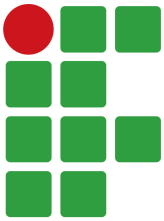


INSTITUTO FEDERAL
Goiás
Câmpus Anápolis

Bruno de Araújo Alves

Desenvolvimento de uma aplicação cliente-servidor para identificação de música

Anápolis - GO

2023

Bruno de Araújo Alves

Desenvolvimento de uma aplicação cliente-servidor para identificação de música

Projeto de Pesquisa apresentado ao curso de Bacharelado em Ciências da Computação, como requisito para obtenção do grau final na disciplina de Projeto Final de Curso.

Instituto Federal de Goiás - Câmpus Anápolis

Orientador: Prof. Dr. Hugo Vinícius Leão e Silva

Coorientador: Prof. Dr. Sergio Daniel Carvalho Canuto

Anápolis - GO

2023

Dados Internacionais de Catalogação na Publicação (CIP)

Alves, Bruno de Araújo

A474d Desenvolvimento de uma aplicação cliente-servidor para identificação de música. / Bruno de Araújo Alves. – Anápolis: IFG, 2023.
34 f. il. color.

Orientador: Prof. Dr. Hugo Vinícius Leão e Silva.

Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Goiás: Campus Anápolis – Bacharelado em Ciência da Computação, 2023.

1. Identificação de música. 2. aprendizado de máquina.
3. processamento de áudio.
I. Silva, Hugo Vinícius Leão e (orient.).
II. Canuto, Sergio Daniel Carvalho (coorient.).
III. Título.

CDD 005.3

Ficha catalográfica elaborada pela bibliotecária Claudineia Pereira de Abreu CRB1/1956
IFG Campus Anápolis

INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[X] TCC - Graduação	[ ] Trabalho Apresentado em Evento

[ ] Produto Técnico e Educacional - Tipo: _____

Nome Completo do Autor: Bruno de Araújo Alves
Matrícula: 20181060140080
Título do Trabalho: Desenvolvimento de uma aplicação cliente-servidor para identificação de música

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ___/___/_____ (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

( ) O documento está sujeito a registro de patente.
( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Anápolis, 16/08/2023.
Local Data

Bruno de Araújo Alves

_____

Assinatura do Autor e/ou Detentor dos Direitos Autorais

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS ANÁPOLIS

ATA DA SESSÃO PÚBLICA DE APRESENTAÇÃO E DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **treze** dias do mês de **julho** de **2023**, às **15h30**, em sessão pública ocorrida na **Sala S-504** do IFG - Campus Anápolis, foi realizada a sessão pública de apresentação e qualificação do Trabalho de Conclusão de Curso do Graduando **Bruno de Araújo Alves (Matrícula n.º 20181060140080)** do curso de **Bacharelado em Ciência da Computação**. A banca foi composta pelos seguintes membros: **Dr. Hugo Vinícius Leão e Silva, Ms. Alexandre Bellezi José, Dr. Sergio Daniel Carvalho Canuto e Dr. Daniel Xavier de Sousa**, sob a presidência do primeiro. O trabalho de conclusão de curso tem como título **Desenvolvimento de uma aplicação cliente-servidor para identificação de música**, sob orientação de **Hugo Vinícius Leão e Silva** e coorientação de **Dr. Sergio Daniel Carvalho Canuto**. Após a apresentação, tendo sido o autor arguido pela Banca Examinadora, a nota obtida foi **8,1** e o Trabalho de Conclusão de Curso foi considerado **Aprovado com ressalvas**, desde que sejam as solicitações da banca examinadora.

Encerra-se a presente sessão às **17** horas e **00** minutos. Eu, **Hugo Vinícius Leão e Silva**, dato e assino a presente ata que segue assinada por todos os membros da Banca e pelo graduando.

Dr. Hugo Vinícius Leão e Silva
(Assinado Eletronicamente)

Ms. Alexandre Bellezi José
(Assinado Eletronicamente)

Dr. Sergio Daniel Carvalho Canuto
(Assinado Eletronicamente)

Dr. Daniel Xavier de Sousa
(Assinado Eletronicamente)

Bruno de Araújo Alves
(Graduando)

Documento assinado eletronicamente por:

- **Hugo Vinicius Leao e Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 15/08/2023 17:03:32.
- **Bruno de Araujo Alves**, **20181060140080 - Discente**, em 15/08/2023 17:01:43.
- **Sergio Daniel Carvalho Canuto**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 02/08/2023 08:09:53.
- **Daniel Xavier de Sousa**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/07/2023 10:11:13.
- **Alexandre Bellezi Jose**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 14/07/2023 09:35:50.

Este documento foi emitido pelo SUAP em 11/07/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifg.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 429769
Código de Autenticação: 6edc049c9f

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ANN	*Artificial Neural Networks*
BiLSTM	*Long Short-Term Memory Bidirectional*
CNN	*Convolutional Neural Network*
dB	*Decibel*
DCT	*Discrete Cosine Transform*
DER	*Diagrama de Entidade e Relacionamento*
FFT	*Fast Fourier Transform*
Hz	*Hertz*
KDD	*Knowledge Discovery in Databases*
KNN	*K-Nearest Neighbours*
LSTM	*Long Short-Term Memory*
MFCC	*Mel-Frequency Cepstral Coefficients*
MIR	*Music Information Retrieval*
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
MVC	*Model View Controller*
REST	*Representation State Transfer*
SNR	*Signal-to-Noise Ratio*
STFT	*Short-Time Fourier Transform*
URL	*Uniform Resource Locator*

Resumo

A crescente disponibilidade de música digital e a ampla adoção de dispositivos móveis aumentaram a demanda por ferramentas que permitam aos usuários identificar instantaneamente músicas desconhecidas. Nesse contexto, a aplicação proposta utiliza técnicas de processamento de áudio e algoritmos de aprendizado de máquina para criar um modelo preditivo capaz de identificar de músicas. A parte central do sistema consiste em um algoritmo de reconhecimento de áudio baseado em extratores de características de áudio, *autoencoder* e um algoritmo classificador. O modelo gerado é implementado em uma aplicação cliente-servidor onde um usuário será capaz de enviar trechos de áudio de uma música, onde este será processado e identificado a música correspondente.

Palavras-chave: Identificação de música; Aprendizado de máquina; Processamento de áudio.

Abstract

The growing availability of digital music and the widespread adoption of mobile devices have increased the demand for tools that allow users to instantly identify unknown songs. In this context, the proposed application employs audio processing techniques and machine learning algorithms to create a predictive model capable of song identification. The central part of the system consists of an audio recognition algorithm based on audio feature extractors, autoencoders, and a classification algorithm. The generated model is implemented in a client-server application where a user will be able to send audio snippets of a song, which will be processed and identified as the corresponding song.

Keywords: Music identification; Machine learning; Audio processing.

Sumário

1 Introdução

Com a expansão da Internet e do desenvolvimento de tecnologias multimídia, a distribuição da música digital cresceu consideravelmente. Aplicações como *YouTube* e *Spotify* oferecem vastas coleções de músicas, tornando cada vez mais disponível às pessoas esse tipo de informação. Essa expansão trouxe a necessidade de desenvolvimento de ferramentas capazes de extrair informações relevantes do áudio para gerenciar essas coleções (PANAGAKIS; BENETOS; KOTROPOULOS, 2008).

Assim, a Recuperação de Informação Musical (*Music Information Retrieval*, MIR), área em ascensão que vem recebendo atenção tanto da comunidade de pesquisadores quanto da industria de música, torna possível o desenvolvimento dessas ferramentas (FU et al., 2011). Com isso diversas informações podem ser extraídas de uma música, incluindo o seu gênero, o seu humor, os artistas, os instrumentos utilizados e notação da música. Utilizando técnicas aplicadas à MIR, também é possível realizar a identificação de uma música. Dentre tais técnicas, a que mais se destaca é a assinatura digital (*fingerprinting*), utilizada atualmente por ferramentas populares como *Shazam* (APPLE, 2002), *SoundHound* (MOHAJER, 2005) e *iTunes* (APPLE, 2001), fazendo com que usuários, através do uso do microfone de seus dispositivos móveis, consigam saber rapidamente informações sobre a música que sendo reproduzida em qualquer ambiente.

Apesar do destaque da técnica de *audio fingerprinting*, é notável a grande quantidade de aplicações em diversas áreas do conhecimento que fazem uso de algoritmos de aprendizado de máquina. Isso se dá, pois esses algoritmos são capazes de utilizar grandes conjuntos de dados para realizar o reconhecimento de padrões, possibilitando gerar classificações, agrupamentos e predições para diversas tarefas. De forma genérica, tal processo de reconhecimento consiste em quatro etapas: a aquisição do conjunto de dados; o pré-processamento e a transformação dos dados; a classificação e, finalmente, a avaliação dos resultados obtidos. Assim, apesar de seguir um plano semelhante para o uso desses algoritmos, cada tarefa exige uma adaptação específica para o problema que se deseja resolver.

Dessa forma, o problema de identificação de música exige deste trabalho procurar uma forma para que uma máquina possa identificá-la automaticamente, baseando-se em pequenos trechos de áudio gravados pelo microfone de um dispositivo em um ambiente que pode ser ruidoso. Assim, este Trabalho de Conclusão de Curso pretende buscar maneiras de combinar métodos extratores de características de áudio e algoritmos de aprendizado de máquina, além de avaliar estratégias que permitam aprimorar essa classificação no que se refere a acurácia. Por fim, pretende-se prototipar uma aplicação cliente-servidor que

fará uso do modelo preditivo gerado e ilustrará um caso de uso real da solução proposta.

1.1 Objetivos gerais e específicos

Este trabalho teve como objetivo principal combinar métodos extratores características de áudio, *autoencoders* e algoritmos de aprendizado de máquina, a fim de criar um sistema identificador de música, comparando-os com a ferramenta Panako (SIX; LEMAN, 2014) que faz uso da estratégia de *audio fingerprinting*. Ainda, objetivou-se desenvolver uma aplicação cliente-servidor que ilustra um caso de uso real dessa ferramenta, onde seja possível o envio da gravação do trecho de uma música, tendo como resposta o nome da música identificada.

Sendo assim, este trabalho teve como objetivos específicos:

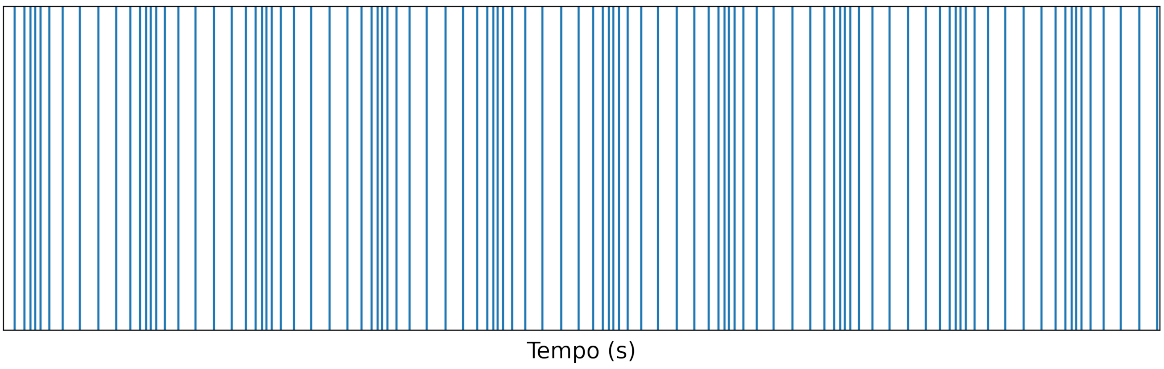
- Gerar um conjunto de dados de acesso privado contendo músicas dos mais variados artistas e gêneros;
- Avaliar a combinação de métodos extratores de características de áudio e algoritmos de aprendizado de máquina, a fim de gerar um modelo preditivo mais assertivo;
- Avaliar fatores que impactam sistemas de aprendizado de máquina no exercício da identificação de música;
- Desenvolver uma aplicação cliente-servidor que ilustre o caso real de uma ferramenta de identificação de música.

2 Fundamentação Teórica

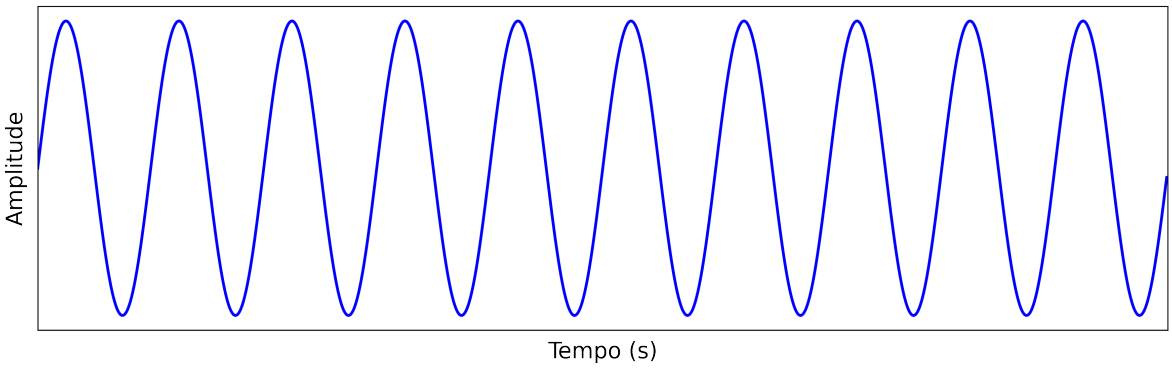
2.1 Processamento de Sinais Digitais

2.1.1 O que é um sinal de áudio e como ele é digitalizado?

O som é uma perturbação que provoca moléculas de algum meio, como o ar, a se moverem. Essa perturbação se propaga pelo meio como uma onda de pressão longitudinal, criando compressões (ou zonas de alta pressão) e rarefações (ou zonas de baixa pressão) no meio em questão como apresentado na Figura 1a (MANTIONE, 2021).



(a) Representação de uma onda de pressão longitudinal de 10 Hz.



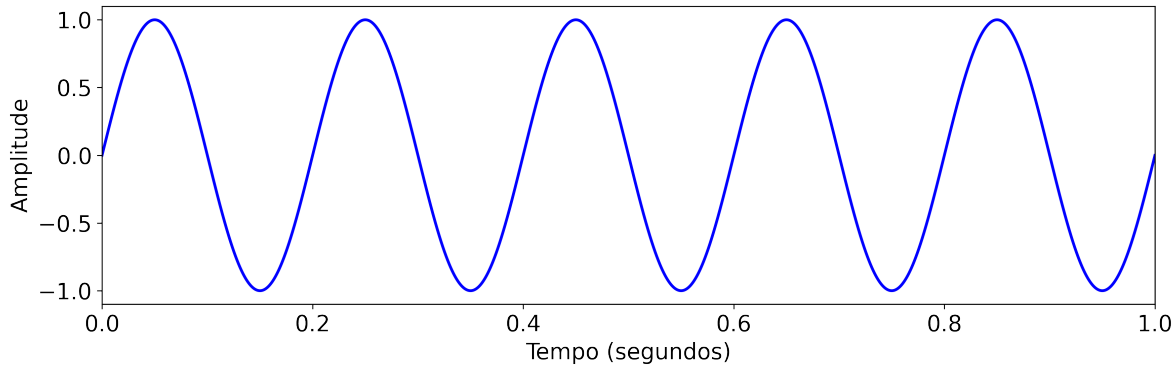
(b) Representação de um sinal gerado por uma onda de pressão longitudinal de 10 Hz.

Figura 1 – Representações de uma onda de pressão longitudinal de 10 Hz e o sinal gerado por ela.

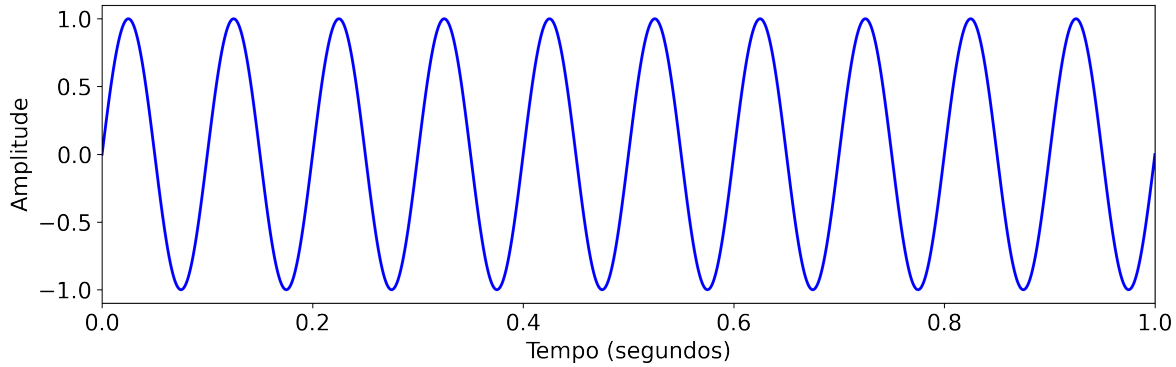
Essas compressões e rarefações podem ser observadas na Figura 1a através da proximidade entre as barras verticais. Quanto maior é a pressão induzida às moléculas do meio, maior é a quantidade de barras. Além disso, é possível verificar que existe uma relação direta entre as Figuras 1a e 1b: quanto mais positiva é essa pressão, maior é o valor assumido pelo sinal senoidal apresentado na Figura 1b. Assim, as compressões são

representadas na Figura 1b por valores positivos e as rarefações, por valores negativos.

Adicionalmente, é importante ressaltar que um sinal senoidal possui três importantes atributos: frequência, amplitude e fase. A frequência é a quantidade de vezes com que esse sinal se repete no intervalo de um segundo, sendo a sua unidade de medida o Hertz (Hz). Dizemos que um sinal de áudio é mais agudo quando o sinal senoidal associado a ele possui uma frequência maior. Por outro lado, quanto menor é essa frequência, mais grave é o som. Em outras palavras, quanto maior a frequência, menor é o comprimento de onda do sinal senoidal. Esse comprimento de onda mede a distância entre repetições do sinal senoidal. As Figuras 2a e 2b ilustram, respectivamente, dois sinais senoidais, onde o primeiro possui frequência de cinco hertz e o segundo, de dez hertz.



(a) Sinal senoidal de 5 Hz.



(b) Sinal senoidal de 10 Hz.

Figura 2 – Exemplos de dois sinais senoidais de 5 e 10 Hz respectivamente.

Para fins de referência, o ouvido humano é capaz de sentir variações numa faixa de frequência que compreende em torno de 20 Hz e 20.000 Hz, dependendo da saúde e da idade do indivíduo. Por sua vez, a amplitude é a distância máxima entre a crista ou vale e o eixo de equilíbrio, dando ao ouvido a percepção do volume de um som. O ouvido humano é sensível a uma grande faixa de amplitudes, que é medida normalmente utilizando uma escala logarítmica conhecida como Decibel (dB) (GIRALDELI, 2009). Por fim, a fase de um sinal é relativa ao deslocamento horizontal do sinal senoidal, assumindo valores entre

zero e 360° (ou 2π).

Quando um som é captado por um microfone, ele converte as ondas de pressão em um sinal elétrico: no caso de alta pressão, o sinal assume tensão positiva e, no caso de baixa pressão, o sinal elétrico assume tensão negativa. Tais valores de tensão são uma representação analógica do som, ou seja, contêm valores contínuos que podem ser gravados em fitas magnéticas como mudanças de força magnética ou em discos de vinil como mudanças no tamanho das ranhuras (ADOBE, 2021). Essa representação, contudo, não pode ser interpretada e manipulada por computadores digitais, havendo a necessidade de tradução do domínio contínuo para o discreto.

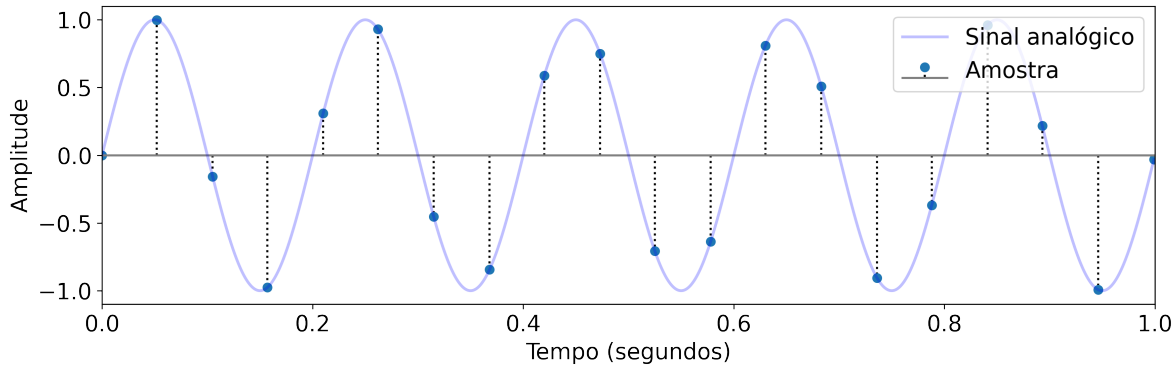
Esse processo é denominado digitalização de som, conhecido como conversão A/D - *Analog to Digital*, e consiste em duas tarefas: a amostragem e a quantização. A amostragem é responsável pela medição da tensão elétrica em vários pontos do sinal em um intervalo regular. Cada um desses pontos é chamado de amostra e convém ao usuário definir a quantidade de pontos que deve haver em um segundo, conhecido como taxa de amostragem. A segunda tarefa é relativa à representação de cada uma dessas amostras em uma sequência de bits segundo uma tabela de quantização. O tamanho dessa tabela está em função da profundidade de bits, que pode ser, como exemplo, quatro, oito ou dezesseis bits. Assim, a tabela de quantização teria, respectivamente, 16, 256 ou 65.536 sequências diferentes, cada uma associada a um valor de tensão de amostra.

Esses dois parâmetros são fundamentais e influenciam a fidelidade de som e o tamanho do arquivo gerado posteriormente. Isso pode ser visto nas Figuras 3 e 4, que ilustram, respectivamente o sinal ilustrado na Figura 2a passando por distintos processos de amostragem e quantização.

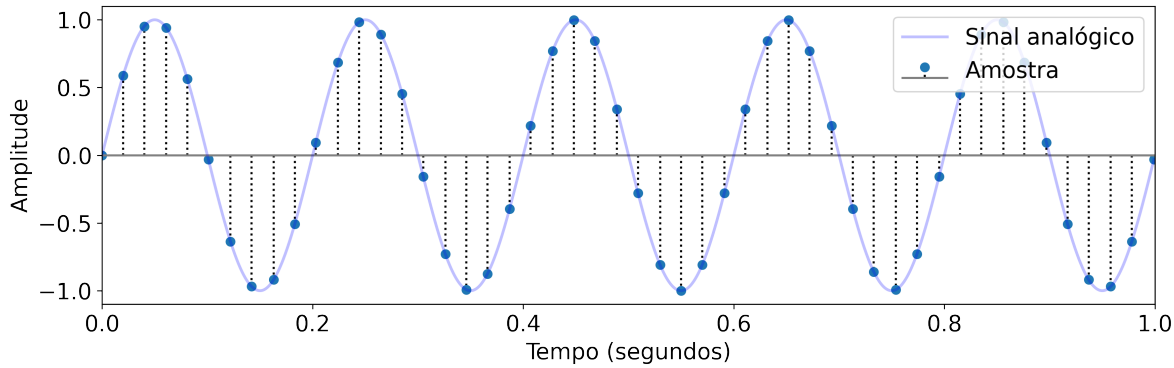
Pode-se observar na Figura 3 que, quanto maior é a taxa de amostragem, o sinal é amostrado mais vezes e, consequentemente, geram-se mais amostras. Adicionalmente, na Figura 4 é possível notar que as amplitudes das amostras do sinal quantizado se aproximam do sinal analógico original à medida que se aumenta a profundidade de bits. Isso é ilustrado na Figura 5, que apresenta os erros de quantização associados às Figura 4a e Figura 4b. É possível verificar que o erro de quantização foi consideravelmente menor na Figura 5b.

Seguindo o Teorema de Nyquist (WESCOTT, 2010), a taxa de amostragem deve ser pelo menos duas vezes maior do que a maior frequência do sinal analógico que se deseja representar. É esse o motivo pelo qual o áudio padrão do CD (*Compact Disc*) é amostrado a uma frequência de 44.100 Hz e quantizado à profundidade de 16 bits, o que cobre uma faixa de frequência que se estende até pouco mais de 20.000 Hz, compatível com a faixa audível aos seres humanos (GIRALDELI, 2009).

Assim, é possível observar que quanto maiores forem a taxa de amostragem e a profundidade de bits, maiores são a qualidade do sinal digital resultante e do arquivo



(a) Amostragem de um sinal usando taxa de amostragem de 20 Hz.



(b) Amostragem de um sinal usando taxa de amostragem de 50 Hz.

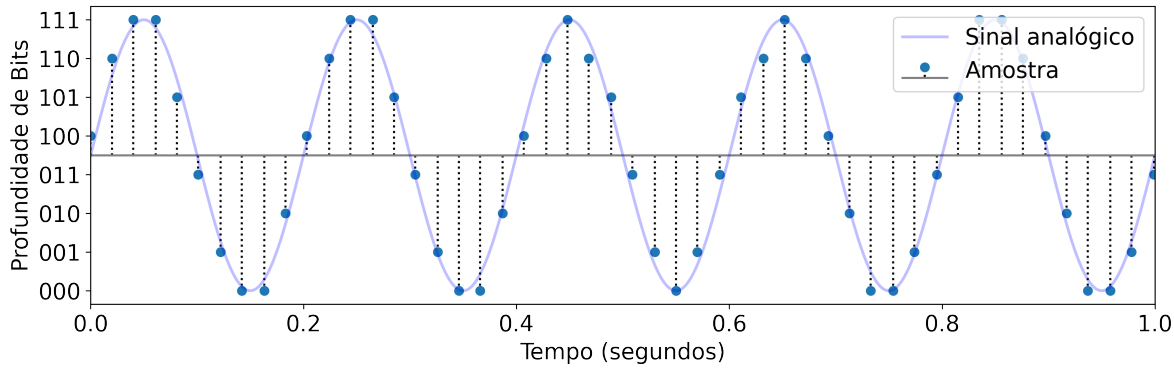
Figura 3 – Exemplos de um sinal sendo amostrado usando taxas de amostragem de 20 e 50 Hz respectivamente.

gerado. Como referência, a Tabela 1 relaciona diversos valores de taxa de amostragem com a qualidade do sinal amostrado.

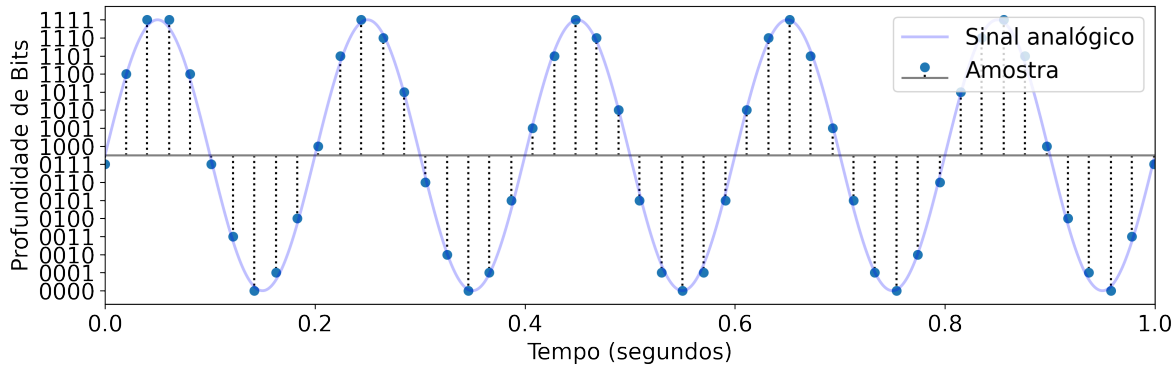
Taxa de amostragem	**Nível de qualidade**	**Faixa de frequência**
11.025 Hz	Rádio AM de baixa qualidade	0 a 5.512 Hz
22.050 Hz	Rádio próximo a FM	0 a 11.025 Hz
32.000 Hz	Melhor que rádio FM	0 a 16.000 Hz
44.100 Hz	CD	0 a 22.050 Hz
48.000 Hz	DVD padrão	0 a 24.000 Hz
96.000 Hz	Blu-ray DVD	0 a 48.000 Hz

Tabela 1 – Tabela de taxa de amostragem comumente utilizada em diferentes aplicações.

Porém, é comum que o sinal de áudio contenha ruído, o que dificulta o processo de digitalização já que ele afeta a amplitude do sinal analógico. Um tipo de ruído comumente encontrado é o ruído branco, que é um sinal aleatório constituído pela combinação de todas frequências independentes identicamente distribuídas. A Figura 6b ilustra um sinal senoidal de 5 Hz e como o ruído branco afeta a sua amplitude quando somados.



(a) Quantização de um sinal usando profundidade de 3 bits (8 valores).



(b) Quantização de um sinal usando profundidade de 4 bits (16 valores).

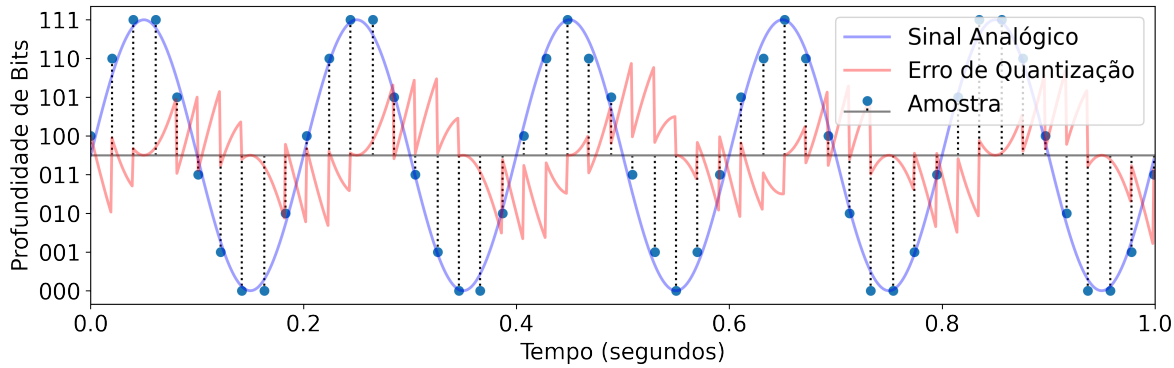
Figura 4 – Exemplos de um sinal sendo amostrado usando taxa de amostragem de 50 Hz usando quantização de 3 e 4 bits respectivamente.

2.2 O que é aprendizado de máquina?

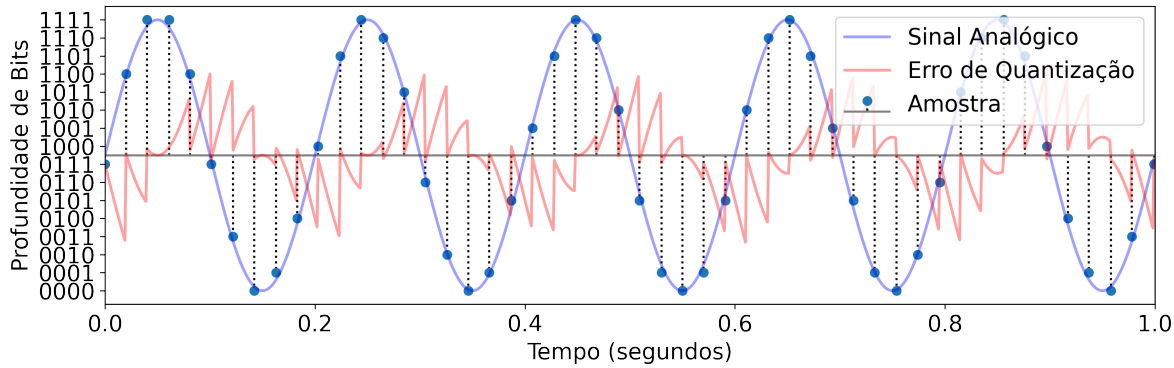
Aprendizado de Máquina faz parte da área de Inteligência Artificial, que faz uso de algoritmos e modelos estatísticos para realizar uma tarefa, sem ter que explicitamente programar as instruções de como realizá-la (NORDBY, 2019). Em vez disso, o algoritmo aprende a realizar determinada função baseando-se em experiências acumuladas através da solução bem sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003). A concepção desse aprendizado se dá de maneira indutiva, que é guiado por uma coleção de dados (SOUZA, 2015), possibilitando a inferência indutiva de exemplos de dados externos à coleção aprendida.

O aprendizado indutivo pode ser dividido em duas categorias: supervisionado e não-supervisionado. No aprendizado supervisionado é fornecido ao algoritmo indutor um conjunto de exemplos de treinamento para os quais o rótulo da classe associada é conhecido (MONARD; BARANAUSKAS, 2003). Já no aprendizado não-supervisionado, esse conjunto de exemplos não contém rótulos e o algoritmo indutor deve determinar esses rótulos formando agrupamentos de exemplos semelhantes.

Em ambos os tipos de algoritmos, há a necessidade de um conjunto de exemplos



(a) Erro de quantização usando quantização com profundidade de 3 bits (8 valores).



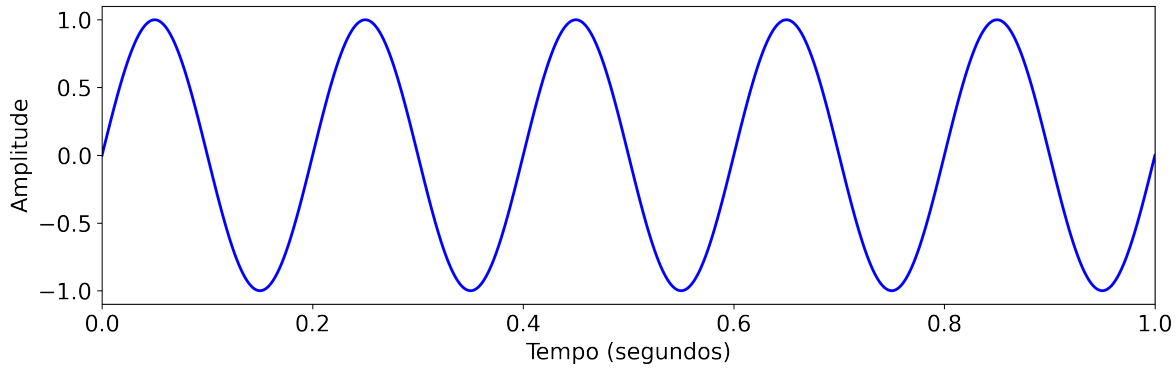
(b) Erro de quantização usando quantização com profundidade de 4 bits (16 valores).

Figura 5 – Exemplos de erro de quantização ao realização amostragem de um sinal utilizando profundidades de 3 e 4 bits.

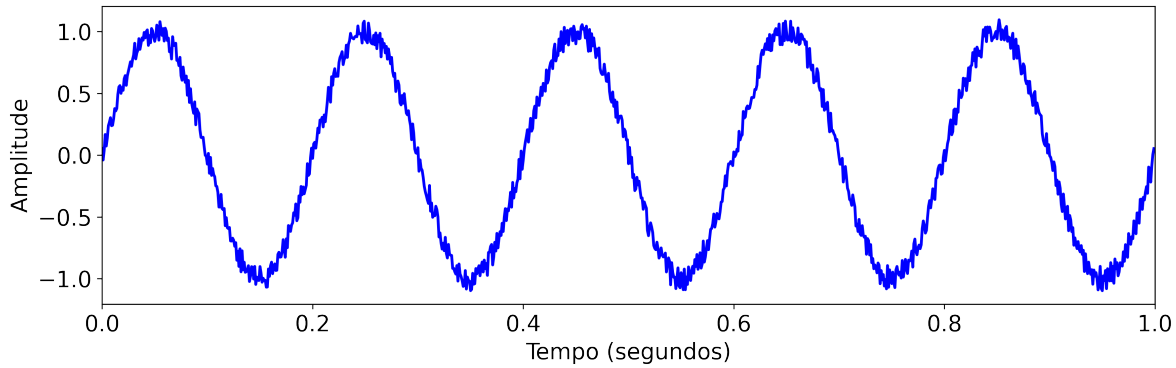
ao qual é extraído o conhecimento. Esse conjunto é conhecido como base de dados. De forma ampla, o processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases*, KDD) é composto pelas fases de aquisição da base de treinamento, do pré-processamento, da transformação, da mineração e da validação (FERRARI; SILVA, 2017), que são explicadas nas próximas seções. É importante ressaltar que esse é um processo iterativo, em que se pode retornar a qualquer fase prévia quando há necessidade.

2.2.1 Aquisição da base de treinamento

A fase de aquisição da base de treinamento consiste na seleção dos dados que, a partir dos quais, será extraído algum tipo de conhecimento. Esses dados podem ser quantitativos e/ou qualitativos e são coletados a partir de qualquer meio que disponha dados sobre a tarefa em questão. Por serem coletados a partir de diferentes meios, é comum que esses dados estejam dispostos em uma forma desorganizada e/ou incompreensível ao algoritmo indutor que se espera utilizar.



(a) Sinal senoidal de 5 Hz.



(b) Sinal senoidal contaminado por ruído branco.

Figura 6 – Exemplo de sinal senoidal contaminado por ruído branco.

2.2.2 Pré-processamento e transformação

Após a seleção do conjunto de dados, é incomum o uso deste em sua forma inicial. Portanto, faz-se necessário o tratamento desses dados realizando a remoção de ruídos e dados inconsistentes, o preenchimento de dados faltantes, além de realizar transformações, utilizando diferentes métodos e representações (FERRARI; SILVA, 2017). Essa fase depende do tipo dos dados sobre os quais estão se trabalhando, variando as técnicas de pré-processamento e transformações para quando são valores numéricos, categóricos, textos, imagens ou áudio. Em aplicações de aprendizado de máquina associadas a processamento de sinal de áudio é comum o uso das técnicas explicadas seções seguintes.

2.2.2.1 Transformada de Fourier

Em 1807 foi proposto pelo pesquisador francês Jean Baptiste Joseph Fourier um artigo que trata do problema prático da propagação do calor em barras e chapas de sódio metálicos. No artigo, Fourier afirma que toda função definida num intervalo finito pode ser ser decomposta numa soma de funções seno e cosseno. Essa afirmação diz que, dada

uma função qualquer definida no intervalo $[-\pi, \pi]$ pode ser representada por:

$$\frac{a_0}{2} \sum_{n=1}^{\infty} (a_n \cos \pi x + b_n \sin \pi x), \tag{2.1}$$

em que os coeficientes a_0, n, a_n e b_n são números reais (NUNES et al., 2002). A Equação (2.1) também é conhecida como Série de Fourier.

A utilização das Séries de Fourier é de fundamental importância a diversas áreas. Elas permitem representar muitas funções periódicas como uma soma infinita de exponenciais complexas. A representação por meio de séries é vantajosa quando se deseja aproximar os valores de uma função. O truncamento da série pode ser realizada para efetuar uma aproximação para casos em que a função seja complexa e que dificilmente chegará em seu cálculo exato. Um exemplo disso é ilustrado na Figura 7, que mostra a aproximação de um sinal de onda quadrada em termos de uma soma de senos.

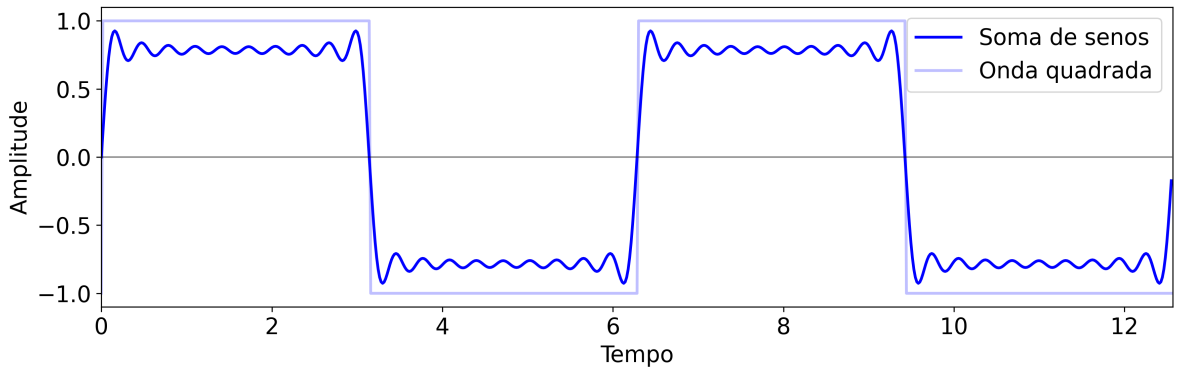


Figura 7 – Aproximação de um sinal de onda quadrada usando a soma de senos.

A Figura 7 ilustra a Série de Fourier, para o sinal de onda quadrada, truncada no quinto termo do somatório da Equação 2.1:

$$f(x) = 0 + \frac{1}{1}\sin(x) + \frac{1}{3}\sin(3x) + \frac{1}{5}\sin(5x) + \frac{1}{7}\sin(7x) + \frac{1}{9}\sin(9x), \tag{2.2}$$

onde $a_0 = 0$, $n = 1, 3, 5, 7, 9$, $a_n = 0$ e $b_n = \frac{1}{n}$. Pode-se observar na Figura 7 que a Equação (2.2) se aproxima de uma onda quadrada. Porém, cabe ressaltar que, quanto maior mais valores assumem-se para n ímpar, mais $f(x)$ se assemelha a essa onda.

A Transformada de Fourier, por outro lado, é considerada uma das ferramentas matemáticas mais importantes do mundo moderno e, apesar de ter surgido no século XVIII, foi com o surgimento das mídias digitais que ela se popularizou, podendo ser utilizada para análise espectral. Uma aplicação onde a Transformada de Fourier é extensivamente utilizada é na codificação de áudio para o formato MP3, onde é possível reduzir significativamente o tamanho de arquivos, removendo determinados componentes do sinal de áudio. Além disso, ela pode ser utilizada na redução de ruído em sinais de áudio, assim como na detecção

de tom ou em aplicações mais avançadas como a classificação com base no som, quando associada às técnicas de aprendizado de máquina.

O objetivo da Transformada de Fourier é, em termos mais simples, mudar o problema original, que é uma função periódica representada no domínio do tempo, representando-o no domínio da frequência. Para realizar isso, contudo, a Transformada de Fourier deve calcular os coeficientes a_0, a_n e b_n da Série de Fourier no domínio do tempo. Em outras palavras, a Transformada de Fourier decompõe um sinal em seus componentes elementares seno e cosseno. Sua aplicação inicial se deu a partir de problemas com condução de calor (lei da condução térmica). A mesma é utilizada em problemas que são difíceis de resolver diretamente no domínio do tempo, mas que podem ser facilmente resolvidos no domínio da frequência. Assim, é mais simples aplicar a Transformada de Fourier nesse problema e solucioná-lo no domínio transformado. Daí, aplica-se a Transformada Inversa de Fourier à solução para ser representada novamente no domínio do tempo (FECHINE, 2010).

Essa transformada pode ser usada em qualquer conjunto de dados analisados diretamente por um espectro de frequências, pois se trata de um conjunto completo e ortonormal de funções. Um ponto considerável da transformada é o critério de Nyquist, que menciona que um sinal precisa ser amostrado pelo menos duas vezes em cada ciclo de variação, ou seja, a frequência de amostragem (frequência de Nyquist) precisa ser pelo menos o dobro da maior frequência presente no sinal. Caso esse critério não seja atendido, os componentes de mais alta frequência do sinal serão incorretamente apontados como de baixa frequência (FILHO, 2011).

2.2.2.2 Espectrograma Mel

A Transformada Rápida de Fourier (*Fast Fourier Transform*, FFT) é um algoritmo que computa mais rapidamente os coeficientes da Transformada de Fourier. Ela é vastamente utilizada em processamento de signais e permite representar no domínio da frequência um sinal que originalmente está no domínio do tempo (THIAGO, 2017).

Contudo, o resultado desse algoritmo são coeficientes que estão apresentados apenas no domínio da frequência, não sendo possível analisar em que momento aparece o coeficiente associado a determinada frequência. Portanto, sinais não-periódicos como sinais de áudio (música e fala), precisam de uma representação que torne possível a análise do sinal variando na frequência e no tempo.

Isso é possível através da divisão do sinal em segmentos de mesma duração, utilizando a superposição de cada segmento – processo denominado janelamento – e, por fim, sobre cada segmento é aplicado o algoritmo FFT (ROBERTS, 2020). Esse algoritmo é conhecido como *Short-Time Fourier Transform* (STFT), tendo como coeficientes resultantes uma representação tempo-frequência, conhecida como espectrograma. No espectrograma, apesar de inicialmente um eixo conter as amplitudes das frequências do sinal amostrado,

deve-se converter para uma escala logarítmica conhecida como Decibel.

Como dito anteriormente, o ouvido humano é sensível a uma grande faixa de amplitudes do som. Contudo, mesmo que o ouvido humano é capaz de ouvir sons entre 20 Hz e 20.000 Hz, ele o faz de maneira diferente, a depender da frequência do som. Para sons entre 0 Hz e 1000 Hz, há uma distinção linear. Para sons com frequência acima de 1.000 Hz, a distinção é logarítmica. Em outras palavras, a diferença entre as frequências 200 Hz e 300 Hz é mais perceptível ao ouvido humano do que a diferença entre as frequências 1.000 Hz e 1.100 Hz, apesar da distância entre elas ser a mesma. Essa escala é conhecida como escala Mel e é utilizada para representar um sinal de áudio de forma semelhante à percepção auditiva humana. A escala Mel é apresentada na Equação (2.3) (MLEARNERE, 2020).

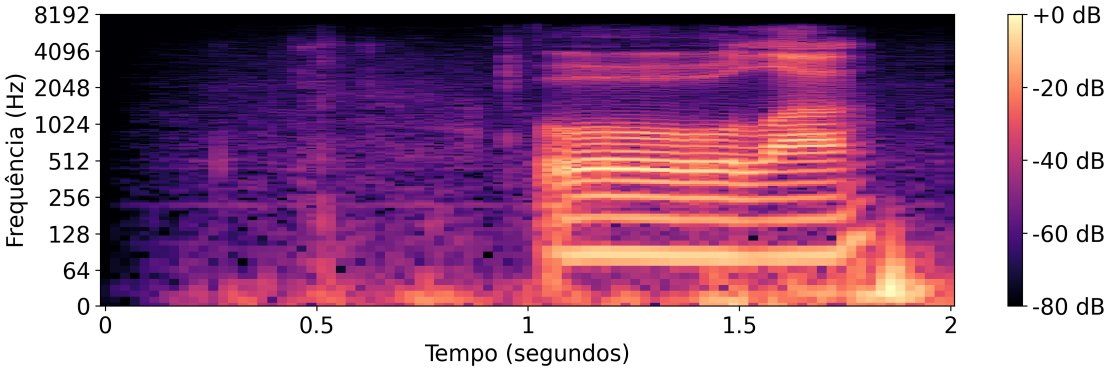
$$f_{mel} = \frac{1000}{\log_{10} 2} \log_{10} \left(1 + \frac{f}{1000}\right) \tag{2.3}$$

Espectrograma Mel é o nome dado ao espectrograma que utiliza a escala Mel em vez da escala logarítmica padrão. O resultado desse espectrograma continua sendo coeficientes resultantes de uma representação tempo-frequência, como em um espectrograma comum. Por outro lado, ele usa faixas de amplitudes na escala Mel, como pode ser visto na Figura 8 (MLEARNERE, 2020). Uma das diversas aplicações dessa representação está em aplicações de reconhecimento de locutor, através do método de extração de características denominado Coeficientes Cepstrais de Frequência Mel (*Mel-Frequency Cepstral Coefficients*, MFCC), onde dado um sinal de áudio e obtido seu espectrograma Mel, calcula-se o logaritmo desse espectrograma e, então, aplica-se a transformada discreta do cosseno (*Discrete Cosine Transform*, DCT). Assim, obtêm-se informações sobre a distribuição de potência do sinal de áudio avaliado (THIAGO, 2017).

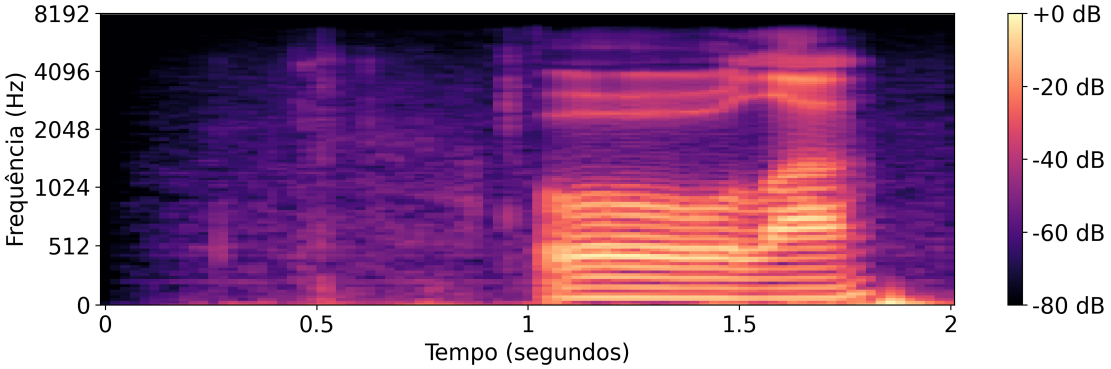
2.2.3 Mineração

Na fase de mineração de dados, tendo como entrada o artefato das fases de pré-processamento e transformação, são utilizados algoritmos indutores a fim de extrair conhecimento. A extração de conhecimento tem como objetivo o aprendizado de padrões a partir da base de dados para, dependendo da tarefa, realizar a predição ou a classificação de novos dados.

Em tarefas de classificação, algoritmos de aprendizado de máquina como K-Vizinhos Mais Próximos (*k-Nearest Neighbours*, KNN), Máquina de Vetores de Suporte (*Support Vector Machine*, SVM), árvores de decisão, *Random Forest* e Redes Neurais Artificiais (*Artificial Neural Networks*, ANN) como Perceptron Multicamadas (Multilayer Perceptron, MLP), Memória Longa de Curto Prazo (Long Short-Term Memory, LSTM) e Rede Neural



(a) Áudio representado usando o Espectrograma Comum.



(b) Áudio representado usando o Espectrograma Mel.

Figura 8 – Exemplos de um sinal de áudio sendo representado respectivamente em Espectrograma Comum e Espectrograma Mel

Convolucional (Convolutional Neural Network, CNN) (GÉRON, 2019) podem ser utilizados a fim de se gerar um modelo.

2.2.4 Validação

Por fim, na fase de validação é realizada a análise de desempenho, onde métricas como acurácia, F1 Macro/Micro (precisão e revocação) e erro médio quadrático (*Mean Square Error*, MSE) (FERRARI; SILVA, 2017) são utilizadas para fornecer resultados quantitativos de desempenho do modelo gerado com base nas escolhas feitas nas fases anteriores. Ainda, diferentes estratégias como divisão de base de dados (em treino, teste e validação), validação cruzada (do inglês *Cross Validation*) e *Data Augmentation* devem ser aplicadas dependendo da tarefa, do tamanho e da base de dados utilizada.

3 Metodologia

Este Trabalho de Conclusão de Curso objetiva a construir uma solução em software capaz de identificar uma música a partir da coleta de um trecho qualquer da mesma, que pode ser afetado por ruído. Este capítulo tem como objetivo apresentar o processo de construção de tal solução.

3.1 Construção da base de dados

A primeira etapa para a realização deste trabalho consistiu na criação de um conjunto de dados. Embora existam bases de dados públicas disponíveis na internet, essas não contêm músicas populares ou disponibilizam apenas fragmentos iniciais das faixas, geralmente limitados aos primeiros trinta segundos. Diante dessa limitação, foi construída uma base de dados privada com 1.000 músicas de diferentes artistas e estilos musicais. Para isso aproveitou-se uma base de dados criada pelo Spotify em um desafio na plataforma AICrowd (SPOTIFY, 2020), onde foi disponibilizada uma lista de metadados de músicas da plataforma, contendo um milhão de *playlists* do Spotify, correspondendo a aproximadamente 65.000 diferentes faixas musicais.

Portanto, para a construção da base de dados foi desenvolvido um *script* em linguagem de programação Python (PYTHON, 2023) e as bibliotecas Spotdl (SPOTDL, 2023), Spotipy (SPOTIPY, 2023) e Mutagen (LIBET, 2023). Esse *script*, ao passar o código identificador de uma música do Spotify, coleta os dados da mesma dentro da plataforma, como o nome, duração, letra e popularidade da música. Em seguida, extrai-se a faixa de áudio correspondente de um vídeo no YouTube associado à música em questão e a converte para o formato mp3.

É então verificado se a duração da faixa está entre dois e oito minutos, garantindo que sejam coletados trechos significativos, descantando as faixas que não contemplem esse requisito. Cada faixa de música baixada tem uma taxa de amostragem de 48.0000 Hz, profundidade de 24 bits e duplo canal de áudio, ou seja, estéreo.

Esse *script* foi executado diversas vezes até que fosse coletada a quantidade de músicas propostas para a construção da base de dados, já que em alguns casos a músicas passadas não possuíam a duração previamente estipulada de dois a oito minutos.

Além das música baixadas, foram coletados também os nomes das músicas, dos artistas e álbuns de cada música, assim como a URL (*Uniform Resource Locator*). Feito isso, todos os dados foram guardados no banco de dados relacional Postgres (GROUP, 1996), utilizando o esquema relacional apresentado no Diagrama de Entidade e Relacionamento

(DER) da Figura 9.

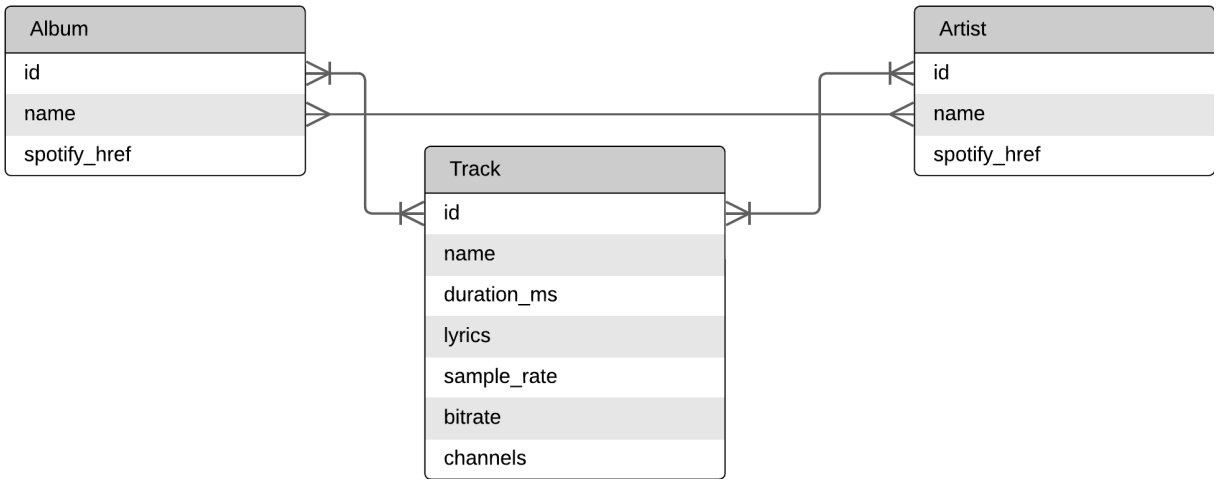


Figura 9 – Diagrama de entidade e relacionamento da base de dados.

O diagrama ilustra a relação entre álbuns, artistas e músicas. Os álbuns são representados pela entidade *Album*, que contém os seguintes atributos:

- **id**: o identificador único do álbum no Spotify.
- **name**: o nome do álbum.
- **spotify_href**: a URL do álbum no Spotify.

Da mesma forma, os artistas são representados pela entidade *Artist*, que possui os seguintes atributos:

- **id**: o identificador único do artista no Spotify.
- **name**: o nome do artista.
- **spotify_href**: a URL do artista no Spotify.

Por fim, as músicas são representadas pela entidade *Track*, que inclui os seguintes atributos:

- **id**: o identificador único da música no Spotify.
- **name**: o nome da música.
- **duration_ms**: a duração da música em microssegundos.
- **lyrics**: a letra da música.
- **sample_rate**: a taxa de amostragem da música.
- **bit_rate**: a taxa de bits da música.

3.2 Pré-processamento

Tendo estabelecida a base de dados e transferida a coleção de músicas, iniciou-se a etapa de tratamento dos dados de áudio, seguindo estratégias distintas.

Primeiramente, optou-se por replicar a base de dados em conjuntos independentes, compreendendo conjuntos de treino, teste e validação. Isso garante que todos os áudios estejam tanto no fase treinamento quanto nas fases de validação e teste. A seguir, cada áudio do conjunto de treino foi segmentado em tamanhos fixos e saltos de mesmo tamanho. Por exemplo, considerando um áudio de dez segundos e escolhendo um tamanho de segmento de três segundos com um salto também de três segundos, a segmentação resulta nas três janelas ilustradas na Figura 10.

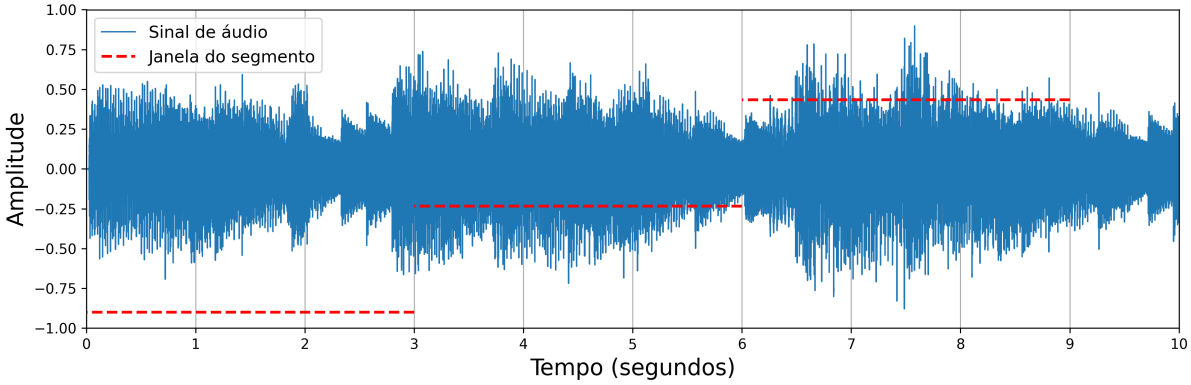


Figura 10 – Segmentação de um áudio de dez segundos em tamanhos fixo de três segundos e saltos de três segundos.

Por outro lado, nos conjuntos de teste e validação, os áudios foram segmentados usando o mesmo tamanho fixo da base de treino, porém, com saltos de um segundo. Seguindo esse processo, também considerando um áudio de dez segundos, resultam-se em oito segmentos, conforme apresentado na Figura 11. Vale ressaltar que em ambos os casos, segmentos que não contemplem o tamanho do segmento definido são descartados.

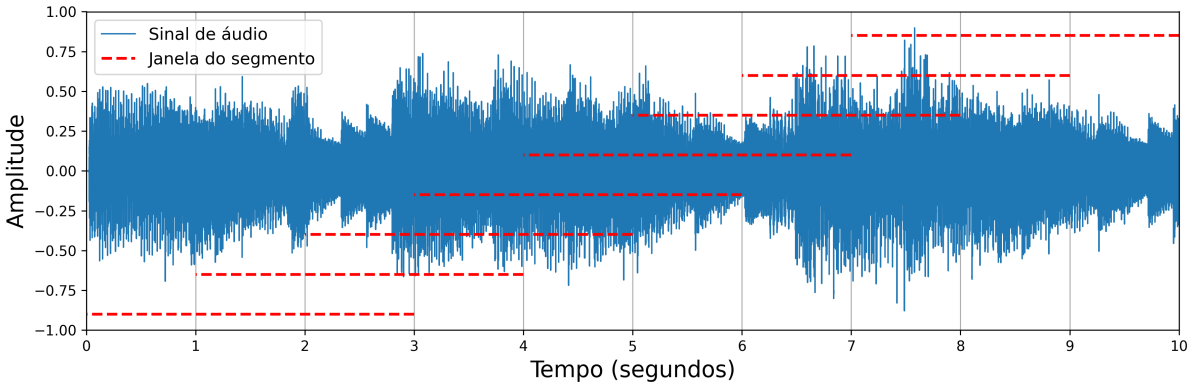


Figura 11 – Segmentação de um áudio de dez segundos em tamanhos fixo de três segundos e saltos de um segundo.

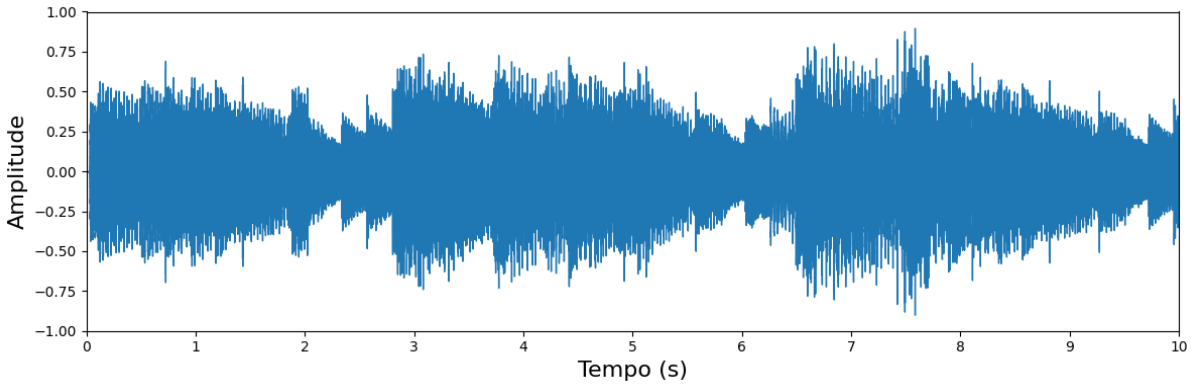
Em seguida, após a replicação dos conjuntos de dados, compreendeu-se a etapa de aplicação de diferentes tarefas de processamento de áudio. Nessa etapa foram aplicadas estratégias de tratamento de áudio, bem como o aumento do conjunto de dados.

Cada uma delas são explicadas respectivamente nas seções 3.2.1 a 3.2.6.

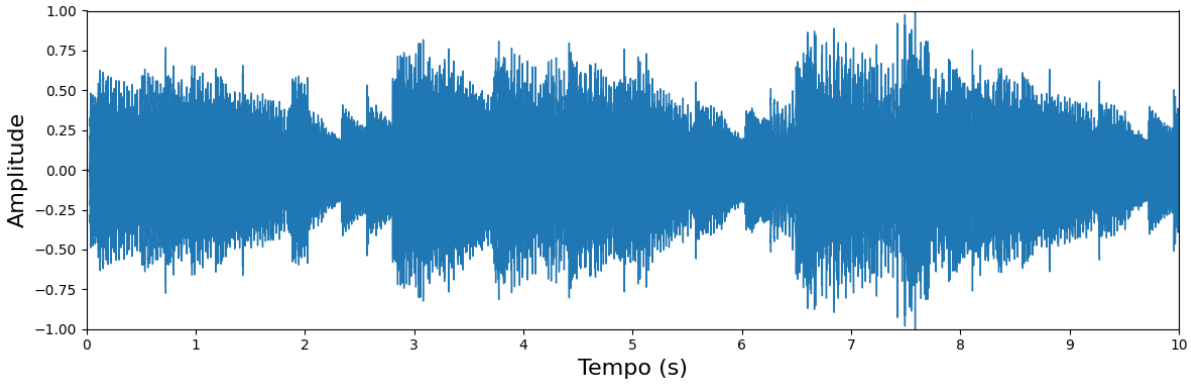
3.2.1 Normalização de pico

Normalização de pico é uma técnica utilizada para ajustar o nível de volume de um sinal de áudio de forma a maximizar a sua amplitude sem distorcer ou realizar recorte do som. O objetivo é levar o pico mais alto do sinal para o nível máximo permitido sem ultrapassá-lo, garantindo que o áudio tenha uma amplitude máxima sem perder detalhes (LOOPMASTERS, 2023).

Um exemplo da aplicação dessa técnica é apresentada na Figura 12, onde dado um sinal de áudio qualquer de dez segundos, é retornado o sinal transformado pela normalização. É possível perceber na Figura 12b que a amplitude do sinal foi aumentada para compreender a amplitude máxima permitida.



(a) Sinal de áudio de 10 segundos representado em forma de onda.



(b) Sinal de áudio de 10 segundos aplicado a normalização de pico

Figura 12 – Exemplos de um sinal de áudio de 10 segundos apresentando a aplicação da normalização de pico

3.2.2 Ruído gaussiano por SNR

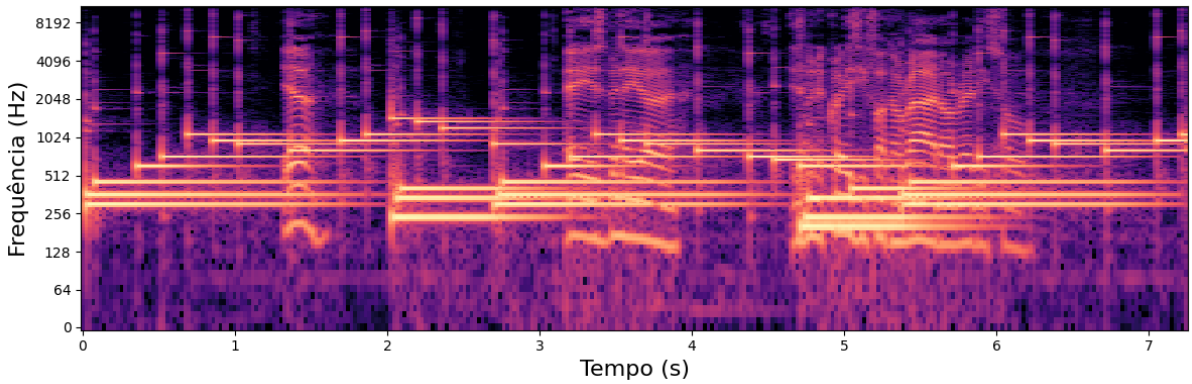
Ruído gaussiano, ou ruído normal, é um tipo de ruído aleatório gerado por uma distribuição normal. Nessa distribuição, os valores do ruído são distribuídos simetricamente (CADENCE, 2020).

Quando junto a relação sinal-ruído (*Signal-Noise Ratio*, SNR), permite aplicação desse ruído a um sinal sem que a amplitude desse ruído ultrapasse à amplitude que o sinal de áudio original permite.

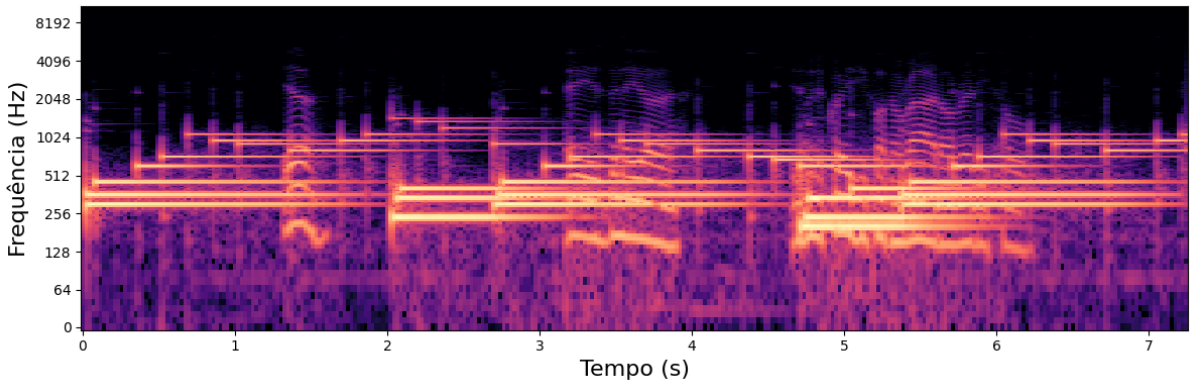
O resultado de tal técnica neste trabalho visa simular condições de ambientes ruidosos.

3.2.3 Filtro passa-baixa

O filtro passa-baixa é um filtro que permite remover componentes de um sinal que estejam acima de um frequência de corte (ROCCHESSO, 2003). Isso pode ser visto na Figura 13b, onde toda frequência acima de 2.000 Hz é filtrada, a parte mais escura, quando comparada a Figura 13a.



(a) Sinal de áudio de 10 segundos representado usando o Espectrograma.



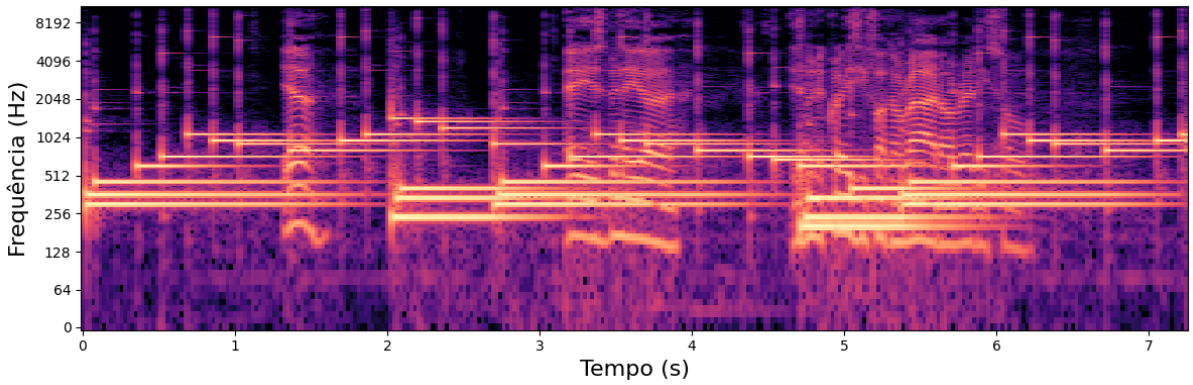
(b) Sinal de áudio de 10 segundos aplicado filtro passa baixa com frequência de corte em 2.000Hz

Figura 13 – Exemplos de um sinal de áudio de 10 segundos apresentando a aplicação de um filtro passa baixa

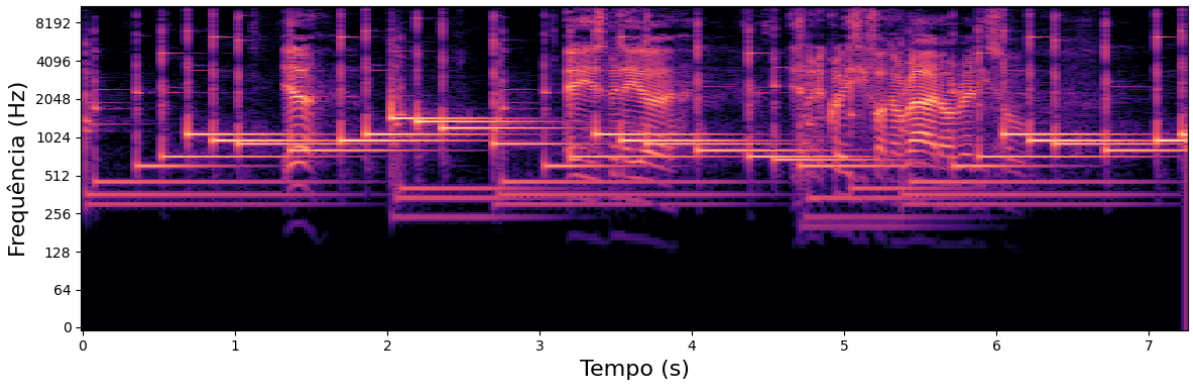
3.2.4 Filtro passa-alta

Ao contrário do filtro anterior, o filtro passa-alta remove componentes de um sinal que estejam abaixo de uma frequência de corte (ROCCHESSO, 2003).

A Figura 14b mostra a aplicação desse filtro a um sinal de áudio de dez segundos, onde a parte mais escura da figura, quando comparada a Figura 14a, são os componentes filtrados.



(a) Sinal de áudio de 10 segundos representado usando o Espectrograma.



(b) Sinal de áudio de 10 segundos aplicado filtro passa alta com ponto de corte em 300Hz

Figura 14 – Exemplos de um sinal de áudio de 10 segundos apresentando a aplicação de um filtro passa baixa

3.2.5 Ruídos ambiente

Nessa técnica utiliza-se um segundo conjunto de dados, que por sua vez, é uma coleção pública de áudios denominada *ESC-50: Dataset for Environmental Sound Classification* (PICZAK, 2015). Essa coleção contém 2.000 arquivos de áudio de 50 diferentes sons ambientes, como sons de latido de cachorro, de chuva, de vento, de gargalhada, de passos, entre outros.

Para a aplicação dessa técnica, os sons são escolhidos aleatoriamente, assim como quando se inicia e termina a inserção dos sons no sinal de áudio que se deseja transformar.

3.2.6 Respostas de impulso - Ruídos Curtos

Para a utilização dessa técnica, foi utilizada outra coleção pública de áudios denominada EchoThief (ECHOTHIEF, 2023). Por sua vez, essa coleção contém 115 arquivos de áudio de resposta de impulsos. Por sua vez, respostas de impulsos são áudios curtos, de milésimos de segundos, utilizadas para capturar características sonoras de um local, como por exemplo teatros.

Assim como nos ruídos ambientes, a escolha dos áudios, onde se inicia e onde termina, foi feita aleatoriamente para a inserção dos áudios no sinal de áudios que se desejava modificar.

3.2.7 Aumento de dados

Escolhidas as estratégias a serem utilizadas, deu-se início à etapa de aumento do conjunto de dados (*data augmentation*) de treino. Essa etapa consistiu na realização da replicação do conjunto segmentado em sete vezes. Ao contrário dos conjuntos de teste e validação, em que todas as técnicas são aplicadas juntas no conjunto de dados, no conjunto de treinamento, cada técnica foi aplicada individualmente em cada réplica, com exceção da última réplica, que por sua vez foram aplicadas todas as técnicas juntas. Isso foi realizado para o algoritmo classificador aprendesse a filtrar cada técnica individualmente e quando todas fossem aplicadas juntas.

É importante destacar que a primeira réplica, ou conjunto inicial, não passou por nenhuma técnica de processamento, mantendo os segmentos originais inalterados. Em seguida, as técnicas de processamento foram incorporadas nas réplicas subsequentes do conjunto de treino. Isso significa que na segunda réplica, somente a técnica de normalização de pico foi aplicada sobre os segmentos de áudio, na terceira réplica apenas o ruído gaussiano por SNR e assim sucessivamente até a penúltima réplica.

Vale ressaltar que em ambos os casos a aplicação das técnicas de processamento são aplicadas a cada segmento de áudio.

3.3 Transformação

Apesar de ter havido um tratamento dos conjuntos de dados, é incomum aplicar algoritmos de aprendizado de máquina diretamente sobre o sinal de áudio no domínio temporal. Como resultado, deu-se início à transformação dos segmentos de áudio com o auxílio da biblioteca Librosa (MCFEE et al., 2015).

Foram então selecionados os seguintes métodos para extrair características do sinal de áudio, que são frequentemente encontrados na literatura de processamento de áudio: a

Transformada de Fourier de Tempo Curto, o Espectrograma Mel e os Coeficientes Cepstrais de Frequência Mel.

Cada um desses métodos foi utilizado com os parâmetros padrões disponibilizados pela biblioteca, sendo aplicado separadamente a cada segmento de áudio dos conjuntos de dados. Essas transformações foram avaliadas em combinação com a redução de dimensionalidade obtida através de estruturas de *autoencoders*.

Com esse propósito, foram desenvolvidas quatro diferentes arquiteturas de *autoencoders*, cada uma empregando uma abordagem de rede neural distinta: MLP, CNN, LSTM e BiLSTM (MARCHI et al., 2015).

3.4 Avaliação

Com o pré-processamento realizado, deu-se início à avaliação de algoritmos de aprendizado de máquina. Para isso avaliou-se o desempenho dos algoritmos KNN e MLP usando a métrica F1-Score, tendo como *baseline* o desempenho da ferramenta Panako, que utiliza a técnica de *fingerprinting*.

Para o algoritmo KNN, usou-se a implementação padrão disponível na biblioteca Scikit-Learn, alterando apenas o parâmetro $K = 2$. Já para o algoritmo MLP criou-se um rede artificial densa com a ajuda da biblioteca Tensorflow, tal arquitetura é descrita na Tabela 2, onde F corresponde ao número de *features* da camada de entrada.

Tabela 2 – Arquitetura do Perceptron Multicamadas

Camada	Número de Neurônios	Função de Ativação
Entrada	F	
Camada densa	$F * 4$	ReLU
Dropout	0.3	
Camada densa	$F * 2$	ReLU
Saída	1.000	

3.5 Aplicação Cliente-Servidor

Por fim, foi desenvolvida uma aplicação cliente-servidor para ilustrar o caso de uso real da solução. Nesse sentido, desenvolveu-se um software *client-side* responsável por coletar uma amostra da música que se deseja identificar e enviar ao servidor. Enquanto o software *server-side*, por sua vez, foi responsável por receber essa amostra e identificar a música, enviando essa resposta ao cliente.

Para isso, do lado do servidor, utilizou-se o *framework* Django (FOUNDATION, 2023) junto ao banco de dados Postgres para a criação de uma aplicação na arquitetura *Model View Controller* (MVC), com interface *Representational State Transfer* (REST) para a conexão com o cliente. Nele foi integrado o *script* de criação do conjunto de dados e aprimorado para que tornasse possível a coleta de mais metadados do Spotify, como o gênero musical, fotos dos álbuns e artistas. Esse procedimento tornou possível guardar os modelos gerados na etapa de experimentação dos algoritmos.

No lado da aplicação cliente, foi criada uma aplicação usando o *metaframework* NextJS (VERCEL, 2023). Ela é responsável por ser a interface do usuário com a aplicação servidora. Criou-se a funcionalidade de gravação ou *upload* de um trecho de áudio, com duração mínima de cinco segundos e máxima de trinta segundos, que é enviada para ser inferida pelo servidor e explicada na seção seguinte.

3.6 Inferência

A inferência, neste trabalho, é etapa onde, dado um trecho qualquer de uma música desconhecida, tenta-se predizer uma música do conjunto da dados que melhor se assemelha. Neste trabalho, tal processo se dá pelo envio de um trecho de áudio utilizando a aplicação cliente. Esse trecho de áudio então é segmentado em tamanhos fixos com saltos de 250ms. Cada segmento é representado utilizando um método extrator de característica e um *autoencoder* e, então, todos os segmentos são passados ao modelo classificador, que irá predizer a música mais provável para cada segmento.

Vale ressaltar que o tamanho de cada segmento, o método extrator de características, o *autoencoder* e o algoritmo modelo classificador são decididos em cada experimentação, quando gerado o modelo classificador final.

4 Resultados e discussão

Neste capítulo são mostrados os resultados obtidos na experimentação de diferentes combinações de métodos extratores de características, *autoencoders* e algoritmos de aprendizado de máquina no exercício de identificação de música dado um trecho de áudio como exemplo. Tendo como referência de comparação, o resultado da métrica F1-Score da execução da ferramenta Panako utilizando o mesmo conjunto de dados foi 0,9983.

A Tabela 3 apresenta os resultados dos experimentos usando o tamanho de segmento de três segundos. Nela pode-se visualizar que a combinação da representação Espectrograma, junto ao *autoencoder* BiLSTM e algoritmo classificador MLP obtiveram o melhor desempenho F1-Score dentre os experimentos executados, resultando em 0,8214. Por outro lado, usando a mesma representação e algoritmo classificador, porém, utilizando o *autoencoder* MLP obteve-se o pior desempenho dentre os experimentos, resultando em 0,0002.

Tabela 3 – Resultados da Experimentação

Representação	*AutoEncoder*	*Algoritmo*	*F1-Score*
Espectrograma	BiLSTM	MLP	0,8214
Espectrograma	LSTM	MLP	0,6611
Espectrograma	BiLSTM	KNN	0,6304
Espectrograma	LSTM	KNN	0,5455
MFCC	BiLSTM	KNN	0,5248
MFCC	CNN	KNN	0,4983
MFCC	LSTM	KNN	0,4382
Espectrograma Mel	CNN	KNN	0,4317
MFCC	LSTM	MLP	0,2081
MFCC	CNN	MLP	0,1800
Espectrograma Mel	BiLSTM	KNN	0,1607
MFCC	BiLSTM	MLP	0,1583
Espectrograma Mel	LSTM	KNN	0,1460
Espectrograma Mel	CNN	MLP	0,0904
Espectrograma Mel	LSTM	MLP	0,0614
Espectrograma Mel	BiLSTM	MLP	0,0570
MFCC	MLP	KNN	0,0445
MFCC	MLP	MLP	0,0219
Espectrograma Mel	MLP	KNN	0,0035
Espectrograma	MLP	KNN	0,0029
Espectrograma Mel	MLP	MLP	0,0002
Espectrograma	MLP	MLP	0,0002

O modelo com melhor desempenho foi utilizado na aplicação servidor para realizar

a inferência de trechos de áudios enviados pelo usuário.

Pelos testes executados, a assertividade do modelo quando passado um trecho de áudio modificado sinteticamente usando as técnicas propostas neste trabalho está de acordo com o resultado demonstrado na Tabela 3. Porém, usando trechos de áudios gravados por um microfone de um celular ou computador, o software apresenta baixa assertividade, raramente apresentando a música correta. O mesmo acontece com a ferramenta Panako, que utiliza a técnica de *audio fingerprinting*.

Em suma, o modelo apresenta resultados ruins na identificação de músicas, devido ao grande números de classes, pequeno número de documentos por classe, sendo necessários mais estudos na área de recuperação de informações musicais. Espera-se em trabalhos futuros que seja feita uma avaliação dos parâmetros das técnicas de representação, assim como a parametrização dos *autoencoders* e algoritmos classificadores. Inclusive, utilizar parte da técnica de *audio fingerprinting* como dado de entrada para um algoritmo classificador pode ser considerada.

Referências

ADOBE. *Digitizing Audio*. 2021. <https://helpx.adobe.com/audition/using/digitizing-audio.html>. Citado na página 12.

APPLE. *iTunes*. 2001. Disponível em: <https://www.apple.com/itunes/>. Citado na página 8.

APPLE, S. E. L. *Shazam*. 2002. Disponível em: <https://www.shazam.com/>. Citado na página 8.

CADENCE. *What is Signal-to-Noise Ratio and How to Calculate It*. 2020. Disponível em: <https://resources.pcb.cadence.com/blog/2020-what-is-signal-to-noise-ratio-and-how-to-calculate-it>. Citado na página 25.

ECHOTHIEF. 2023. <http://www.echothief.com/>. Citado na página 27.

FECHINE, J. M. *A transformada de Fourier e Suas Aplicações*. 2010. <http://www.dsc.ufcg.edu.br/~pet/ciclo_seminarios/tecnicos/2010/TransformadaDeFourier.pdf>. Citado na página 18.

FERRARI, D. G.; SILVA, L. N. D. C. *Introdução a mineração de dados*. [S.l.]: Saraiva Educaçao SA, 2017. Citado 3 vezes nas páginas 15, 16 e 20.

FILHO, K. de S. O. *Transformada de Fourier*. 2011. <http://astro.if.ufrgs.br/med/imagens/fourier.htm>. Citado na página 18.

FOUNDATION, D. S. *Django*. 2023. Disponível em: <https://djangoproject.com>. Citado na página 29.

FU, Z. et al. A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, v. 13, n. 2, p. 303–319, 2011. Citado na página 8.

GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. [S.l.]: "O'Reilly Media, Inc.", 2019. Citado na página 20.

GIRALDELI, F. B. Avaliação de técnicas espectrais aplicadas à remoção de ruído em sinais de Áudio musical. 2009. Citado 2 vezes nas páginas 11 e 12.

GROUP, P. G. D. *PostgreSQL Database Management System*. 1996. <https://www.postgresql.org/>. Citado na página 21.

LIBET, Q. *mutagen*. 2023. Disponível em: <https://mutagen.readthedocs.io/>. Citado na página 21.

LOOPMASTERS. 2023. Disponível em: <https://www.loopmasters.com/articles/2993-Peak-Normalization-Explained>. Citado na página 24.

MANTIONE, P. *Digital Audio 101: The Basics*. 2021. <https://theproaudiofiles.com/digital-audio-101-the-basics/>. Citado na página 10.

MARCHI, E. et al. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2015. p. 1996–2000. Citado na página 28.

MCFEE, B. et al. librosa: Audio and music signal analysis in python. In: *Proceedings of the 14th python in science conference*. [S.l.: s.n.], 2015. v. 8. Citado na página 27.

MLEARNERE. *Learning from Audio: The Mel Scale, Mel Spectrograms, and Mel Frequency Cepstral Coefficients*. 2020. <https://towardsdatascience.com/learning-from-audio-the-mel-scale-mel-spectrograms-and-mel-frequency-cepstral-coefficients-f5752b63 Citado na página 19.

MOHAJER, K. *SoundHound*. 2005. Disponível em: <https://www.soundhound.com/>. Citado na página 8.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, Manole, v. 1, n. 1, p. 32, 2003. Citado na página 14.

NORDBY, J. *Environmental Sound Classification on Microcontrollers using Convolutional Neural Networks*. Dissertação (Mestrado) — Norwegian University of Life Sciences, 5 2019. Disponível em: <http://hdl.handle.net/11250/2611624>. Citado na página 14.

NUNES, J. et al. Noções sobre as séries de fourier. 2002. Citado na página 17.

PANAGAKIS, Y.; BENETOS, E.; KOTROPOULOS, C. Music genre classification: A multilinear approach. In: . [S.l.: s.n.], 2008. p. 583–588. Citado na página 8.

PICZAK, K. J. ESC: Dataset for Environmental Sound Classification. In: *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015. p. 1015–1018. ISBN 978-1-4503-3459-4. Disponível em: <http://dl.acm.org/citation.cfm?doid=2733373.2806390>. Citado na página 26.

PYTHON, S. F. *Python*. 2023. Disponível em: <https://www.python.org/>. Citado na página 21.

ROBERTS, L. *Understanding the Mel Spectrogram*. 2020. <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53>. Citado na página 18.

ROCCHESSO, D. *Introduction to sound processing*. [S.l.]: Mondo estremo, 2003. Citado 2 vezes nas páginas 25 e 26.

SIX, J.; LEMAN, M. Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification. In: *Proceedings of the 15th ISMIR Conference (ISMIR 2014)*. [S.l.: s.n.], 2014. Citado na página 9.

SOUZA, M. N. V. de. Comparaçao de algoritmos do aprendizado de máquina aplicados na mineraç ao de dados educacionais. *Tcc, Universidade Federal Rural de Pernambuco*, 2015. Citado na página 14.

SPOTDL. *spotDL*. 2023. Disponível em: <https://spotdl.readthedocs.io/>. Citado na página 21.

SPOTIFY. *Spotify Million Playlist Dataset Challenge*. 2020. Disponível em: <https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge>. Citado na página 21.

SPOTIPY. *Spotipy*. 2023. Disponível em: <https://spotipy.readthedocs.io/>. Citado na página 21.

THIAGO, E. R. S. Reconhecimento de voz utilizando extração de coeficientes mel-cepstrais e redes neurais artificiais. 2017. Citado 2 vezes nas páginas 18 e 19.

VERCEL. *Next.js*. 2023. Disponível em: <https://nextjs.org/>. Citado na página 29.

WESCOTT, T. Sampling: what nyquist didn't say, and what to do about it. *Wescott Design Services, Oregon City, OR*, 2010. Citado na página 12.