UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO ALMEIDA COSTA

IDENTIFICAÇÃO DE ACORDES DE VIOLÃO UTILIZANDO MACHINE LEARNING

CHAPECÓ
2021

RODRIGO ALMEIDA COSTA

IDENTIFICAÇÃO DE ACORDES DE VIOLÃO UTILIZANDO MACHINE LEARNING

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.
Orientador: Dr. Denio Duarte

CHAPECÓ
2021

Costa, Rodrigo Almeida

Identificação de acordes de violão utilizando Machine Learning / Rodrigo Almeida Costa. – 2021.
45 f.: il.

Orientador: Dr. Denio Duarte.
Trabalho de conclusão de curso (graduação) – Universidade Federal da Fronteira Sul, curso de Ciência da Computação, Chapecó, SC, 2021.

1. Aprendizado de Máquina. 2. Música. 3. Violão. 4. Acordes. 5. Reconhecimento. I. Duarte, Dr. Denio, orientador. II. Universidade Federal da Fronteira Sul. III. Título.

E-mail: rodrigo.acosta@estudante.uffs.edu.br

RODRIGO ALMEIDA COSTA

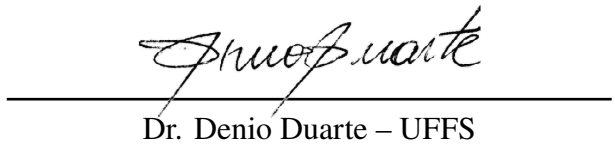
IDENTIFICAÇÃO DE ACORDES DE VIOLÃO UTILIZANDO MACHINE LEARNING

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Dr. Denio Duarte

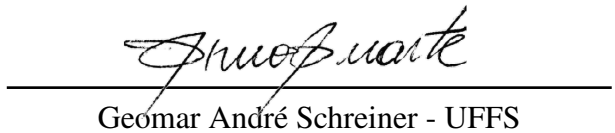
Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 18/10/2021.

BANCA AVALIADORA



Dr. Denio Duarte – UFFS

Dr. Guilherme Dal Bianco - UFFS



Geomar André Schreiner - UFFS

AGRADECIMENTOS

Dedico este trabalho especialmente aos meus avós Ana Szura Costa e Raimundo Graff. Lembro da felicidade quando vocês souberam que eu havia conseguindo ingressar na UFFS, e agora, lá de cima, estão me vendo concluir o curso. Só Deus sabe como eu queria que vocês estivessem aqui para compartilhar este momento da minha vida.

Agradeço à minha família, em especial meus pais Mauro e Janete e minha irmã Renata, que sempre me apoiaram nos momentos difíceis, obrigado pelos conselhos e pelas tantas vezes em que me ajudaram sem medir esforços. A minha namorada e companheira Simone que me puxava a orelha quando não me via estudando para o TCC e que me acalmava nos momentos de pânico, devo esta conquista a você, muito obrigado por tudo.

Aos tantos amigos feitos na UFFS, em especial ao Murillo, Bruno, Lucas, Natan, Eduardo, Rafael e Rodolfo, que foram essenciais durante esta jornada. Obrigado pelos finais de semana estudando cálculo e revezando os trabalhos, pelas conversas fiadas no intervalo e pelas tantas vezes em que me socorreram nas matérias. Foi um grande honra tê-los como colegas.

Agradeço à todos os professores que tive o privilégio de ser aluno. Creio que uma faculdade inteira foi pouco para usufruir de todo o conhecimento que vocês possuem. Obrigado por estarem sempre disponíveis, por serem comprometidos com o aprendizado e por descomplicar assuntos tão complexos envolvidos neste curso. Um muito obrigado ao saudoso professor Dr. José Carlos Bins Filho, seu legado será eterno.

Um agradecimento especial meu orientador Dr. Denio Duarte. Obrigado pela paciência e por acreditar no meu potencial. Sua maestria na forma de organizar as tarefas e os seus conselhos foram essenciais para tornar o TCC mais fácil e prazeroso de se fazer.

RESUMO

A transcrição musical é o ato de anotar um som ou uma peça gerado por um instrumento que não tenha sido anotada anteriormente. Trata-se de um trabalho manual que necessita de um conhecimento em teoria musical. A tarefa de anotação dos acordes faz parte da transcrição musical. Esta tarefa pode ser feita utilizando aprendizado de máquina. Estudos para criação de ferramentas de anotações de acordes de forma automática utilizando aprendizado de máquina têm mantido o foco na tarefa de classificação e pós-processamento. Recentemente, trabalhos relacionados à área mostraram que o foco no pré-processamento, com algoritmos que tornam os exemplos do *dataset* mais descritivos sobre o acorde, pode trazer resultados melhores. O trabalho mostra a criação de um *dataset* utilizando um método eficaz na extração de *features* dos áudios gravados de 41 acordes. Após construção do dataset, foram testados modelos baseados em regressão logística, redes neurais e florestas aleatórias. Entre todos os modelos testados, a floresta aleatória obteve os melhores resultados, aliado a um algoritmo de extração de features que utiliza a técnica Constant-Q transform. A média de precisão do modelo chegou a 99%.

Palavras-chave: Aprendizado de Máquina. Música. Violão. Acordes. Reconhecimento.

ABSTRACT

Music transcription is the act of annotating a sound or piece generated by an instrument that has not been previously annotated. It is a manual work that requires knowledge in music theory. Chord annotation task is part of musical transcription. This task can be done using machine learning. Studies to create automatic chord annotation tools using machine learning have focused on the task of classification and post-processing. Recently, it has been shown that focusing on pre-processing, with algorithms that make dataset examples more descriptive about the chord, can yield better results. The work shows the creation of a dataset using an efficient method for extracting features from 41-chord recorded audio. After construction of the dataset, models based on logistic regression, neural networks and random forests were tested. Among all the models tested, the random forest obtained the best results, combined with a feature extraction algorithm that uses the Constant-Q transform technique. The model's average accuracy reached 99%.

Keywords: Machine learning. Guitar. Music Transcription. Music Information Retrieval

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Music Information Retrieval - MIR é a área dos estudos relacionados a extração de dados em músicas para diversos fins, pode-se destacar detecção dos acordes, estudos destinados a classificação de gênero musical, classificação de instrumentos e identificação do gosto musical para recomendação de músicas (SCHEDL; GÓMEZ GUTIÉRREZ; URBANO, 2014). Existem várias formas de se trabalhar com MIR, uma delas é utilizando a aprendizado de máquina.

Aprendizado de Máquina é uma área de 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. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003).

Técnicas para MIR utilizando aprendizado de máquina têm demonstrado resultados satisfatórios (CHOI et al., 2017). A detecção de acordes musicais utilizando aprendizado de máquina é uma das áreas de estudo associadas à MIR. Um acorde é uma combinação de duas ou mais notas tocadas simultaneamente. O principal desafio deste campo é criar modelos sensíveis as pequenas variações de notas, tornando-os capazes de distinguir entre as várias configurações disponíveis (CLETO et al., 2010). Em um violão é possível formar muitos acordes, por isso, foi necessário limitar a quantidade de acordes considerados no aprendizado. Por exemplo, considerando as 12 notas base e as variações menor (m), com sétima (7), com nona (9), com sexta (6), diminuto (º) e aumentado (+), é possível formar 768 acordes ($12 * 2^6$). Para este trabalho, a quantidade de acordes foi limitado a 12 notas e duas variações, menor (m) e com sétima (7), totalizando 48 acordes.

Existem diversas técnicas de aprendizado de máquina. Modelos classificadores baseados em algorítimos como regressão linear e florestas aleatórias podem ser eficazes para o reconhecimento de acordes. Para isto, deve ser utilizado exemplos extraídos por um método robusto de extração de *features*, como visto nos trabalhos de (KORZENIOWSKI; WIDMER, 2016), (MCFEE; BELLO, 2017) e (NADAR; ABESSER; GROLLMISCH, 2019). Os autores em (KORZENIOWSKI; WIDMER) mostram a importância do pré-processamento dos dados antes de submeter ao treinamento de um modelo, tirando o foco da aplicação de métodos artesanais no pós processamento. Os autores também defendem que um processo robusto para extração de *features* pode simplificar a etapa do treinamento, tornando viável a utilização de métodos classificadores menos complexos para a tarefa de criação do modelo e pós-processamento.

Este trabalho trata do problema da classificação de acordes de violão utilizando algoritmos de aprendizado de máquina. Para realizá-lo, foram gravados os acordes utilizando um violão e um celular, processadas as gravações e extraídas as *features* baseadas em cromagramas para criar um arquivo CSV com os dados dessas gravações. O motivo que levou à criação de um *dataset* foi a dificuldade em encontrar um *dataset* no formato desejado na web. A característica do *dataset* desejado é possuir gravações de acordes isolados, sem a presença de outros

instrumentos. Também deve haver exemplos padronizados para todos os acordes das variações desejadas. A escolha por construir um *dataset* é perigosa pois nos trabalhos observados, os autores testaram sua eficácia em sons gerados por sintetizadores e músicas gravadas em alta qualidade em estúdios.

As *features* foram extraídas utilizando a biblioteca librosa[1], implementada na linguagem python. Foram testadas as Transformadas Short-Time Fourier Transform - STFT e Constant-Q Transform - CQT, mostrando-se mais adequada a Transformada CQT para o objetivo deste trabalho.

1.1 ESTRUTURA DO TRABALHO

O Capítulo 2 apresenta brevemente os conceitos de notas e acordes musicais. São conceituadas as formações de acordes pela combinação de notas, a representação do som através de ondas, e a exibição das ondas sonoras através de Transformadas. O Capítulo 3 descreve a biblioteca librosa e as ferramentas nela implementadas que foram empregadas neste trabalho. O Capítulo 4 apresenta os conceitos de Machine Learning e os tipos de aprendizado. O Capítulo 5 aprofunda três trabalhos relacionados ao tema. O Capítulo 6 contém detalhes sobre a criação do *dataset*. Também apresenta o resultado dos experimentos com modelos baseados em florestas aleatórias. O Capítulo 7 encerra o trabalho e indica trabalhos futuros.

[1] https://librosa.org/

2 ACORDES MUSICAIS

Para entender o que são os acordes musicais que este trabalho pretende identificar, se faz necessário descrever algumas definições. Nas seções seguintes serão mostrados os elementos relacionadas as nota musicais, a física dos sons, harmônicas e quando essas definições estão empregadas no violão.

2.1 NOTAS MUSICAIS

As notas musicais são notações dadas para determinadas frequências a partir de uma frequência base e podem ser representadas por uma progressão geométrica de razão $\sqrt[12]{2}$ (DODGE; JERSE, 1997). As notas musicais naturais existentes são Dó, Ré, Mi, Fá, Sol, Lá e Si, os símbolos que representam essas notas são, respectivamente, C, D, E, F, G, A e B, símbolos esses que musicalmente são conhecidos como cifras. A Figura 1 mostra as sete notas musicais na escala natural e as cifras que as denotam.

Dó ou **C**	Ré Ou **D**	Mi Ou **E**	Fá Ou **F**	Sol Ou **G**	Lá Ou **A**	Si Ou **B**

Figura 1 – Notas musicais naturais de uma oitava.

As notas musicais estão distribuídas na faixa audível (que será explicado na seção 2.2) em intervalos repetidos chamados de oitavas. A Figura 1 apresenta uma sequência com todas as notas naturais de uma oitava, partindo da nota mais grave a esquerda e mais aguda a direita. A última nota de uma oitava (Si ou "B") antecede a primeira nota da próxima oitava (Dó ou "C"). A frequência de uma mesma nota em diferentes oitavas será sempre um valor múltiplo, de modo que, na oitava acima, a mesma nota possui o dobro da frequência, e, na oitava abaixo, possua metade da frequência. Por exemplo, a nota Lá na quarta oitava do intervalo audível possui frequência de $440Hz$, consequentemente, a nota Lá da oitava acima tem frequência de $440 \cdot 2Hz$ e da oitava abaixo $\frac{440}{2}Hz$.

Além das notas naturais, existem notas entre algumas das notas naturais. Estas notas são chamadas de acidentes musicais, representados pelos símbolos ♯ chamado de "sustenido"para aumentar a nota natural em um semitom e ♭ chamado do "bemol"para diminuir uma nota natural em um semitom. Na Figura 2 são colocadas em escala todas as notas e seus respectivos nomes, sendo os acidentes considerados como notas. As notas E e B não possuem o acidente ♯, pois a nota E♯ é equivalente a nota F, e B♯ é equivalente a nota C da próxima oitava. Pelo mesmo motivo, as notas C e F não possuem o acidente ♭. Com isso, a quantidade de notas de uma oitava é aumentada de sete para doze notas, que é o somatório de sete notas naturais mais cinco acidentes.

Nota → / ↓Nome	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª	11ª	12ª
Principal	Dó ou **C**	Dó ♯ Ou **C** ♯	Ré Ou **D**	Ré ♯ Ou **D** ♯	Mi Ou **E**	Fá Ou **F**	Fá ♯ Ou **F** ♯	Sol Ou **G**	Sol ♯ Ou **G** ♯	Lá Ou **A**	Lá ♯ Ou **A** ♯	Si Ou **B**
Secundário		Ré ♭ Ou **D** ♭		Mi ♭ Ou **E** ♭			Sol ♭ Ou **G** ♭		Lá ♭ Ou **A** ♭		Si ♭ Ou **B** ♭	

Figura 2 – Todas as doze notas musicais de uma oitava considerando os acidentes. Notas acidentadas podem ter dois nomes, tendo preferência o nome principal nas notações de musicas.

2.2 A FÍSICA DOS SONS

O som é uma vibração mecânica na faixa dos 20 Hz até 20.000 Hz, considerando a faixa na qual o ouvido humano pode ouvir. Essa vibração é propagada por um meio elástico, como o ar, e sentida pelos tímpanos. A Figura 3 apresenta diagramaticamente a propagação do som. O intervalo de uma vibração mecânica é medido pela sua frequência (OGASAWARA et al., 2008). Frequências sonoras podem ser representadas por senoides, como será apresentado na seção 2.2.1.

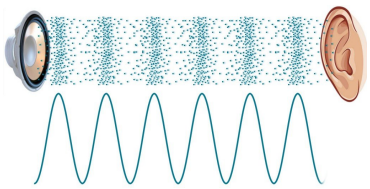


Figura 3 – Como o som produzido por um auto-falante chega aos nossos ouvidos. Fonte: scienceabc.com/pure-sciences/movement-of-sound-waves-through-different-media

2.2.1 Representação das Ondas Sonoras

Toda frequência pode ser representada por uma senoide no domínio do tempo. No caso das frequências sonoras, os eixos X e Y representam o tempo e a amplitude, respectivamente. Na Figura 4 são apresentados dois gráficos: no gráfico a esquerda, está uma onda senoidal composta pela frequência da nota C, e no gráfico a direita, está representada uma onda com as frequências das notas que compõem o acorde de C maior somadas. Quando um som é formado por somente uma nota, como no caso do gráfico a esquerda da Figura 4, é chamado de som monofônico. Já sons formados por várias notas tocadas simultaneamente, como do gráfico a direita da Figura 4, são chamados de sons polifônicos.

Para que seja possível descobrir quais as notas compõem um determinado som polifônico, é necessário aplicar um cálculo chamado de transformada (CLETO et al., 2010). Modelos de transformadas serão descritos na subseção 2.2.5

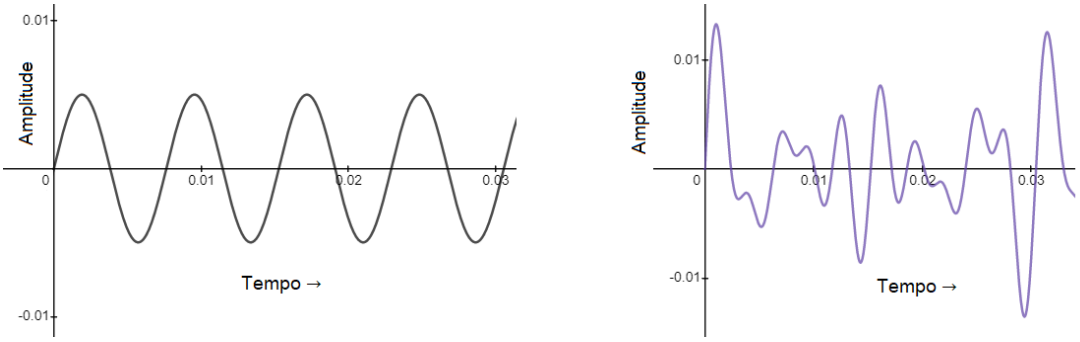


Figura 4 – Representação senoidal da nota C e do acorde maior de C.

2.2.2 Representação digital das Ondas Sonoras

Um sinal sonoro pode ser representado por uma sequência de valores reais, o tamanho dessa sequência está diretamente ligado a fidelidade do áudio, quanto mais informações essa sequência armazenar, mais fiel a realidade o sinal sonoro é. Na música essa densidade da sequência é chamada de Taxa de Amostragem, queé medida em Hz. Em uma música de frequência de sinal 44,1 KHz, por exemplo, cada segundo da música é representado por uma sequência de tamanho n onde $n = 44100$ e cada posição representa a altura da onda sonora (amplitude). Uma música com dois minutos será representada por $44100 * 120 = 5.292.000$ valores reais.

Para converter um arquivo de áudio em um arquivo interpretável a um modelo de reconhecimento, pode-se utilizar o software Sonic Visualiser, esse software, possibilita gerar um arquivo CSV - *Comma-separated values* a partir de um arquivo de áudio. Na Figura 5, é exibido um exemplo de um arquivo de áudio de 2s em 44100 Hz convertido para CSV, a primeira coluna indica o índice do sinal e a segunda a amplitude do sinal sonoro.

Índice do Sinal	Amplitude
0	0.0000000
1	0.0249023
2	0.0233459
3	0.0219116
4	0.0225525
⋮	⋮
88198	-0.0430335
88190	-0.0450745
88200	-0.0536499

Figura 5 – Representação de um arquivo de áudio por um arquivo CSV.

O arquivo CSV gerado pode ser usado para representação gráfica por ondas, similar as ondas senoidais vistas anteriormente.

2.2.3 Notas musicais no violão

Um violão emite som a partir do ar deslocado pelo seu interior pela vibração das cordas. A frequência da vibração é determinada pela espessura e comprimento da corda. Ao vibrar uma corda com espessura maior, na tensão recomendada para a corda, a tendência é que o som seja mais grave do que as cordas com espessuras menores. Da mesma forma que, ao pressionar, a corda em uma determinada casa, o comprimento da corda é reduzido, resultando em um som mais agudo na medida em que o comprimento da corda é diminuído. Na Figura 6, pode-se observar como as frequências estão distribuídas ao longo do braço em intervalos chamados "Casas".

7ª Casa	6ª Casa	5ª Casa	4ª Casa	3ª Casa	2ª Casa	1ª Casa	Corda solta	
B – 123,47	A# - 116,54	A – 110	G# - 103,83	G – 98	F# - 92,5	F – 87,31	E – 82,41	6ª Corda
E – 164,81	D# - 155,56	D -146,83	C# - 138,59	C – 130,81	B – 123,47	A# - 116,54	A – 110	5ª Corda
A – 220	G# - 207,65	G – 196	F# – 185	F – 174,61	E – 164,81	D# - 155,56	D – 146,83	4ª Corda
C# - 277,18	C – 261,63	C - 261,63	B – 246,94	A# - 233,08	A – 220	G# - 207,65	G – 196	3ª Corda
F# - 369,99	F -349,23	E – 329,63	D# - 311,13	D – 293,66	C# - 277,18	C – 261,63	B – 246,94	2ª Corda
B – 493,88	A# - 466,16	A – 440	G# - 415,30	G – 392	F# - 369,99	F -349,23	E – 329,63	1ª Corda

Figura 6 – Representação do braço do violão até a sétima casa com suas determinadas frequências em Hz.

Os violões podem conter diversas configurações. As características que podem influenciar neste trabalho são o número de cordas e a quantidade de casas, pois, isso influencia diretamente nas notas mais graves e agudas emitidas por ele. Partindo dessa afirmativa, é necessário definir as características do violão que será utilizado para as gravações. O violão escolhido contém as seguintes características:

- Possui 6 Cordas;
- Está afinado em 440 Hz;
- Tem como nota mais grave a E2 - $(88,41 Hz)$;
- Tem como nota mais aguda a E6 - $(1.318,52 Hz)$;
- Possui 24 casas;

2.2.4 Acordes

Acordes são comumente definidos como duas ou mais notas tocadas simultaneamente e harmonicamente, por esse motivo, um acorde por si só é um som polifônico. As formações de acordes podem ser classificadas como maior, menor, aumentado e diminuto. O que diferencia cada tipo de acorde é a distancia entre as notas que o compõem.

As notas que formam os acordes maiores são as notas de primeiro, terceiro e o quinto graus com a diferença de 4 semitons entre a primeira e a terceira e de 3 semitons entre a terceira e a quinta (veja a Figura 2). Para exemplificar, toma-se o acorde de C como sendo a formação das notas C (primeiro grau), E (resultado de C + 4 semitons) e G (resultado de E + 3 semitons). Na Figura 7 são mostrados todos os acordes maiores em pelo menos uma de suas formações, considerando que o braço do violão permite que o mesmo acorde seja formado em lugares e formações diferentes.

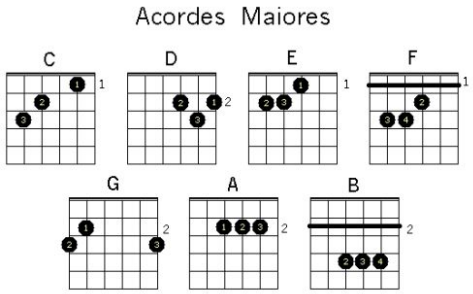


Figura 7 – Um exemplo de formação para cada acorde maior. Um acorde pode ter mais que uma formação.

Os acordes menores são como os acordes maiores, porém, a terceira nota tem distância de 3 semitons da primeira e a quinta tem distância de 4 semitons da terceira, ou seja, possuem a terceiro grau diminuído em um semitom (Figura 2). No caso do acorde C, a nota do terceiro grau E é substituída pela nota $E\flat$. Os acordes menores são representados pela cifra da nota de primeiro grau com um m indicando "menor". No exemplo do acorde C menor, sua cifra é Cm. Na figura 8 é ilustrado ao menos uma formação de cada acorde menor.

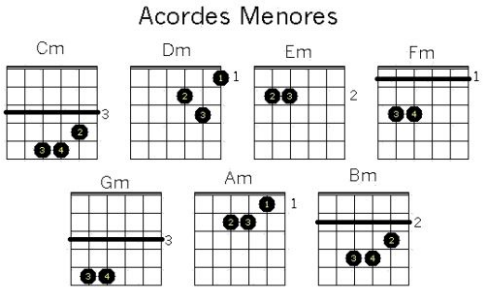


Figura 8 – Um exemplo de formação para cada acorde menor. Um acorde pode ter mais que uma formação.

2.2.5 Transformadas

Como visto na seção 2.2.1, os sons podem ser representados por ondas senoidais no domínio amplitude/tempo. Para que se encontre quais frequências compõem uma onda polifônica, é necessário transformar para o domínio de frequência. Um cálculo que cumpre essa tarefa é

a Transformada de Fourier. Computacionalmente, este cálculo é obtido através do algoritmo *Fast Fourier Transform - FFT*. Na Figura 9 é mostrado no gráfico superior uma linha senoidal composta pelas frequências $5Hz$ e $15Hz$ somadas. No gráfico inferior, é mostrado o resultado da FFT, onde as frequências são destacadas com maior amplitude.

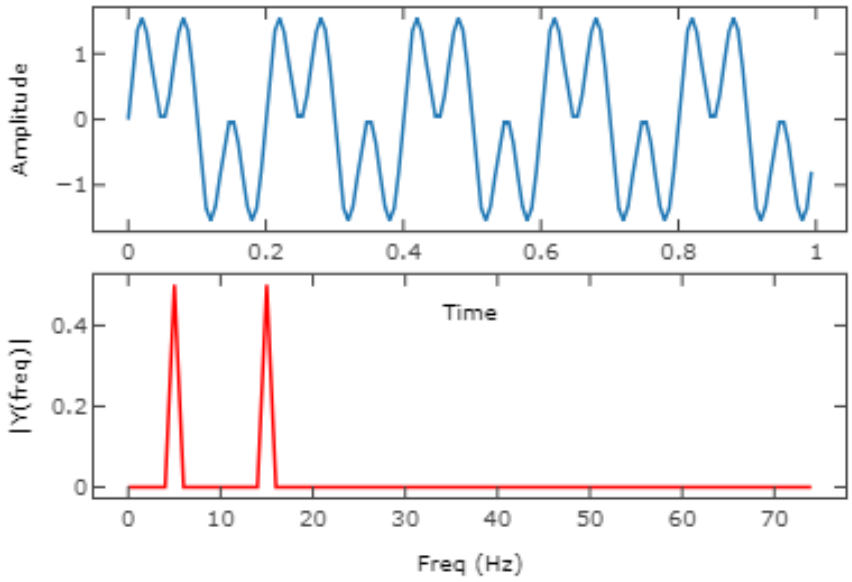


Figura 9 – Transformada de Fourier aplicada a uma onda senoidal composta pelas frequências $5Hz$ e $15Hz$.

Comumente a análise espectral de sinais digitais é associada à aplicação da transformada discreta de Fourier (DFT) em sua implementação rápida (FFT). Todavia, na análise de sinais musicais, a FFT apresenta pouca eficiência nos resultados, devido a distribuição linear de amostras no domínio da frequência, enquanto a música sugere uma escala logarítmica, em função do espaçamento geométrico entre seus semitons (OGASAWARA et al., 2008).

No trabalho de (OGASAWARA et al., 2008) são testadas diversas transformadas e escolhida a mais adequada ao problema proposto. Os autores utilizam a *Transformada de Q Limitado* por distribuir de forma logarítmica o espectro de frequências, com maior resolução nas frequências baixas e menor nas frequências altas.

O conceito consiste em realizar a FFT por partes, variando o tamanho do intervalo de frequência a ser analisado (oitava por oitava por exemplo). Na Figura 10 é mostrada a comparação de uma mesma nota nos dois modelos de transformação.

A partir da FFT e da distribuição logarítmica, é possível obter diversos outras representações das frequências de uma faixa de áudio, podendo-se destacar o *mel-spectrogram* que é mostrado na Figura 11 e o cromagrama que é uma representação obtida a partir de um espectrograma. A Figura 12 mostra um cromagrama que representa mais que uma oitava, porém, para o reconhecimento de acordes, não interessa em qual oitava se encontra a nota, desde que esteja dentro da faixa de frequência emitida pelo violão. Portanto, um cromagrama pode ser

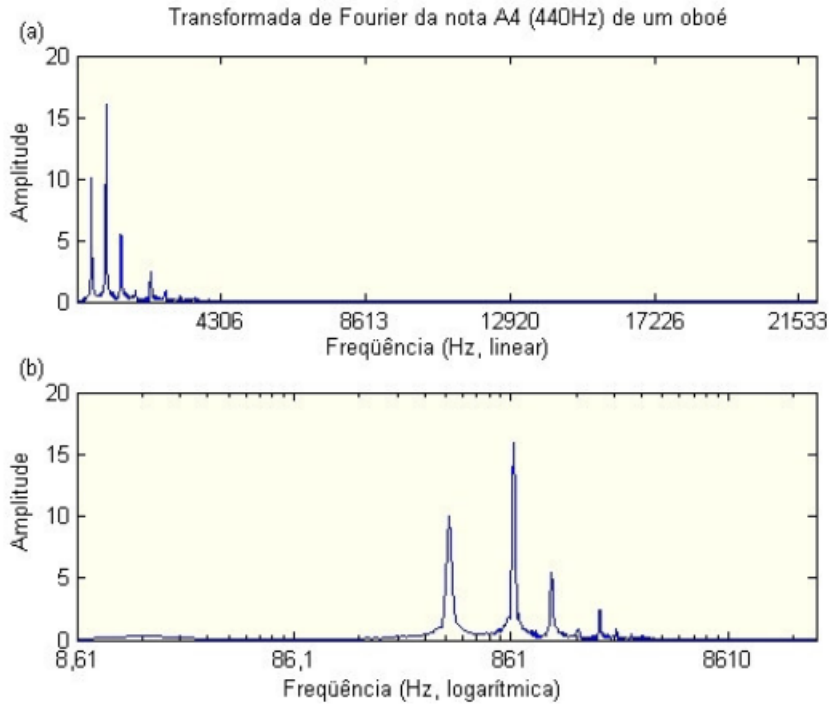


Figura 10 – Comparação entre a FFT e a Transformada de Q Limitado. (OGASAWARA et al., 2008).

simplificado em uma única oitava, somando-se as notas de todas as oitavas em uma única faixa, como pode ser observado na Figura 13.

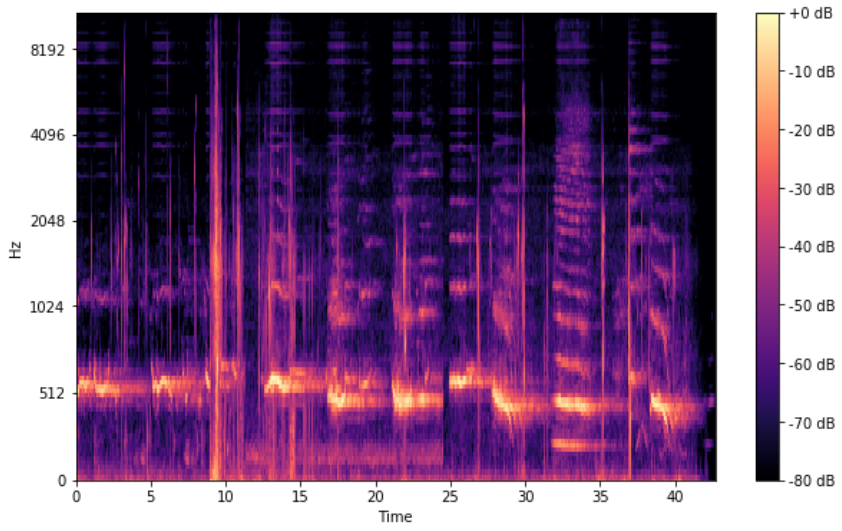


Figura 11 – Mel-spectrogram para representação das ondas sonoras no domínio da frequência em uma distribuição logarítmica.

Neste capítulo foram conceituados os acordes musicais. Foi mostrado como eles são formados no violão, como eles podem ser representados por ondas, espectrogramas e cromagramas. A seguir é apresentada a biblioteca librosa, feita em python. Esta biblioteca possui

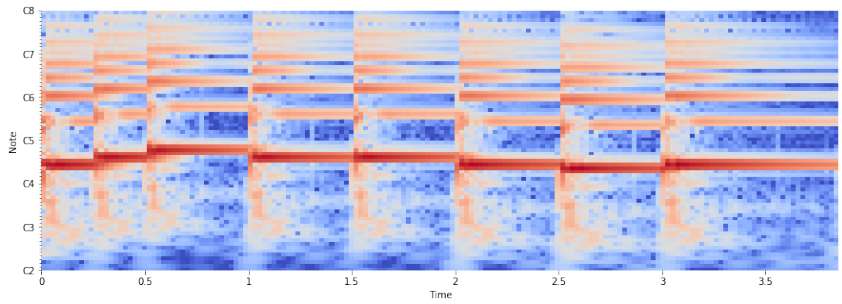


Figura 12 – Cromagrama para representação das ondas sonoras no domínio da frequência, já rotuladas como notas.

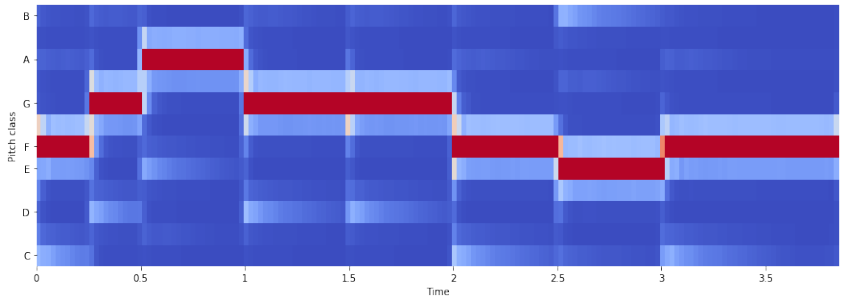


Figura 13 – Agrupamento de um cromagrama para representação das ondas sonoras no domínio de frequência em uma única oitava.

as ferramentas necessárias realizar as transformadas e para extrair os cromagramas mostrados neste capítulo.

3 LIBROSA

O librosa é uma biblioteca desenvolvida na linguagem python para análise de musicas e áudios (MCFEE; RAFFEL et al., 2015). Esta biblioteca disponibiliza as ferramentas necessárias para criar sistemas MIR. Entre as ferramentas disponíveis, foram escolhidas a de extração de cromagramas, utilizando as funções `features.chroma_stft` e `features.chroma_cqt`.

A funções `chroma_stft` e `features.chroma_cqt` computam um cromagrama a partir de um sinal de entrada *y*, este sinal de entrada são os valores de um arquivo de áudio no formato `wav`. Com esta entrada, é aplicada uma Tranformada para processar o sinal e retornar um Array contendo a intensidade (um valor real entre 0 e 1) de cada um dos doze semitons. Todo cromagrama extraído é normalizado para limitar os valores no intevalo de 0 e 1. A nota com maior intensidade identificada ficará com o valor 1 enquanto as demais são recalculadas para manter a proporção. Nos cromagramas extraídos e normalizados utilizando as funções citadas, sempre haverá ao menos um semitom com intensidade de 1. A Figura 14 mostra o cromagrama do acorde de C computado utilizando a função `chroma_stft`. O eixo *y* à esquerda apresenta as notas primários e o da direita a intensidade do acorde no tempo *t*. Perceba que quanto mais clara, maior a intensidade das notas. Já o eixo *x* apresenta o tempo de execução da nota (em segundos). Assim, pode-se observar nas cores mais claras, em geral, os valores mais intensos nas notas C, E, G e B. As notas C, E e G são pertencentes ao acorde de C, já a nota B foi identificada erroneamente pela função em alguns trechos do áudio.

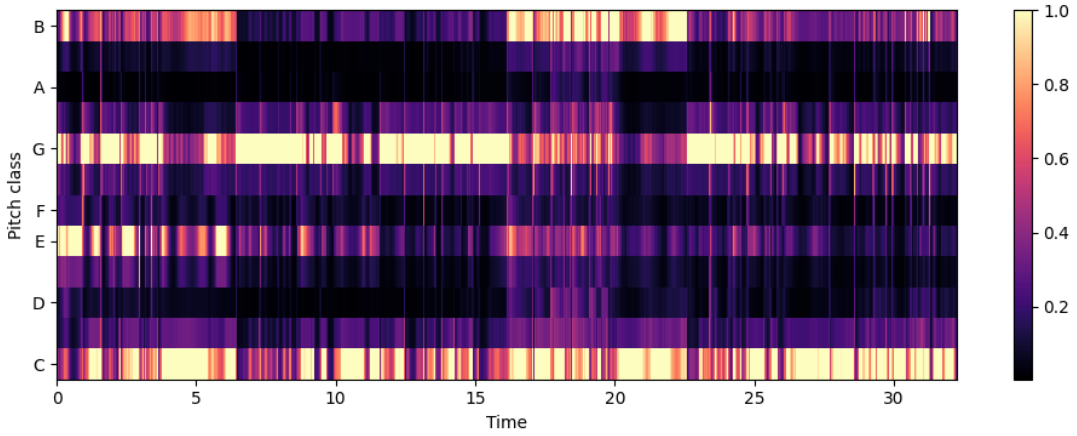


Figura 14 – Cromagrama do acorde de C maior resultante da função stft.

A Figura 15 mostra o cromagrama resultante da função `features.chroma_cqt`. Nela, percebe-se que somente as notas pertencentes ao acorde foram destacadas. A diferença entre as duas funções se dá principalmente pela forma como é realizada a transformada do sinal de áudio. Enquanto a função `chroma_stft` utiliza o algoritmo Short-Time Fourier Transform, a função `chroma_cqt` utiliza a Transformada de Q Constante, assemelha-se ao algoritmo Fast Fourier Transform (FFT), porém, no caso da librosa, aplica-se a FFT oitava por oitava, para fins

de performance. O áudio utilizado pertence ao *dataset* próprio, e nesse arquivo estão presentes de duas à quatro formações do mesmo acorde. Observando o rótulo B de ambas as figuras, percebe-se que o cromagrama gerado pela `features.chroma_stft` (Figura 14) é mais intenso em B que o cromagrama gerado pela `features.chroma_cqt` na Figura 15.

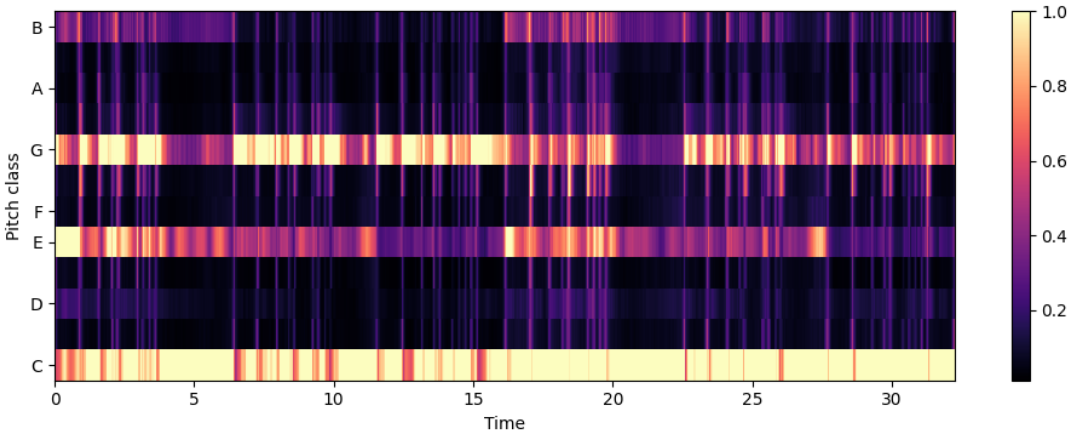


Figura 15 – Cromagrama do acorde de C maior resultante da função CQT.

Este trabalho irá realizar experimentos que as duas transformadas para verificar qual delas é a mais adequada para extrair as *features* dos acordes gerados.

4 APRENDIZADO DE MÁQUINA

O aprendizado de máquina (*Machine learning* - ML) é o campo da computação relacionada à inteligência artificial que estuda formas para que as máquinas, através de uma base de dados para treinamento, seja capaz de tomar decisões precisas a cerca de um determinado problema (MONARD; BARANAUSKAS, 2003). Como exemplo pode-se citar a estimação do preço de um imóvel ou a classificação de espécies de plantas. Este capítulo introduz os alguns conceitos de aprendizado de máquina utilizados neste trabalho.

Pode-se dividir o aprendizado de máquina em três subcategorias: Aprendizado supervisionado, não supervisionado e por reforço. A escolha do tipo de aprendizado a ser utilizado depende, principalmente, se a base de dados é rotulada ou não. Este trabalho utilizará um conjunto de dados com rótulo, portanto, será empregado o aprendizado supervisionado descrito brevemente a seguir.

O aprendizado supervisionado é um tipo de aprendizado em que o conjunto de exemplos $E = [E_1, E_2, ..., E_n]$ onde cada exemplo $E_i \subset E$ é composto por um conjunto de atributos x e um rótulo y. O objetivo nesse caso é estimar valores $\hat{y}$ para novos valores de x baseando-se nos exemplos rotulados de treinamento E. Dentro do aprendizado supervisionado pode-se subdividir em classificação e regressão, dependendo do tipo do rótulo ser contínuo ou discreto. No caso desse trabalho, por tratar-se de identificação de classes (valores discretos ex: A, $B7$, $C\sharp$), é utilizada a classificação.

4.1 CLASSIFICAÇÃO

As funções dos algorítimos de aprendizado de máquina podem ser categorizadas como classificação e regressão. Os algoritmos de classificação tem como objetivo identificar a qual classe pertencem os conjuntos de dados de teste a partir de um treinamento em classes predeterminadas. Nesse tipo de abordagem, o resultado final será sempre algum valor que represente alguma classe. Existem diversos algoritmos para gerar modelos de classificação. A seguir, são apresentados alguns tipos de classificadores.

4.1.1 Modelos Classificadores

Os algoritmos considerados mais importantes e utilizados em problemas de aprendizado supervisionado são: k-Nearest Neighbors (k-NN), Regressão Linear, Regressão logística, Máquinas de Vetores de Suporte (SVM), Árvores de decisão, Random Forests e Redes Neurais Géron (2017).

4.1.2 K-Nearest Neighbors (k-NN)

O método k-Nearest Neighbor (k-ésimo vizinho mais próximo) tem como objetivo classificar um novo elemento, atribuindo a ele o rótulo representado mais frequentemente dentre as k amostras mais próximas e utilizando um esquema de votação. O que determina essa "proximidade entre vizinhos" é uma distância calculada entre pontos em um espaço euclidiano (LEMOS et al., 2019). As fórmulas mais comuns são a distância Euclidiana e a distância Manhattan. A Distância Euclidiana (Eq. 4.1) é definida como a soma da raiz quadrada da diferença entre x e y em suas respectivas dimensões. Já a Distância Manhattan (Eq.4.1) é a soma das diferenças entre x e y em cada dimensão.

$$d = \sqrt{x^2 + y^2} \tag{4.1}$$

$$d = |x_1 - x_2| + |y_1 - y_2| \tag{4.2}$$

4.1.3 Regressão logística

A Regressão Logística é utilizada quando o valor que se deseja predizer é binário. É um modelo probabilístico que indica a probabilidade de um exemplo pertencer a uma classe ou não a partir de uma função sigmoide (Eq. 4.3).

$$p = \frac{e^x}{1 + e^x} \tag{4.3}$$

A função sigmoide retorna valores entre 0 e 1. Com um determinado valor de entrada x, o valor de p, $0 < p < 1$ é aproximado a 1 ou 0 a partir de um limiar a ser definido (Eq. 4.4).

$$f(p) = \begin{cases} 0, se(p) < 0,5 \\ 1, se(p) >= 0,5 \end{cases} \tag{4.4}$$

Os dados de treinamento são utilizados para encontrar valores contínuos que, em relação a x, posicione a linha resultante da função logística de um modo que melhor indique a probabilidade de um valor ser um ou zero. Por exemplo, na Figura 16 é mostrado um gráfico resultante de uma função cujo expoente é a expressão $(ax + b)$ e os valores a e b encontrados pelo treinamento são 2.1 e -1, respectivamente. Perceba, na figura, que o limiar de x é definido pela derivada de segunda ordem da função em que $f''(x) = 0$. No caso do exemplo em questão, o limiar de x é igual a 0.476.

4.1.4 Support Vector Machines (SVM)

Máquina de vetores de suporte é um método utilizado para regressão e classificação. Quando utilizado em classificação, seu objetivo é encontrar um hiperplano que melhor separe

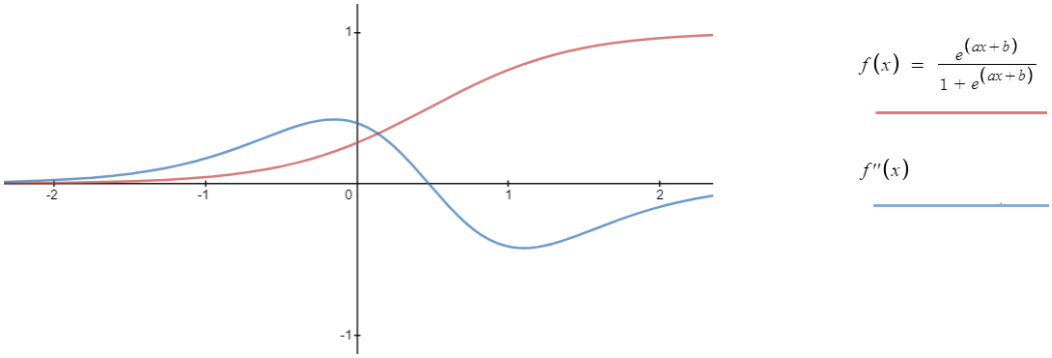


Figura 16 – Gráfico de uma função logística e sua derivada de segunda ordem.

duas classes (Figura 17). A separação pelo hiperplano resulta em duas áreas distintas, cada uma representando uma classe. A classificação é dada ao identificar em qual área o objeto se encontra.

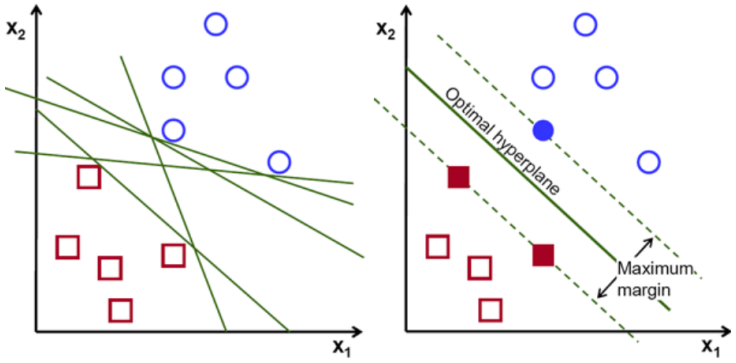


Figura 17 – Algoritmo SVM encontrando hiperplano que separe as duas classes. Fonte: towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c

4.1.5 Random Forest

Traduzido do inglês, florestas aleatórias ou florestas de decisão aleatória, é um modelo que consiste na combinação de árvores de decisão. Cada árvore tenta estimar uma classe e o resultado de cada uma é chamado de voto. A classe mais votada entre as árvores de decisão é definida como a decisão da floresta. Para a construção das árvores de decisão, procura-se colocar os atributos mais descritivos próximos a raiz e os menos descritivos nas extremidades. Uma boa abordagem para encontrar os atributos mais descritivos de um conjunto de dados rotulado é utilizando o *Indice Gini*.

Na Figura 18 é exemplificada uma floresta com quatro árvores. Neste exemplo, todas as árvores receberam um mesmo conjunto de dados chamado X, as árvores #1 e #4 decidiram pela classe C, enquanto as árvores #2 e #3 decidiram pelas classes D e B, respectivamente. No sistema democrático das Florestas Aleatórias, a classe escolhida definida pela maioria dos votos é a C.

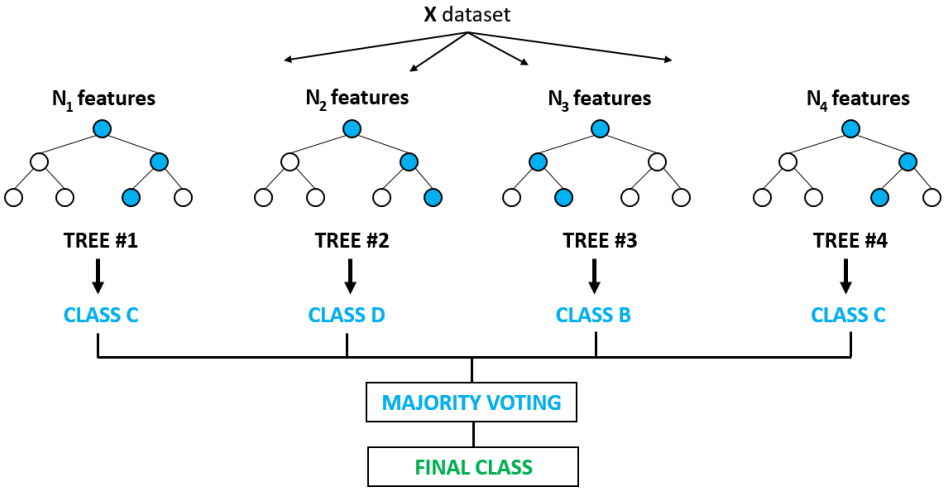


Figura 18 – Exemplo de uma floresta com quatro árvores de decisão. Fonte: medium.com

4.1.6 Redes Neurais Artificiais

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

Uma rede neural artificial é composta por várias unidades de processamento, chamados de Neurônios (Figura 19). Essas unidades geralmente são conectadas por canais de comunicação que estão associados a determinado peso sináptico W. As unidades fazem operações apenas sobre seus dados locais que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede. Na Figura 20 é apresentada uma estrutura simples de rede neural.

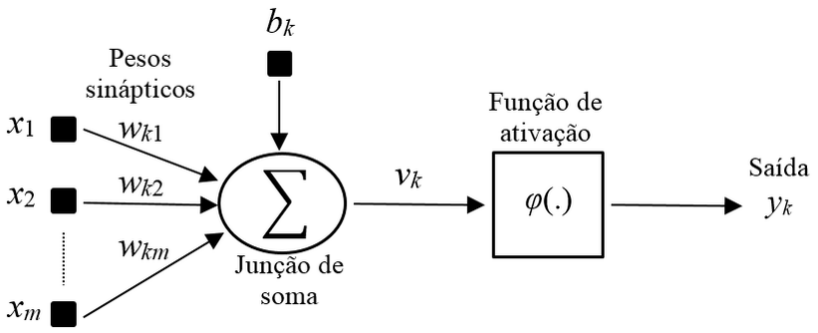


Figura 19 – Estrutura de um neurônio artificial. Fonte: Simon Haykin[2001]

Na Figura 20 é possível observar as camadas da rede. Estas camadas são compostas por um número de neurônios interconectados que contêm uma "função de ativação". Os padrões são apresentados à rede através da "camada de entrada", que se comunica com uma ou mais

“camadas ocultas”, onde o processamento atual é feito através de um sistema de “conexões” ponderadas. As camadas ocultas vinculam-se a uma “camada de saída” na qual a resposta é exibida. O treinamento da rede ocorre no ajuste dos pesos através da etapa chamada de *backpropagation*, onde o rótulo dos dados de entrada são comparados com os resultados obtidos da rede para calcular um erro.

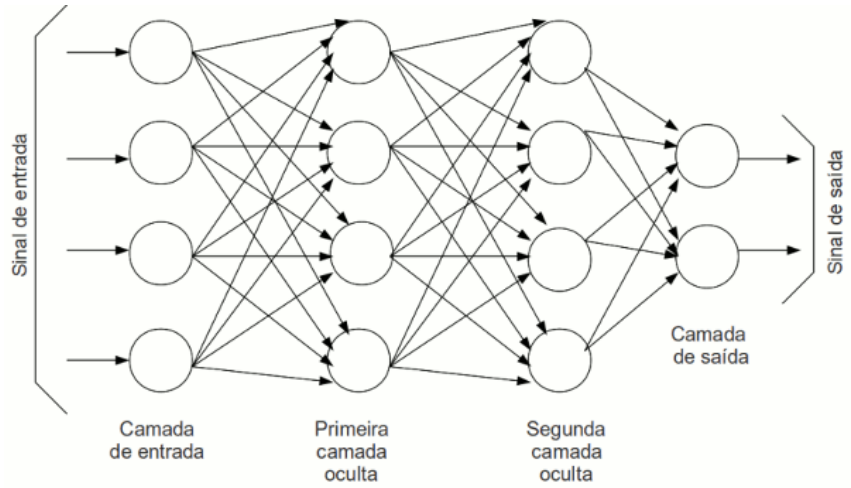


Figura 20 – Exemplo de estrutura de uma Rede Neural. Fonte: monolitonimbus.com.br/redes-neurais-artificiais

Neste capítulo sobre *machine learning* foram mostrados os tipos de aprendizado, aprofundando em aprendizado supervisionado. Foram apresentados brevemente as principais técnicas de aprendizado de máquina voltadas à classificação. Todas são aptas para o reconhecimento de acordes pois identificam a qual grupo ou classe cada novo exemplo pertence, baseado em exemplos anteriores. No caso dos acordes, cada acorde é uma classe. Para este trabalho foram criados modelos de florestas aleatórias. Os demais modelos não foram aplicados.

A seguir serão mostrados os trabalhos relacionados, esclarecendo como os modelos aqui citados foram aplicados pelos autores.

5 TRABALHOS RELACIONADOS

Como existem muitos trabalhos relacionados a esta área (CHOI et al., 2017), é preciso limitar o campo de busca e aprofundar-se em somente uma de tantas tarefas que compõem os aprendizados supervisionados em si. Portanto, limitou-se o aprofundamento em bibliografias que priorizassem a fase de pré-processamento como (KORZENIOWSKI; WIDMER, 2016) e (MCFEE; BELLO, 2017). Também foi aprofundado um trabalho mais antigo sobre o assunto (CLETO et al., 2010), com a finalidade de entender como esse campo de pesquisa tem evoluído na última década.

No trabalho de (CLETO et al., 2010) é explorado o problema da detecção polifônica por meio de uma técnica utilizada em redes neurais artificiais chamada *Multilayer Perceptron* e foi validada utilizando uma base de dados criada com vários acordes produzidos com diferentes timbres por um sintetizador.

O objetivo desse trabalho é fornecer a cifra correspondente ao acorde identificado, recebendo como entrada um arquivo de som contendo um ou mais acordes tocados individualmente, isto é, sem sobreposição de acordes com outros acordes, melodia, percussão ou voz.

São identificadas pelo trabalho apenas acordes maiores. A rede neural utilizada possui 61 nós de entrada, cada um representando uma frequência que é obtida por uma Transformada de Fourier, 61 nós na camada oculta e um nó para a saída com um valor discreto que indica o acorde sendo tocado, a estrutura pode ser visto na Figura 21.

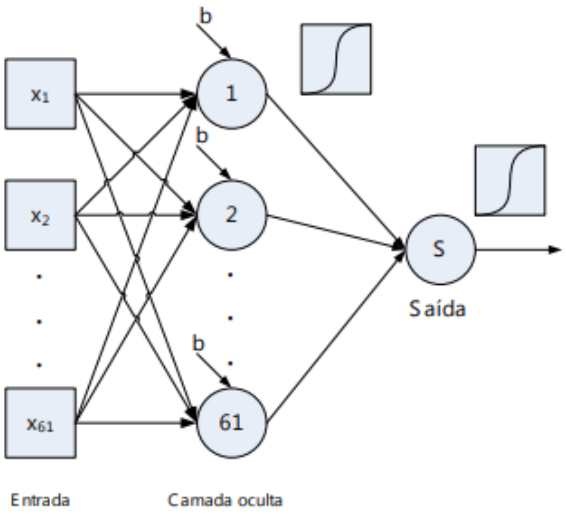


Figura 21 – Estrutura da rede utilizada. Fonte: (CLETO et al., 2010).

A metodologia alcançou um taxa média de reconhecimento de acordes de 84%, como pode ser visto na Figura 22.

No artigo (KORZENIOWSKI; WIDMER, 2016) é enfatizada a importância do pré-processamento dos dados antes de submeter a um modelo de treinamento, alterando o foco da literatura que recentemente era a aplicação de métodos artesanais no pós processamento (como

	Número de acordes	Número de acordes não reconhecidos	Taxa de acerto
Teste 1	12	0	100%
Teste 2	11	3	73%
Teste 3	12	4	67%
Teste 4	48	5	90%
Teste 5	7	2	71%
Total	90	14	84%

Figura 22 – Resultados obtidos pelo trabalho. Fonte: (CLETO et al., 2010).

por exemplo a utilização de modelos estatísticos que indicam a probabilidade de um acorde anteceder o outro) e defendendo que um bom pré-processamento pode simplificar a etapa do treinamento, sendo possível utilizar métodos classificadores menos robustos para a tarefa de criação do modelo e pós-processamento.

Os autores propõem um modelo de pré-processamento utilizando uma Rede Neural Convolucional profunda - CNN, essa rede neural tem como entrada um espectrograma oriundo de uma transformada CQT de uma amostra do sinal sonoro. Esta entrada é chamada por eles de C.

Os áudios utilizados são fragmentados em amostras e intervalos, cada amostra contém 8192 sinais de áudio e para cada amostra existe um intervalo chamado de "salto"com um tamanho de 4410 sinais de áudio. A taxa de amostragem utilizada é de 44100 Hz.

Na tentativa de aprimorar os resultados, um segundo espectrograma chamado pelos autores de "espectrograma de quarto de tom"S_{Log} é criado com frequências de 30 Hz a 5500 Hz, com 24 compartimentos por oitava. Isso resulta em uma dimensionalidade de 178 posições. A rede neural profunda tem como saída um cromagrama reduzido em apenas 12 posições.

Os *datasets* utilizados são *Beatles*, *Queen and Zweieck*, the *RWC pop dataset 4* e o *Robbie Williams dataset*, totalizando 383 faixas ou aproximadamente 21 horas e 39 minutos de músicas. São reconhecidos apenas acordes maiores e menores, totalizando 24 classes. Para comprovar a eficácia dessa abordagem, o treinamento de reconhecimento de acorde é feito utilizando uma regressão logística e não é realizado nenhum pós processamento. Na Figura 23, é mostrado os resultados obtidos pelo mesmo modelo de classificação a partir de diferentes dados de entrada, onde, C é o próprio espectrograma calculado a partir de uma CQT, C^W_{Log} é o cromagrama em distribuição logarítmica obtido através de C e S_{Log} é o espectrograma de quarto de tom, com as frequências limitadas conforme descrito acima e C_D é o cromagrama resultante do pré processamento utilizando a Rede Neural.

Aplicando métodos semelhantes ao (KORZENIOWSKI; WIDMER, 2016), no artigo (MCFEE; BELLO, 2017), é discutido a importância e as dificuldades encontradas em se classificar uma grande gama de acordes, mostrando que na literatura até então os métodos encontrados não são eficientes para classificar individualmente classes com muitas similaridades como o C e o $C7$, por exemplo. Aborda-se o problema do reconhecimento de acordes com vocabulário amplo, introduzindo uma representação estruturada das qualidades dos acordes.

A representação estruturada proposta é classificar em modelos separados a raiz do acorde

	Btls	Iso	RWC	RW	Total
C	71.0±0.1	69.5 ±0.1	67.4±0.2	71.1±0.1	69.2±0.1
C^W_{Log}	76.0±0.1	74.2 ±0.1	70.3±0.3	74.4±0.2	73.0±0.1
S_{Log}	78.0±0.2	76.5 ±0.2	74.4±0.4	77.8±0.4	76.1±0.2
C_D	**80.2**±0.1	**79.3**±0.1	**77.3**±0.1	**80.1**±0.1	**78.8**±0.1

Figura 23 – Resultados obtidos pelo trabalho. De modo geral, observa-se uma melhora de 9,6% do pré-processamento menos robusto (C) ao mais robusto (C_D) Fonte: (KORZENIOWSKI; WIDMER, 2016).

de sua variação. Enquanto um treinamento é feito para indicar qual a nota que está sendo tocada (exemplo: C, $D\#$, B) classificando todas as 12 classes possíveis, o outro indica qual a variação (exemplo: maior, menor, com sétima) classificando 12 variações distintas. A combinação da saída dos dois modelos possibilita a classificação de 168 acordes diferentes. São utilizados 1217 exemplos de um *datasets* próprio composto pela combinação dos *datasets Isophonics, Billboard, RWC Pop, e MARL collections.* Cada faixa foi representada como a distribuição logarítmica da frequência utilizando um spectrograma da função *CQT* com 36 slots por oitava, abrangendo um total de 6 oitavas a partir da nota C da segunda oitava. Os sinais foram analisados em 44,1KHz com um comprimento de salto de 4096 amostras, resultando em uma taxa de quadros de aproximadamente 10,8Hz. Os resultados obtidos pelo experimento é mostrado na Figura 24, na parte superior, estão as estruturas utilizadas e na parte de baixo os modelos de comparação.

Method	Root	Thirds	Triads	Sevenths	Tetrads	Maj-Min	MIREX
CR2+S+A	0.861	0.836	0.812	0.729	0.671	0.855	0.852
CR2+A	0.850	0.828	0.801	0.719	0.659	0.845	0.837
CR1+S+A	0.850	0.824	0.801	0.716	0.648	0.842	0.832
CR1+A	0.841	0.815	0.791	0.702	0.647	0.834	0.829
KHMM [5]	0.849	0.822	0.785	0.674	0.629	0.817	0.827
DNN [11]	0.838	0.809	0.766	0.654	0.605	0.803	0.812

Figura 24 – Resultados obtidos pelo trabalho. Fonte: (MCFEE; BELLO, 2017).

Os modelos propostos alcançam uma precisão substancialmente mais alta do que os modelos anteriores, baseados em redes convolucionais e modelos ocultos de Markov, resultando em ganhos absolutos de 4% a 5% nas classes mais difíceis como as sétimas.

Seguindo os modelos de pré-processamento apresentados por (KORZENIOWSKI; WIDMER, 2016) e (MCFEE; BELLO, 2017), o objetivo nesse trabalho é criar um dataset com 48 acordes e extrair as informações utilizando dois modelos diferentes de transformada, Short-Time Fourier Transform e Constant-Q Transform. Com o dataset próprio foram criados e avaliados os modelos de classificação conforme a sua precisão.

6 PROJETO E EXPERIMENTO

Para realizar o treinamento do um modelo é necessário ter um *dataset* com exemplos de acordes. Em um violão é possível tocar muitos acordes diferentes. Uma forma de se estimar a quantidade de acordes possíveis é multiplicando a quantidade de notas bases pelas suas variações. Tudo depende de quantas variações são consideradas no cálculo. Considerando as 12 notas base e as variações menor (m), com sétima (7), com nona (9), com sexta (6), diminuto (º) e aumentado (+) já são 768 acordes ($12 * 2^6$). Para este trabalho, a quantidade de acordes foi limitado a 12 notas e duas variações, menor (m) e com sétima (7), totalizando 48 acordes. Na Web não foram encontradas gravações que atendessem ao trabalho proposto. A característica do *dataset* desejado é possuir gravações de acordes isolados, sem a presença de outros instrumentos. Também deve haver exemplos padronizados para todos os acordes das variações desejadas. Os detalhes da criação do *dataset* são mostrados na próxima seção.

6.1 CRIAÇÃO DO *DATASET*

Para criar um *dataset* foi necessário realizar desde a gravação dos acordes até a criação de um arquivo CSV com os cromagramas destas gravações devidamente rotuladas. A Figura 25 ilustra todas as etapas necessárias para a criação do *dataset*. Na figura, a criação do *dataset* é apresentada em duas partes. A primeira parte consiste em gravar os acordes, realizar os tratamentos necessários nos áudios e rotulá-los. A segunda parte consiste em ler os arquivos de áudio rotulados, transformá-los em cromagramas e gerar um único arquivo CSV com todas as gravações. Todas as etapas presentes na figura serão detalhadas nos parágrafos a seguir.

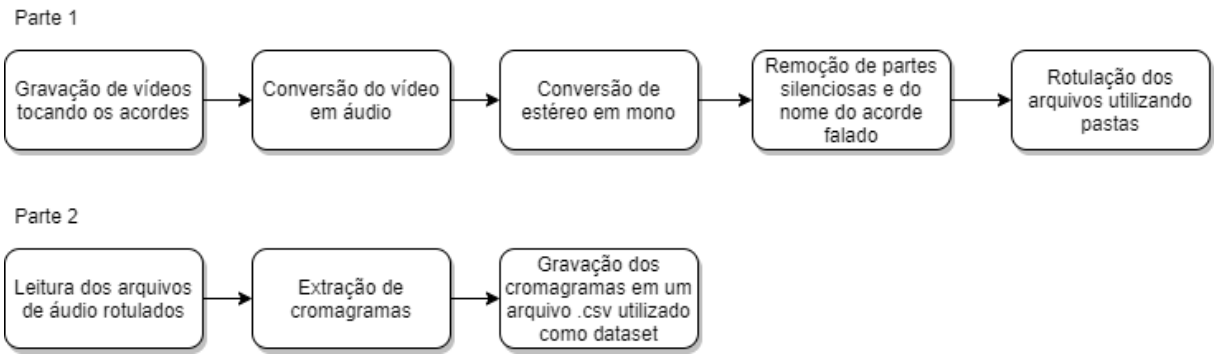


Figura 25 – Diagrama com todas as etapas de criação do *dataset*.

Para gravar os acordes, foi utilizado um violão de seis cordas afinado na frequência base $440Hz$. Os acordes foram gravados tocando as cordas de cima para baixo $\downarrow$, sem empregar um ritmo, em cada vez, esperava-se o som diminuir até o silêncio e então toca-se novamente o acorde. Esta sequência de gravação foi repetida em média 10 vezes para cada acorde. Foram gravados ao todo 48 acordes, dos quais somente 41 puderam ser aproveitados. O motivo que levou a algumas gravações serem descartadas foi o ruído do ambientes presente no momento da

gravação, tal ruído confundiria o treinamento. Cada acorde foi gravado explorando ao menos duas formações diferentes no braço do violão.

O dispositivo utilizado foi um celular modelo Moto X (2ª geração), utilizando os próprios microfones do celular através da gravação de vídeo. Para cada vídeo, falava-se o nome do acorde e posteriormente tocava-se o acorde citado no violão, separando um arquivo para cada acorde. No modelo do celular utilizado, a qualidade de gravação do áudio em vídeo é melhor comparado a outros meios de gravação.

Os vídeos gravados foram convertidos em áudio no formato Waveform Audio File Format - (wav) pelo do software aTube Catcher [1]. A taxa de amostragem utilizada foi de 44.100 Hz. Os vídeos foram gravados em dois canais (estéreo), foi preciso então separar os áudios dos canais para tornar possível a utilização em uma transformada. A separação dos canais de áudio foi feita no software Audacity [2]. No Audacity é possível separar os dois canais em duas faixas. Estas faixas foram concatenadas para gerar um áudio mono com o dobro de duração.

Por exemplo, suponha que um áudio foi gravado utilizando-se a tecnologia estéreo. Durante a execução deste áudio, o som será distribuído entre os auto falantes, cada um ficará responsável por reproduzir uma canal de áudio, tornando a experiência imersiva. Neste trabalho, o áudio estéreo não pôde ser utilizado, pois a soma das ondas dos dois canais impossibilita o uso de uma Transformada. Para possibilitar o seu uso, o áudio foi transformado em mono. Assim, todos os canais de reprodução recebem o mesmo sinal, executando todo áudio gravado pelo primeiro canal e posteriormente o áudio do outro canal. Este aumento de duração proporcionado pela transformação de estéreo em mono, fez com que o *dataset* dobrasse de tamanho, contribuindo para a criação de mais exemplos. A exclusão de partes silenciosas foi feita manualmente no Audacity. Foram excluídas as partes onde eram falados os nomes dos acordes. Isto foi feito para que os áudios fossem nomeados adequadamente. Para cada acorde gerado, foi criado um arquivo correspondente ao acorde. Empregou-se esta estratégia para facilitar a rotulação dos acordes no próximo passo.

Para ler os arquivos já rotulados e transformá-los em um *dataset* foi desenvolvido um programa chamado Extrator librosa. O Extrator librosa percorre todos os diretórios buscando por arquivos .wav. Cada arquivo .wav encontrado é importado e submetido as funções `chroma_sftf` e `chroma_cqt` do `librosa`, essas funções, tem como saída um cromagrama. Lembrando que um cromagrama é a representação do áudio em 12 valores que vão de 0 até 1, cada um dos 12 valores representa um semitom.

Nas funções `chroma_sftf` e `chroma_cqt` define-se um tamanho de janela, que é a quantidade de valores que serão extraídos do áudio para gerar o cromagrama. Por padrão, o tamanho da janela é de 512. A librosa trabalha com a taxa de amostragem em 22.050Hz. Assim utilizando o tamanho de janela padrão, e a taxa de amostragem fornecida pelo librosa, é possível extrair de cada segundo de áudio até 43 cromagramas (*i.e.*, $\frac{22050}{512}$). A Figura 26

[1] https://www.atube.me

[2] https://www.audacityteam.org

mostra um arquivo de áudio do acorde *Amaj* representado em um conjunto de cromagramas pela função `chroma_sftf`. A legenda da direita é uma gradiente de cores onde as cores mais claras representam valores próximos a 1 e mais escuras próximas a 0. Na legenda à esquerda, são nomeadas cadas uma das notas. A biblioteca librosa nomeia somente as notas naturais A, B, C, D, E, F e G. Os espaços vazios entre as notas anotadas são os valores de intensidade dos acidentes, de baixo para cima, $C\sharp, D\sharp, F\sharp, G\sharp$ e $A\sharp$. Ainda na figura, pode-se observar, na maioria das vezes, uma intensidade maior nas notas A, E e C$\sharp$, exatamente as notas que compõem o acorde Lá Maior ($Amaj$).

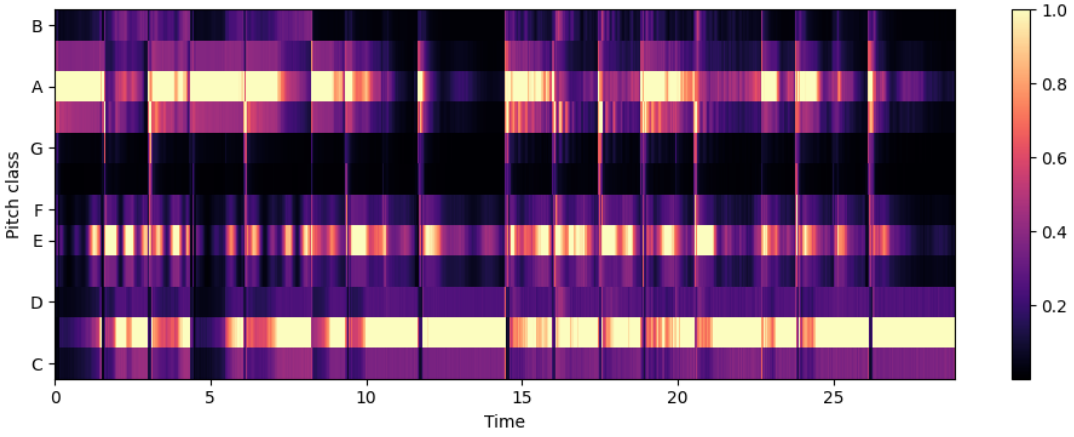


Figura 26 – Arquivo de áudio representado em um cromagrama.

Todos os cromagramas extraídos de todas as classes foram salvos em dois arquivos CSV, cada um contendo o conjunto de dados extraído utilizando as funções `chroma_sftf` e `chroma_cqt`. Cada arquivo CSV contém 13 colunas, as 12 primeiras colunas representam a intensidade de cada semitom e a última coluna representa o nome da classe. Na Figura 27 é ilustrada uma parte do arquivo gerado utilizando a função `stft`. Nela é possível ver as 12 notas e o valor representando a intensidade de cada uma. Na última coluna está a classe. Ao final da extração de *features* de todas as gravações foram gerados 72.564 exemplos em cada arquivo.

A qualidade dos dados obtidos pode ser prejudicada pelos seguintes fatores:

- Afinação: O violão precisa estar afinado em 440Hz. Todas as suas cordas devem estar tensionadas na medida certa para que quando tocadas emitam as frequências conforme visto na Figura 6 do Capítulo 2.
- Cordas em más condições: Cordas em más condições de uso reduzem significativamente o volume e a duração do som emitido, podendo assim uma corda se sobrepor ao som de outra e ter mais destaque.
- Distanciamento entre as trastes incorreto: Se a distância entre as trastes que separam as casas do violão não obedecerem exatamente a razão $\sqrt[12]{2}$, as frequências emitidas terão uma leve alteração, fazendo com que as notas vizinhas à nota correta apareçam na gravação.

- Desgaste nas trastes: Este defeito faz as cordas tocarem em trastes menos desgastadas, emitindo frequências totalmente erradas dependendo de como a nota é executada. Um outro problema decorrente do desgaste das trastes é o ruído característico causado por este problema.

- Má execução do acorde: A habilidade do instrumentista e até mesmo alguns vícios podem interferir no som emitido pelo violão, fazendo com que o mesmo acorde soe diferente ao ser tocado por duas pessoas distintas. Todas as cordas precisam ser tocadas na mesma intensidade, os dedos precisam estar posicionados corretamente para não esticar a corda para cima ou para baixo, o que aumenta a frequência da nota pela técnica conhecida como *bend*. As cordas devem ser pressionadas com a mesma força contra as trastes e somente as cordas pertencentes ao acorde devem ser tocadas. Qualquer erro cometido pela pessoa interfere diretamente na qualidade da gravação obtida.

- Ruídos externos: Como as gravações foram feitas utilizando microfones, está suscetível a interferências do som ambiente. O som ambiente é captado e considerado erroneamente como uma nota pertencente ao acorde.

Nos experimentos realizados, apenas o desgaste nas trastes, ruídos externos e possível má execução de alguns acordes foram considerados.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Nota
0,0236	0,0189	0,0287	0,0517	0,0625	0,0450	0,0574	0,1162	0,4748	1,0000	0,4621	0,0725	[AMAJ]
0,0255	0,0226	0,0508	0,1514	0,2144	0,1861	0,1255	0,1422	0,6004	1,0000	0,4134	0,0972	[AMAJ]
0,0436	0,0181	0,0432	0,2663	0,5111	0,3581	0,1012	0,1419	0,5568	1,0000	0,4881	0,1934	[AMAJ]
0,0393	0,0246	0,0316	0,1583	0,2989	0,1961	0,1076	0,2048	0,5555	1,0000	0,4587	0,1337	[AMAJ]
0,0232	0,0251	0,0235	0,0663	0,0911	0,0361	0,0512	0,1046	0,5742	1,0000	0,3667	0,0574	[AMAJ]
0,0534	0,0971	0,0455	0,0573	0,1060	0,0749	0,0502	0,0496	0,4829	1,0000	0,3808	0,0446	[AMAJ]
0,0759	0,1663	0,0520	0,1168	0,2381	0,0790	0,0425	0,0556	0,4772	1,0000	0,3711	0,0488	[AMAJ]
0,0641	0,1609	0,0502	0,1146	0,3267	0,1136	0,0346	0,0461	0,4810	1,0000	0,3707	0,0559	[AMAJ]
0,0634	0,1553	0,0478	0,0981	0,2765	0,0807	0,0156	0,0442	0,4947	1,0000	0,3757	0,0562	[AMAJ]
0,0697	0,1664	0,0518	0,0909	0,2347	0,0703	0,0125	0,0411	0,5031	1,0000	0,3797	0,0555	[AMAJ]
0,0672	0,1693	0,0496	0,0741	0,1939	0,0600	0,0089	0,0370	0,4977	1,0000	0,3801	0,0536	[AMAJ]
0,0641	0,1530	0,0420	0,0570	0,1542	0,0509	0,0085	0,0372	0,5002	1,0000	0,3749	0,0508	[AMAJ]
0,0591	0,1472	0,0418	0,0480	0,1222	0,0411	0,0079	0,0357	0,4910	1,0000	0,3754	0,0538	[AMAJ]
0,0608	0,1492	0,0394	0,0375	0,0977	0,0362	0,0078	0,0326	0,4788	1,0000	0,3789	0,0586	[AMAJ]
0,0658	0,1505	0,0381	0,0320	0,0817	0,0305	0,0067	0,0309	0,4778	1,0000	0,3720	0,0608	[AMAJ]
0,0599	0,1411	0,0371	0,0245	0,0601	0,0231	0,0065	0,0294	0,4666	1,0000	0,3800	0,0639	[AMAJ]
0,0594	0,1454	0,0398	0,0186	0,0452	0,0184	0,0055	0,0286	0,4634	1,0000	0,3792	0,0651	[AMAJ]
0,0702	0,1583	0,0409	0,0154	0,0383	0,0169	0,0053	0,0274	0,4628	1,0000	0,3784	0,0672	[AMAJ]
0,0714	0,1599	0,0409	0,0134	0,0342	0,0136	0,0048	0,0321	0,4818	1,0000	0,3715	0,0664	[AMAJ]
0,0649	0,1581	0,0416	0,0139	0,0371	0,0136	0,0047	0,0306	0,4761	1,0000	0,3739	0,0629	[AMAJ]

Figura 27 – Parte do arquivo .csv gerado pelo Extrator librosa.

6.2 EXPERIMENTO COM SHORT-TIME FOURIER TRANSFORM - STFT

Inicialmente, foram feitos experimentos para escolher os melhores hiperparâmetros. Para tal, utilizou-se o algoritimo `GridSearchCV` disponível na biblioteca `sklearn.model_selection`. O `GridSearchCV` é um módulo do *Scikit Learn* e é amplamente usado para automatizar grande

parte do processo de *tuning*. O objetivo primário do `GridSearchCV` é a criação de combinações de parâmetros para posteriormente avaliá-las. Foram submetidos ao `GridSearchCV` os hiperparâmetros pertinentes ao treinamento de uma floresta aleatória, cada um com no mínimo dois candidatos. Os melhores hiperparâmetros encontrados utilizando esta técnica podem ser vistos na Tabela 1.

A tabela é dividida em três colunas, o nome do hiperparâmetro, os candidatos e o hiperparâmetro escolhido entre os candidatos através do `GridSearchCV`. Por exemplo, para o hiperparâmetro `max_features`, foram dados os candidatos "auto", "sqrt"e "log", e foi escolhido o "log2".

Tabela 1 – Hiperpârametros encontrados pelo GridSearchCV

Hiperparâmetro	Candidatos	Escolhido
`bootstrap`	`True,False`	`False`
`criterion`	`gini,entropy`	`gini`
`n_estimators`	`50,100,150,200,250`	`250`
`max_features`	`auto, sqrt, log2`	`log2`
`class_weight`	`None,balanced,balanced_subsample`	`balanced`

Com os hiperparâmetros definidos, foi treinado um modelo utilizando a ferramenta `RandomForestClassifier` da biblioteca `sklearn.ensemble`. Como entrada é dado o *dataset* já devidamente separado em dados para treinamento e dados para teste. Os dados foram separados em 70% para treinamento e 30% para teste. O desempenho do modelo treinado utilizando o *dataset* criado pela função `chroma_stft` pode ser visto na Tabela 2. Esta tabela lista todos as classes treinadas pelo modelo. A coluna `support` indica a quantidade de exemplos utilizados por cada classe. O resultado obtido é apresentado nas colunas `precision`, `recall` e `f1-score`.

O modelo teve uma maior precisão nas classes *Bmaj*, *Cm*, *Dm*, *F♯m* e *Fmaj*, todos estes atingindo precisão de 100%. Já o pior resultado foi observado na classe *G7*, que obteve 96% de precisão. A média da coluna *precision* considerando o treinamento de todas as classes foi de 98,56%. Na Figura 28 é mostrado o cromagrama do arquivo utilizado para treinar a classe *G7*. O acorde *G7* é composto pelas notas *G*, *B*, *D*, *F*, na figura, as notas com maior intensidade são *B* e *D*. A nota *G* possui intensidade acentuada somente em alguns pontos, sendo inclusive até menos percebida que as notas *C♯* e *D♯* no decorrer da gravação. Notas estas que originalmente não fazem parte do acorde mas acabaram sendo percebidas ao realizar a `STFT`.

Na Figura 28 no segundo 15 até o segundo 23, aproximadamente, a nota *F* tem mais intensidade. Isto se deve à formação do acorde no braço do violão ser diferente nos trechos citados. Isto mostra que o mesmo acorde em formações diferentes pode destacar mais algumas notas do que em outras formações. Esta situação pode interferir no modelo, pois uma pequena variação como essa pode ser interpretado como outro acorde. Na Figura 29 é mostrada, à direita, a formação utilizada entre os segundos 15 até o 23 e do segundo 37 em diante, à esquerda, a

formação utilizada no restante da gravação. Apesar das posições dos dedos serem diferentes, elas representam o mesmo acorde, no caso *G7*.

A formação a direita do acorde *G7* na Figura 29 está localizada em casas mais altas do braço do violão. Para o algorítimo da `stft`, é melhor de distinguir, já que sua desvantagem está principalmente na baixa resolução de frequências menores. Por isso, o acorde da Figura 28 foi melhor representado no cromagrama da segunda formação de acorde. Além dos outras interferências que possam ter ocorrido na gravação do `dataset` conforme a seção 6.1.

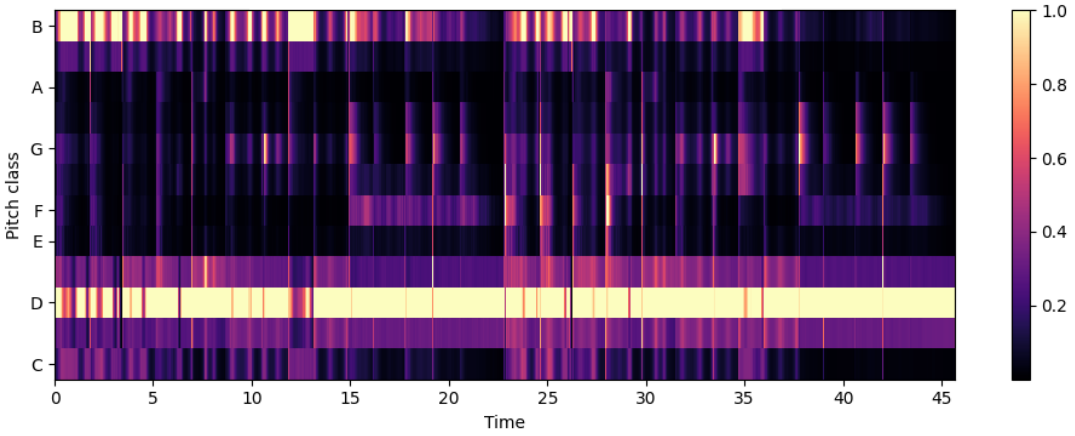


Figura 28 – Cromagrama do arquivo utilizado para o treinamento da classe *G7*.

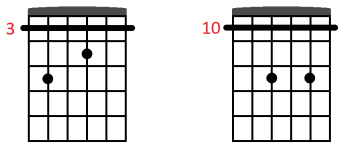


Figura 29 – Formações do acorde *G7* utilizados na gravação do *dataset*

Com o objetivo de verificar se a transformada de CQT pode melhorar o desempenho do modelo, um novo experimento foi conduzido e é apresentado na próxima seção.

6.3 EXPERIMENTO COM *CONSTANT-Q TRANSFORM - CQT*

Após o experimento com *STFT* mostrar que a transformada escolhida pode interferir na precisão do modelo, foi realizado um novo experimento utilizando um *dataset* criado extraindo as *features* pela Transformada de Q Constante, resultante da função `chroma_cqt`.

O resultado do modelo criado se mostrou melhor que no primeiro experimento, conforme pode ser visto na Tabela 3. Nesta tabela observa-se que, apesar de alguns acordes terem piorado a sua precisão em relação ao STFT, como os acordes Em e Fm, no geral os acordes tiveram precisão maior do que no experimento anterior. Configurando, assim, que o `CQT` tem melhor desempenho.

O acorde de *G7* teve precisão de 96% utilizando o *dataset* elaborado com a técnica STFT, e 99% utilizando o *dataset* elaborado pela técnica CQT. A Figura 30 compara, a esquerda, o cromagrama extraído utilizando STFT, com o da direita, utilizando CQT. No cromagrama à direta, existe um contraste maior entre as notas que realmente compõe o acorde em relação as notas não pertencentes ao acorde, o que garante dados muito mais representativos.

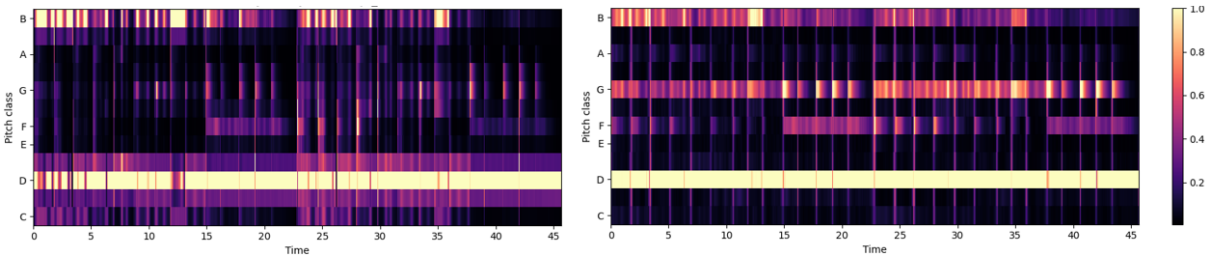


Figura 30 – Comparação de cromagramas gerados a partir do STFT e CQT do acorde G7.

6.4 EXPERIMENTO COM NOVA SEQUÊNCIA DE ACORDES

Com o objetivo de testar a robustez do modelo, é proposto um teste de reconhecimento utilizando um arquivo de áudio gravado nos mesmos moldes dos arquivos utilizados no *dataset* anterior, mas que não foi utilizado para treinar e nem validar o modelo. Este teste visa analisar o comportamento do modelo quando no arquivo de áudio existirem trechos silenciosos provenientes do inicio e final do arquivo. Também é pretendido observar a interferência dos ruídos causados pela troca de acorde, pois por mais rápida que seja esta troca, sempre haverá um pequeno intervalo entre um acorde e outro. O passo a passo deste experimento será descrito a seguir.

Primeiro, foram gravados três acordes em sequência. Os acordes escolhidos foram Am7, Cmaj e Gmaj. Esta mesma sequência se repetiu duas vezes, então cada acorde foi tocado, ritmicamente, duas vezes. O ritmo empregado na gravação destes acordes foi ↓↓↑↑↓↓↑↓↑ para cada acorde. A seta para baixo indica o acorde sendo tocado de cima para baixo, e vice-versa.

Com a gravação feita, foi submetida a transformação de estéreo em mono. Essa transformação, assim como na elaboração dos *datasets*, duplicou o tamanho do arquivo de gravação. O arquivo de testes ficou com uma duração total de 59 segundos. O arquivo de áudio então foi submetido aos algoritmos de extração de *features*. Foram obtidos dois arquivos CSV, um para testar o modelo baseado em STFT e outro para testar o modelo baseado em CQT. Do arquivo de áudio foram extraídos 2.576 exemplos.

Para realizar o experimento, o arquivo precisa ser fragmentado em conjuntos menores. A fragmentação é necessária pois não se pode submeter um arquivo com vários acordes ao modelo de uma única vez. Foi definida uma quantidade de 100 exemplos para cada conjunto. Ao todo, foram extraídos 25 conjuntos com 100 exemplos e um conjunto com 76 exemplos.

Totalizando 2.576 exemplos distribuídos em 26 conjuntos. A divisão dos conjuntos não coincide com as trocas de acorde da gravação, então um conjunto pode ter mais de um acorde correto. Na Figura 31, é mostrado o passo a passo do experimento e qual a saída desejada. À esquerda, é mostrado um cromagrama representando o arquivo de áudio gravado para este experimento. Logo a direita, este mesmo arquivo de áudio foi fragmentado em conjuntos com 100 exemplos cada um. No centro da figura, está a classificação dos conjuntos utilizando os modelos criados neste trabalho. Como saída é gerada uma sequência de acordes. Cada etapa do experimento será detalhada a seguir.

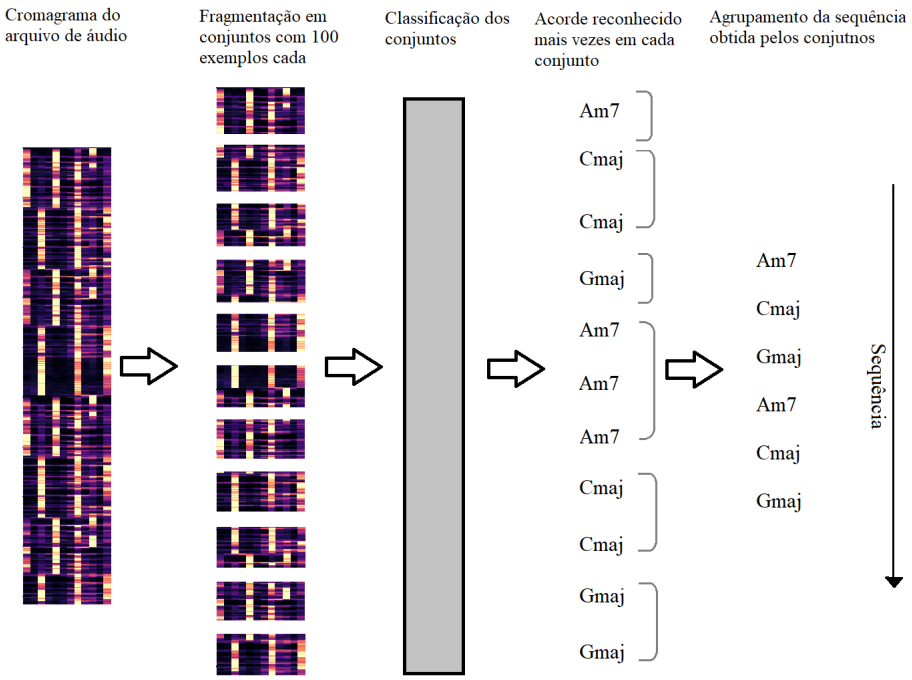


Figura 31 – Experimento realizado utlizando áudios que contenham uma sequencia de acordes.

No experimento, cada conjunto é submetido ao modelo, sendo testados cada um dos 100 exemplos. Após testar um conjunto inteiro, é verificado qual foi o acorde reconhecido mais vezes. Cada conjunto então é classificado pelo acorde mais reconhecido entre os seus exemplos. Na Figura 32, é mostrado graficamente os acordes reconhecidos no primeiro conjunto utilizando o modelo baseado em `CQT`. O conjunto da figura contém 100 exemplos. Os exemplos do conjunto foram classificados em Am 7%, Amaj 1%, C7 2%, Am7 88% e Cmaj 2%. Como a maioria dos exemplos do conjunto foi classificado como Am7, é considerado o acorde Am7 como o primeiro acorde da sequência.

Todos os 26 conjuntos foram submetidos ao procedimento. Na Figura 33 é ilustrada o reconhecimento dos acordes do segundo conjunto, tendo como resultado o acorde Cmaj. Nas Figuras 34 e 35 são representados graficamente o reconhecimento dos acordes do terceiro e quarto conjuntos. Foram reconhecidos mais vezes, respectivamente, os acordes Cmaj e Gmaj.

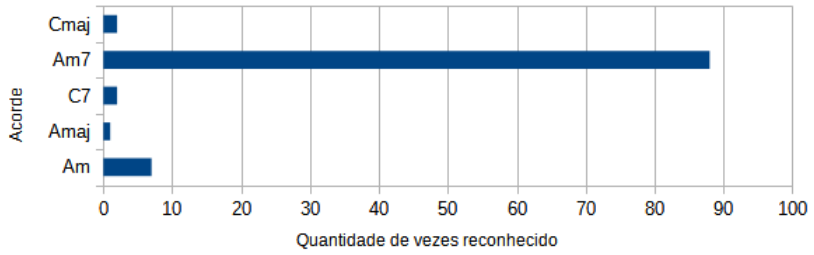


Figura 32 – Acordes classificados pelo modelo CQT no primeiro conjunto.

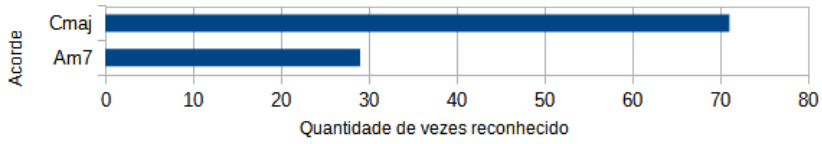


Figura 33 – Acordes classificados pelo modelo CQT no segundo conjunto.

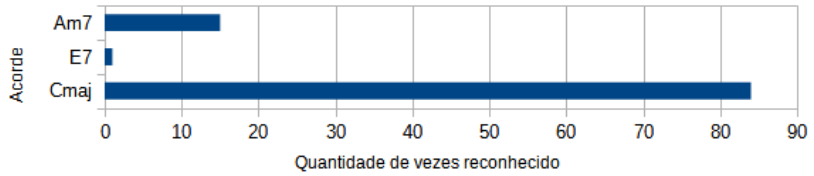


Figura 34 – Acordes classificados pelo modelo CQT no terceiro conjunto.

Na medida que os conjutos são classificados, é construída uma sequência. Para os quatro primeiros conjuntos, foi construída a sequência Am7, Cmaj, Cmaj e Gmaj. Ao final da classificação pelo modelo, é esperada uma sequência composta por 26 acordes. Na sequência referente aos quatro primeiros conjuntos, o acorde Cmaj aparece duas vezes consecutivas. Neste caso, entende-se que no áudio não houve troca de acorde. Estes dois elementos da sequência podem ser representados como um só.

Agrupando os acordes iguais e consecutivos, é obtida a sequência agrupada Am7, Cmaj e Gmaj. Este agrupamento é ilustrado na Figura 31. Após realizar o agrupamento para toda a sequência de 26 acordes, restaram apenas 12 elementos. Com o tamanho de sequência igual a 12, pode-se comparar a sequência de acordes correta e a sequência estimada pelos modelos. As comparações entre as sequências corretas e as estimadas são mostradas a seguir.

Na Figura 36 são comparados os acordes corretos e os acordes estimados utilizando o *dataset* e modelo baseado em `STFT`. Dos 12 acordes da sequência, sete foram reconhecidos corretamente e cinco foram reconhecidos com acordes semelhantes, mas não corretos.

Na Figura 37 são comparados os acordes corretos e os acordes reconhecidos utilizando o *dataset* e modelo baseado em `CQT`. Todos os acordes foram reconhecidos corretamente.

O experimento trouxe resultados melhores utilizando `CQT` comparado ao modelo que utiliza `STFT`. Apesar do resultado final exibir a sequência correta, não significa essencialmente

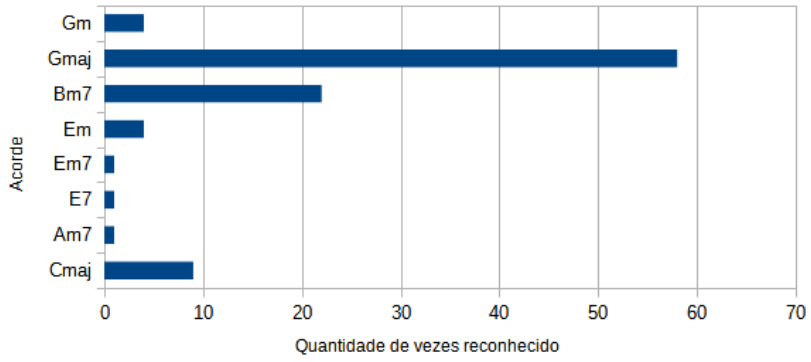


Figura 35 – Acordes classificados pelo modelo CQT no quarto conjunto.

Acorde correto	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj
Acorde reconhecido	Am7	Cmaj	Gmaj	Am7	Am7	Gmaj	Am	Am7	Gmaj	Am	Am7	Gmaj
	✓	✓	✓	✓	×	✓	×	×	✓	×	×	✓

Figura 36 – Comparação entre os acordes da sequência e os acordes reconhecidos - STFT.

Acorde correto	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj
Acorde reconhecido	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj	Am7	Cmaj	Gmaj
	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figura 37 – Comparação entre os acordes da sequência e os acordes reconhecidos - CQT.

que o classificador reconheceu todos os exemplos sem cometer equívocos. Pode-se dizer apenas que a maioria dos exemplos foram classificados com o acorde correto. As Figuras 32, 33, 34 e 35 mostram respectivamente os acordes reconhecidos nos quatro primeiros conjuntos tomados do arquivo de teste. Na Figura 34 é notado que os acordes Cmaj e Am7 foram reconhecidos mais vezes, mas ambos pertencem à sequência correta. Diferentemente do resultado obtido nas Figuras 32, 33 e 35, onde acordes não pertencentes à sequência foram reconhecidos.

Os resultados observados do reconhecimento dos quatro primeiros conjuntos mostram que a maioria dos exemplos foram classificados com a classe correta. Define-se como classe correta um acorde pertencente à sequência. As partes silenciosas e ruídos da gravação não interferiram significativamente no reconhecimento da sequência.

6.5 CONSIDERAÇÕES FINAIS

O trabalho feito a partir de *datasets* próprios proporcionou um ambiente de experimentos confiável para a tarefa de classificação de acordes isolados, ou seja, sem a presença de outros instrumentos. Tendo todos os áudios gravados e rotulados de acordo com os nomes dados às formações, a tarefa de criação do modelo se torna mais assertiva. Apesar disso, com uma qualidade de gravação inferior aos *datasets* baseados em músicas, não tinha se a certeza de que um *dataset* próprio até então nunca experimentado seria adequado para o treinamento de um modelo. Por isso, optou-se por criar um *dataset* considerado pequeno, com apenas 41 acordes.

Com esta quantidade foram obtidos resultados com precisão de 99 %.

O modelo gerado utilizando `CQT` se mostrou robusto reconhecendo corretamente a sequência de acordes proposta no experimento. Mesmo sendo testado um áudio sem cortes. Já o resultado com o modelo `STFT` mostrou que esse modelo é mais sensível a ruídos e menos preciso. Isto comprova a teoria de que um método mais robusto de extração de *features*, como a `CQT`, interfere diretamente na capacidade de reconhecimento do modelo. A extração de features mais descritivas contribuiu para o resultado dos experimentos realizados no modelo baseado em florestas aleatórias.

Tabela 2 – Resultado do treinamento para cada uma das 41 classes utilizando o *dataset* `STFT`

Classe	precision	recall	f1-score	support
A#7	0.98	0.99	0.99	724
A#m7	0.98	0.99	0.99	557
A#m	0.99	0.99	0.99	453
A7	0.98	0.99	0.99	595
Amaj	0.97	0.99	0.98	381
Am7	0.98	1.00	0.99	626
Am	0.99	0.99	0.99	501
B7	0.97	1.00	0.98	648
Bmaj	1.00	0.97	0.99	300
Bm7	0.99	0.98	0.98	654
Bm	0.99	0.98	0.99	495
C#7	0.99	0.98	0.98	606
C#m7	0.99	0.98	0.98	675
C#m	0.99	0.99	0.99	378
C7	0.99	0.99	0.99	654
Cmaj	0.98	0.99	0.98	406
Cm7	0.99	0.99	0.99	541
Cm	1.00	0.99	0.99	411
D#7	0.99	0.99	0.99	654
D#m7	0.99	0.99	0.99	561
D#m	0.99	0.99	0.99	595
D7	0.99	0.99	0.99	856
Dmaj	0.99	0.98	0.98	484
Dm7	0.99	0.98	0.98	642
Dm	1.00	0.98	0.99	490
E7	0.99	0.98	0.99	634
Emaj	0.98	0.98	0.98	285
Em7	0.99	0.97	0.98	507
Em	0.99	0.98	0.98	467
F#7	0.98	0.99	0.99	695
F#m7	0.98	0.99	0.99	570
F#m	1.00	0.99	0.99	371
F7	0.98	0.99	0.98	425
Fmaj	1.00	0.98	0.99	275
Fm7	0.97	0.98	0.97	635
Fm	0.98	0.98	0.98	381
G#7	0.97	0.98	0.98	394
G#m7	0.99	0.99	0.99	596
G7	0.96	1.00	0.98	627
Gm7	0.97	0.98	0.98	529
Gm	0.99	0.98	0.99	492

Tabela 3 – Resultado do treinamento para cada uma das 41 classes utilizando o *dataset* CQT

Classe	precision	recall	f1-score	support
A#7	0.99	1.00	1.00	352
A#m7	1.00	1.00	1.00	288
A#m	1.00	0.99	0.99	230
A7	0.99	1.00	1.00	316
Amaj	0.98	0.99	0.99	193
Am7	1.00	0.99	0.99	326
Am	0.99	0.99	0.99	271
B7	0.99	1.00	0.99	315
Bmaj	1.00	0.98	0.99	162
Bm7	1.00	0.99	1.00	346
Bm	0.99	1.00	0.99	243
C#7	1.00	0.99	0.99	295
C#m7	1.00	0.99	1.00	327
C#m	0.98	1.00	0.99	215
C7	1.00	0.99	0.99	327
Cmaj	1.00	1.00	1.00	224
Cm7	1.00	1.00	1.00	273
Cm	1.00	0.99	1.00	188
D#7	0.99	1.00	1.00	310
D#m7	1.00	0.99	0.99	253
D#m	0.99	1.00	1.00	302
D7	1.00	1.00	1.00	407
Dmaj	0.99	1.00	0.99	236
Dm7	1.00	0.98	0.99	322
Dm	0.98	0.99	0.98	234
E7	0.98	1.00	0.99	297
Emaj	1.00	0.97	0.99	143
Em7	0.99	0.95	0.97	261
Em	0.95	0.99	0.97	224
F#7	0.99	1.00	0.99	300
F#m7	1.00	1.00	1.00	288
F#m	1.00	1.00	1.00	167
F7	0.99	1.00	0.99	215
Fmaj	0.98	0.99	0.98	150
Fm7	0.99	0.97	0.98	346
Fm	0.95	1.00	0.98	177
G#7	0.99	0.99	0.99	201
G#m7	1.00	0.99	1.00	305
G7	0.99	0.99	0.99	304
Gm7	0.99	0.98	0.98	299
Gm	0.99	0.99	0.99	256

7 CONCLUSÃO

Neste trabalho relacionado a Music Information Retrieval - MIR, explorou-se a área dedicada ao reconhecimento de acordes. Foram levantados trabalhos relacionados a fim de se observar o estado da arte desta área de pesquisa. Após este levantamento, concluiu-se que modelos robustos para extração de *features* como os que utilizam Deep Learning se mostram mais eficazes no reconhecimento de acordes.

Métodos para a extração de cromagramas como o Deep Chroma Extractor visto nos trabalhos de (KORZENIOWSKI; WIDMER, 2016), (MCFEE; BELLO, 2017) e (NADAR; ABESSER; GROLLMISCH, 2019) garantem dados mais descritivos acerca dos acordes. Consequentemente, os acordes são reconhecidos com mais assertividade por modelos de aprendizado de máquina, como por exemplo florestas aleatórias.

Os *datasets* utilizados nos trabalhos relacionados são baseados em músicas, porém, a anotação dos acordes nem sempre estão corretas. Com a finalidade de se ter um ambiente totalmente controlado e menos suscetível a falhas, optou-se pela criação de um *dataset* próprio utilizando gravações feitas pelo autor. Foram escolhidos 48 acordes para serem gravados, dos quais 41 foram de fato considerados no *dataset*.

Para a extração de features e criação dos modelos, foram pesquisadas tecnologias capazes de realizar estas tarefas. Escolheu-se como linguagem de programação base o *Python*. No *Python*, encontra-se a biblioteca *librosa*, que também é aplicada nos trabalhos relacionados. Na biblioteca *librosa*, as funções utilizadas referem-se à extração de cromagramas pelas transformadas `STFT` e `CQT`. Já para a criação do modelo, explorou-se a biblioteca `sklearn` para a identificação de hiperparâmetros e para a criação do modelo baseado em florestas aleatórias.

Após a construção dos dois *datasets* e dos dois modelos, foram realizados experimentos com a finalidade de testar a precisão de cada um. O *dataset* extraído utilizando Constant-Q Transform mostrou-se mais preciso que o dataset extraído utilizado STFT, com uma precisão média de 99% que é muito além do esperado para um *dataset* composto por 41 acordes distribuídos em 72.564 exemplos. Inicialmente o objetivo do trabalho era treinar diferentes modelos de aprendizado de máquina, porém, como os resultados com florestas aleatórias mostraram-se promissores, foi optado por manter o foco em otimizar esse modelo.

Como trabalhos futuros, pode-se explorar este mesmo conceito de florestas aleatórias e extração de *features* com `CQT` utilizando mais acordes para a criação do *dataset*. Também podem-se incluir no *dataset* áudios que contenham não somente um violão, mas acompanhamento de outros instrumentos e vozes. A biblioteca *librosa* contém ferramentas para supressão de voz e de percussão, tais ferramentas podem ser utilizadas para separar o áudio e gerar cromagramas somente com os instrumentos harmônicos. Esta modificação tornaria o reconhecimento ainda mais robusto abrindo a possibilidade de se começar a trabalhar com a extração de cifras de músicas.

REFERÊNCIAS

CHOI, Keunwoo et al. A tutorial on deep learning for music information retrieval. **arXiv preprint arXiv:1709.04396**, 2017.

CLETO, Pedro et al. Reconhecimento de Acordes Musicais: Uma Abordagem Via Perceptron Multicamadas. **Mecánica Computacional**, v. 29, n. 93, p. 9169–9175, 2010.

DODGE, Charles; JERSE, Thomas A. **Computer music: synthesis, composition and performance**. [S.l.]: Macmillan Library Reference, 1997.

GÉRON, Aurélien. **Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems**. [S.l.]: "O Reilly Media, Inc.", 2017.

KORZENIOWSKI, Filip; WIDMER, Gerhard. Feature learning for chord recognition: The deep chroma extractor. **arXiv preprint arXiv:1612.05065**, 2016.

LEMOS, Julio Cesar et al. Aplicando aprendizado de máquina para identificação do gosto musical de um indivíduo. **Revista Brasileira de Computação Aplicada**, v. 11, n. 3, p. 88–98, 2019.

MCFEE, Brian; BELLO, Juan Pablo. Structured Training for Large-Vocabulary Chord Recognition. In: ISMIR. [S.l.: s.n.], 2017. P. 188–194.

MCFEE, Brian; RAFFEL, Colin et al. librosa: Audio and music signal analysis in python. In: CITESEER. PROCEEDINGS of the 14th python in science conference. [S.l.: s.n.], 2015. v. 8, p. 18–25.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, Manole Ltda, v. 1, n. 1, p. 32, 2003.

NADAR, Christon-Ragavan; ABESSER, Jakob; GROLLMISCH, Sascha. Towards CNN-based Acoustic Modeling of Seventh Chords for Automatic Chord Recognition. In: PROCEEDINGS of the International Conference on Sound and Music Computing, Málaga, Spain. [S.l.: s.n.], 2019.

OGASAWARA, Angélica Soares et al. Reconhecedor de notas musicais em sons polifônicos. **Trabalho de Conclusao de Curso, Graduacao em Engenharia Eletrica. Universidade Federal do Rio de Janeiro, Rio de Janeiro**, 2008.

SCHEDL, Markus; GÓMEZ GUTIÉRREZ, Emilia; URBANO, Julián. Music information retrieval: Recent developments and applications. **Foundations and Trends in Information Retrieval. 2014 Sept 12; 8 (2-3): 127-261.**, Now Publishers Inc., 2014.