outlier de 1 bilhão representa um mega-hit
- Valores entre 0-3.5 provavelmente representam músicas com pouca reprodução



Este é um comportamento muito típico em dados culturais e de consumo:

- Muitos itens com baixa popularidade (long tail)
- Poucos itens muito populares (hits)

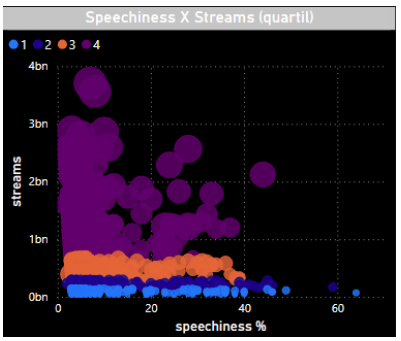
O gráfico reforça a existência de uma distribuição de Pareto:

- Cerca de 20% das músicas devem estar em 80% das playlists.

c. Aplicar técnica de análise

Refere-se à avaliação de declarações ou suposições sobre uma população, ou fenômeno, através da análise dos dados coletados de uma amostra dessa população. Em essência, é tentar determinar se as suposições feitas sobre uma população são apoiadas por evidências empíricas fornecidas pelos dados.

- Análise das categorias criadas através dos quartis para as características da música em relação à variável streams.



- Validar as hipóteses levantadas através de correlação e gráfico de dispersão

Validação de hipóteses

Hipótese	Validação	Coeficiente de correlação de Pearson
Músicas com BPM (Batidas Por Minuto)	Não confere	**-0.00797112408...** Esse coeficiente é **praticamente zero**, indicando **nenhuma correlação**

Validação de hipóteses

Hipótese	Validação	Coeficiente de correlação de Pearson
mais altos fazem mais sucesso em termos de número de streams no Spotify.		**linear** entre o BPM das músicas e o número de streams. Ou seja, **o BPM não influencia significativamente o sucesso das músicas** em termos de streams no Spotify.
As músicas mais populares no ranking do Spotify também possuem um comportamento semelhante em outras plataformas, como a Deezer.	Validado	**0.550822431355...** Esse valor representa uma **correlação positiva moderada**. Indica que, de forma geral, **músicas bem posicionadas no Spotify tendem também a estar bem colocadas na Deezer**. Assim, a hipótese é **parcialmente confirmada**: há um padrão de comportamento semelhante entre as duas plataformas, mas não é uma correlação perfeita.
A presença de uma música em um maior número de playlists está correlacionada com um maior número de streams	Validado	**0.7810034570444** O coeficiente de **0,78** indica uma **correlação positiva forte** entre o número de playlists em que uma música aparece e o total de streams. O gráfico confirma essa relação: à medida que a quantidade de playlists aumenta, também aumenta o volume de streams, como mostra a **linha de tendência ascendente**. Embora haja alguma dispersão, a **maioria das músicas com muitos streams aparece em um número elevado de playlists**. Ou seja, estar em mais playlists **potencializa a exposição** e favorece o acúmulo de streams.
Artistas com um maior número de músicas no Spotify têm mais streams.	Validado	**0.77894414453142** O coeficiente de **0,77** indica uma **correlação positiva forte** entre o número de músicas lançadas por um artista e o total de streams acumulados Ainda que existam exceções, artistas com poucas músicas e muitos streams ou

Validação de hipóteses		
Hipótese	Validação	Coeficiente de correlação de Pearson
		artistas com muitas músicas e menos streams, a **tendência geral é clara**: mais músicas, mais streams.
As características da música influenciam o sucesso em termos de número de streams no Spotify.	Não confere	Os gráficos indicam que características como energia, dançabilidade ou acústica não determinam o número de streams de forma consistente. Outros fatores podem ser mais relevantes para o sucesso no Spotify. Portanto, a hipótese não é validada por esses dados.

d. Resumir informações em um dashboard ou relatório

O Power BI se tornou uma das soluções líderes do mercado para criar relatórios e painéis interativos que permitem que as organizações tomem decisões informadas.

Assim, foi criado um dashboard desse projeto para melhor visualização dos resultados.

e. Apresentar resultados

Vídeo de 5 minutos gravado para apresentação dos resultados dessa análise.

5. Cronograma

Etapa	Status	Sprints
Processar e preparar a base de dados	Feito	2
Fazer uma análise exploratória	Feito	1
Aplicar técnica de análise	Feito	1
Resumir informações em um dashboard	Feito	1/2
Apresentar resultados	Feito	1/2

6. Recursos Necessários

- Google Sheets para extração dos dados
- Linguagem SQL para tratamento dos dados
- BigQuery como plataforma de tratamento de dados usando SQL
- Power BI para visualização e análise exploratória dos dados por meio de tabelas e gráficos
- Python para geração de histogramas com código de programação

7. Entregas

- Slide da apresentação
- Vídeos de apresentação

8. Resultados e Conclusões

A análise dos dados permitiu validar hipóteses relacionadas ao desempenho das músicas.

Foi observado que as seguintes estratégias contribuem significativamente para o aumento do alcance e visibilidade:

- Lançamento de um maior número de faixas
- Presença em diversas playlists
- Disponibilização das músicas em múltiplas plataformas (Spotify, Dezer, Apple, Shazam)

Por outro lado, características específicas das músicas, como estilo, duração ou estrutura, não demonstraram correlação direta com o sucesso em número de streams.

Com base nos resultados, recomenda-se focar em lançamentos frequentes, divulgação ativa em playlists variadas e distribuição ampla em várias plataformas, garantindo que o conteúdo esteja acessível ao maior número possível de ouvintes.

Ainda que as características musicais não estejam diretamente ligadas ao sucesso, é fundamental manter a consistência e qualidade, buscando atrair curadores, conquistar espaço em playlists e incentivar o engajamento do público.