Relatório Técnico - Análise de Letras de Músicas

Autor: Hugo de Souza Almeida

1. Introdução

Este relatório descreve o desenvolvimento de um projeto de análise textual aplicado a uma base de letras de músicas. O objetivo principal foi aplicar técnicas de pré-processamento, análise de polaridade e análise exploratória (EDA) para compreender padrões linguísticos, emocionais e temporais nas composições musicais. A tarefa escolhida, conforme proposto, foi a análise de polaridade das letras ao longo do tempo por artista.

2. Desafios Técnicos e Estratégias

Inicialmente, enfrentei um desafio técnico relacionado ao volume da base de dados. O ambiente local apresentou limitações de memória RAM, o que dificultou o uso de ferramentas como o Jupyter Notebook. Como alternativa, foi utilizado o Google Colab, que oferece mais recursos computacionais para processamento em nuvem, porém o código foi testado no jupyter e com base de dados bastante reduzida funcionou normalmente.

Além disso, no começo havia tentado realizar o processamento de dados em chunks independentes, porém estava demorando tempo demais. Assim, optei por carregar uma amostra representativa de 100.000 registros, suficiente para garantir análises estatísticas robustas sem comprometer o desempenho.

Outros desafios que enfrentei serão explicados nos outros tópicos como a questão dos dados que não eram música, o tipo de ferramenta apropriado para a tarefa escolhida e a análise dos resultados.

3. Ferramentas e Bibliotecas Utilizadas

pandas — manipulação e análise de dados em formato tabular (DataFrame)

matplotlib — criação de gráficos e visualizações estáticas

seaborn — geração de gráficos estatísticos com apelo visual

re — expressões regulares para identificação de padrões textuais

vaderSentiment — análise de sentimentos em textos curtos (ótimo desempenho em inglês, com suporte imediato e sem necessidade de treinamento)

4. Pré-processamento dos Dados

Após a leitura dos dados, descartei colunas irrelevantes para a tarefa proposta. A análise inicial revelou a presença de muitos registros não relacionados a músicas, como trechos de livros e discursos. Estes dados se concentravam majoritariamente na tag 'misc'.

No entanto, observei que parte das entradas em 'misc' eram de fato músicas. Para garantir a qualidade da análise, realizei um processo de filtragem estruturado:

Dividi o DataFrame em dois subconjuntos: um contendo apenas os registros marcados como 'misc' (df_misc) e outro com os demais registros, presumivelmente já classificados corretamente como músicas (df_musicas). Para identificar quais registros em 'misc' eram realmente músicas, criei uma função denominada contem_estruturas_de_musica. Essa função verifica a presença de padrões estruturais típicos de letras de música, como [intro], [verse], [chorus], entre outros, dentro do campo lyrics. Registros que continham esses elementos foram marcados como potenciais músicas.

Os registros identificados como músicas legítimas dentro de 'misc' foram separados em um novo DataFrame (df_misc_musicas). Em seguida, uni esses registros ao DataFrame

original de músicas (df_musicas), formando um conjunto consolidado (df_todas_musicas) que representa de modo mais fiel as músicas presentes na base de dados.

Após a união, removi possíveis duplicatas considerando os campos 'title', 'artist', 'year' e 'lyrics'. Finalmente, excluí a coluna auxiliar utilizada para identificação (parece_musica) e conferi o resultado final, contabilizando quantas músicas foram recuperadas da tag 'misc' e o total de músicas na amostra.

5. Análise Exploratória dos Dados (EDA)

Com a base limpa e unificada, foram realizadas análises para compreender melhor os dados:

- Identificação das colunas e seus valores únicos

- Frequência de idiomas presentes

- Distribuição de músicas por tag

- Top 10 artistas mais recorrentes

- Distribuição de músicas ao longo dos anos

- Distribuição de visualizações (views)

Essas análises permitiram compreender tendências temporais, concentração de conteúdo e perfis de popularidade.

6. Limpeza dos Dados

Realizei uma limpeza dos dados, principalmente em ‘lyrics’ para preparar o df para a análise de sentimentos. O código faz uma limpeza minuciosa no DataFrame de letras de músicas, removendo linhas incompletas, padronizando o texto das letras, eliminando anotações estruturais e corrigindo espaçamentos. O resultado é um conjunto de dados limpo, consistente e pronto para análises textuais.

7. Análise de Polaridade

Para avaliar o sentimento presente nas letras das músicas, utilizei a biblioteca VADER, amplamente reconhecida para análise de sentimentos em textos curtos. O processo foi dividido em duas etapas principais: obtenção da pontuação de sentimento e categorização da polaridade.

- Cálculo da pontuação de sentimento

  Inicialmente, instanciei o analisador de sentimentos da VADER (SentimentIntensityAnalyzer). Para cada letra de música, apliquei uma função chamada get_sentiment_score, que calcula a pontuação composta de sentimento (compound score), variando de -1 (sentimento muito negativo) até +1 (sentimento muito positivo). Letras vazias ou nulas foram tratadas e marcadas como None.

- Classificação da polaridade

  Com a pontuação composta em mãos, defini faixas para categorizar cada letra em três classes de polaridade:

  Positivo: score ≥ 0.5

  Negativo: score ≤ -0.5

  Neutro: score entre -0.5 e 0.5

  Para isso, utilizei a função get_polaridade_precisa, que recebe o score e retorna a polaridade correspondente. O resultado foi armazenado em uma nova coluna chamada polaridade.

Ao final deste processo, cada música passou a ter associada uma pontuação de sentimento e uma classificação de polaridade (positivo, negativo ou neutro), tornando possível investigar padrões emocionais nas composições, analisar tendências ao longo do tempo e comparar artistas ou estilos musicais sob a ótica do sentimento. Além disso, por meio dos gráficos e dos resultados apresentados, foi possível identificar os artistas

que apresentaram maior variação de polaridade em suas letras ao longo dos anos, evidenciando mudanças significativas em suas produções. Para ilustrar essa análise de forma mais aprofundada, selecionei um dos artistas com maior número de músicas na base e demonstrei como a polaridade das composições desse artista evoluiu de acordo com o período de lançamento, permitindo visualizar possíveis transformações em sua expressão artística e na carga emocional transmitida ao longo da carreira.

8. Conclusão

O projeto foi bem-sucedido na extração e estruturação de uma base textual complexa, permitindo análises relevantes a partir de dados inicialmente misturados e ruidosos. A recuperação de músicas na tag 'misc' foi um ponto chave para garantir qualidade à base final. A análise de polaridade realizada fornece insights valiosos sobre a tonalidade emocional das composições.

Esse processo foi fundamental para aumentar a precisão da análise, evitando tanto a exclusão de músicas reais quanto a inclusão de textos irrelevantes, e garantindo que as etapas posteriores fossem realizadas sobre um conjunto de dados realmente representativo do universo musical. Essa abordagem permitiu uma avaliação objetiva e padronizada do conteúdo emocional das letras, viabilizando análises quantitativas e comparativas de polaridade para todo o conjunto de músicas limpas da base.

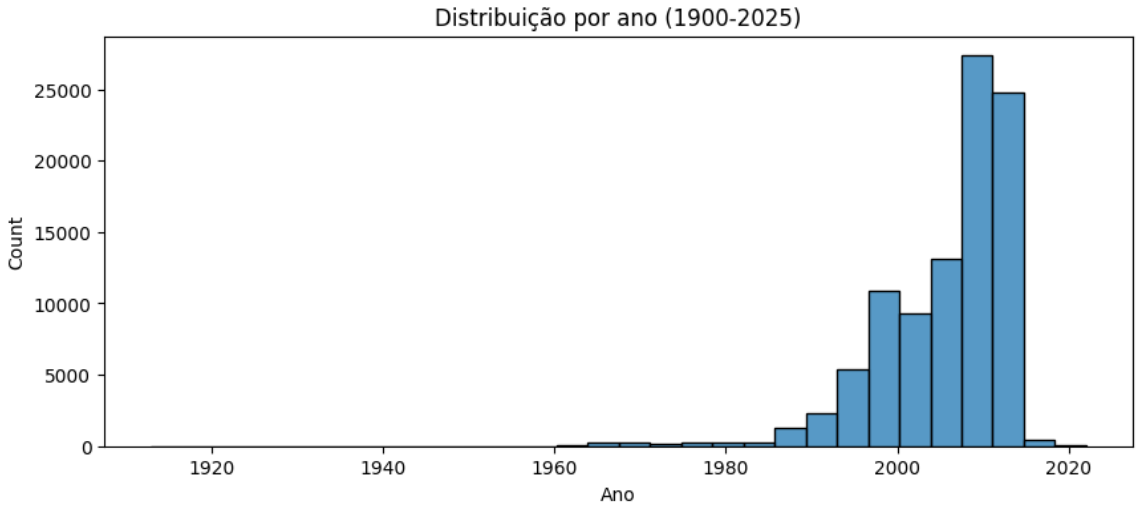
Além disso, a metodologia adotada demonstrou a importância do pré-processamento rigoroso em projetos de ciência de dados textuais, especialmente quando se trata de bases heterogêneas. O uso da biblioteca VADER foi apropriado para a tarefa, possibilitando a categorização eficiente das letras em sentimentos positivos, negativos ou neutros, e permitindo a identificação de tendências temporais e variações entre artistas. Ao analisar os resultados percebi que a correlação entre polaridade e views se aproxima cada vez mais de 0 conforme a base de dados aumenta, concluindo assim que a relação entre sentimentos “negativos”, “positivos” ou “neutros” das letras não tem relação com a quantidade de views.

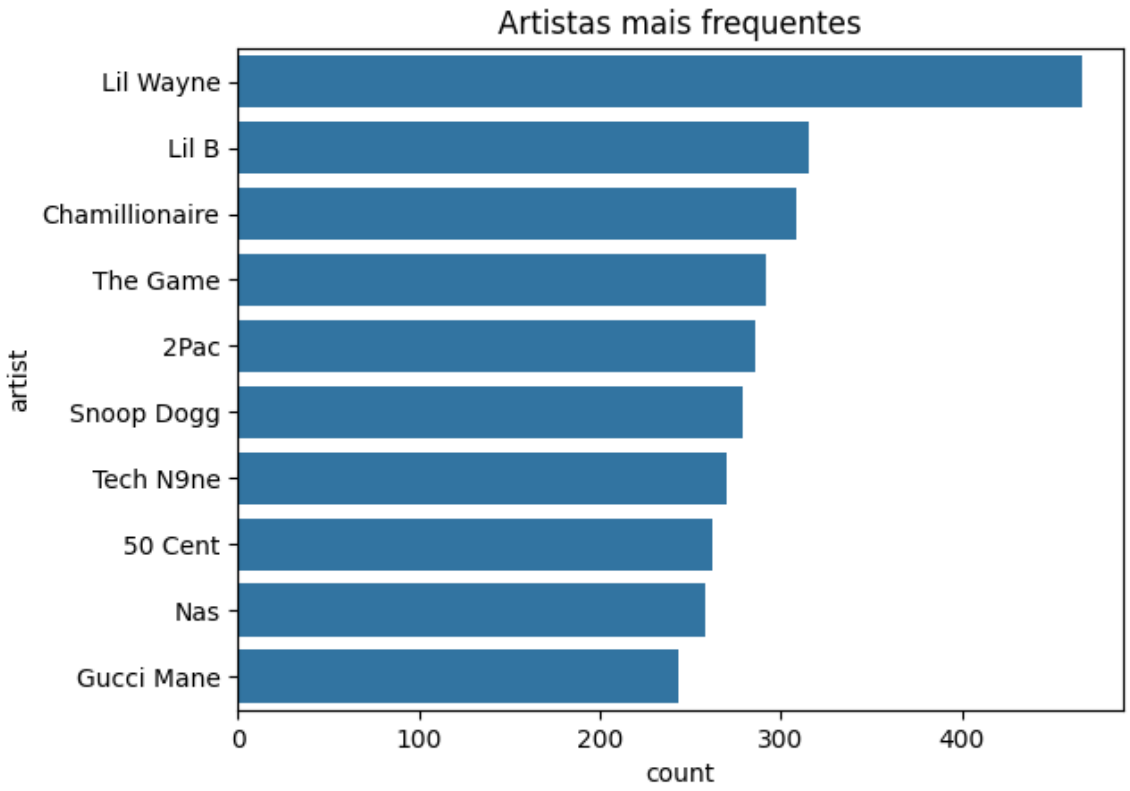
Acredito que quanto mais dados mais sólidos serão os resultados e apesar dos avanços, a ferramenta VADER é boa na minha opinião, ela não consegue apenas definir se uma música é “negativa” ou “positiva”, mas reconheço limitações no uso do VADER para

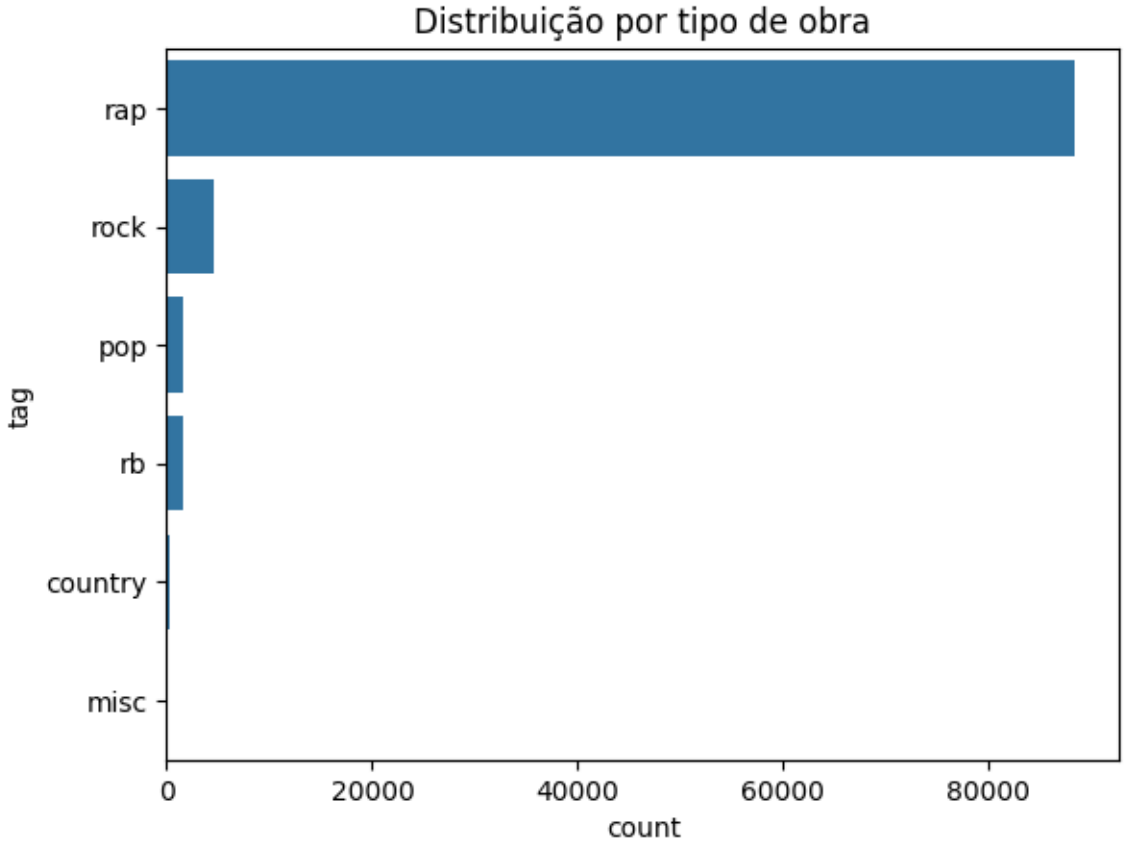
músicas em outras línguas que não a inglesa por exemplo. Porém fiz alguns testes por fora e tiveram resultados plausíveis quanto a músicas dessa base de dados que são de outros idiomas, mas ainda assim foram inferiores as da linguagem inglesa. Além de lacunas na análise de sentimentos mais complexos, como ironia ou ambiguidade.

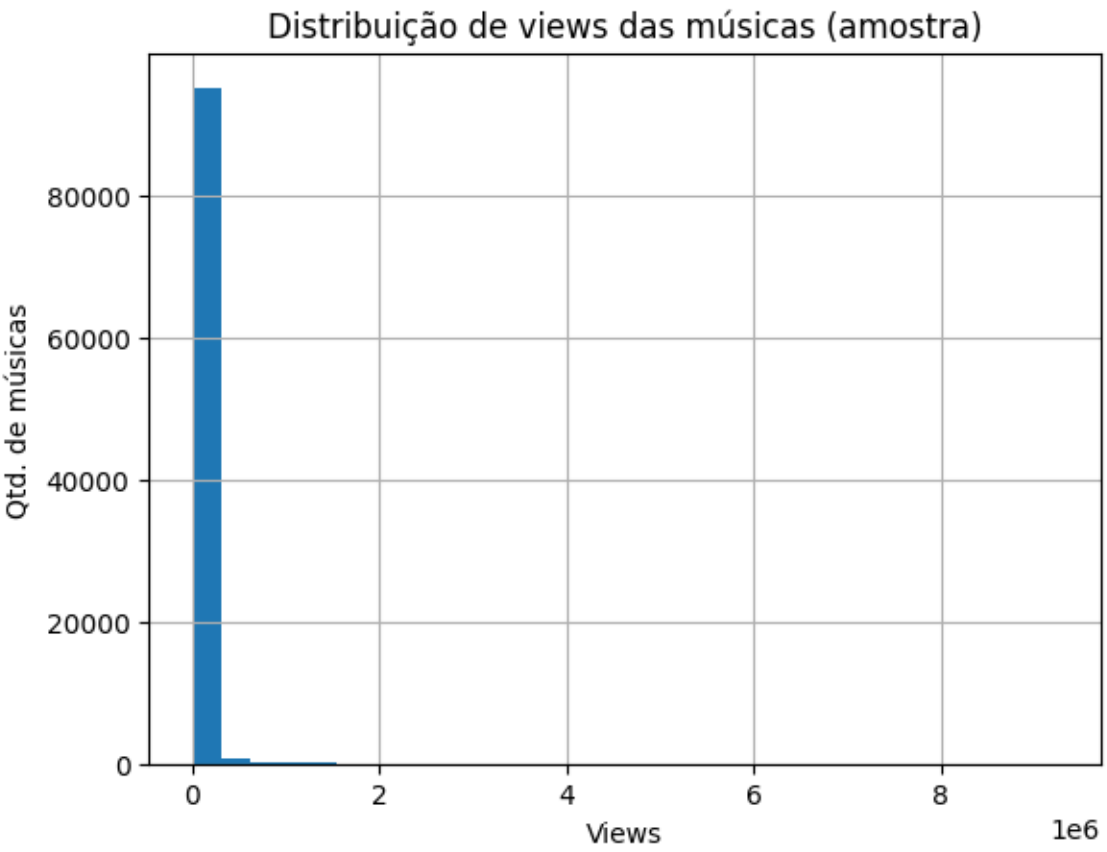
Por fim, acredito ter obtido os resultados desejados, deixei alguns comentários no código para melhor compreensão e gráficos simples de serem interpretados. Abaixo finalizo este relatório com alguns dos gráficos dentro dos “100000” primeiros dados que analisei.

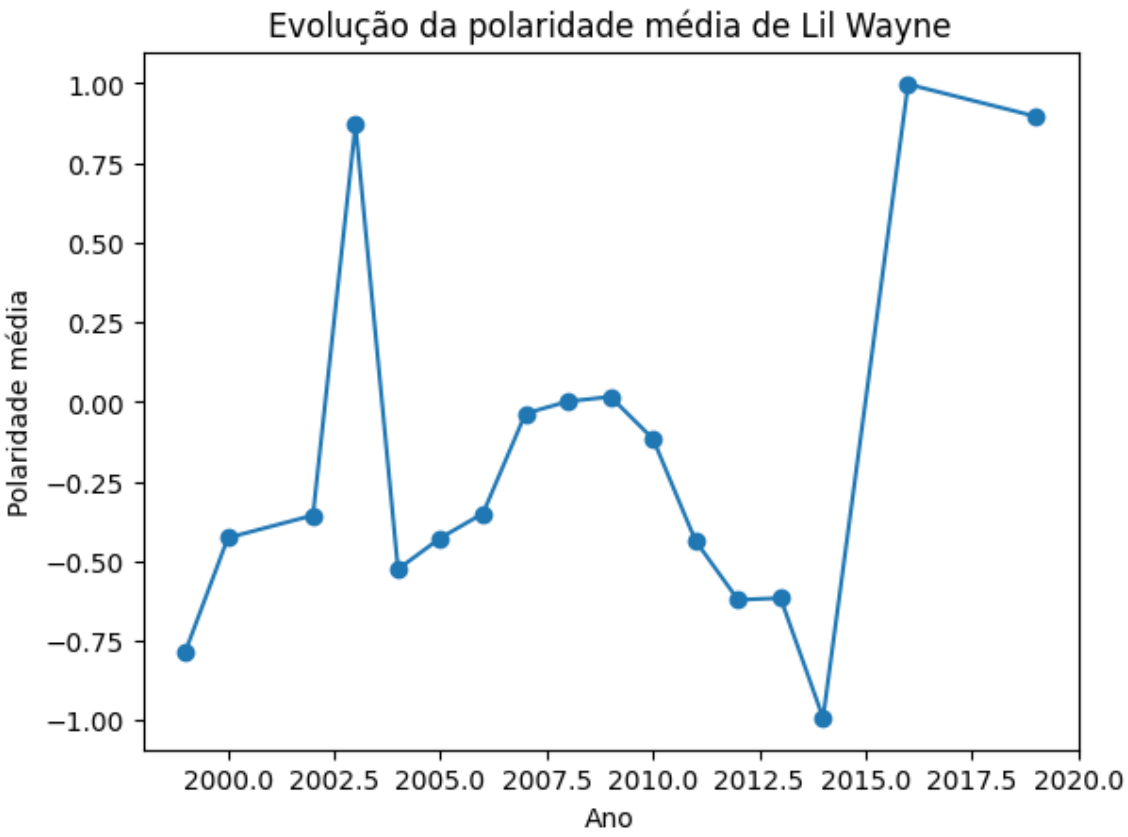
9. Anexos (Gráficos)

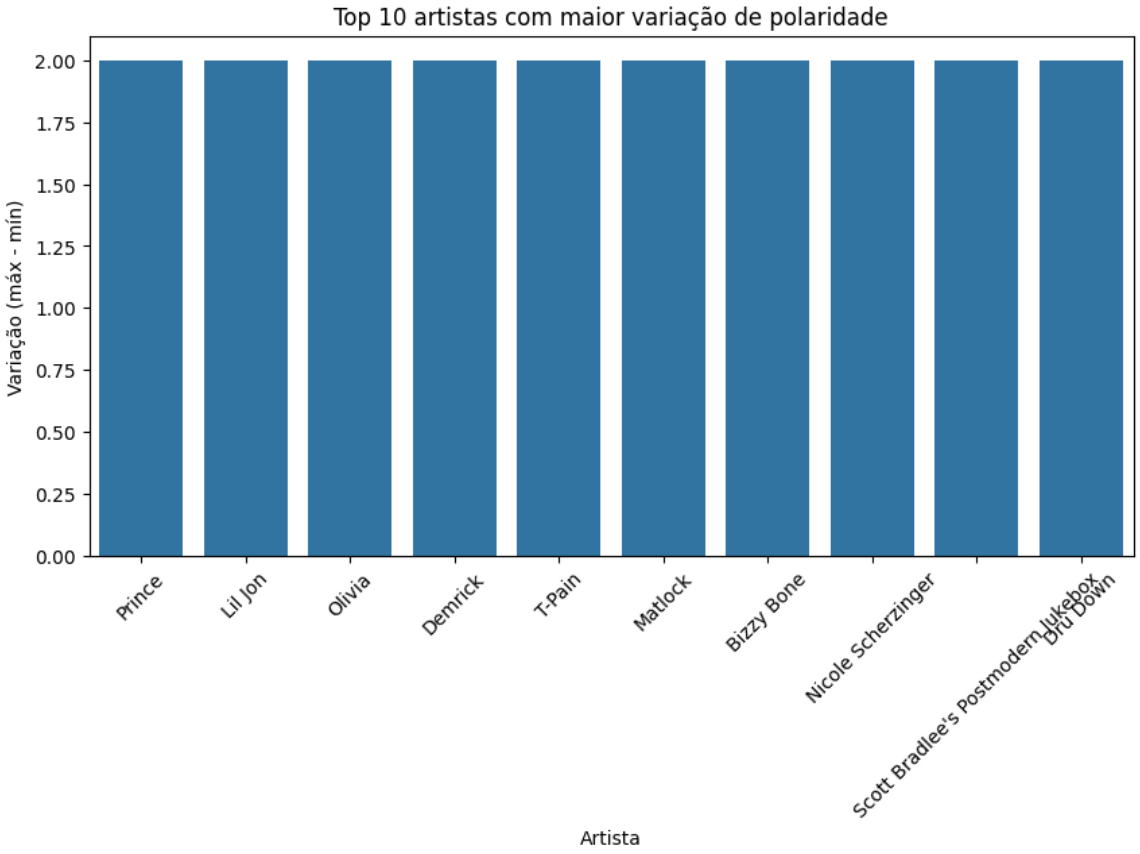


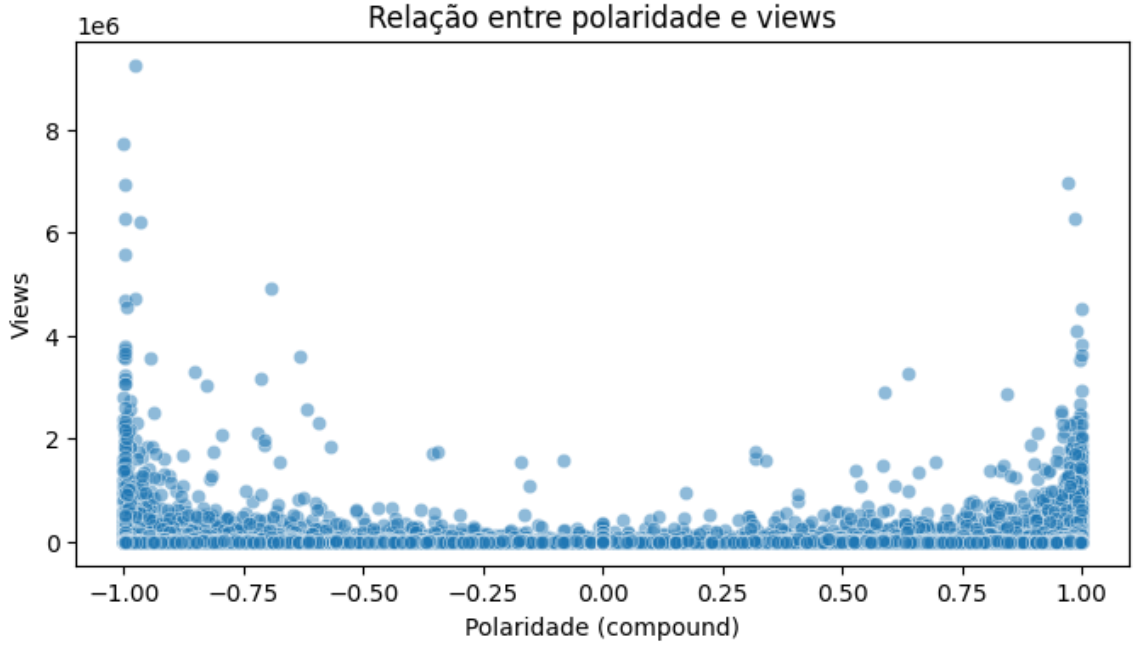

Artistas mais frequentes
Lil Wayne
Lil B
Chamillionaire
The Game
2Pac
Snoop Dogg
Tech N9ne
50 Cent
Nas
Gucci Mane
artist
0
100
200
300
400
count

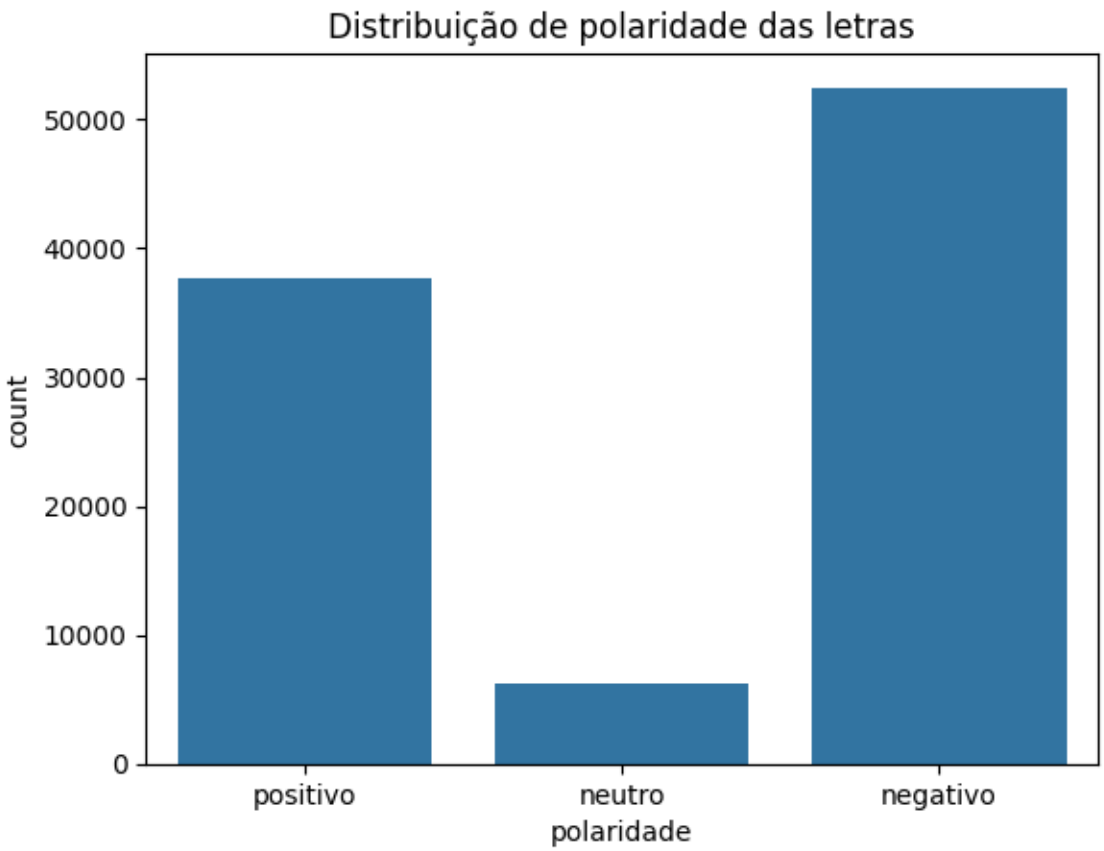

Distribuição por tipo de obra
rap
rock
pop
rb
country
misc
tag
0
20000
40000
60000
80000
count

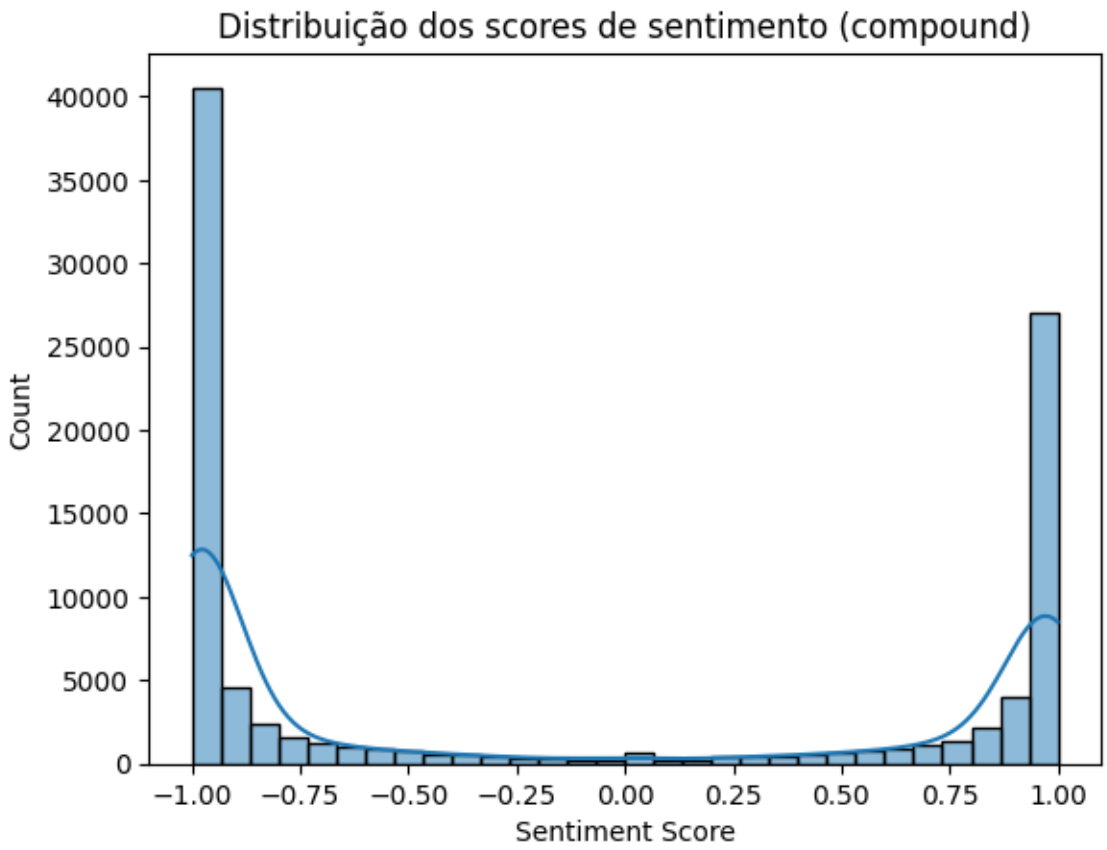

Distribuição de views das músicas (amostra)
Qtd. de músicas
80000
60000
40000
20000
0
0
2
4
6
8
Views
1e6


Evolução da polaridade média de Lil Wayne
Polaridade média
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
2015.0
2017.5
2020.0
Ano


Top 10 artistas com maior variação de polaridade
Variação (máx - mín)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Prince
Lil Jon
Olivia
Demrick
T-Pain
Matlock
Bizzy Bone
Nicole Scherzinger
Scott Bradlee's Postmodern Jukebox
Dru Down
Artista


Relação entre polaridade e views
1e6
Views
8
6
4
2
0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Polaridade (compound)


Distribuição de polaridade das letras
50000
40000
30000
20000
10000
0
count
positivo
neutro
negativo
polaridade


Distribuição dos scores de sentimento (compound)
Count
Sentiment Score
40000
35000
30000
25000
20000
15000
10000
5000
0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00