

RECONHECIMENTO DE GÊNEROS MUSICAIS COM TÉCNICAS DE APRENDIZAGEM DE MÁQUINA SUPERVISIONADA

Maria Fernanda Pimenta
Graduanda em Ciência da Computação – Uni-FACEF
maria.fpimenta7@gmail.com

Jaqueline Brigladori Pugliesi
Doutora em Ciência da Computação – USP São Carlos
jbpugliesi@gmail.com

Resumo

A música está presente na vida de pessoas do mundo todo, em diferentes meios, sendo o principal deles, o multimídia. Devido ao maior contato com a música ser por meio de áudio, não há muitos estudos envolvendo o reconhecimento de gêneros musicais utilizando letras de música, o qual é o objetivo deste trabalho. Esse projeto busca desenvolver um modelo preditivo, a fim de classificar gêneros musicais brasileiros, sendo eles: Sertanejo, MPB e Funk, utilizando a linguagem Python, processamento de linguagem natural, aprendizado de máquina e modelos de classificação multiclasse. O objetivo é construir um modelo com a maior precisão para mostrar que é possível reconhecer gêneros musicais por meio das letras das músicas e mostrar que eles não são definidos apenas pelo som. O resultado foi obtido por meio de um algoritmo de Regressão Logística com mais de 80% de acurácia na predição dos gêneros.

Palavras-chave: Música. Aprendizado de Máquina. Gênero Musical. Processamento de Linguagem Natural.

Abstract

Music is present in the lives of people all over the world, through different media, the main one being multimedia. Due to the contact with music with a greater number of songs, there are not many studies on song recognition using song lyrics, what is the purpose of this work. This project seeks to develop a predictive model in order to classify Brazilian musical genres, namely: Sertanejo, MPB and Funk, using the Python language, natural language processing, machine learning and multiclass classification models. The objective was to build a model with the greatest possible precision to show that it is possible to recognize musical genres through the lyrics of the songs and to show that they are not defined only by the sound. The result was obtained through a Logistic Regression algorithm with more than 80% accuracy in gender prediction.

Keywords: Music. Machine Learning. Musical Genre. Natural Language Processing.

1 Introdução

A música está presente no cotidiano de milhares de pessoas, seja direta ou indiretamente, sempre há algum recurso sonoro chegando a seus ouvidos. Hoje, com a grande quantidade de conteúdo multimídia disponível online, se faz necessário criar ferramentas capazes de organizar e gerenciar essa quantidade de informações, diz Fingerhut (1999).

A maior parte de dados musicais digitais são associados a arquivos de áudio e, por esse motivo, tais tipos de dados são o foco no que se refere a

classificação, organização, indexação, busca e recuperação da música. Neste cenário, os gêneros musicais possuem descrições fundamentais para o desenvolvimento de soluções para tal propósito, continua o autor.

Os trabalhos voltados para o reconhecimento de gêneros musicais se baseiam sempre em arquivos de áudio e características do som que o determinam ou utilizam de bases de dados com atributos musicais para tal tarefa. Ao pensar em utilizar as letras das músicas, logo aparecem diversos trabalhos para criação ou geração automática de letras, mas não para a classificação de um gênero musical.

O objetivo deste projeto é apresentar que é possível fazer essa classificação utilizando as letras das músicas de três gêneros musicais brasileiros, sendo eles: Sertanejo, MPB e Funk, visto que a música é muito mais que apenas a reprodução de um som, é um recurso de expressão, de comunicação, manifestação de crenças, singular e abstrata, como diz Sekeff (2002).

O estudo envolveu uma contextualização acerca da música, seus diversos significados, os gêneros musicais e quais elementos os constituem. Além de uma fundamentação a respeito de aprendizagem de máquina supervisionada, seus algoritmos e processamento de linguagem natural para a construção dos modelos preditivos.

Para o desenvolvimento do projeto foi utilizada a base de dados *Song lyrics from 79 musical data genres* (Neisse, 2019), a linguagem de programação Python e ambiente de desenvolvimento Google Colab.

Este artigo está organizado da seguinte maneira: na Seção 2 é abordado o referencial teórico, e na Seção 3 as ferramentas e métodos utilizados. Na Seção 4 está o passo a passo do desenvolvimento, como pré-processamento e treinamento dos modelos. Na Seção 5 são apresentados os resultados e, na Seção 6, são feitas as considerações finais acerca deste projeto.

2 Referencial Teórico

Nesta seção serão abordados o tema do projeto, suas aplicações, conceitos, fundamentos e técnicas utilizadas em seu desenvolvimento.

2.1 Música

Existem inúmeras as abordagens para definir o significado de música. Para Edgard Varése (apud GOLDMAN, 1961, p.133) o som é uma matéria viva e de espaço musical mais aberto do que limitado. Sendo assim, a música nada mais é do que um som organizado.

Segundo Esteves (2019), a música é a arte originada da organização e manipulação dos sons em um discurso lógico, ou seja, que tem um começo e um fim. Tais sons podem ser produzidos pelo corpo humano, por instrumentos musicais elétricos ou digitais e até mesmo por sons vindos da natureza, desde que organizados de maneira lógica.

Apesar de diferentes, as definições dadas por ambos os autores são bastante semelhantes. Estes afirmam que a música é originada a partir de qualquer som organizado, mas que para aqueles que possuem ouvidos condicionados, um som novo sempre é chamado de ruído. Com isso, é possível observar que o conceito de música também depende daqueles que a ouvem e de sua interpretação pessoal.

A palavra música é originada da expressão grega *musiké téchne*, que significa a arte das musas. Na mitologia clássica, as musas são ditas como qualquer uma das nove deusas irmãs que presidiam as artes e as ciências. Sendo assim, é possível dizer que para os gregos, a música estava relacionada às atividades que dissessem respeito à busca da beleza e da verdade, além de uma ligação direta com o mundo divino e a harmonia. (SOUZA, 2011).

Para Sekeff (2002), a linguagem musical não é apenas um recurso de combinação e exploração de ruídos e sons, mas é também um recurso de expressão, de comunicação, de gratificação, de mobilização e de autorrealização, o que abrange diversas partes de um indivíduo, como seus sentimentos, valores, cultura, emocional e físico.

De acordo com Pinto (2001), música é a manifestação de crenças, de identidades, e é considerada universal no que se refere à sua existência e importância em qualquer sociedade. Ao mesmo tempo, é singular e de difícil tradução quando apresentada fora de seu contexto ou meio cultural. Ela é abstrata e imaterial, mas tem um efeito corporal e emocional, podendo causar diversas sensações a quem a escuta.

Sendo assim, é possível afirmar que a música é a combinação de quaisquer sons e de um conjunto de informações que podem ser listadas como elementos musicais, tais como melodia, harmonia, ritmo, timbre, forma e textura. Junto a isso, há também os aspectos sociais e culturais referente ao local, época e ao povo, que trazem o toque humano sobre a musicalidade.

Esteves (2019) diz que qualquer som produzido possui sempre quatro propriedades, sendo especificadas a seguir:

1. Intensidade: caracteriza se um som é forte ou fraco de acordo com a energia que a onda sonora transfere.

2. Duração: é o parâmetro que permite diferenciar sons longos de sons curtos.

3. Altura: caracteriza se um som é agudo ou grave pela frequência do som, ou seja, quanto maior, mais agudo (alto) e quanto menor, mais grave (baixo).

4. Timbre: é o parâmetro que permite distinguir sons de mesma frequência, produzidos por instrumentos diferentes.

Segundo Esteves (2019), é a partir dessas propriedades e da manipulação dos sons que a música começa a criar forma. Além dessas propriedades, há também três elementos básicos que fazem parte da estrutura musical, que são:

1. Melodia: é uma sequência linear de notas musicais.

2. Harmonia: se refere ao encontro simultâneo das notas musicais, o qual é denominado acorde.

3. Ritmo: diz respeito à duração de uma série de notas, assim como se agrupam as unidades.

No entanto, não basta apenas unir esses elementos, é necessário organizá-los em partes ou seções para que a música faça sentido. A essa organização é dado o nome de forma, a qual é um elemento fundamental no sentido de organizar a escuta. Formas comuns utilizam alguns nomes como introdução, verso, refrão e sonatas, de acordo com Esteves (2019).

2.2 Gênero Musical

De acordo com Correa (2019, p.2),

Os gêneros musicais são reconhecidos como as diversas nomenclaturas utilizadas para agrupar peças musicais que conservam alguma similaridade entre si. No mercado musical, por exemplo, termos como samba, pagode, sertanejo, rock, pop, blues, entre outros, dão nome a agrupamentos de grandes obras musicais semelhantes, perfazendo aquilo que comumente é entendido como gênero musical.

Para Denizeau (2005), o gênero não deve ser confundido com a forma musical, pois ele é o objeto musical, o resultado dos elementos constituintes das composições e interpretações musicais que os diferenciam aos ouvidos do ouvinte, enquanto a forma serve para estruturar e dar coerência à composição, mas não é suficiente para designar o gênero.

Outros autores, tais como Fabbri (2017), Andrade (1989) e Zamacois (1984), também demonstram que forma e gênero são conceitos diferenciados por meio da afirmação de que a forma musical é um dos elementos constituintes em um gênero, mas que somente um conjunto de elementos musicais, culturais, sociais, políticos e mercadológicos, juntamente com a forma, pode caracterizá-lo.

Segundo Bennett (1986), quando um compositor está escrevendo uma peça de música, ele está combinando diversos elementos musicais importantes, dentre eles: melodia, harmonia, ritmo, timbre, e forma. Há também a textura, que determina a qualidade sonora global de uma peça, de acordo com a forma em que os materiais melódicos, rítmicos e harmônicos se combinam na composição.

Para designar a maneira pela qual compositores de épocas, países e períodos diferentes apresentam os elementos citados em suas obras, emprega-se a palavra estilo, que é a maneira particular como esses componentes são equilibrados, tratados e combinados e faz com que uma peça tenha o estilo de determinado período, além de fornecer itens para compor sua ficha de identificação, diz Bennett (1986).

Dessa maneira, pode-se concluir que não há um consenso a respeito do que é gênero musical, mas que este deve ser diferenciado de forma musical e que é um conceito atrelado a um conjunto de funções e significados decorrentes de diversos interesses, atribuídos às manifestações musicais, as quais englobam o lado social, cultural e mercadológico da música.

2.3 Inteligência Artificial

Segundo Kaufman (2018, p.16) a inteligência artificial (IA) “refere-se a um campo de conhecimento associado à linguagem e à inteligência, ao raciocínio, à aprendizagem e à resolução de problemas”. Suas aplicações são diversas, como veículos autônomos, robótica, diagnósticos médicos, entre outros. Atualmente, os sistemas inteligentes estão em quase toda a vida em sociedade.

Para os autores Russel e Norvig (2013), a inteligência artificial é dividida em duas dimensões:

1. Processos de pensamento e raciocínio, na qual há a intenção de automatizar atividades associadas ao pensamento humano, atividades como tomada de decisões, a resolução de problemas, o aprendizado, além do estudo das computações que tornam possível agir e raciocinar como um ser humano.

2. Processos comportamentais, tais como a criação de máquinas que podem executar funções e tarefas que exigem inteligência e são mais bem desempenhadas pelas pessoas.

Alguns cientistas e pesquisadores atribuem a primeira referência à inteligência artificial ao matemático e cientista da computação inglês Alan Turing, em seu famoso artigo *Computing Machinery and Intelligence* (1950), no qual ele questionava se máquinas poderiam pensar. Turing se empenhou em descobrir se seria possível construir uma máquina capaz de simular inteligência, ideia concretizada anos depois por Von Neumann.

Conclui-se que apesar das diferentes definições, é possível afirmar que a inteligência artificial revela um campo de estudo amplo e promissor no que se refere a entender e aplicar técnicas inteligentes para a solução de problemas, reconhecimento de padrões, execução de raciocínio e de aprendizado, como descrito por Luger (2013).

2.4 Aprendizado de Máquina

Para Rezende (2003), aprendizado de máquina (AM) é uma área da inteligência artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

De acordo com Faceli et al. (2011, p.2), aprendizado de máquina é o nome dado ao “processo de indução de uma hipótese (ou aproximação de função) a partir da experiência passada”, o qual faz parte de uma área de pesquisa da IA que visa o desenvolvimento de programas de computador com a capacidade de aprender a executar uma dada tarefa com sua própria experiência.

Utilizando um conjunto de dados que representam experiências passadas, os programas são capazes de aprender por si só. Trata-se de uma área de pesquisa multidisciplinar que engloba probabilidade e estatística, psicologia, teoria computacional, entre outros.

O aprendizado de máquina pode ser dividido em duas categorias: não supervisionado e supervisionado, que será utilizado neste projeto.

2.5 Aprendizado Supervisionado

Segundo Fontana (2020), os algoritmos de aprendizado de máquina supervisionado relacionam a entrada e saída de um sistema, baseado em um conjunto de amostras de treinamento com dados rotulados. Ou seja, já se sabe qual a saída correta e, baseado na entrada informada, o algoritmo deve prever qual o rótulo de saída.

Nesse aprendizado se encontram os chamados algoritmos preditivos, os quais proveem uma descrição compacta de um conjunto de dados e podem ser divididos em algoritmos de classificação, que são aqueles em que os rótulos são pré-definidos (o que chamamos de classe) e os algoritmos de regressão, nos quais o rótulo de saída pode assumir qualquer valor real.

2.5.1 Algoritmos de Classificação

Segundo Silva (2007), classificação é uma técnica de inteligência artificial caracterizada por ter entre seus arquivos de treinamento e teste um atributo chamado classe. Normalmente, este atributo é o objetivo esperado quando se utiliza esta técnica.

Existem diversos algoritmos classificadores para utilização, tendo as redes neurais e as árvores de decisão (puras e derivadas) entre as mais conhecidas. Ambas as técnicas são de grande praticidade e bastante utilizadas no que se refere a classificação de gêneros musicais.

2.5.2 Algoritmos de Regressão

Os algoritmos de regressão têm por objetivo prever um valor numérico, ou seja, identificar uma categoria a partir de uma faixa contínua de valores. Na regressão, é preciso identificar a relação entre os parâmetros de entrada e saída e sua exatidão é calculada com base nos dados de saída precisos e previstos, diz Fontana (2020).

Alguns exemplos de algoritmos desta categoria são a regressão linear e a regressão logística, podendo esta ser binominal, ordinal ou multinominal.

2.6 Aprendizado Não Supervisionado

De acordo com Luger (2013), o aprendizado não supervisionado requer que o próprio algoritmo de aprendizado avalie os conceitos apresentados. Neste caso, não há categorização ou rotulação dos dados e o algoritmo deve tentar entendê-los por conta própria.

Dessa forma, a máquina busca agrupar os dados de acordo com semelhanças, hierarquias e encontrar estruturas e padrões ocultos. Ao contrário do aprendizado supervisionado não há um professor, nem caminho certo ou errado, apenas uma análise pura dos dados.

2.7 Processamento de Linguagem Natural

De acordo com Vieira e Lopes (2010, p.184), processamento de linguagem natural (PLN) é uma área da computação que “estuda o desenvolvimento de programas de computador que analisam, reconhecem e/ou geram textos em linguagens humanas, ou linguagens naturais”.

A linguagem natural é o meio de comunicação dos seres humanos, seja por texto ou como um ato de fala. Segundo Luger (2013), a compreensão da linguagem não é apenas a transmissão de palavras, ela depende do conhecimento, das expectativas dentro do domínio do discurso, das suposições do locutor e do contexto da interação.

Devido à complexidade desse assunto, diferentes níveis de análise citados pelo autor foram definidos para a linguagem natural, sendo eles:

1. Prosódia: trata-se do ritmo e entonação da linguagem, mas é difícil de ser formalizada e normalmente é negligenciada.

2. Fonologia: examina os sons que são combinados para formar a linguagem. Importante para reconhecimento e geração computadorizada da fala.

3. Morfologia: é a preocupação com os componentes que constituem uma palavra. Importante para determinar o papel da palavra em uma sentença.

4. Sintaxe: é o estudo de regras para a estruturação de uma frase de acordo com a combinação de palavras. Esse componente obteve a automatização mais bem-sucedida da análise linguística.

5. Semântica: considera o significado de palavras, frases e sentenças e as formas pelas quais ele é transmitido em expressões.

6. Pragmática: estuda as formas de uso da linguagem e seus efeitos sobre o interlocutor.

7. Conhecimento do mundo: inclui o conhecimento do mundo físico, da interação social humana e do papel de objetivos e intenções na comunicação. Tal conhecimento é essencial para o entendimento completo.

A rica ambiguidade da linguagem natural faz com que o PLN não seja uma tarefa fácil e o torna diferente do processamento das linguagens de programação de computador, as quais são formalmente definidas para que não exista ambiguidade, dizem Vieira e Lopes (2010).

3 Ferramentas e Métodos

Nesta seção serão apresentadas as ferramentas e as tecnologias utilizadas para o desenvolvimento do projeto.

3.1 Python

Python é uma linguagem de programação de alto nível, dinâmica, interpretada, multiplataforma e orientada a objetos. Conhecida por ter a sintaxe relativamente simples e de fácil compreensão, tem popularidade entre profissionais da indústria tecnológica que não são especificamente programadores, como cientistas de dados, engenheiros, matemáticos, entre outros.

Um de seus maiores atrativos é possuir muitas bibliotecas, nativas e de terceiros, o que a tornou muito difundida e útil em uma grande variedade de setores dentro do desenvolvimento *web* e em áreas como análise de dados, aprendizado de máquina e inteligência artificial, segundo Roveda (2020). Para este projeto, foram utilizadas as seguintes bibliotecas:

1. Pandas: *open-source* e de uso gratuito, fornece inúmeras ferramentas para manipulação e análise de dados. Suas vantagens são o desempenho e alta produtividade.

2. Scikit-learn: desenvolvida especificamente para aplicação prática de aprendizado de máquina, dispõe de ferramentas simples e eficientes para análise preditiva de dados.

3. Nltk (*Natural Language Toolkit*): utilizada para realizar tarefas de PLN, nela se encontram *datasets* e alguns algoritmos essenciais para análise e pré-processamento textual.

3.2 Google Colaboratory

O Google Colaboratory, mais conhecido como Colab, é um serviço de nuvem gratuito hospedado pelo Google com o objetivo de incentivar pesquisas de

Aprendizado de Máquina e Inteligência Artificial, sendo possível escrever e executar códigos Python (COLABORATORY, 2021).

Essa ferramenta tem o formato de *notebooks* (cadernos) e permite combinar código executável com texto rico no mesmo documento, além de imagens, HTML (Linguagem de Marcação de HiperTexto), entre outros. Ao serem criados, os *notebooks* são automaticamente armazenados na conta do usuário no Google Drive e é possível compartilhá-los facilmente para que outras pessoas façam comentários ou até mesmo editem o documento.

Com o Colab é possível importar um conjunto de dados e utilizar as diversas bibliotecas Python para analisar, treinar, classificar e avaliar o modelo de dados com poucas linhas de código. Além disso, a ferramenta é amplamente utilizada pela comunidade de aprendizado de máquina em diversas aplicações (Colaboratory, 2021).

3.3 Kaggle

O Kaggle é uma plataforma de aprendizagem e prática de ciência de dados que abrange níveis variados de conhecimento entre seus usuários. A plataforma é bastante conhecida por suas competições, as quais podem ser públicas, privadas ou acadêmicas, algumas delas oferecem prêmios em dinheiro pela melhor solução (KAGGLE, *sd, online*).

Além disso, o Kaggle também funciona como um grande repositório de *datasets* (conjuntos de dados) com variados tipos de dados, os quais podem ser utilizados para criar algoritmos de aprendizado de máquina, treinar habilidades em análise de dados, entre outros. Qualquer pessoa pode criar um *dataset* ou uma competição na plataforma.

A ferramenta também conta com um ecossistema para se envolver, conectar, colaborar e trocar informações com outros usuários. Por conta de todos esses itens, o Kaggle se torna uma ótima opção para encontrar conjuntos de dados de diversos assuntos e se aprofundar no aprendizado de máquina, além de ser possível desenvolver os códigos dentro de *notebooks* na própria plataforma (KAGGLE, *sd, online*).

3.4 Base de Dados

Para realizar a análise, foi selecionada a base de dados *Song lyrics from 79 musical data genres* (NEISSE, 2019), obtida da plataforma Kaggle. A base é dividida em dois *datasets*, um contendo os dados dos artistas, como nome, gênero musical, quantidade de músicas e popularidade, e outro contendo os dados das letras das músicas dos artistas.

O *dataset Artists* possui 5 atributos, sendo eles: *Artists*, *Genres*, *Songs*, *Popularity* e *Link*, e 4.168 dados de artistas do mundo todo, abrangendo mais de 50 idiomas e 79 gêneros musicais. O *dataset Lyrics* também possui 5 atributos, os quais são: *ALink*, *SName*, *SLink*, *Lyric* e *language*, e dados de 379.931 músicas.

Ao somar os dados de ambos os *datasets*, obtêm-se um conjunto de dados extremamente volumoso e com grande quantidade de informações.

3.5 Mineração de Textos

De acordo com Morais e Ambrósio (2007), a tecnologia de mineração de textos (*Text Mining*) vem das técnicas de Recuperação de Informações (RI) e utiliza técnicas de análise e extração de dados a partir de bases textuais, de frases ou apenas palavras. Com base no conhecimento extraído, a mineração de textos define técnicas de extração de padrões ou tendências de grandes volumes de textos em linguagem natural.

Para a melhor utilização dos dados selecionados para o desenvolvimento do projeto, foi necessária a aplicação de algumas técnicas de pré-processamento da mineração de texto, as quais são descritas a seguir:

1. Tokenização: consiste em percorrer todo o texto identificando cada palavra entre as sequências de caracteres e fazendo sua decomposição. Cada palavra ou termo identificado se torna um *token*, que possuem significados quando analisados de forma isolada, diz Soares (2008).

2. Remoção de *Stopwords*: de acordo com Neves, Corrêa e Cavalcanti (2013), trata-se de remover palavras funcionais que geralmente não agregam valor ao texto, como artigos, conjunções, pronomes, verbos auxiliares, entre outros.

3. TF-IDF (*Term Frequency – Inverse Document Frequency*): é um modelo que cria um representativo numérico, levando em consideração a frequência do termo no documento e o número de documentos da coleção em que o termo aparece, segundo Sanga (2017).

3.6 Modelos de Aprendizado

Após o tratamento dos dados, foi necessário realizar o treinamento dos algoritmos e, para isso, foram escolhidos os modelos que se enquadravam melhor na classificação multiclasse, visto que a base de dados utilizada é composta por mais de dois gêneros musicais. Sendo assim, foram escolhidos os seguintes algoritmos:

1. Análise Discriminante Linear (LDA): de acordo com Barbosa, Filho e Teles (2019, p. 353), “LDA é um método usado em estatística e aprendizado de máquina para encontrar uma combinação linear de características que caracterizam ou separam duas ou mais classes de objetos ou eventos”.

2. Regressão Logística: é uma técnica que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica, em função de uma ou mais variáveis independentes, diz Gonzalez (2018).

3. K-Vizinhos Mais Próximos (KNN): atua na classificação de objetos desconhecidos, ou seja, para cada objeto ainda não classificado (desconhecido), o algoritmo verifica as classes dos *k* objetos mais próximos e aloca esse objeto a uma das classes, segundo Semaan et al. (2019).

4. Árvores de Decisão: de acordo com Crepaldi et al. (2010), nesse algoritmo, o atributo mais importante é apresentado na árvore como o primeiro nó, e os atributos menos relevantes são mostrados nos nós subsequentes. Sua principal vantagem é a tomada de decisões levando em consideração tais atributos.

5. Máquinas de Vetores de Suporte (SVM): podem resolver problemas de classificação e regressão, adquirindo a capacidade de generalização durante a etapa de treinamento, diz Oliveira (2010).

6. Floresta Aleatória: baseado em árvores de decisão, lida bem com conjuntos de dados de alta dimensão. Pode ser usado para classificação e regressão,

estudo de importância e seleção de variáveis, e detecção de *outliers* (valor atípico), segundo Júnior (2018).

4 Desenvolvimento

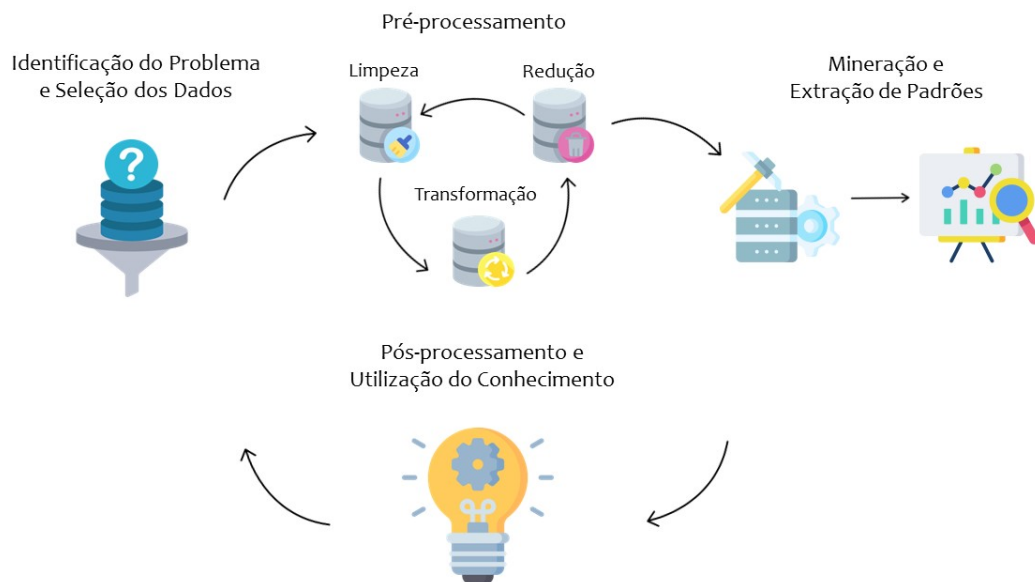
Esta seção tem por objetivo apresentar o ciclo de desenvolvimento seguido no projeto e o detalhamento de cada uma das etapas a serem apresentadas, como a seleção dos dados, o pré-processamento, a mineração e utilização do conhecimento.

4.1 Ciclo de Desenvolvimento

A primeira etapa do desenvolvimento é a identificação do problema e a seleção dos dados a serem utilizados. Em seguida, é realizado o pré-processamento, no qual será feita a limpeza, transformação e redução do volume de dados, para que estes estejam adequados para a extração de padrões, que é a próxima etapa do processo.

Na extração de padrões é feita a extração e a transformação de dados não estruturados, além da seleção dos algoritmos de aprendizagem a serem utilizados. Por fim, no pós-processamento, é feita a análise dos resultados de treinamento e a utilização do conhecimento, podendo repetir todo o processo caso os resultados não sejam os esperados.

Figura 1 - Ciclo de Desenvolvimento



Fonte: a autora.

4.2 Seleção dos Dados

Conforme citado na Subseção 3.4, foi selecionada a base *Song lyrics from 79 musical data genres* (NEISSE, 2019) para o desenvolvimento do projeto. Os

critérios para a escolha foram a consistência, o volume, a diversidade e a facilidade de manipulação dos dados.

Os dois datasets (*artists e lyrics*) foram baixados na plataforma Kaggle e depois armazenados no Google Drive de forma que pudessem ser integrados mais facilmente ao Google Colab. Já na ferramenta, os arquivos puderam ser extraídos para dar início ao pré-processamento.

4.3 Pré-Processamento

Uma vez que foram utilizados dois *datasets* diferentes, o pré-processamento foi dividido em 4 partes: a limpeza do *dataset artists*, do *lyrics* e do novo *dataset* criado a partir da junção dos dois anteriores, o qual deve conter apenas as letras e os gêneros e, por último, o tratamento do texto.

4.3.1 Artists

Visto que um mesmo artista poderia ser classificado em mais de um gênero musical, a primeira etapa do pré-processamento desse conjunto foi remover a multiplicidade de gêneros da coluna Genres, mantendo apenas um gênero por artista, de modo a facilitar a classificação.

Em seguida, foi feita a remoção das colunas não relevantes para a aprendizagem, como o nome do artista, a quantidade de músicas e a popularidade, além da exclusão de valores duplicados ou redundantes. As Figuras 2 e 3 ilustram o conjunto antes e depois da limpeza, respectivamente.

Figura 2 - Cinco primeiros exemplos do conjunto *Artists* antes da limpeza

	Artist	Genres	Songs	Popularity	Link
0	Ivete Sangalo	Pop; Axé; Romântico	313.0	4.4	/ivete-sangalo/
1	Chiclete com Banana	Axé	268.0	3.8	/chiclete-com-banana/
2	Banda Eva	Axé; Romântico; Reggae	215.0	2.3	/banda-eva/
3	É O Tchan	Axé	129.0	1.6	/e-o-tchan/
4	Claudia Leitte	Pop; Axé; Romântico	167.0	1.5	/claudia-leitte/

Fonte: a autora.

Figura 3 - Cinco primeiros exemplos do conjunto *Artists* após a limpeza

	Genres	Link
0	Pop	/ivete-sangalo/
1	Axé	/chiclete-com-banana/
2	Axé	/banda-eva/
3	Axé	/e-o-tchan/
4	Pop	/claudia-leitte/

Fonte: a autora.

4.3.2 Lyrics

Devido ao grande volume da base de dados, foi necessário escolher apenas um idioma para a análise e o reconhecimento dos gêneros musicais, visto que está sendo utilizado o plano gratuito do Google Colab, o qual possui um limite de memória RAM a ser utilizado. Sendo assim, todos os gêneros diferentes de português foram excluídos da base, passando de 379.931 músicas para 157.393. Em seguida, foi feito o mesmo processo do conjunto anterior, a remoção de valores duplicados ou redundantes e das colunas não relevantes, as quais foram o nome da música, do artista e o idioma.

Inicialmente o conjunto estava conforme a Figura 4 e, após a limpeza, ficou como mostrado na Figura 5.

Figura 4 - Cinco primeiros exemplos do conjunto *Lyrics* antes da limpeza

	ALink	SName	SLink	Lyric	language
0	/ivete-sangalo/	Arerê	/ivete-sangalo/arere.html	Tudo o que eu quero nessa vida,\nToda vida, é\...	pt
1	/ivete-sangalo/	Se Eu Não Te Amasse Tanto Assim	/ivete-sangalo/se-eu-nao-te-amasse-tanto-assim...	Meu coração\nSem direção\nVoando só por voar\n...	pt
2	/ivete-sangalo/	Céu da Boca	/ivete-sangalo/chupa-toda.html	É de babaixá!\nÉ de balacubaca!\nÉ de babaixá!...	pt
3	/ivete-sangalo/	Quando A Chuva Passar	/ivete-sangalo/quando-a-chuva-passar.html	Quando a chuva passar\n\nPra quê falar\nSe voc...	pt
4	/ivete-sangalo/	Sorte Grande	/ivete-sangalo/sorte-grande.html	A minha sorte grande foi você cair do céu\nMin...	pt

Fonte: a autora.

Figura 5 - Cinco primeiros exemplos do conjunto *Lyrics* após a limpeza

	ALink	Lyric
0	/ivete-sangalo/	Tudo o que eu quero nessa vida,\nToda vida, é\...
1	/ivete-sangalo/	Meu coração\nSem direção\nVoando só por voar\n...
2	/ivete-sangalo/	É de babaixá!\nÉ de balacubaca!\nÉ de babaixá!...
3	/ivete-sangalo/	Quando a chuva passar\n\nPra quê falar\nSe voc...
4	/ivete-sangalo/	A minha sorte grande foi você cair do céu\nMin...

Fonte: a autora.

4.3.3 Junção dos *datasets*

Finalizado o pré-processamento separado de cada um dos conjuntos, foi feita a junção dos dois *datasets* por meio das colunas *Link* e *ALink*. Essas colunas tem a mesma função de um id ou código de referência, mas com o nome dos artistas e servem para fazer a ligação entre os *datasets*. Após isso, ambas foram removidas da tabela, e realizada nova verificação de valores nulos, redundantes ou duplicados e os que se enquadravam nessas categorias foram excluídos. A Figura 6 ilustra o *dataset* final.

Figura 6 - Cinco primeiros exemplos do conjunto final após a junção

	Lyric	Genres
0	Tudo o que eu quero nessa vida,\nToda vida, é\...	Pop
1	Meu coração\nSem direção\nVoando só por voar\n...	Pop
2	É de babaixá!\nÉ de balacubaca!\nÉ de babaixá!...	Pop
3	Quando a chuva passar\n\nPra quê falar\nSe voc...	Pop
4	A minha sorte grande foi você cair do céu\nMin...	Pop

Fonte: a autora.

Para facilitar a classificação e o tratamento dos dados, foram selecionados apenas 3 gêneros para o treinamento, sendo eles: Sertanejo, MPB e Funk (Figura 7). Após essa definição, foi criado um novo *dataset* com 4.000 exemplos de cada um desses gêneros, totalizando 16.000.

Figura 7 - Selecionando os gêneros musicais

```
sertanejo = df2[df2['Genres']=="Sertanejo"].sample(n=4000, random_state=1)
mpb = df2[df2['Genres']=="MPB"].sample(n=4000, random_state=1)
funk = df2[df2['Genres']=="Funk"].sample(n=4000, random_state=1)

df_final = pd.concat([sertanejo, mpb, funk]).reset_index()
```

Fonte: a autora.

4.3.4 Tratamento do Texto

Para um treinamento eficiente dos modelos, foi necessário realizar algumas tratativas no texto das letras da coluna *Lyric*. O primeiro passo foi transformar todas as letras para o tipo *string* (sequência de caracteres) e em seguida colocá-las em minúsculo (Figura 8), de modo que o modelo não identifique palavras iguais como diferentes.

Figura 8 - Utilização da função *lower*

```
[8] # colocando todas as letras em minúsculo
    low = [lyr.lower() for lyr in lyrics]
    low
```

Fonte: a autora.

O próximo passo foi realizar a tokenização do texto, utilizando o método *word_tokenize* da biblioteca *nlk* (Figura 9), para que cada palavra ou termo se torne um *token* e possa ser realizada a remoção das *stopwords* (Figura 10) por meio do método *stopwords.words*, na qual seu parâmetro é o idioma do texto.

Figura 9 - Utilização da função `word_tokenize`

```
# tokenizando as letras
tokenized = [word_tokenize(lyr) for lyr in low]
```

Fonte: a autora.

Figura 10 - Utilização da função `stopwords.word`

```
# filtrando stopwords
stop_words = set(stopwords.words('portuguese'))
stop_vec = []
for tok in tokenized:
    filtered_stop = [w for w in tok if w not in stop_words]
    stop_vec.append(filtered_stop)
```

Fonte: a autora.

Após isso, foi feita a remoção de acentos e caracteres especiais do texto, utilizando as funções `unidecode` e `isalpha` (Figura 11). Em seguida, as palavras são separadas por espaço. A Figura 12 exemplifica o resultado final.

Figura 11 - Utilização das funções `unidecode` e `isalpha`

```
# removendo a pontuação

clean_vec = []
for lyr in stop_vec:
    clean_lyr = [unidecode(word) for word in lyr if word.isalpha()]
    clean_vec.append(clean_lyr)
```

Fonte: a autora.

Figura 12 - Resultado após tratamento do texto

```
'tentei levar serio prometi apaixonar sempre soube relacionamento queria respeitar deu passei vida inteira procurando encontrei beijo
o prometeu sinto perdeu duvido corpo encaixa igual encaixa comigo passo dia ver mando mensagem ei sumido tarde passa la casa hoje dia
ido corpo encaixa igual encaixa comigo passo dia ver mando mensagem ei sumido tarde passa la casa hoje dia dar perdido dar perdido'
```

Fonte: a autora.

O último passo realizado é a vetorização das palavras com a utilização do TF-IDF (Figura 13), e são passados dois parâmetros, para que palavras que apareçam menos de 5 vezes ou em mais de 80% das letras não sejam adicionadas ao vetor. Tal ação foi necessária visto que palavras que aparecem em todos os gêneros podem atrapalhar o modelo de treinamento, pois este não será capaz de definir a qual classe a palavra pertence e palavras que pouco aparecem não terão relevância para o treinamento do algoritmo.

Após isso, um *dataset* pronto para o treinamento do modelo, contendo os dados da vetorização é criado (Figura 14), no qual cada coluna é uma palavra, as linhas continuam sendo as letras e a coluna final é a classe (*Genres*).

Figura 13 - Utilização do TF-IDF

```
# não adiciona palavras que aparecem menos de 5 vezes e que aparecem em mais de 80% das letras
vectorize = TfidfVectorizer(min_df=5, max_df=0.8)
vectors = vectorize.fit_transform(lyrics)
dense = vectors.todense()
dense_list = dense.tolist()
df_words = pd.DataFrame(dense_list)
```

Fonte: a autora.

Figura 14 – Cinco primeiros exemplos do dataset pronto para treino do modelo

	aa	aaah	aah	aahh	aai	aba	abaixa	abaixo	abajur	abala	...	zoologico	zorra	zua	zuando	zuar	zueira	zuera	zum	zumbi	Genres		
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Sertanejo	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Sertanejo
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Sertanejo
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Sertanejo
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	Sertanejo

Fonte: a autora.

4.4 Mineração e Treinamento dos Modelos

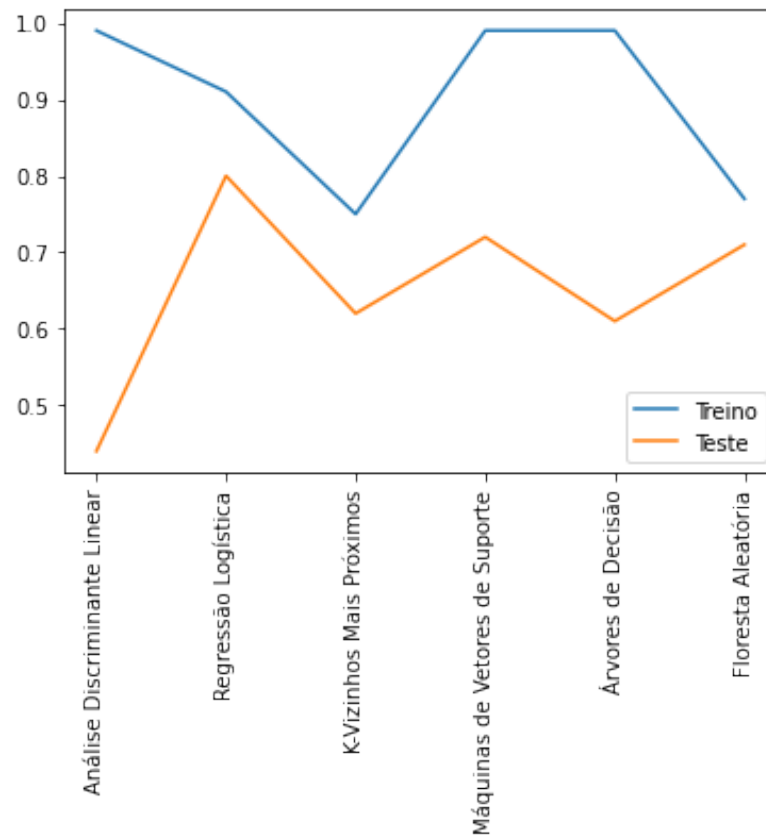
Para obter os melhores resultados na classificação dos gêneros musicais, 6 modelos de algoritmos foram treinados e testados. A Tabela 1 contém os números detalhados do resultado do treinamento. Já o gráfico da Figura 15 foi criado para que seja possível visualizar com mais facilidade os resultados de precisão dos modelos da diferença entre cada um dos algoritmos.

Tabela 1 - Resultados de precisão dos modelos após o treinamento

Modelo	Treino	Teste
Análise Discriminante Linear	0,99	0,44
Regressão Logística	0,91	0,80
K-Vizinhos Mais Próximos	0,75	0,62
Máquinas de Vetores de Suporte	0,99	0,72
Árvores de Decisão	0,99	0,62
Floresta Aleatória	0,77	0,71

Fonte: a autora.

Figura 15 - Gráfico dos resultados de precisão dos modelos após o treinamento



Fonte: a autora.

Com a tabela e o gráfico, é possível analisar qual modelo obteve o melhor resultado, o qual foi o algoritmo de Regressão Logística com os maiores valores e menor distância entre treinamento e teste. O gráfico também indica a possibilidade de que os modelos de LDA, SVM e Árvores de Decisão tenham se encaixado no quadro de *overfitting*, o que significa que eles podem ter decorado os padrões dos dados no treinamento e não acertaram os testes. Já o KNN e Floresta Aleatória ficaram com valores medianos

Sendo assim, o modelo escolhido para dar continuidade aos testes e classificação foi o de Regressão Logística. Utilizando as métricas da biblioteca scikit-learn, foi possível gerar a matriz de confusão para o algoritmo, com o detalhamento de quais foram os acertos e os erros da predição (Figura 16).

Figura 16 - Matriz de confusão

col_0	Funk	MPB	Sertanejo
Genres			
Funk	1005	86	87
MPB	50	969	167
Sertanejo	95	234	907

Fonte: a autora.

Com base na matriz de confusão, foi possível gerar algumas métricas adicionais e igualmente importantes para a análise do modelo, como a acurácia, que diz quanto o modelo acertou das previsões possíveis, a precisão, que define o quão bem o modelo trabalhou, o *recall*, que define o quão bom o modelo é para prever positivos e o *F1-score*, que mostra o balanço entre a precisão e o *recall*. Tais informações podem ser visualizadas na Figura 17.

Figura 17 – Resultados

	precision	recall	f1-score	support
Funk	0.87	0.85	0.86	1178
MPB	0.75	0.82	0.78	1186
Sertanejo	0.78	0.73	0.76	1236
accuracy			0.80	3600
macro avg	0.80	0.80	0.80	3600
weighted avg	0.80	0.80	0.80	3600

Fonte: a autora.

4.5 Pós-Processamento

Feito o treinamento dos modelos, foi o momento de realizar previsões com dados diferentes dos já utilizados para teste e treinamento. Foi criada uma função chamada *test*, na qual é feito o pré-processamento do texto, assim como foi realizado no treinamento. Na função também é realizada a transformação das palavras em vetores TF-IDF. A função pode ser visualizada na Figura 18.

Figura 18 – Função para tratamento do texto na predição

```
def test(teste):  
  
    # colocando todas as letras em minúsculo  
    teste = teste.lower()  
  
    tokenized = [word_tokenize(teste)]  
  
    # remove as stopwords  
    stop_words = set(stopwords.words('portuguese'))  
    stop_vec = []  
    for tok in tokenized:  
        filtered_stop = [w for w in tok if w not in stop_words]  
        stop_vec.append(filtered_stop)  
  
    # remove a pontuação e acentos  
    clean_vec = []  
    for lyr in stop_vec:  
        clean_lyr = [unicode(word) for word in lyr if word.isalpha()]  
        clean_vec.append(clean_lyr)  
  
    lyrics = []  
    for lyr in clean_vec:  
        lyr_text = ' '.join(lyr)  
        lyrics.append(lyr_text)  
  
    single_entry = vectorize.transform(lyrics)  
    s_e = single_entry.todense().tolist()  
    return s_e
```

Fonte: a autora.

5 Resultados

Foram realizados diversos testes, com diferentes músicas dentro dos gêneros classificados: Sertanejo, MPB e Funk. São músicas que não foram retiradas dos *datasets* utilizados para treinamento e sim, de outras fontes ou conhecidas. Foi possível analisar que os resultados do modelo se mantiveram nos mesmos padrões dos resultados obtidos anteriormente, dentro da precisão esperada e, com isso, o desenvolvimento foi concluído. É possível analisar alguns desses testes nas Figuras 19, 20 e 21, respectivamente.

Figura 19 - Predição com Sertanejo

```
teste = test("""Quando eu digo que deixei de te amar
É porque eu te amo
Quando eu digo que não quero mais você
É porque eu te quero
Eu tenho medo de te dar meu coração
E confessar que eu estou em tuas mãos
Mas não posso imaginar
O que vai ser de mim
Se eu te perder um dia

Eu me afasto e me defendo de você
Mas depois me entrego
Faço tipo, falo coisas que eu não sou
Mas depois eu nego
Mas a verdade
É que eu sou louco por você
E tenho medo de pensar em te perder
Eu preciso aceitar que não dá mais
Pra separar as nossas vidas

E nessa loucura de dizer que não te quero
Vou negando as aparências
Disfarçando as evidências
Mas pra que viver fingindo
Se eu não posso enganar meu coração?
Eu sei que te amo!
""")

pred = lr.predict(teste)
pred

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:421:
X does not have valid feature names, but LogisticRegression
array(['Sertanejo'], dtype=object)
```

Fonte: a autora.

Figura 20 - Predição com MPB

```

▶ teste = test("""Estranho seria se eu não me apaixonasse por você
O sal viria doce para os novos lábios

Colombo procurou as Índias
Mas a Terra avisto em você
O som que eu ouço
São as gírias do seu vocabulário

Estranho é gostar tanto do seu All Star azul
Estranho é pensar que o bairro das Laranjeiras
Satisfeito, sorri, quando chego ali
E entro no elevador

Aperto o 12, que é o seu andar
Não vejo a hora de te encontrar
Continuar aquela conversa
Que não terminamos ontem
Ficou pra hoje

Estranho, mas já me sinto
Como um velho amigo seu
Seu All Star azul combina
Com o meu, preto, de cano alto
""")

▶ pred = lr.predict(teste)
pred

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted
array(['MPB'], dtype=object)

```

Fonte: a autora.

Figura 21 - Predição com Funk

```

▶ teste = test("""Me perguntaram qual que era meu sonho
Eu não sabia então nem respondi
Era pequeno demais pra saber
Então Deus respondeu pra mim

Calma, Meu filho, eu já tenho um plano
Passar por isso faz parte do jogo
Na madrugada nem sentia sono
Com aquele cheiro de beira do esgoto

Dificuldade, nós passou
Mas não deixou abalar nossa mente
Fiz amizade com ela
Felicidade, é ver o sorriso das crianças
E a paz de canto em canto nas favelas

Nóis é o gueto, pobre, louco
Maloqueiro, fui servente de pedreiro
Cansei de bater prego
Minha mãe queria que eu fosse um engenheiro
Mas eu falei pra ela
Que eu vou ser dono de um prédio
""")

▶ pred = lr.predict(teste)
pred

/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning:
X does not have valid feature names, but LogisticRegression was fitted
array(['Funk'], dtype=object)

```

Fonte: a autora.

6 Conclusão

Com o objetivo de mostrar que é possível realizar a classificação de gêneros musicais por meio das letras das músicas, o projeto visa utilizar de técnicas de aprendizado de máquina supervisionado e processamento de linguagem natural, para classificar três gêneros musicais brasileiros: Sertanejo, MPB e Funk, o que foi seguido e desenvolvido nos mesmos termos iniciais.

O desenvolvimento foi realizado com a linguagem Python e os recursos disponíveis nas bibliotecas scikit-learn, nltk e pandas. Foi feito o pré-processamento, que obteve bons resultados na limpeza, organização e redução do volume de dados dos *datasets* utilizados.

Houve algumas dificuldades durante a utilização da ferramenta Colab. Esta foi fácil de configurar, importar bibliotecas e utilizar, mas não tinha os recursos de memória RAM e CPU necessários para o nível de processamento que a análise dos dados estava exigindo.

Visto que nem todos os algoritmos utilizados eram especificamente para classificação multiclasse, os resultados foram muito bons, com alguns possíveis casos de *overfitting* e com o modelo de Regressão Logística tendo o melhor resultado para o andamento da análise e predição. O desenvolvimento foi realizado conforme os passos definidos anteriormente e teve bons resultados.

Como trabalhos futuros, pretende-se aumentar a quantidade de gêneros musicais do conjunto de dados e melhorar as precisões e resultados obtidos por meio de um pré-processamento mais detalhado, utilização de outras técnicas de aprendizagem e configurações diferentes dos modelos.

Referências

ANDRADE, Mário. **Dicionário Musical Brasileiro**. São Paulo: Editora USP, 1989.

BARBOSA, Ana; FILHO, Gilberto; TELES, Flávio. Aplicação da Análise Discriminante Linear (LDA) para Classificação de Sinais Eletromiográficos (EMG) de Movimentos da Mão. **Anais da VII Escola Regional de Informática de Goiás**. Porto Alegre: Sociedade Brasileira de Computação, p. 351-360, 2019.

BENNETT, Roy. **Uma Breve História da Música**. Rio de Janeiro: Jorge Zahar Editor, 1986.

COLABORATORY, Google. **Conheça o Colab**. 2021. Disponível em: <<https://colab.research.google.com/>>. Acesso em: 15 mai. 2022.

CORREA, Marcio. O conceito de gênero musical no repertório e nas áreas de antropologia, comunicação, etnomusicologia e musicologia. **ARJ – Art Research Journal: Revista de Pesquisa em Artes**, v. 5, n. 2, 2019. Disponível em: <<https://periodicos.ufrn.br/artresearchjournal/article/view/17796>>. Acesso em: 18 abr. 2022.

CREPALDI, Paola et al. **Um estudo sobre a árvore de decisão e sua importância na habilidade de aprendizado**. 2010. Disponível em: <https://www.inesul.edu.br/revista/arquivos/arq-idvol_15_1320100263.pdf> Acesso em: 10 out. 2022.

DENIZEAU, Gérard. **Los géneros musicales**: Una visión diferente de la historia de la música. Barcelona: Robinbook, 2005.

ESTEVEES, Wilson. **O que é a música?** 2019. Disponível em: <<https://www.musicdot.com.br/artigos/saiba-o-que-e-musica>>. Acesso em: 14 abr. 2022.

FABBRI, Franco. Uma teoria dos gêneros musicais: duas aplicações. **Revista Vórtex**, Curitiba, v. 5, n. 3, 2017. Disponível em: <http://vortex.unespar.edu.br/fabbri_v5_n3.pdf>. Acesso em: 18 abr. 2022.

FACELI, Katti et al. **Inteligência Artificial**: Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro: LTC, 2011.

FINGERHUT, M. The ircam multimedia library: A digital music library. **IEEE Forum on Research and Technology Advances in Digital Libraries**, p.19-21, 1999.

FONTANA, Éliton. **Introdução aos Algoritmos de Aprendizagem Supervisionada**. Departamento de Engenharia Química, Universidade Federal do Paraná, 2020.

GOLDMAN, Richard. VARÈSE: Ionisation; Density 21.5; Intégrales; Octandre; Hyperprism; Poème Electronique. **The Musical Quarterly**, Oxford, jan. 1961. Disponível em: <<https://academic.oup.com/mq/article-abstract/XLVII/1/133/1267372>>. Acesso em: 19 abr. 2022.

GONZALEZ, Leandro. **Regressão Logística e suas Aplicações**. Monografia (Ciência da Computação). Universidade Federal do Maranhão, São Luís, 2018.

JÚNIOR, Wagner. **Métodos de Otimização Hiperparamétrica**: um estudo comparativo utilizando árvores de decisão e florestas aleatórias na classificação binária. Dissertação (Engenharia Elétrica). Universidade Federal de Minas Gerais, Belo Horizonte, 2018.

KAGGLE. **About Kaggle**. Disponível em: <<https://www.kaggle.com/>>. sd. Acesso em: 15 mai. 2022.

KAUFMAN, Dora. **A inteligência artificial irá suplantará a inteligência humana?** São Paulo: Estação das Letras e Cores, 2018.

LUGER, George. **Inteligência Artificial**. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

MORAIS, Edison; AMBRÓSIO, Ana. Mineração de Textos. **INF**. Goiás, v.5, n.7, 2007. Disponível em: <https://ww2.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_005-07.pdf>. Acesso em: 10 out. 2022.

NEISSE, Anderson. **Song lyrics from 79 musical genres**. 2019. Disponível em: <<https://www.kaggle.com/datasets/neisse/scrapped-lyrics-from-6-genres>>. Acesso em: 05 set. 2022.

NEVES, Priscilla; CORRÊA, Débora; CAVALCANTI, Maria. Uma análise sobre abordagens e ferramentas para extração de informação. **Revista Militar de Ciência e Tecnologia**. Rio de Janeiro, 2013. Disponível em: <https://rmct.ime.eb.br/arquivos/RMCT_3_tri_2013/RMCT_123_E8A_13.pdf>. Acesso em: 10 out. 2022.

OLIVEIRA, Gilson. **Máquina de Vetores Suporte**: estudo e análise de parâmetros para otimização de resultado. Monografia (Ciência da Computação). Universidade Federal de Pernambuco, Recife, 2010.

PINTO, Tiago. Som e Música. Questões de uma Antropologia Sonora. **Revista de Antropologia**. São Paulo, v. 44, n. 1, 2001. Disponível em: <<https://www.revistas.usp.br/ra/article/view/27128>>. Acesso em: 14 abr. 2022.

REZENDE, Solange. **Sistemas Inteligentes**: Fundamentos e Aplicações. São Paulo: Editora Manole Ltda, 2003.

ROVEDA, Ugo. **O que é Python, para que serve e por que aprender?** Curitiba, 2020. Disponível em: <<https://kenzie.com.br/blog/o-que-e-python/>>. Acesso em: 15 mai. 2022.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2013.

SANGA, Dione. **Mineração de textos para o tratamento automático em sistemas de atendimento ao usuário**. Dissertação (Computação Aplicada). Universidade Tecnológica Federal do Paraná, Curitiba, 2017.

SEMAAN, Gustavo et al. Um método para classificação de dados baseando nos k- vizinhos mais próximos para o reconhecimento de caracteres. **XIX Simpósio de Pesquisa Operacional e Logística da Marinha**, 2019.

SEKEFF, Maria. **Da música**: seus usos e recursos. São Paulo: Editora UNESP, 2002.

SILVA, João. **Algoritmos de Classificação baseados em Análise Formal de Conceitos**. Dissertação (Ciência da Computação). Universidade Federal de Minas Gerais, Belo Horizonte, 2007.

SOARES, Fábio. **Mineração de textos na coleta inteligente de dados na web**. Dissertação (Engenharia Elétrica). Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

SOUZA, Ronaldes. O Corpo de Baile da Linguagem da Vida. **Terceira Margem**, Rio de Janeiro, n. 25, p.191-209, jul./dez. 2011.

TURING, Alan. Computing Machinery and Intelligence. **Mind**: A quarterly review of psychology and philosophy. Oxford, v. 59, p. 236, out. 1950.

VIEIRA, Renata; LOPES, Lucelene. Processamento de Linguagem Natural e o Tratamento Computacional de Linguagens Científicas. *In*: PERNA, Cristina;

DELGADO, Heloísa; FINATTO, Maria (Orgs.). **Linguagens especializadas em corpora**: modos de dizer e interfaces de pesquisa. Porto Alegre: EDIPUCRS, 2010. p.183.

ZAMACOIS, Joaquín. **Temas de Estética y de Historia de la Música**. Barcelona: Labor, 1984.