

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS DE RIO PARANAÍBA
SISTEMAS DE INFORMAÇÃO

VINÍCIUS SILVA GRILO

**ANÁLISE DE SENTIMENTOS DE LETRAS DE MÚSICAS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM
NATURAL**

RIO PARANAÍBA

2022

VINÍCIUS SILVA GRILO

ANÁLISE DE SENTIMENTOS DE LETRAS DE MÚSICAS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Monografia apresentada à Universidade Federal de Viçosa como parte das exigências para a aprovação na disciplina Trabalho de Conclusão de Curso II

Orientador: Prof. Dr. Clausius Duque Gonçalves Reis

RIO PARANAÍBA

2022

VINÍCIUS SILVA GRILO

**ANÁLISE DE SENTIMENTOS DE LETRAS DE MÚSICAS
UTILIZANDO PROCESSAMENTO DE LINGUAGEM
NATURAL**

Monografia apresentada à Universidade Federal de Viçosa como parte das exigências para a aprovação na disciplina Trabalho de Conclusão de Curso II

APROVADA EM: 15 de julho de 2022

**Prof. Dr. Clausius Duque Gonçalves
Reis**
Orientador
UFV-CRP

Prof. Dr. João Fernando Mari
UFV-CRP

Prof. Dr. Guilherme de Casto Pena
UFV-CRP

RIO PARANAÍBA
2022

“It’s a long way to the top, if you wanna rock ‘n’ roll.”
(AC/DC, 1975)

Resumo

A inteligência computacional de uma máquina digital é estudada desde a década de 50, com as pesquisas de Alan Turing. Elas deram embasamento para o que é conhecido hoje como Processamento de Linguagem Natural (PLN), a capacidade de um computador compreender e analisar a língua humana. Por meio dela, é possível extrair e gerar informações relevantes, como a Análise de Sentimentos, uma subárea do PLN. Uma vez que letras de músicas são uma forma de autores expressarem sensibilidade, é possível analisar e extrair o sentimento destes textos usando métodos de Aprendizado de Máquina. Letras de músicas podem possuir bastante sarcasmo e ironia, e por isso também foi realizada a análise da polaridade das palavras contidas na letra. Foi utilizado um método de classificação *Bayesiano* para realizar a Análise de Sentimentos e uma base de conhecimento de palavras para extrair a polaridade de cada uma. Como os resultados dessas análises são comumente numéricos, é possível aplicar técnicas de Visualização de Dados para auxiliar na compreensão deles. Com isso, foi construída uma ferramenta que faz a coleta, análise e visualização dos sentimentos de letras de músicas do gênero *Heavy Metal*, que possuem particularidades sobre o sentimento que é expressado por elas.

Palavras-chaves: Processamento de Linguagem Natural, Análise de Sentimentos, Aprendizado de Máquina, Visualização de Dados.

Abstract

The computational intelligence of a digital machine has been studied since the 1950s, with research by Alan Turing. They underpinned what is known today as Natural Language Processing (NLP), the capability of a computer to comprehend and analyze human language. With it, it is possible to extract and generate relevant information, such as Sentiment Analysis, a field of NLP . Since song lyrics are a way for authors to express sensitivity, it is possible to analyze and extract the sentiment from these texts using Machine Learning methods. Song lyrics can have a lot of sarcasm and irony, and that's why the analysis of the polarity of the words contained in the lyrics was also carried out. A *Bayesian* classification method was used to perform Sentiment Analysis and a word knowledge base to extract the polarity of each one. As the results of these analysis are commonly numerical, it is possible to apply Data Visualization techniques to aid in their understading. With this, a tool was built that collects, analyzes and generates the visualization of lyric sentiment from the Heavy Metal genre, which have particularities about the sentiment that is expressed by them.

Key-words: Natural Language Processing, Sentiment Analysis, Machine Learning, Data Visualization.

Lista de ilustrações

Figura 1 – Estágios da análise no PLN	16
Figura 2 – Visão do classificador <i>Naive Bayes</i>	17
Figura 3 – Exemplo de retorno de uma consulta utilizando a API do Deezer . . .	20
Figura 4 – Exemplo de retorno de uma consulta utilizando a API do Vagalume . .	22
Figura 5 – Técnicas de Visualização de Dados comumente usadas	23
Figura 6 – Exemplo de visualização de respostas de uma população por área . . .	25
Figura 7 – Exemplo de visualização de funções matemáticas	26
Figura 8 – Código fonte que gera a visualização de funções matemáticas	26
Figura 9 – Painel principal de um sistema de Visualização de Dados	29
Figura 10 – Conjunto de dados criado	37
Figura 11 – Tela inicial do sistema final	38
Figura 12 – Filtro com todos os períodos	39
Figura 13 – Filtro de 1984 até 1997	40
Figura 14 – Nuvens de palavras e gráficos das polaridades	41
Figura 15 – Análise individual	42
Figura 16 – Estrofe da música “ <i>It’s Still Rock And Roll To Me</i> ”, de Billy Joel . . .	43
Figura 17 – Análise de Slipknot - <i>Psychosocial</i>	45
Figura 18 – Análise de Megadeth - <i>Symphony Of Destruction</i>	46
Figura 19 – Análise de Ozzy Osbourne - <i>Hellraiser</i>	47
Figura 20 – Análise de Slipknot - <i>Left Behind</i>	48

Lista de tabelas

Tabela 1 – Parâmetros de consulta da API do Vagalume	21
--	----

Lista de abreviaturas e siglas

UFV	Universidade Federal de Viçosa
CRP	<i>Campus</i> Rio Paranaíba
IA	Inteligência Artificial
PLN	Processamento de Linguagem Natural
NLP	<i>Natural Language Processing</i>
NLTK	<i>Natural Language Toolkit</i>
API	<i>Application Programming Interface</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
URL	<i>Uniform Resource Locator</i>
HTML	<i>Hypertext Markup Language</i>
CSS	<i>Cascade Style Sheet</i>
CSV	<i>Comma-separated Values</i>

Sumário

1	Introdução	11
1.1	Descrição do Problema	11
1.2	Motivações e Desafios	12
1.3	Objetivo Geral	13
1.4	Objetivos Específicos	13
1.5	Organização do Texto	13
2	Referencial Teórico	15
2.1	A Abordagem Clássica do Processamento de Linguagem Natural	15
2.2	Análise de Sentimentos	16
2.2.1	Classificador <i>Naive Bayes</i>	16
2.3	Análise de Polaridade	18
2.4	<i>Heavy Metal</i>	18
2.5	Coleta de Letras de Músicas	19
2.5.1	API do Deezer	19
2.5.2	API do Vagalume	20
2.6	Visualização de Dados	22
3	Trabalhos Relacionados	27
4	Métodos	30
4.1	Coleta e Armazenamento dos Dados	30
4.2	Pré-processamento do Texto	30
4.2.1	Limpeza dos Dados	31
4.2.2	Tokenização	31
4.2.3	Remoção das <i>Stopwords</i>	31
4.2.4	Derivação Regressiva	31
4.3	Construção do Modelo de Análise de Sentimentos	31
4.3.1	Dados de Treinamento	31
4.3.2	Seleção de Características	32
4.3.3	Treinamento do Modelo	32
4.4	Construção do Modelo de Análise de Polaridade	32
4.5	Visualização de Dados	33
5	Resultados	34
5.1	Conjunto de Dados	34
5.2	<i>Software</i> Final	37
5.3	Discussão	43
5.4	Trabalhos Futuros	49

Referências 50

1 Introdução

Em 1950, Alan Turing publicou o artigo “*Computing Machinery and Intelligence*”, e criou uma polêmica sobre os computadores digitais com a seguinte frase: “*Can machines think?*” (TURING, 1950). A partir daí, o desenvolvimento da tecnologia conhecida como Inteligência Artificial (IA) começou a crescer. Com a evolução da área ao longo das décadas, diversas subáreas da IA foram criadas, e uma delas, que será abordada neste trabalho, é o Processamento de Linguagem Natural (PLN).

1.1 Descrição do Problema

Trabalhar com a interpretação da língua natural do ser humano é o principal objetivo do PLN. Algoritmos de Aprendizado de Máquina utilizam fundamentações matemáticas e estatísticas para classificar os dados de entrada (JURAFSKY; MARTIN, 2000). Uma grande aplicação dessa área é a criação de robôs de autoatendimento, os “*Chatbots*”. Um grande exemplo é a Inteligência Artificial do Banco Bradesco, a BIA¹.

Uma das subáreas do PLN é a Análise de Sentimentos (*Sentiment Analysis*, em Inglês), que busca classificar os textos de entrada em emoções e sentimentos que elas emitem, a fim de criar conhecimento estruturado que possa ser utilizado como base para outros estudos. Essa análise é descrita no Capítulo 4 de Jurafsky e Martin (2000). Um dos grandes interesses do autor deste trabalho é a música e como ela afeta as emoções e sentimentos do ouvinte, conforme será detalhado na seção 1.2.

Com o passar dos anos, as letras de músicas ganharam mais significado, tanto para os ouvintes, quanto para os autores que as escreveram. A combinação das letras com o instrumental pode causar diversas sensações no ser humano, e até alterar o humor repentinamente.

Com isso, este trabalho propõe uma discussão sobre a Análise de Sentimentos de letras de músicas de *Heavy Metal*, gerada à partir de visualizações gráficas dos resultados do algoritmo de Análise de Sentimento utilizado.

¹ <<https://banco.bradesco/canaisdigitais/conheca-bia.shtm>>

1.2 Motivações e Desafios

A motivação para a Análise de Sentimentos de letras de músicas vem da publicação de [Cowen et al. \(2020\)](#), que descreve as sensações que músicas causam nos seres humanos, as quais pelo menos treze dimensões organizam experiências subjetivas. Essas sensações podem ser interpretadas pelo ouvinte de maneiras diferentes, de acordo com a subjetividade sobre as letras. As sensações (dimensões) listadas são:

1. *Amusing* (divertida)
2. *Annoying* (irritante)
3. *Anxious/tense* (Ansiosa/tensa)
4. *Beautiful* (Bela)
5. *Calm/relaxing/serene* (Calma/relaxante/serena)
6. *Dreamy* (Sonhadora)
7. *Energizing/pump-up* (Energizante/bombástica)
8. *Erotic/desirous* (Erótica/desejosa)
9. *Indignant/defiant* (Indignante/desafiadora)
10. *Joyful/cheerful* (Alegre/animado)
11. *Sad/depressing* (Triste/Depressivo)
12. *Scary/fearful* (Assustador/temeroso)
13. *Triumphant/heroic* (Triunfante/heroico)

Outra motivação relevante é o trabalho de [Souza e Café \(2018\)](#), em que criou-se um banco de dados com informações de músicas coletadas (como as letras, os autor(es) e o(s) gênero(s)). Em cima disso, foi feita a Análise de Sentimentos das músicas e armazenou-se a sua polaridade: Um valor numérico que varia de -1 até 1, em que -1 seriam sentimentos negativos, como raiva e medo, e 1 seriam sentimentos positivos, como felicidade e alegria.

Para este trabalho, o conjunto de dados a ser obtido possuirá apenas músicas referentes ao gênero *Heavy Metal*, que além de ser um interesse particular do autor, limita o escopo de pesquisa para um tema mais específico.

O principal desafio ao analisar as letras do gênero citado é o fato de conterem muito sarcasmo e ironia, o que pode gerar resultados inconsistentes ao analisar os sentimentos. A forma de expressão varia muito entre compositores, mas muitos costumam utilizar as

figuras de linguagem citadas, ou seja, criam ambiguidade e podem até deixar múltiplas interpretações. Compreender a ironia e o sarcasmo é difícil até para humanos, e apresenta problemas em tarefas de Processamento de Linguagem Natural ([FARIAS; ROSSO, 2017](#)).

1.3 Objetivo Geral

Este trabalho tem como objetivo extrair e analisar os sentimentos de letras de músicas de *Heavy Metal* utilizando Processamento de Linguagem Natural (PLN) para construir visualizações dos sentimentos e das polaridades das palavras individuais, com a finalidade de gerar uma discussão sobre os resultados.

1.4 Objetivos Específicos

- Obter letras de músicas de *Heavy Metal* da *Web*.
- Criar um conjunto de dados com os artistas e as músicas coletadas.
- Realizar a Análise de Sentimentos de cada uma das letras do conjunto utilizando PLN.
- Realizar a análise de polaridade das palavras das letras.
- Gerar visualizações dos dados e resultados obtidos, bem como discussão dos mesmos.

1.5 Organização do Texto

O Capítulo 2 apresenta todo o embasamento teórico para o que será desenvolvido neste trabalho. Descreve uma visão sobre os métodos que serão utilizados, assim como as fundamentações matemáticas da Análise de Sentimentos, a história do PLN e técnicas de extração de dados da *Web*. Também é apresentado a principal fonte de informações utilizadas na pesquisa, que são as letras de músicas de *Heavy Metal*, descrevendo um pouco o gênero e mostrando os obstáculos encontrados ao analisar tais letras.

O Capítulo 3 apresenta trabalhos que serviram de inspiração e embasamento para esta pesquisa. São descritas, brevemente, publicações que envolvem o Processamento de Linguagem Natural, referentes especificamente a Análise de Sentimentos e trabalhos que servem como exemplo de implementação da Visualização de Dados como forma de representação da informação.

O Capítulo 4 descreve como este trabalho foi realizado e quais ferramentas são utilizadas para determinadas tarefas. Tudo o que foi desenvolvido pelo autor também é descrito, informando como é a interação de ferramentas prontas com códigos autorais.

Também é apresentado o cronograma do projeto em meses, apresentando a distribuição de tarefas ao longo do tempo.

O Capítulo 5 apresenta os resultados gerados, e descreve os artefatos (de *software*) gerados, que englobam todo o conhecimento adquirido até a finalização da pesquisa.

2 Referencial Teórico

O PLN é embasado no artigo “*Computing Machinery and Intelligence*”, de Alan Turing, cujo autor propôs o que é chamado de “Teste de Turing” como critério de inteligência de um computador digital. Como apresentado na publicação, o Teste de Turing consiste em testar a capacidade de uma máquina em pensar como um cérebro humano, ou seja, define se ela possui um comportamento inteligente ([TURING, 1950](#)). Através dessa capacidade de inteligência, surgiram estudos de como ela poderia compreender a linguagem humana e conseguir interpretar textos escritos em linguagens naturais, como o Português, por exemplo. Com a evolução da tecnologia, surgiram diversas aplicações do PLN, como a robótica, ferramentas de tradução de textos e análises sintáticas de frases. O PLN foi descrito com mais detalhes no livro “*Speech and Language Processing*”, de [Jurafsky e Martin \(2000\)](#). Na obra, os autores descrevem as bases que o compõem, e o classificam como uma subárea do Aprendizado de Máquina.

2.1 A Abordagem Clássica do Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é dividido em várias etapas. A Figura 1 ilustra (de baixo para cima) os passos realizados por um algoritmo de PLN. Há uma dificuldade em relacionar dados distintos de uma oração ou discurso. Logo, tal separação serve como uma organização da arquitetura da análise, e também constitui a base para modelos que tornam a tarefa mais gerenciável no ponto de vista da engenharia de software ([BARBOSA et al., 2017](#)).

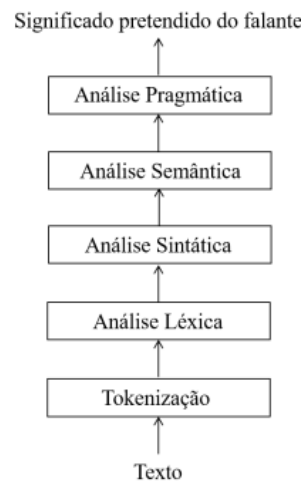


Figura 1 – Estágios da análise no PLN

Fonte: [Barbosa et al. \(2017\)](#), adaptado de [Dale \(2010\)](#)

2.2 Análise de Sentimentos

A Análise de Sentimentos é uma das aplicações do Processamento de Linguagem Natural, que consiste em analisar o sentimento produzido de uma determinada sentença ou texto, cujo resultado extraído, conhecido como polaridade, pode ser positivo, negativo ou neutro. A extração e a Análise de Sentimento gera uma orientação positiva ou negativa que um escritor expressa em relação a algum objeto. Uma crítica de filme, livro ou produto na *Web* expressa o sentimento em relação ao produto, enquanto um texto editorial ou político expressa o sentimento em relação a um candidato ou ação política. Existem diversos algoritmos de Aprendizado de Máquina que realizam a Análise de Sentimento. A tarefa de analisar um sentimento pode ser interpretada como uma simples classificação de texto binária, ou seja, a tarefa de atribuir um rótulo à um texto. ([JURAFSKY; MARTIN, 2000](#), cap.4 - p. 55).

2.2.1 Classificador *Naive Bayes*

O classificador utilizado neste trabalho é o *Naive Bayes* multinomial, um classificador probabilístico Bayesiano, que faz uma suposição simples (ingênua) sobre como os recursos interagem. Essa interação é mensurada pela probabilidade obtida pelo teorema de [Bayes \(1763\)](#).

Para o classificador, documentos de texto são representados como uma “sacola de palavras” (*bag-of-words*), ou seja, um conjunto desordenado de palavras com suas posições ignoradas. Um exemplo de aplicação é a obtenção das frequências das palavras em um documento. Na Figura 2, ao invés de representar as frases como “*I love this movie*”, e

“*I would recommend it*”, apenas notamos que a palavra *I* ocorreu 5 vezes, a palavra *it* ocorreu 6 vezes, as palavras *love*, *recommend* e *movie* ocorreram apenas uma vez, e assim por diante (JURAFSKY; MARTIN, 2000, cap.4 - p. 57).

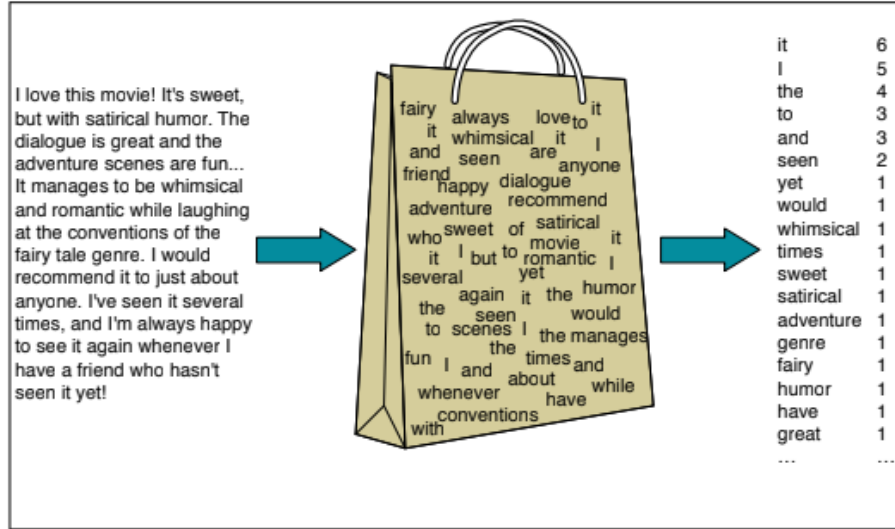


Figura 2 – Visão do classificador aplicado à uma crítica de um filme. A posição das palavras é ignorada (representação da sacola de palavras) e é feito o uso apenas da frequência de cada palavra

Fonte: Jurafsky e Martin (2000, cap.4 - p. 57)

A probabilidade posterior, também conhecida como inferência Bayesiana, consiste em aplicar o teorema de Bayes (1763) para obter os resultados probabilísticos esperados para determinado documento, representada pela Equação 2.1:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \quad | \quad P(Y) \neq 0 \quad (2.1)$$

Logo, o classificador funciona da seguinte maneira: para um documento d , de todas as classes $c \in C$, o classificador retorna a classe \hat{c} , gerada pela maximização da fórmula da probabilidade, representada por $\text{argmax}P$. É utilizado a notação $\hat{}$ para expressar “a estimativa da classe correta” (JURAFSKY; MARTIN, 2000, p. 57), obtida pela Equação 2.2:

$$\hat{c} = \text{argmax}P(c|d) \quad | \quad c \in C \quad (2.2)$$

2.3 Análise de Polaridade

Existem duas abordagens principais para a Análise de Sentimentos:

1. Aprendizado de máquina supervisionado
2. Abordagens não supervisionadas baseadas em análise léxica

Neste trabalho, a discussão gerada é feita à partir da comparação dos dois métodos. O aspecto chave da Análise de Sentimento é analisar um corpo de texto para entender a opinião expressa por ele. Conforme descrito por [Jurafsky e Martin \(2000\)](#), a tarefa de analisar um sentimento utilizando aprendizado de máquina supervisionado pode ser interpretada como uma classificação de texto. Porém, a segunda abordagem é baseada em análise léxica, ou seja, a análise realizada por palavras (denominadas *tokens*) individualmente. Quantificamos esse sentimento com um valor positivo ou negativo, chamado de polaridade. O sentimento geral é muitas vezes inferido como positivo, neutro ou negativo a partir do sinal da pontuação de polaridade. Para essa tarefa, foi utilizada uma técnica não supervisionada para prever o sentimento usando uma base de conhecimento que possui informações detalhadas e especialmente preparadas para Análise de Sentimentos. A base utilizada foi a do projeto SenticNet¹, uma iniciativa concebida no MIT *Media Laboratory* em 2009 dentro de um projeto de uma pesquisa industrial para a *Cooperative Awards in Science and Engineering* (CASE). Seu objetivo é tornar informações conceituais e afetivas transmitidas pela linguagem natural (destinada ao consumo humano) acessíveis e interpretáveis para as máquinas. Isso é feito substituindo o modelo “sacola de palavras” por um novo modelo que vê o texto como uma “sacola” de conceitos e narrativas.

2.4 *Heavy Metal*

O *Heavy Metal* é um gênero do *Rock* que surgiu no final dos anos 1960, início dos anos 1970. A presença da guitarra elétrica com distorção tornava o som mais pesado e sombrio, diferente do *Blues* e do *Jazz* da época. Alguns especialistas sugerem que o gênero surgiu com a música *Helter Skelter* de 1968, dos *Beatles* ([MCKINNEY, 2003](#)). Outros dizem que o gênero começou com os vocais demoníacos de Ozzy Osbourne no álbum homônimo de estréia do *Black Sabbath* ([COPE, 2016](#)), também em 1968. A trajetória da banda começou em 1968, e em 1969 lançaram seu disco de estreia. Os temas de suas músicas contrariavam a “Paz e Amor” da época, incitando personas bíblicas controversas, como Lúcifer e Satanás, a morte de Deus e ida ao inferno, além de criticar sistemas políticos, guerras, o capitalismo e revoluções armadas.

¹ <<https://sentic.net/>>

Hoje em dia, o *Heavy Metal* mudou bastante em relação às décadas passadas, e tem se adaptado às mudanças sociais e aos contextos políticos. Costumam incitar temas como a depressão, a solidão, a tristeza e o suicídio, sempre com um tom herdado de bandas mais antigas.

2.5 Coleta de Letras de Músicas

As letras de músicas de *Heavy Metal* são coletadas da *Web* através de consultas em sites escolhidos previamente, utilizando APIs (Interface de Programação de Aplicação). Neste trabalho são utilizadas duas interfaces: *Deezer for Developers* para a coleta de artistas e músicas e Vagalume API para a coleta das letras das respectivas músicas.

2.5.1 API do Deezer

O Deezer é um sistema de *streaming* de músicas com mais de 73 milhões de obras, além de podcasts, audiobooks e canais de rádio, e oferecem um dos maiores catálogos de áudio disponíveis. Seu serviço é integrado no hardware de mais de 80 marcas de tecnologia². O sistema disponibiliza uma API com acesso ilimitado e sem identificação, e fornece um conjunto de serviços para construir aplicativos da *Web*, permitindo a descoberta do catálogo de músicas do Deezer via uma requisição HTTP³ GET comun⁴. A URL base para cada método de API segue o padrão:

<https://api.deezer.com/version/service/id/method/?parameters>

Os parâmetros são opcionais, mas deve-se especificar o que será buscado através do caminhos *service* e *id*. Para recuperar as *playlists* do usuário com ID 2529, por exemplo, deve-se utilizar a URL:

<https://api.deezer.com/user/2529/playlists>

A resposta pode ser formatada em vários formatos. Neste trabalho, é utilizado o formato JSON⁵, como é mostrado na Figura 3.

² <<https://www.deezer.com/br/company>>

³ <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>>

⁴ <<https://developers.deezer.com/api>>

⁵ <<https://www.json.org/json-en.html>>

```
{
  "id":7620623,
  "title":"Electro yeahh",
  "duration":522,
  "public":true,
  "is_loved_track":false,
  "collaborative":false,
  "nb_tracks":7,
  "fans":18,
  "link":"https://www.deezer.com/playlist/7620623",
  "picture":"https://api.deezer.com/playlist/7620623/image",
  "picture_small":"https://e-cdns-images.dzcdn.net/images/cover/9059eb81e575b1540e86ade0e1852158/56x56-000000-80-0-0.jpg",
  "picture_medium":"https://e-cdns-images.dzcdn.net/images/cover/9059eb81e575b1540e86ade0e1852158/250x250-000000-80-0-0.jpg",
  "picture_big":"https://e-cdns-images.dzcdn.net/images/cover/9059eb81e575b1540e86ade0e1852158/500x500-000000-80-0-0.jpg",
  "picture_xl":"https://e-cdns-images.dzcdn.net/images/cover/9059eb81e575b1540e86ade0e1852158/1000x1000-000000-80-0-0.jpg",
  "checksum":"4575fea47f5c5e1f2aad50fd4018b750",
  "tracklist":"https://api.deezer.com/playlist/7620623/tracks",
  "creation_date":"2013-11-06 15:00:00",
  "md5_image":"9059eb81e575b1540e86ade0e1852158",
  "picture_type":"cover",
  "time_add":0,
  "time_mod":1383750000,
  "creator":{
    "id":2529,
    "name":"dadbond",
    "tracklist":"https://api.deezer.com/user/2529/flow",
    "type":"user"
  },
  "type":"playlist"
}
```

Figura 3 – Exemplo de retorno de uma consulta utilizando a API do Deezer

Fonte: Próprio autor

2.5.2 API do Vagalume

O Vagalume é um portal de música brasileiro, lançado em 2002. Seu site possui um grande acervo de discografias, traduções, cifras, fotos, estilos musicais, notícias, informações de biografia sobre os artistas, medidor de popularidade, cálculo de proximidade musical (de músicas e artistas) e tempo de sincronização de legendas das músicas⁶. O site disponibiliza uma API que permite a integração de informações de músicas e artistas em outras aplicações. A consulta pode ser realizada via os métodos HTTP POST ou GET para obter informações do artista e/ou músicas. A URL da requisição é:

<https://api.vagalume.com.br/search.php?>

Após o símbolo `?`, deve-se inserir um ou mais parâmetros de consulta, listados na Tabela 1, separados pelo símbolo `&`.

⁶ <<https://api.vagalume.com.br/docs/>>

Tabela 1 – Parâmetros de consulta da API do Vagalume

Parâmetro	Descrição
art	Nome do artista.
mus	Título da música (opcional).
musid	ID da música. Quando este parâmetro for passado, art e mus são desnecessários.
udig	Caso o retorno seja o captcha por conta do <i>rate limit</i> de acessos, a requisição seguinte deve conter o parâmetro udig e serial. Veja o exemplo completo no tratamento do captcha.
serial	Usado em conjunto com udig para fazer a requisição pós-captcha.
callback	Função que será chamada em seu código para requisição JSONP (“JSON with padding”). Como permitimos requisição de qualquer origem, este parâmetro não é necessário.
nolyrics	Não irá trazer o texto da letra da música, consequentemente não passará pela verificação de captcha.
extra	Indica quais informações extras podem vir na resposta. É uma lista de opções separadas por vírgulas que podem ser livremente combinadas. As opções possíveis são: alb , relart , relmus e rank .

Fonte: <<https://api.vagalume.com.br/docs/letras/>>

Todas as chamadas devem incluir a chave de API na requisição utilizando o parâmetro **apikey**, que é gerada pelo próprio Vagalume e requer um cadastro no mesmo. Para recuperar informações da música *Holiday*, de Madonna, por exemplo, deve-se utilizar a URL:

<https://api.vagalume.com.br/search.php?art=madonna&mus=holiday&apikey=CHAVE>

O resultado da chamada é formatado em JSON, como é mostrado na Figura 4.

```
1  // Exemplo de retorno da requisição
2  { "type": "exact",
3    "art": {
4      "id": "3ade68b3g1f86eda3",
5      "name": "Madonna",
6      "url": "https://www.vagalume.com.br/madonna/"
7    },
8    "mus": [{
9      "id": "3ade68b6g8e27fda3",
10     "name": "Holiday",
11     "lang": 2,
12     "url": "https://www.vagalume.com.br/madonna/holiday.html",
13     "text": "Holiday Celebrate\nHoliday Celebrate(..cut..)"
14     "translate": [{
15       "id": "3ade68b6g417afda3",
16       "lang": 1,
17       "url": "https://www.vagalume.com.br/madonna/holiday-traducao.html",
18       "text": "[Feriado]\nFeriado, comemore\nFeriado comemore(..cut..)"
19     }]
20   }]
21 };
```

Figura 4 – Exemplo de retorno de uma consulta utilizando a API do Vagalume

Fonte: <<https://api.vagalume.com.br/docs/letras/>>

2.6 Visualização de Dados

Visualização de Dados é uma subárea da Análise de Dados, que foca em mostrar ao analista uma forma acessível e compreensível de visualizar grandes volumes de dados, ou seja, é uma representação gráfica da informação. Preocupa-se com o design, desenvolvimento e aplicação gráfica dos dados gerados por computador. Diferente da visualização estática, como planilhas, por exemplo, a visualização interativa permite o usuário especificar o formato usado para apresentar os dados (SADIKU et al., 2016). As formas mais comuns de visualização, segundo Sadiku et al. (2016), são:

- A) Gráfico de linha: Mostra o relacionamento entre itens, e pode ser usado para comparar mudanças ao longo do tempo.
- B) Gráfico de barras: Usado para comparar quantidades de diferentes categorias.
- C) Gráfico de dispersão: Representação bi-dimensional mostrando a variação entre dois itens.
- D) Gráfico de pizza: Usado para comparar partes de um todo.

A Figura 5 mostra um exemplo de cada um dos gráficos citados acima.

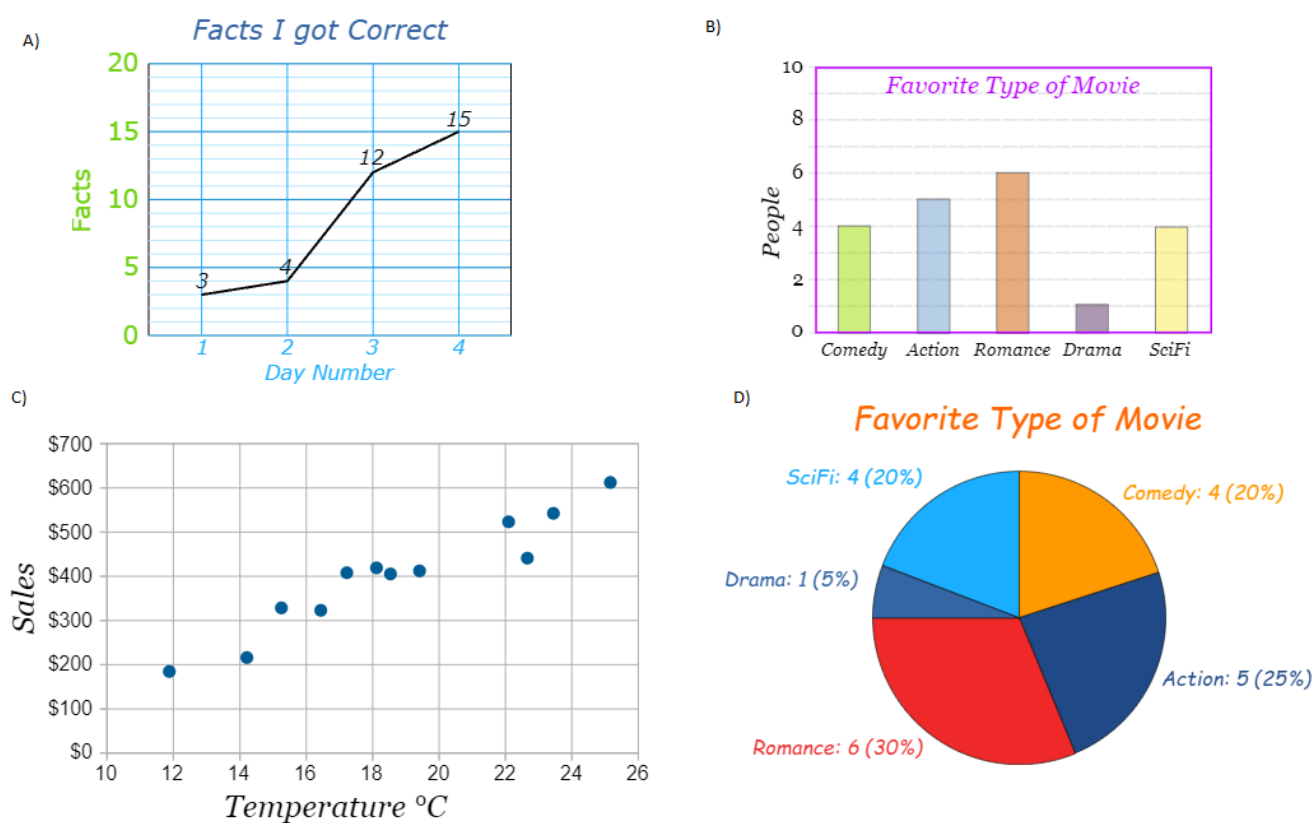


Figura 5 – Técnicas de Visualização de Dados comumente usadas:

- A) Gráfico de linha.
- B) Gráfico de barras.
- C) Gráfico de dispersão.
- D) Gráfico de pizza.

Fonte: Adaptado de <<https://www.mathsisfun.com/>>

Azzam et al. (2013) define que a Visualização de Dados é embasada em três critérios:

1. Visualização de Dados é um processo baseado ou em dados qualitativos ou em dados quantitativos.
2. Resulta em uma imagem representativa dos dados não tratados.
3. Legível pelos leitores e auxilia na exploração, examinação e comunicação dos dados.

A Figura 6 mostra uma visualização de respostas de uma pesquisa feita sobre um programa qualquer em algumas áreas da cidade de Jacksonville, no estado da Flórida nos Estados Unidos da América. Cada um dos gráficos demonstra uma região da cidade, e as barras mostram a porcentagem de habitantes de acordo com cada uma das respostas coletadas, separadas em:

- A) Já ouviu falar do programa.
- B) Sente que a existência do programa é importante.
- C) Pensa que a cidade deve pagar pelo programa.
- D) Está otimista que o programa irá funcionar.

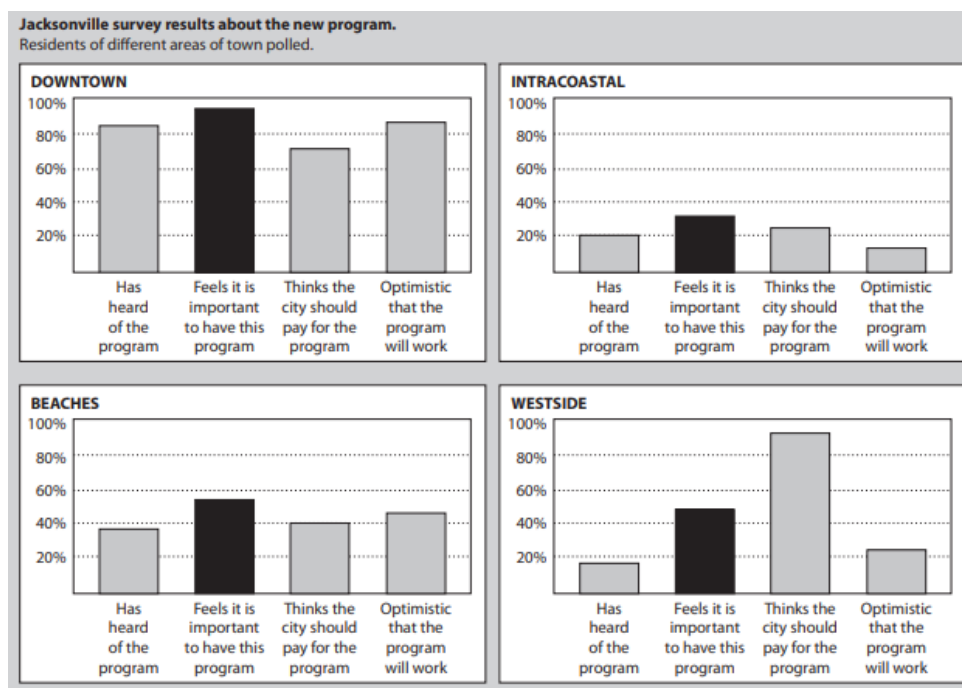


Figura 6 – Exemplo de visualização de respostas de uma população por área

Fonte: Azzam et al. (2013)

Kumar (2019) demonstra diversos exemplos de como gerar Visualização de Dados utilizando a linguagem de programação Python⁷ e a biblioteca Matplotlib⁸. A Figura 7 mostra uma representação da duas funções matemáticas, gerada pelo código fonte na Figura 8.

⁷ <<https://python.org>>

⁸ <<https://matplotlib.org/>>

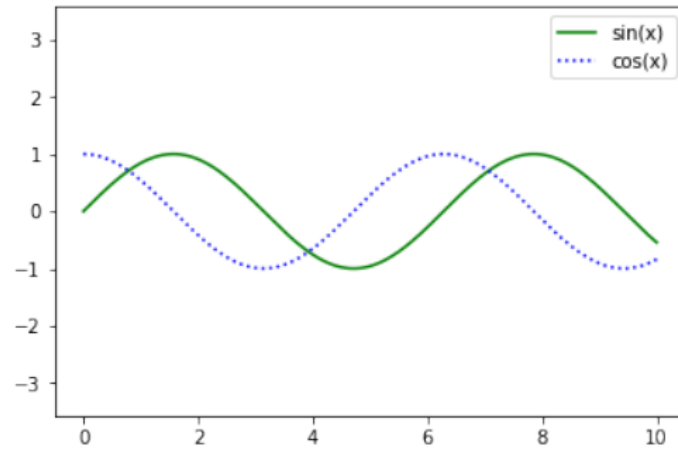


Figura 7 – Exemplo de visualização das funções matemáticas $y = \text{sen}(x)$ e $y = \text{cos}(x)$

Fonte: [Kumar \(2019\)](#)

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.linspace(0, 10, 1000) #Returns number spaces evenly w.r.t interval
4 plt.plot(x, np.sin(x), '-g', label='sin(x)')
5 plt.plot(x, np.cos(x), ':b', label='cos(x)')
6 plt.axis('equal')
7 plt.legend()
8 plt.show()
```

Figura 8 – Código fonte em Python que gera a visualização das funções matemáticas $y = \text{sen}(x)$ e $y = \text{cos}(x)$

Fonte: [Kumar \(2019\)](#)

3 Trabalhos Relacionados

Barbosa et al. (2017) publicaram diversas formas de se trabalhar com PLN utilizando a linguagem Python. No artigo, são descritos os principais conceitos básicos dessa linguagem de programação, e como utilizar a biblioteca NLTK¹. São dados vários exemplos de utilização da ferramenta de Loper e Bird (2002), todos em português, demonstrando o poder computacional da ferramenta. Os autores utilizaram a mesma ferramenta que será utilizada neste trabalho, e serve como uma espécie de passo a passo, com informações retiradas da própria documentação do *Natural Language Toolkit*.

Oudenne, Swarthmore e Chasins (2010) trabalham com a polaridade de emoções produzidas por letras de músicas utilizando Análise de Sentimentos, porém sem utilizar o NLTK. Na publicação, demonstram que as letras são uma grande fonte de informação, pois são disponibilizadas de forma semi-estruturada na Internet. Também focam na simples, porém não trivial tarefa de categorizar as músicas pela sua polaridade. Elas são classificadas como positivas ou negativas dependendo da emoção geral extraída. São testados diferentes algoritmos que analisam a frequência, a presença, e a semelhança de cosseno para classificar um conjunto de dados de 420 letras únicas. Também avaliam a subjetividade das letras para fazer uma comparação entre a Análise de Sentimentos de letras de músicas com a Análise de Sentimentos tradicional. Neste trabalho também pretende-se extrair a polaridade das emoções das letras de músicas, assim como foi feito em Oudenne, Swarthmore e Chasins (2010), porém utilizando o classificador *Naive Bayes*, ao invés da Similaridade por Cosseno, utilizada pelos autores citados.

Já no trabalho de Souza e Café (2018), os autores desenvolveram um *Web Crawler* utilizando a linguagem Python, e os dados coletados foram armazenados em um banco de dados relacional em SQLite². Após a coleta e estruturação, as informações são complementadas com marcação morfo sintática para o português e resultado da Análise de Sentimentos (polaridade). Contudo, na publicação, os autores não descrevem a Análise de Sentimento em si, apenas mostram os resultados gerados. O trabalho citado é a inspiração para esta pesquisa, que inclui a forma de como as informações são coletadas, armazenadas e analisadas.

¹ <<https://www.nltk.org/>>

² <<https://www.sqlite.org/index.html>>

Seguindo com a Análise de Sentimentos, [Weiand e Weiand \(2018\)](#) propuseram um algoritmo de Análise de Sentimentos do microblog Twitter³, utilizando o modelo probabilístico *Naive Bayes*, com a finalidade de classificá-los em positivos ou negativos. Foram utilizados dados pré-treinados de outro autor para realizar uma análise cruzada dos resultados, e foram utilizadas as bibliotecas NLTK e Scikit-learn⁴. Os autores utilizaram um modelo de pesquisa bem parecido com o que é proposto neste trabalho, diferenciando apenas em como as informações são coletadas e a fonte de informações.

Já relacionado com a forma de como representar a informação, [Ribas, Noda e Marques \(2018\)](#) criaram uma ferramenta *Web* para Visualização de Dados obtidos através de *Web Scraping*, utilizando a linguagem Python. A aplicação, denominada *BigPy*, faz a captura, tratamento e análise de dados de defesa do consumidor, utilizando como exemplo os dados abertos do portal público <https://consumidor.gov.br>. Após pesquisar pelo nome da empresa, é disponibilizada ao usuário a interface principal do painel do sistema, contendo a visualização dos dados da empresa, conforme Figura 9. Foram implementados gráficos para visualização da evolução do número de reclamações por ano, evolução na nota dada pelo consumidor ao atendimento da empresa, reclamações por gênero, reclamações por região do país e reclamações por faixa de idade do consumidor. As visualizações foram produzidas com a biblioteca Chart.js⁵, da linguagem JavaScript⁶, ferramenta que pretende ser utilizada nesta pesquisa.

³ [<https://twitter.com>](https://twitter.com)

⁴ [<https://scikit-learn.org/>](https://scikit-learn.org/)

⁵ [<https://www.chartjs.org/>](https://www.chartjs.org/)

⁶ [<https://www.javascript.com/>](https://www.javascript.com/)



Figura 9 – Painel principal do BigPy com a visualização dos dados

Fonte: Ribas, Noda e Marques (2018)

4 Métodos

Este trabalho foi desenvolvido utilizando a linguagem de programação Python. A escolha dessa linguagem se deu devido à sua grande versatilidade para o desenvolvimento de diversos tipos de aplicações (desde simples *scripts* até aplicações *Web* completas). Além disso, a linguagem possui diversas bibliotecas para o tratamento, manipulação e análise de dados (MCKINNEY, 2017). O desenvolvimento do *software* foi dividido nas seguintes etapas:

1. Coleta e Armazenamento dos Dados
2. Pré-processamento do Texto
3. Construção do Modelo de Análise de Sentimentos
4. Construção do Modelo de Análise de Polaridade
5. Visualização

4.1 Coleta e Armazenamento dos Dados

As informações que serão utilizadas para gerar os resultados da Análise de Sentimentos e as visualizações serão as letras de músicas de bandas de *Heavy Metal*. As bandas foram escolhidas da *playlist Metal Essentials* do Deezer, uma lista de reprodução escolhida pelo autor que conta com um conjunto de cem músicas de diversos artistas do gênero. Para recuperar esses dados, utiliza-se a API do Deezer, que retorna todos os dados necessários. Após coletar o nome dos artistas e das músicas, são coletadas as letras utilizando API do Vagalume. Todas as músicas cujas letras foram encontradas no acervo do Vagalume são armazenadas em um arquivo em formato CSV, e são lidas pela biblioteca Pandas¹, da linguagem Python, que permite a análise e manipulação dos dados.

4.2 Pré-processamento do Texto

Antes de utilizar a letra como entrada do algoritmo, é necessário pré-processar os dados, à fim de obter apenas o que agregue sentido e relevância para a análise.

¹ <<https://pandas.pydata.org/>>

4.2.1 Limpeza dos Dados

Para a limpeza, é utilizada uma biblioteca nativa da linguagem Python, a *Regular Expression Operations*, que fornece a implementação de Expressões Regulares. Com ela, são removidas *strings* vazias (espaços em branco), caracteres especiais, sinais de pontuação e dígitos, deixando apenas as palavras.

4.2.2 Tokenização

Tokenização consiste em transformar a letra limpa em uma lista de palavras, que antes era uma única *string*. Isso ajuda a trabalhar diretamente com as palavras. O Tokenizador utilizado é implementado utilizando a biblioteca NLTK, de Loper e Bird (2002). O conceito de “sacola de palavras” (*bag-of-words*) foi aplicado nessa etapa pois ignoramos qualquer ordem que elas possuam dentro de um texto. A em nível de programação, elas possuem uma ordem pois são armazenadas em um vetor para fácil iteração, mas a nível de análise, a ordem é totalmente ignorada.

4.2.3 Remoção das *Stopwords*

Stopwords são as palavras que são usadas para construir uma frase. Palavras, principalmente conectivos, como “the”, “a”, “of”, etc., são as mais comuns. Essas palavras constituem a frequência máxima de ocorrência quando comparadas com todas as outras palavras. Como não agregam nenhum valor ao sentimento da frase, parágrafo ou da letra, são removidas da lista. As *stopwords* são obtidas da biblioteca NLTK.

4.2.4 Derivação Regressiva

Na linguística e no Processamento de Linguagem Natural, a Derivação Regressiva (*Stemming*) é o processo de reduzir palavras flexionadas e derivadas para a raiz da palavra, base ou forma raiz. Palavras como “cats”, “catlike”, “catty” são reduzidas para “cat”. A ferramenta de derivação é implementada utilizando a biblioteca NLTK.

4.3 Construção do Modelo de Análise de Sentimentos

4.3.1 Dados de Treinamento

O conjunto de letras para treinamento do modelo é obtido do projeto “*Music-Mood*”², uma abordagem de aprendizado de máquina para classificar músicas por humor com base nas letras das músicas, onde o autor criou um conjunto pré-classificado com o humor da letra.

² <<https://github.com/rasbt/musicmood>>

4.3.2 Seleção de Características

Modelos de aprendizado de máquina não funcionam com recursos textuais, apenas com características (*features*) numéricas. Isso significa que é necessário transformar os dados a serem classificados em números. Mas não é qualquer informação que deve ser levada em conta. Para isso, foi utilizada a técnica conhecida como Vetorização por Contagem (*Count Vectorization*). Sua função é considerar a frequência das palavras na letra da música e criar um vocabulário com palavras Felizes e Tristes.

O vocabulário produzido é salvo em um arquivo binário utilizando a biblioteca Joblib³, pois ele deve ser utilizado sempre pelo mesmo modelo de Análise de Sentimentos.

4.3.3 Treinamento do Modelo

O modelo utilizado neste trabalho é o classificador *Naive Bayes* multinomial (*Multinomial Naive Bayes*). Suas funções de treinamento, classificação e predição são importadas da biblioteca Scikit-learn. Como entrada, são passadas as classes de sentimento (*Happy* e *Sad*) e as letras das músicas contidas no conjunto de treinamento

Após o treinamento, o modelo treinado é salvo em um arquivo binário utilizando a biblioteca Joblib. Sempre que o classificador for utilizado, o sistema lê o arquivo, ao invés de treinar o modelo novamente a cada iteração.

4.4 Construção do Modelo de Análise de Polaridade

A abordagem para analisar a polaridade neste trabalho não utiliza nenhum modelo de Aprendizado de Máquina explícito, mas a base de conhecimento disponibilizada pela Sentic Team, desenvolvedores do projeto, foi construída utilizando técnicas de Inteligência Computacional. Essa base, com mais de 50.000 palavras do vocabulário da língua inglesa, e cada palavra possui sua polaridade, que varia de -1 até 1, e seu sentimento, que pode ser positivo, negativo ou neutro. O sentimento da palavra é atribuído pela análise do valor da polaridade, realizando a análise léxica, ou seja, não envolve o algoritmo de classificação deste trabalho. O algoritmo criado consiste em iterar o texto de entrada (letra de uma música) e verificar se as palavras estão na base do Senticnet. Se verdadeiro, o sistema armazena sua polaridade. Se falso, ela não possui relevância para a análise, ou ela ainda não foi estudada pela equipe da Sentic Team.

³ <<https://joblib.readthedocs.io/>>

4.5 Visualização de Dados

O *front-end* do sistema foi construído com o *framework* Streamlit⁴, que permite criar componentes como botões, caixas de texto, menus, barras laterais, etc. O Streamlit também fornece componentes para renderização de gráficos e imagens, que possibilita a visualização dos dados no navegador. As visualizações de sentimento e polaridade foram feitas utilizando gráficos de barra, construídos com a biblioteca Plotly⁵, que produz gráficos interativos com qualidade. Para a visualização da frequência das palavras, é utilizada a técnica de nuvem de palavras, uma imagem que contém as palavras mais frequentes de um texto; enquanto maior sua frequência, maior seu tamanho na imagem. A biblioteca para gerar nuvens de palavras se encontra disponível em um repositório⁶ *online*, e a imagem final é renderizada com a biblioteca Matplotlib⁷. Tanto as nuvens de palavras quanto os gráficos de barra com as polaridades da análise da *playlist* levam em consideração apenas as cem palavras mais faladas pois o conjunto de palavras analisadas é volumoso, e caso todas fossem consideradas, não seria possível analisar os resultados com precisão. Na análise individual, são mostradas apenas as palavras encontradas na base.

⁴ <<https://streamlit.io/>>

⁵ <<https://plotly.com/python/>>

⁶ <https://amueller.github.io/word_cloud/>

⁷ <<https://matplotlib.org/>>

5 Resultados

5.1 Conjunto de Dados

Utilizando a *playlist Metal Essentials* do Deezer, foram coletadas apenas as músicas nas quais suas letras foram encontradas na base do Vagalume, dentre elas:

1. Slayer - *Raining Blood*
2. Pantera - *Walk*
3. Megadeth - *Symphony Of Destruction*
4. Metallica - *Enter Sandman*
5. Ozzy Osbourne - *Hellraiser*
6. Iron Maiden - *Fear of the Dark*
7. System of a Down - *ATWA*
8. Slipknot - *Left Behind*
9. Type O Negative - *Red Water (Christmas Mourning)*
10. Linkin Park - *In the End*
11. DOPE - *Die MF Die*
12. Fear Factory - *Demanufacture*
13. Slipknot - *Wait and Bleed*
14. Sepultura - *Refuse / Resist*
15. Alice in Chains - *Them Bones*
16. Ghost - *Square Hammer*
17. Mastodon - *Steambreather*
18. Iron Maiden - *Days Of Future Past*
19. Black Sabbath - *Children of the Grave*
20. Judas Priest - *Painkiller*

21. Anthrax - *Madhouse*
22. Metallica - *One*
23. Pantera - *Cowboys from Hell*
24. Machine Head - *Davidian*
25. Slipknot - *Psychosocial*
26. Sepultura - *Roots Bloody Roots*
27. Rage Against the Machine - *Bombtrack*
28. System of a Down - *Chop Suey!*
29. Soulfly - *Jumpdafuckup*
30. Drowning Pool - *Bodies*
31. Linkin Park - *One Step Closer*
32. Evanescence - *Going Under*
33. P.O.D - *Youth Of The Nation*
34. Papa Roach - *Last Resort*
35. Children of Bodom - *Are You Dead Yet?*
36. Type O Negative - *Black No. 1*
37. Ozzy Osbourne - *Mr. Crowley*
38. Megadeth - *Tornado Of Souls*
39. Iron Maiden - *Run to the Hills*
40. Judas Priest - *Electric Eye*
41. Metallica - *Master Of Puppets*
42. Pantera - *Cemetery Gates*
43. In Flames - *Only for the Weak*
44. Gojira - *Born For One Thing*
45. System of a Down - *Toxicity*
46. TOOL - *Sober*

-
47. Metallica - *Holier Than Thou*
 48. Dream Theater - *Pull Me Under*
 49. Amon Amarth - *Twilight Of The Thunder God*
 50. Nightwish - *Nemo*
 51. Trivium - *Pull Harder on the Strings of Your Martyr*
 52. Slayer - *Dead Skin Mask*
 53. Megadeth - *Hangar 18*
 54. Sepultura - *Territory*
 55. Five Finger Death Punch - *Living The Dream*
 56. Down - *Lifer*
 57. TOOL - *Stinkfist*
 58. Mötley Crüe - *Shout At The Devil*
 59. Judas Priest - *You've Got Another Thing Coming*
 60. Behemoth - *O Father O Satan O Sun!*
 61. Lamb of God - *Redneck*
 62. Volbeat - *Shotgun Blues*
 63. Life of Agony - *River Runs Red*
 64. Fear Factory - *Linchpin*
 65. Killswitch Engage - *My Curse*
 66. Mastodon - *Pushing the Tides*
 67. Entombed - *Sinners Bleed*
 68. Cannibal Corpse - *Hammer Smashed Face*
 69. Dio - *Rainbow In The Dark*
 70. Mötley Crüe - *Too Fast For Love*
 71. Black Sabbath - *Iron Man*
 72. Saxon - *Princess of the Night*

- 73. Metallica - *Fade To Black*
- 74. Megadeth - *Peace Sells*
- 75. Testament - *Practice What You Preach*
- 76. TOOL - *Pneuma*
- 77. Type O Negative - *The Profit of Doom*

Cada linha do conjunto de dados representa uma música diferente, e as colunas são formadas por: ID do Deezer, Artista, Nome da Música, Letra, Letra Explícita (verdadeiro ou falso) e ano de lançamento. Uma observação importante é que o ano da música no Deezer corresponde ao ano que a mesma foi inserida na plataforma, logo alguns anos tiveram que ser pesquisados manualmente pelo autor. A Figura 10 representa as cinco primeiras entradas do conjunto de dados construído.

song_id	artist	song	lyric	explicit_lyrics	release_date
65690449	Slayer	Raining Blood	Now I shall reign in blood!	False	1986
662879	Pantera	Walk	Walk on home, boy	True	1994
61382107	Megadeth	Symphony Of Destruction	(Chorus)	False	1992
2114399	Metallica	Enter Sandman	We're off to never-never land	False	1991
1487860222	Ozzy Osbourne	Hellraiser	I'll put a spell on you	False	1991
585608872	Iron Maiden	Fear of the Dark	I am a man who walks alone	False	1992

Figura 10 – Conjunto de dados criado

Fonte: Próprio autor

5.2 Software Final

Como artefato final, foi desenvolvido um software que contempla todos os conceitos pesquisados. A ferramenta coleta as letras de músicas e as armazena em um arquivo em formato CSV. O *back-end* foi escrito em Python, que realiza a análise das letras coletadas, e todos os resultados numéricos com suas devidas descrições serão passados para o *front-end*, que irá gerar as visualizações dos dados. A Figura 11 mostra a tela inicial do sistema assim que os *scripts* são iniciados.

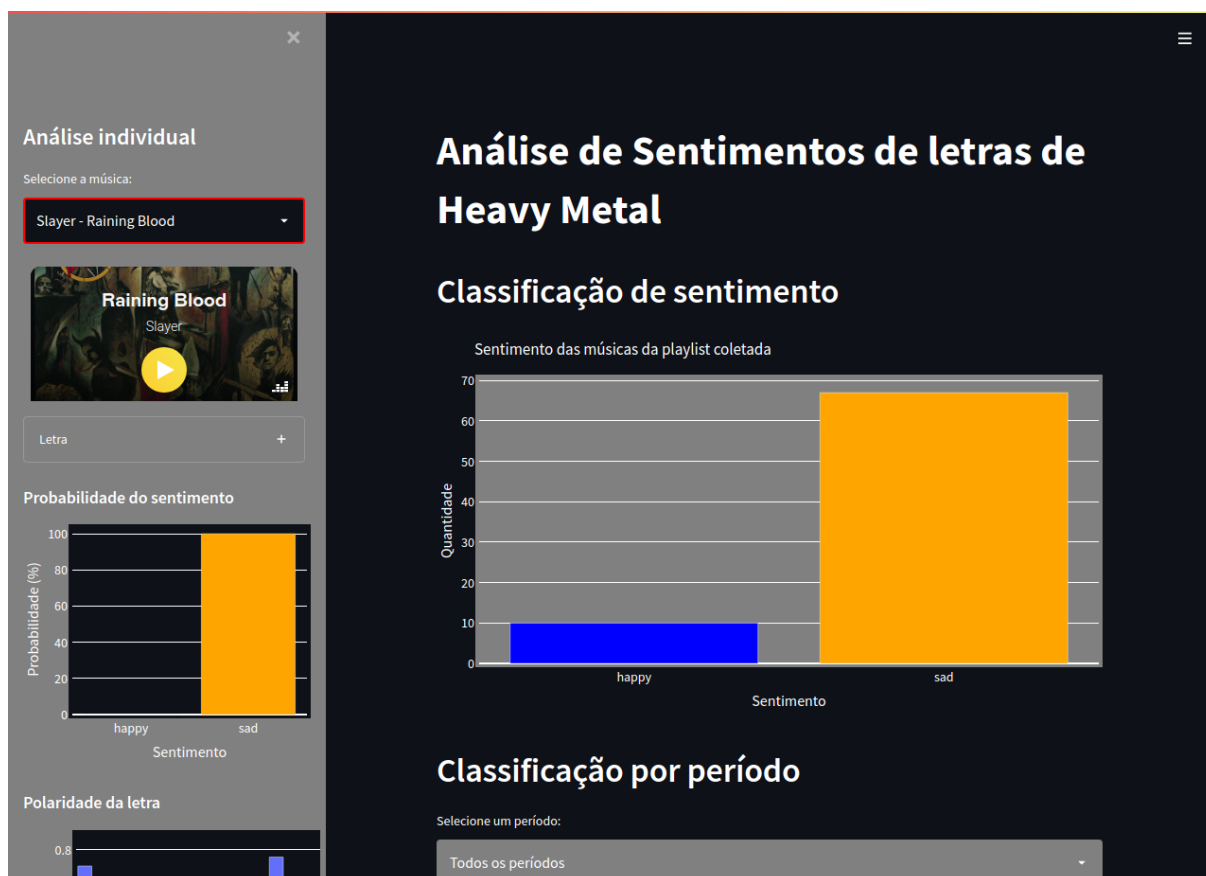


Figura 11 – Tela inicial do sistema final

Fonte: Próprio autor

É possível filtrar os períodos de visualização, de 1970 até 2021, conforme Figura 12 e Figura 13. Uma vez escolhido o período, as nuvens de palavras e os gráficos de polaridade aparecem em seguida, conforme Figura 14. O sistema também apresenta a análise individual de sentimentos e polaridade das palavras das músicas encontradas, conforme Figura 15.

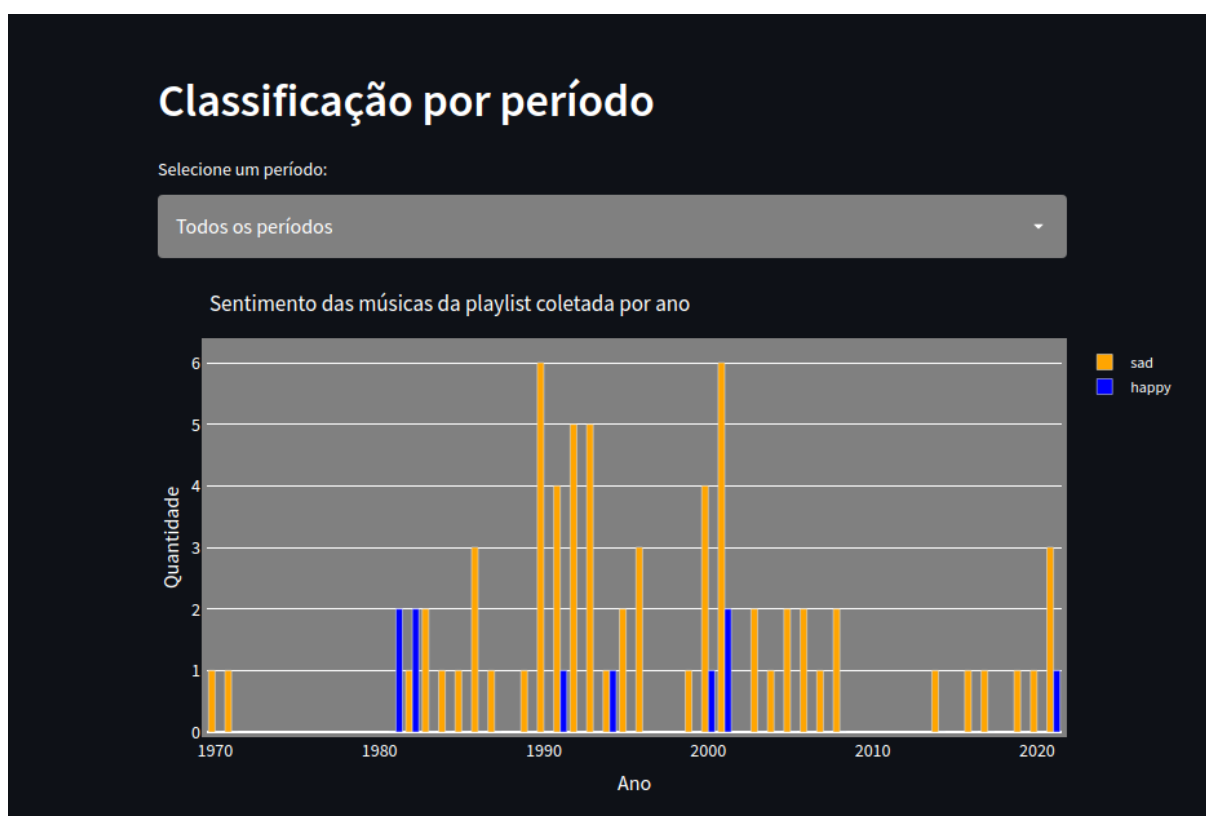


Figura 12 – Filtro com todos os períodos

Fonte: Próprio autor

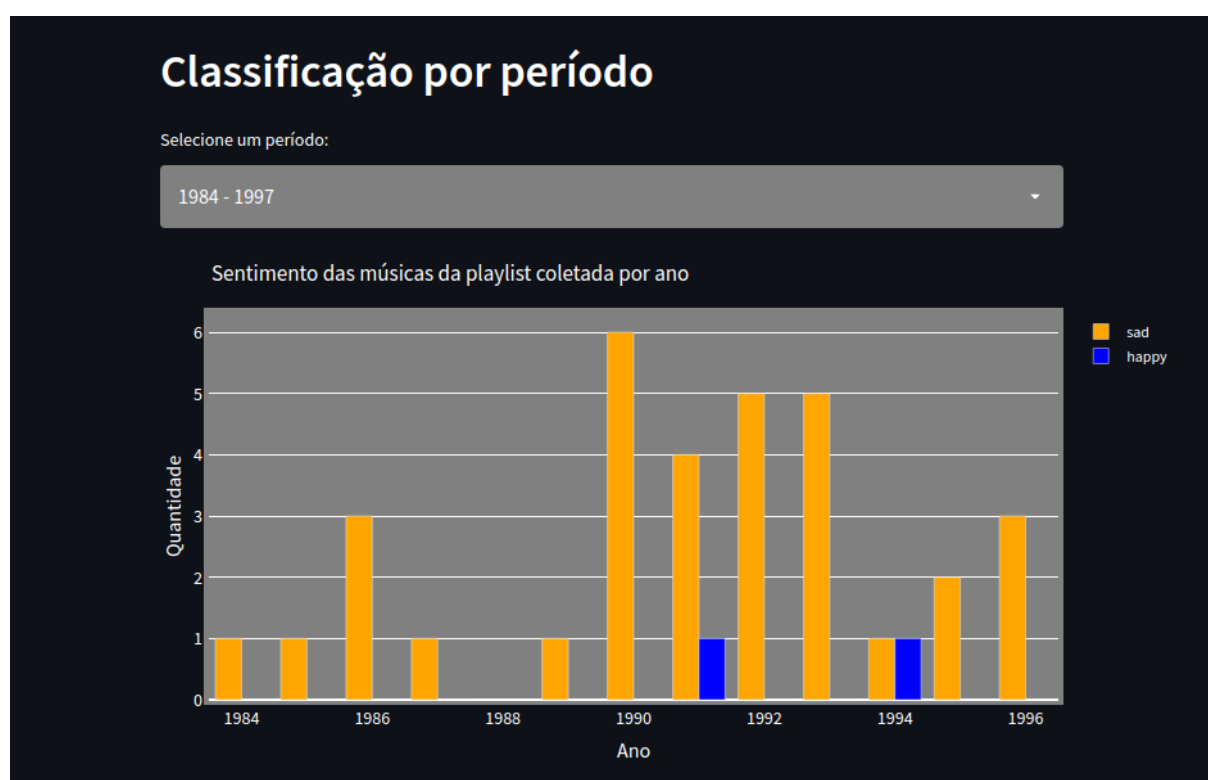


Figura 13 – Filtro de 1984 até 1997

Fonte: Próprio autor

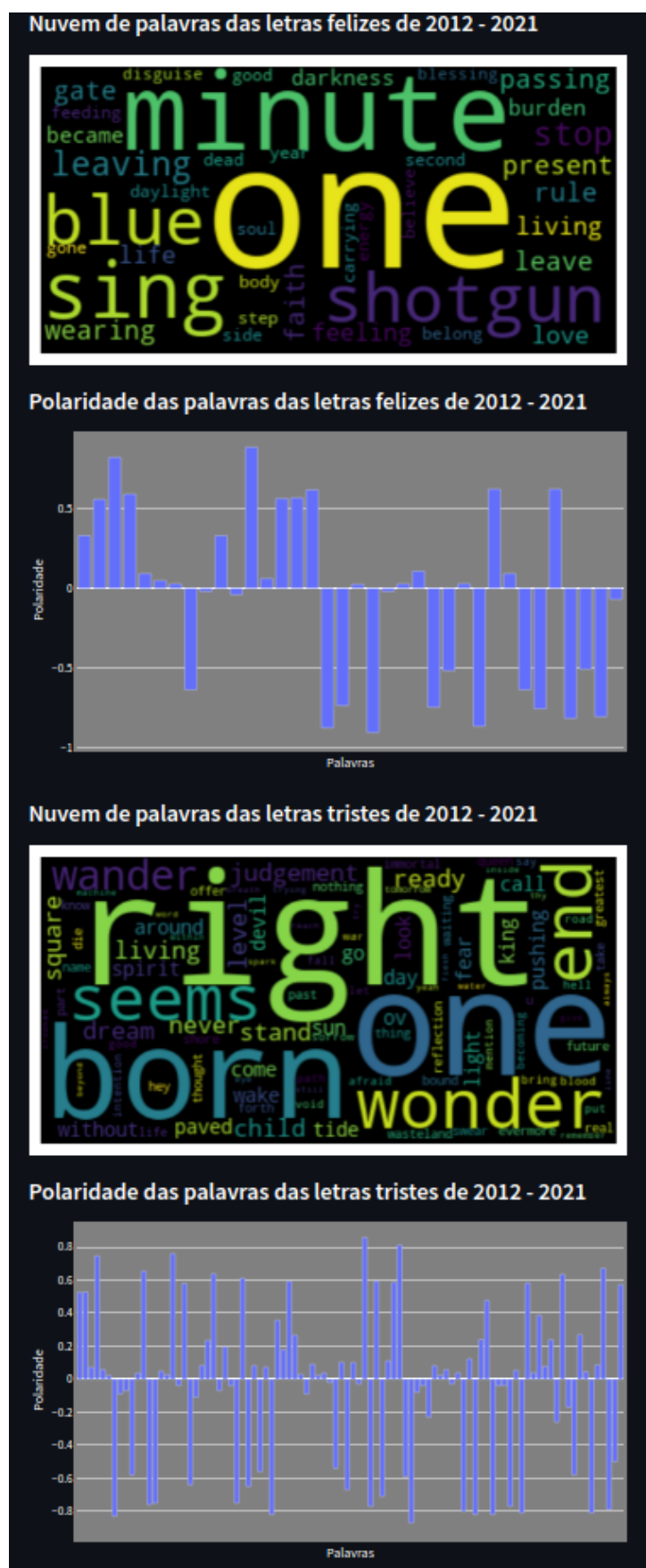


Figura 14 – Nuvens de palavras e gráficos das polaridades

Fonte: Próprio autor

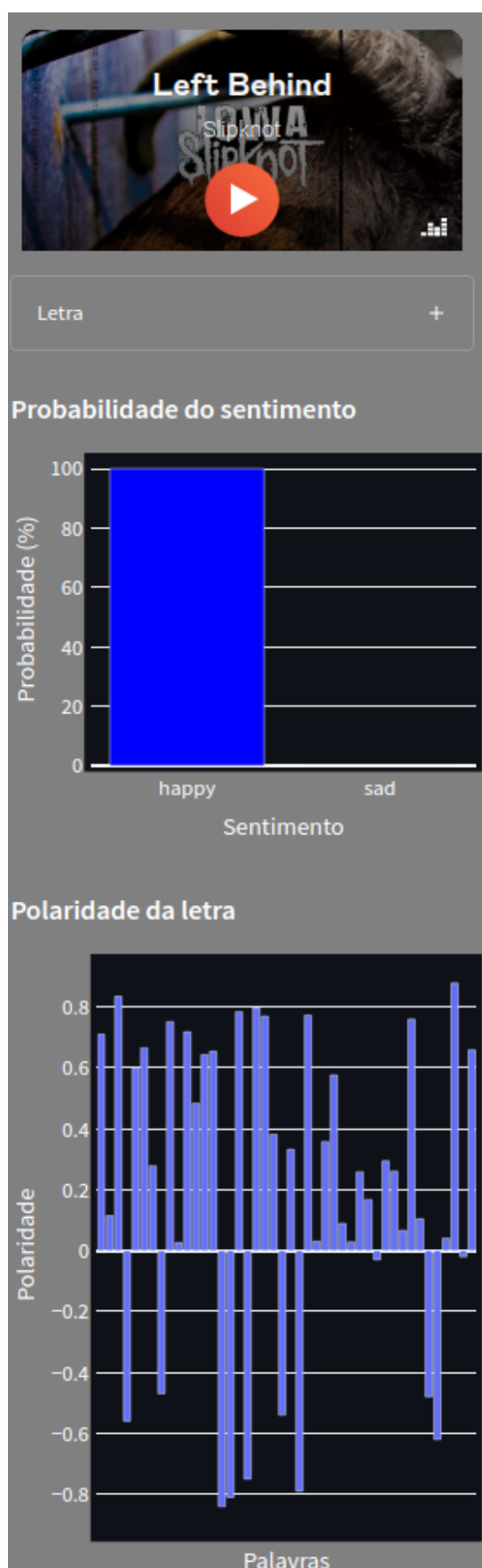


Figura 15 – Análise individual

Fonte: Próprio autor

5.3 Discussão

Músicas de todos os gêneros desencadeiam consistentemente diversos impulsos e emoções, que são interpretados pelo cérebro. Essas percepções induzem experiências mentais particulares, conhecidas como sentimentos. Os sentimentos costumam ser prazerosos, mas não necessariamente são (HABIBI; DAMASIO, 2014). Além disso, as letras das músicas fornecem informações importantes sobre a canção. Os humanos, simplesmente ao ler as letras, captam essas informações em forma de gênero, sentimento e tema. Contudo, automatizar essa tarefa é muito complicada (OUDENNE; SWARTHMORE; CHASINS, 2010).

Por outro lado, ao comparar com a Análise de Sentimentos de avaliações de filmes, por exemplo, é esperado que os resultados não sejam obtidos com tanta precisão quanto ao analisar letras de músicas. Críticos de filmes utilizam frases como “Eu gostei deste filme pois...” ou “Aquela cena é terrível” em suas críticas, que podem ser extraídas com facilidade a partir da subjetividade léxica adequada (OUDENNE; SWARTHMORE; CHASINS, 2010). De acordo com Oudenne, Swarthmore e Chasins (2010), existem três principais dificuldades com a Análise de Sentimentos de letras de músicas:

1. Canções podem conter uma série de versos e frases negativas que acabam em notas positivas e edificantes, e vice-versa. Músicas românticas, particularmente, podem gerar resultados errôneos, pois podem expressar o quão feliz o autor estava durante um romance, e, no final, expressar sua tristeza sobre um término repentino.
2. Canções podem não conter nenhuma subjetividade léxica, e ainda sim expressar sentimentos positivos, negativos ou neutros. Na Figura 16 é mostrado uma estrofe da música “*It’s Still Rock And Roll To Me*” de Billy Joel:

What's the matter with the clothes I'm wearing?
"Can't you tell that your tie's too wide?"
Maybe I should buy some old tab collars?
"Welcome back to the age of jive.

Figura 16 – Billy Joel - “*It’s Still Rock And Roll To Me*”, 1980

Fonte: <<https://www.vagalume.com.br/billy-joel/its-still-rock-and-roll-to-me.html>>

Não é evidente quais palavras dessa estrofe poderiam representar conotações positivas; porém, quando analisadas juntas, a estrofe expressa uma emoção positiva. Isso ocorre tanto em canções positivas e negativas, e pode ser difícil separar a emoção da letra inteira da emoção produzida por cada estrofe individualmente.

3. Canções podem expressar emoções positivas sobre coisas negativas, e vice-versa. Músicas do gênero Rap particularmente sofrem deste problema: suas letras costumam expressar emoções positivas sobre eventos negativos como tiroteios e assaltos, que gera um certo nível de confusão em um sistema de classificação.

Esse tipo de confusão pode ser observado no *software* final. O sarcasmo, a ironia e a ausência de contexto fazem com que a máquina produza resultados “ao pé da letra”. De todas as músicas analisadas, alguns resultados se sobressaíram.

A música *Psychosocial*, da banda Slipknot é classificada como uma letra triste, conforme Figura 17. A palavra *psychosocial* aparece na nuvem de palavras tristes que incluem 2008, ano de lançamento da música. Isso ocorre pois o vocalista a repete diversas vezes na letra, porém ela não agrega nenhum valor na análise.

Outro resultado interessante é na análise de *Symphony Of Destruction*, de Megadeth. Mesmo com as polaridades das letras sendo bem distribuídas O algoritmo Bayesiano atrelou 86% de probabilidade da música ser triste, que foi resultado final, mostrado na Figura 18.

Já a música *Hellraiser*, do cantor Ozzy Osbourne, foi classificada como música feliz com mais de 99% de probabilidade, conforme Figura 19, mas possui palavras mais palavras com polaridade negativa como *fire*, *spell* e *liar*. A frequência do título também é alta na nuvem de palavras pois, assim como *Psychosocial*, o vocalista a repete diversas vezes na letra, porém ela não agrega nenhum valor na análise.

Um exemplo que demonstra bastante precisão do algoritmo é mostrado na Figura 20. A música *Left Behind*, do Slipknot, foi classificada como música feliz, e suas palavras em sua maioria possuem uma polaridade positiva.

Analisando a *playlist* de uma forma geral (sem selecionar músicas individualmente), é possível perceber que muitas músicas são classificadas como tristes, mas possuem palavras felizes (ou vice-versa). Isso demonstra as dificuldades descritas por Oudenne, Swarthmore e Chasins (2010), mostrando que não é evidente quais palavras poderiam representar conotações positivas, mas quando analisadas juntas, expressam um sentimento positivo.

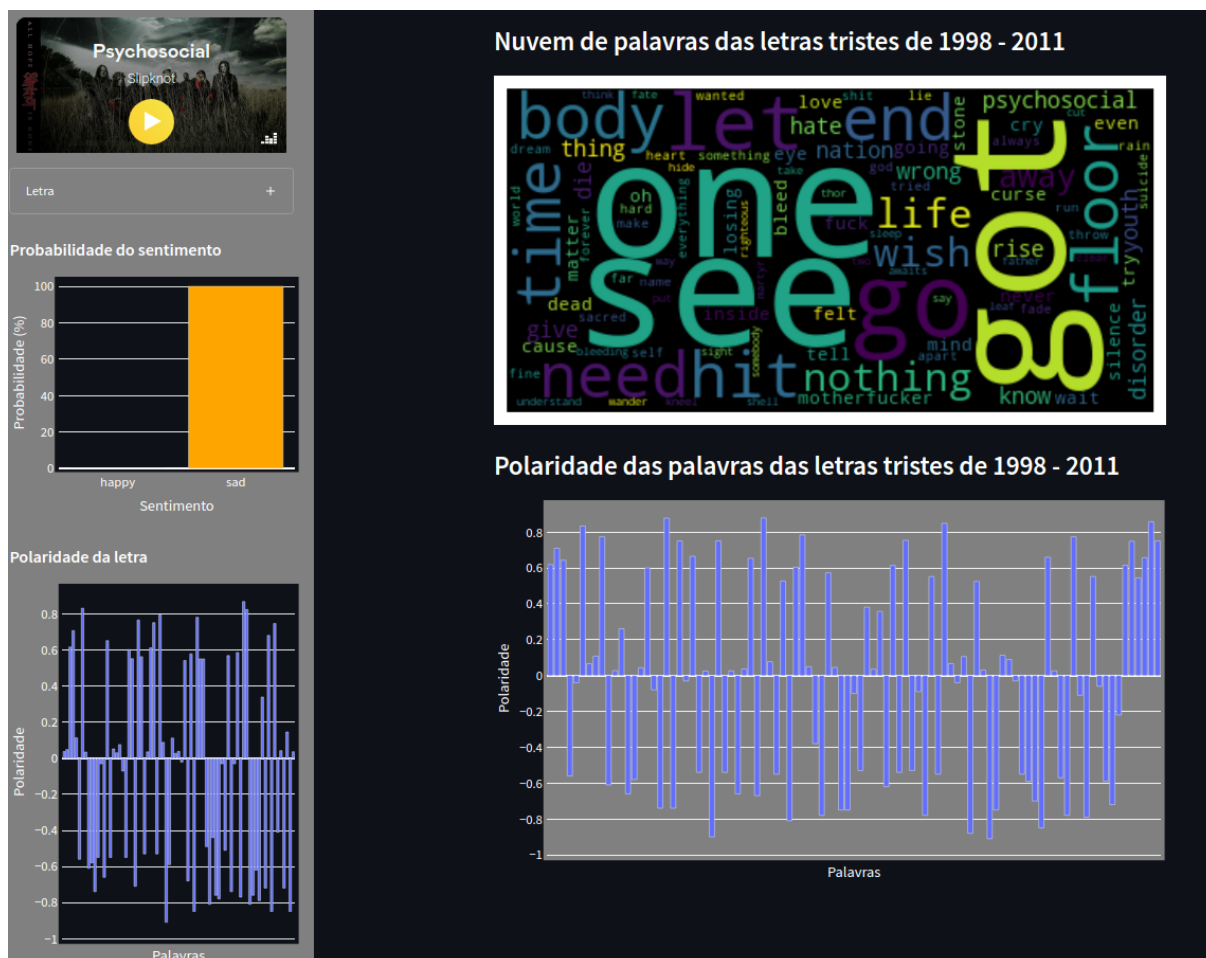
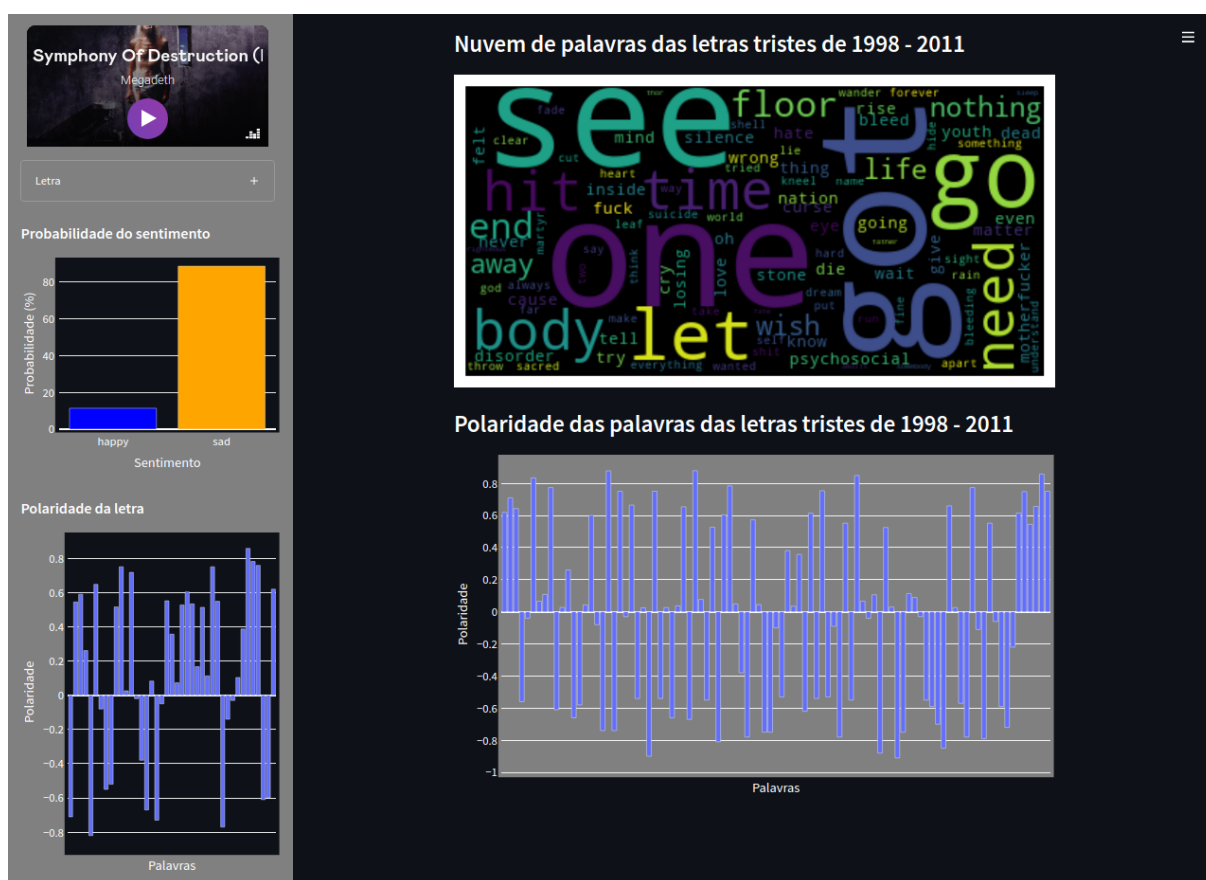


Figura 17 – Análise de Slipknot - *Psychosocial*

Fonte: Próprio autor

Figura 18 – Análise de Megadeth - *Symphony Of Destruction*

Fonte: Próprio autor



Figura 19 – Análise de Ozzy Osbourne - *Hellraiser*

Fonte: Próprio autor



Figura 20 – Análise de Slipknot - *Left Behind*

Fonte: Próprio autor

5.4 Trabalhos Futuros

A Análise de Sentimentos utilizando Aprendizado de Máquina considera o texto como uma sacola de palavras, e as classifica de acordo com a interação que produzem no texto. Como foi descrito nas dificuldades e na discussão, o contexto das letras não é levado em conta. Esse fator, unido com sarcasmo e ironia, podem gerar muitos falsos-positivos no sistema de classificação. Para classificações binárias como a deste trabalho, não há justificativa para aplicar técnicas mais complexas de análise, como Redes Neurais Artificiais, que por mais que dariam resultados muito mais precisos, são computacionalmente muito caros.

Todavia, a análise de contexto das letras pode, futuramente, ser levada em consideração, uma vez que uma letra de música é uma forma de arte de expressão do compositor, e não apenas uma forma de comunicação por texto.

Referências

AZZAM, T. et al. Data visualization and evaluation. **New Directions for Evaluation**, v. 2013, n. 139, p. 7–32, 2013. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/ev.20065>>.

BARBOSA, J. et al. Introdução ao processamento de linguagem natural usando python. **III Escola Regional de Informatica do Piauí**, v. 1, p. 336–360, 2017. Disponível em: <<http://www.facom.ufu.br/~wendelmelo/terceiros/>>.

BAYES, T. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. **Philosophical transactions of the Royal Society of London**, The Royal Society London, n. 53, p. 370–418, 1763. Disponível em: <<https://doi.org/10.1098/rstl.1763.0053>>.

COPE, A. L. **Black Sabbath and the rise of heavy metal music**. [S.l.]: Routledge, 2016.

COWEN, A. S. et al. What music makes us feel: At least 13 dimensions organize subjective experiences associated with music across different cultures. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 117, n. 4, p. 1924–1934, 2020. ISSN 0027-8424. Disponível em: <<https://www.pnas.org/content/117/4/1924>>.

DALE, R. Classical approaches to natural language processing. In: **Handbook of natural language processing, second edition**. [S.l.]: CRC Press, Taylor & Francis Group, 2010. p. 3–7.

FARIAS, D. H.; ROSSO, P. Chapter 7 - irony, sarcasm, and sentiment analysis. In: POZZI, F. A. et al. (Ed.). **Sentiment Analysis in Social Networks**. Boston: Morgan Kaufmann, 2017. p. 113–128. ISBN 978-0-12-804412-4. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128044124000073>>.

HABIBI, A.; DAMASIO, A. Music, feelings, and the human brain. **Psychomusicology: Music, Mind, and Brain**, v. 24, p. 92, 03 2014. Disponível em: <<https://psycnet.apa.org/record/2014-13444-009>>.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 1st. ed. USA: Prentice Hall PTR, 2000. ISBN 0130950696.

KUMAR, G. S. Data visualization-python. 2019. Disponível em: <http://dspace.cusat.ac.in/jspui/handle/123456789/13946?mode=simple&submit_simple>Show+simple+item+record>.

LOPER, E.; BIRD, S. Nltk: The natural language toolkit. In: **Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1**. USA: Association for Computational Linguistics, 2002. (ETMTNLP '02), p. 63–70. Disponível em: <<https://doi.org/10.3115/1118108.1118117>>.

MCKINNEY, D. **Magic circles: the Beatles in dream and history**. Cambridge, Massachusetts: Harvard University Press., 2003.

MCKINNEY, W. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491957662.

OUDENNE, A. M.; SWARTHMORE, P.; CHASINS, S. E. Identifying the emotional polarity of song lyrics through natural language processing. **Swarthmore College**, p. 1–14, 2010. Disponível em: <https://www.sccs.swarthmore.edu/users/11/aoudenn1/Ashley_M._Oudenne/Research.html>.

RIBAS, W. A.; NODA, E.; MARQUES, D. Bigpy: um sistema web para captura, tratamento e visualização de dados de defesa do consumidor utilizando python. **Trabalho de Conclusão de Curso - Instituto Federal de Educação, Ciência e Tecnologia de São Paulo**, 2018. Disponível em: <https://hto.ifsp.edu.br/portal/index.php/ensino/cursos/ads/ads-tcc#Sem1_2018>.

SADIKU, M. et al. Data visualization. **International Journal of Engineering Research And Advanced Technology (IJERAT)**, v. 2, n. 12, p. 11–16, 2016. Disponível em: <<https://ijerat.com/index.php/ijerat/article/view/191>>.

SOUZA, R. R.; CAFÉ, L. M. A. Análise de sentimento aplicada ao estudo de letras de música. **Informação & Sociedade: Estudos**, v. 28, n. 3, dez. 2018. Disponível em: <<https://periodicos.ufpb.br/ojs/index.php/ies/article/view/34884>>.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: <<https://doi.org/10.1093/mind/LIX.236.433>>.

WEIAND, A.; WEIAND, F. Análise de sentimentos do twitter com naïve bayes e nltk. **ScientiaTec**, v. 4, p. 46–57, 04 2018. Disponível em: <<https://periodicos.ifrs.edu.br/index.php/ScientiaTec/article/view/2188>>.