

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CLASSIFICAÇÃO AUTOMÁTICA DE GÊNEROS MUSICAIS UTILIZANDO
TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Emerson Demetrio Plácido

Florianópolis - SC
2018/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CLASSIFICAÇÃO AUTOMÁTICA DE GÊNEROS MUSICAIS UTILIZANDO
TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Emerson Demetrio Plácido

Trabalho de conclusão de curso apresentado como
parte dos requisitos para obtenção do grau de
Bacharel em Sistemas de Informação

Florianópolis - SC
2018/1

Emerson Demetrio Plácido

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador:

Prof. Dr. Elder Rizzon Santos

Banca examinadora:

Prof. Dr. João Candido Dovicchi

Prof. Dr. Ricardo Azambuja Silveira

Florianópolis - SC
2018/1

*Dedicado a Chester Bennington, cuja vida e obra
inspiraram e inspiram até hoje.*

AGRADECIMENTOS

Mariah Barros Cardoso
Lindiomar Ávila Borges Jr.
Débora da Silva
Rafael Spricigo

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE ABREVIACÕES.....	9
RESUMO	10
1 - INTRODUÇÃO.....	11
1.1 - OBJETIVOS	12
1.1.1 - OBJETIVOS PRINCIPAIS.....	12
1.1.2 - OBJETIVOS SECUNDÁRIOS.....	12
2 - FUNDAMENTAÇÃO TEÓRICA.....	13
2.1 - MÚSICA	13
2.2 - INTELIGÊNCIA ARTIFICIAL	18
2.3 - COMPUTAÇÃO MUSICAL E RECUPERAÇÃO DA INFORMAÇÃO MUSICAL	19
2.4 - TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL E CLASSIFICAÇÃO	23
2.4.1 - ÁRVORES DE DECISÃO: ALGORITMO J48.....	24
2.4.2 - ÁRVORES DE DECISÃO: ALGORITMO RANDOM FOREST	25
2.4.1 - REDES NEURAIS.....	25
2.5 – TRABALHOS RELACIONADOS.....	27
3 - EXTRAÇÃO DE ATRIBUTOS E CLASSIFICAÇÃO DE GÊNEROS MUSICAIS	28
3.1 - ETAPA 1 - EXTRAÇÃO DAS INFORMAÇÕES	32
3.1.1 - ESCOLHA DOS ATRIBUTOS E GÊNEROS.....	34
3.1.2 - PRÉ-PROCESSAMENTO.....	44
3.6 - ETAPA 2 - CLASSIFICAÇÃO	45
3.6.1 - ALGORITMO J48.....	47
3.6.2 - RANDOM FOREST	51
3.6.3 - REDES NEURAIS MLP - MULTILAYER PERCEPTRON.....	55
3.7 - AVALIAÇÃO DOS RESULTADOS	59
4 - CONCLUSÃO	60
4.1 - TRABALHOS FUTUROS	62
REFERÊNCIAS BIBLIOGRÁFICAS	63
APÊNDICES	66
APÊNDICE 1 - PERFIL UTILIZADO PARA EXTRAÇÃO	66
APÊNDICE 2 - EXEMPLO DE ARQUIVO USADO PARA TREINAMENTOS.....	68
APÊNDICE 3 - EXTRATOR DE INFORMAÇÕES.....	69
APÊNDICE 4 - INTERFACE ENTRE O SIS. OPERACIONAL E ESSENTIA	71

APÊNDICE 5 - CRIADOR DE ARQUIVOS DE TREINO E TESTE	73
APENDICE 6 – ARTIGO SBC	76

LISTA DE FIGURAS

Figura 1 - Partitura com claves de Sol e Fá	15
Figura 2 - Sustenido e Bemol em um sistema temperado.....	16
Figura 3 - Etapas de Extração e Classificação.....	29
Figura 4 - Parâmetros para extração.....	33
Figura 5 - average_loudness (Altura Média)	36
Figura 6 - BPM (Batidas por Minuto)	37
Figura 7- beats loudness mean (Altura média das batidas)	38
Figura 8 - danceability (Danceabilidade)	39
Figura 9 - chords_change_rate (Taxa de alteração dos acordes).....	40
Figura 10 - Distribuição média MFCC 1	41
Figura 11 – Contagem de Gêneros.....	43
Figura 12 - Melhoria dos resultados do J48	50
Figura 13 - Resultados do Random Forest	54
Figura 14 - Resultados MLP	58
Figura 15 - Comparação entre os algoritmos J48, RF e MLP	59

LISTA DE TABELAS

Tabela 1 - Atributos utilizados	34
Tabela 2 - Testes executados com o J48.....	49
Tabela 3 - Testes com o Random Forest	53
Tabela 4 - Testes com MLP	57

LISTA DE ABREVIACÕES

IA - Inteligência Artificial

CM - Computação Musical

Csv - *Comma Separated Values* (Valores separados por vírgula)

RIM - Recuperação da Informação Musical

MIR - *Music information retrieval*, termo em inglês para RIM

hz – Hertz, uma medida de frequências sonoras

wav - WAVEform audio format, Formato de áudio em onda

FLAC - *Free Lossless Audio Codec* - Codec de Áudio Livre Sem Perdas

mp3 - *MPEG Layer 3* - Um formato de áudio da Moving Picture Express Group

MIDI - *Musical Instrumental Digital Interface* - Interface Digital de Instrumentos

Musicais

MEL - Uma medida (escala) musical proposta por Volksmann e Newman

MFCC - *Mel frequency Cepstrum Coefficients* – Coeficientes Cepstrum MEL

MLP - Multi Layer Perceptron - Perceptron de Múltiplas Camadas

ID3 Tags - Conjunto de informações mais usado junto com o formato de áudio MP3

YAML - *YAML Ain't Markup Language* - Formato para serialização de dados

Javascript – Uma linguagem de programação

Python – Uma linguagem de programação

JSON - *Javascript Object Notation* - Notação de Objetos Javascript

DFA - *Detrended Fluctuation Analysis* - Análise de Flutuação Detectada

MDL – *Minimum description length* - Comprimento da Descrição Mínima

RESUMO

De uma maneira geral, definir o que é música não é uma tarefa simples, e classificar gêneros musicais é um problema ainda mais discutido. Agrupar músicas por meio de reconhecimento de padrões não é, todavia, uma grande novidade. A ciência cujo foco principal recai sobre o estudo da música e suas relações tecnológicas é a Computação Musical.

O presente trabalho contempla os processos de recuperação da informação musical, por meio de diversas etapas e técnicas, abordando ferramentas para extração e algoritmos de classificação aplicados aos gêneros e também aborda a teoria musical por trás de diversas características informacionais das músicas. Por fim, uma avaliação sobre os resultados das classificações é apresentado, demonstrando qual técnica de IA melhor classificou os dados.

Palavras-chave: Classificação, Inteligência Artificial, Recuperação da Informação Musical, Computação Musical, Redes Neurais, Árvores de Decisão, Similaridade

1 - INTRODUÇÃO

O cenário musical é um dos mais relevantes na indústria do entretenimento, tendo apresentado crescimento contínuo nos últimos anos. Impulsionado por novas tecnologias que surgem todos os dias, possibilitando acesso rápido e disseminação de conteúdo como nunca antes visto¹. Junto ao crescimento do setor, observa-se também uma queda gigantesca na influência de grandes gravadoras na produção de conteúdo independente e o surgimento de novas mídias de compartilhamento, sejam redes sociais especificamente voltadas ao compartilhamento de músicas, ou outros canais de divulgação de conteúdo.

O presente trabalho propõe uma análise de algoritmos de classificação e de extração da informação musical, a fim de classificar o gênero de um arquivo de áudio (músicas), utilizando técnicas de recuperação de informação musical (RIM) e técnicas de inteligência artificial (IA). Os desafios da computação musical abordados neste trabalho são justamente obter informações sobre faixas de áudio (músicas), de maneira que seja possível comparar faixas de áudio distintas em busca de suas semelhanças. Uma vez obtidas tais informações, o passo seguinte é demonstrar de forma simples os resultados e melhorar o processamento do sistema com novos padrões reconhecidos - tendo em mente que o motor de classificação será construído por meio de técnicas de inteligência artificial capazes de aperfeiçoar suas próximas saídas por meio de resultados previamente obtidos e reutilizados como entrada.

¹ Crescimento do setor musical disponível em <<https://veja.abril.com.br/entretenimento/industria-musical-ve-novo-crescimento-gracas-ao-streaming/>>

1.1 - OBJETIVOS

1.1.1 - OBJETIVOS PRINCIPAIS

1.1.1.1 - Construir, utilizando IA, um modelo capaz de classificar o gênero musical de um arquivo de áudio de entrada, valendo-se de seus atributos e de um treinamento prévio, utilizando informações extraídas de outros arquivos de áudio.

1.1.2 - OBJETIVOS SECUNDÁRIOS

1.1.2.1 - Extrair informações de arquivos de áudio, utilizando técnicas de recuperação da informação musical.

1.1.2.2 - Analisar algoritmos para classificação de gênero de faixas de áudio, definindo quais dados serão adotados para a etapa de classificação, com base nas capacidades de cada algoritmo, uma vez que as faixas podem ser parecidas em diversos aspectos.

1.1.2.3 - Implementar um protótipo do modelo proposto

1.1.2.4 - Testar o protótipo quanto à classificação dos áudios

1.1.2.5 - Analisar os resultados obtidos

2 - FUNDAMENTAÇÃO TEÓRICA

2.1 - MÚSICA

Existem diversas abordagens para definir o significado de música. Para Santini e Souza (2007), uma música é uma concepção intelectual e sensível ao som, cuja finalidade é transmitir a concepção intelectual e sensível a outras pessoas. Os autores acima citados afirmam que há uma aproximação entre a música e informação, todavia, diferente da informação textual, a informação musical é mais pessoal, relativa a quem recebe o sinal (o ouvinte) contido na composição musical, demonstrando que a relação entre a cultura e a música é praticamente indissociável. Abraham (1974) afirma que “O ethos estético é o produto final de uma tradição em um dado tempo e, quanto mais longa e complexa uma tradição, mais profundo e sutil é o ethos.”

Para Almeida (2007), a obra musical é abstrata e só se concretiza na sua interpretação ou execução. A autora também ressalta a diferença entre a música propriamente dita e a representação musical, propondo que a representação musical acontece durante a criação dos registros, gravações, partituras, etc.

É possível afirmar que a música é uma junção de diversos fatores culturais e informacionais. Os aspectos sociais e culturais são referentes ao local, época e povo no qual estão inseridos. Já os aspectos informacionais ou sonoros da música podem ser listados como os elementos musicais ritmo, melodia, harmonia, timbre, textura musical, estrutura musical e estilo musical.

Downie (2003) divide a música em outras sete facetas da descrição musical, são elas: tonal, temporal, harmônica, de timbre, editorial, textual e faceta bibliográfica. Estas facetas são subdivididas em: tratamento, representação (gráfica e sonora) e informações bibliográficas do produto musical. Existem ainda diversos outros

aspectos culturais relacionados à música, tais como expressão de época, fatores humanos e outros, mas que não serão abordados neste trabalho. Todavia, é importante ressaltar que a recepção da informação musical ainda é um tema muito debatido, uma vez que sua interpretação varia de indivíduo para indivíduo.

Soares (2005) é mais sucinto, subdividindo a música em quatro grandes características, listadas da seguinte maneira: ritmo, melodia, harmonia e andamento. Segundo o autor, ritmo é a sequência de sons em intervalos regulares e melodia é uma sequência de sons em intervalos irregulares, harmonia é a junção das partes e o andamento seria uma variação da harmonia, caracterizada pela definição do intervalo de tempo. É sabido que as músicas como um todo podem ser descritas como junção de notas musicais em um determinado intervalo de tempo, escritas em uma folha pautada com cinco linhas e quatro espaços, chamada pauta ou pentagrama (conforme figura 1 abaixo), respeitando uma dinâmica e uma marcação para o início da disposição das notas e um intervalo de tempo.

O autor aborda o sistema de notação universal das notas musicais, evidenciando sua disposição na pauta, elucidando o conceito de clave que, por sua vez é um elemento simbólico utilizado para definir o local na partitura onde a nota que a organiza se encontrará. De uma forma geral, pode-se ter, por exemplo, uma partitura descrita com base na clave de Sol. Conforme mostrado na figura 1 abaixo, na clave de Sol (denotada pelo símbolo G), a nota Sol apresenta-se por definição na segunda linha (de baixo para cima).



Figura 1 - Partitura com claves de Sol e Fá

É importante ressaltar que existem outras claves no contexto musical. Cada nota (conforme descrito mais adiante no texto) corresponde a uma vibração em uma determinada frequência. Por exemplo, a nota Lá vibra em 440 Hz. Ainda segundo Soares (2005), as músicas são compostas pela disposição das notas naturais e não naturais nas partituras, junto com os outros elementos descritos. Segundo o autor, as notas naturais são: Dó, Ré, Mi, Fá, Sol, Lá e Si, agrupadas em intervalos de vibração chamados tons e semitons (ver figura 2), já as notas não naturais são os chamados “acidentes” (variações) das notas naturais, sendo os sustenidos (denotados pelo símbolo \sharp) notas naturais acrescidas de meio tom e os bemóis (denotados pelo símbolo \flat), notas naturais decrescidas de meio tom.

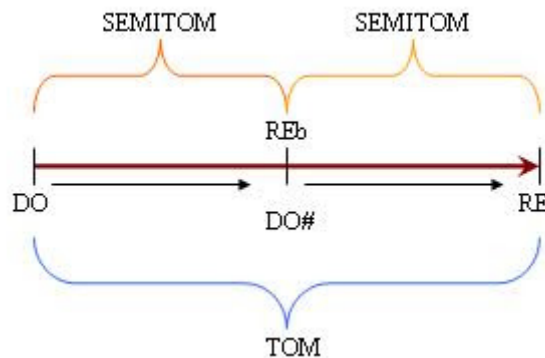


Figura 2 - Sustenido e Bemol em um sistema temperado

Por fim, Correa (2012) afirma que a música pode ser entendida de maneira geral como a arte de organizar e combinar sons no espaço e no tempo. Para a autora, o som e o tempo estão relacionados com os elementos internos da música, como a harmonia e sequenciamento das notas.

Para fins de descrição musical, o presente trabalho adota a estrutura musical proposta por Downie (2003) que sugere as sete facetas da descrição musical, especificadas a seguir.

Tonal: Relacionada ao tom da música, sendo o tom a distância ou intervalo entre duas notas. Tais intervalos, segundo o autor, são definidos por semitons, que variam em tamanho de acordo com as notas em questão. Por exemplo, da nota Dó para a nota Ré existe o intervalo de um tom, já para da nota Mi para a nota Fá, o intervalo é meio tom ou um semitom. Monteiro (2012) traz uma segunda definição para o tom: quando este também é o centro tonal da música, ou seja, se uma música está no tom de Ré Maior, isto significa que a composição da música é guiada pelas notas que fazem parte da escala de Ré Maior. A música também pode ser atonal, quando não possui um centro tonal definido.

Temporal: Caracteriza a determinação rítmica de uma obra, sendo o aspecto que descreve o período de duração de eventos musicais. O ritmo geral da obra caracteriza

a unidade fundamental do compasso, da pulsação básica subjacente à música (GROVE; SADIE, 1994).

Harmônica: Aspecto complexo da música, pode ser entendida como a simultaneidade da sonoridade de notas musicais (polifonia) tal como um acorde de violão por exemplo, caracterizado pela junção de notas executadas ao mesmo tempo, ou ainda uma nota soando por vez (monofonia) que, em conjunto com a faceta temporal, cria a melodia (Monteiro, 2012).

Timbre: o timbre é o aspecto sonoro propriamente dito, ou seja, a identidade do som. Permite ao receptor diferenciar o que está recebendo, assim como vozes humanas têm diferentes timbres, instrumentos musicais também.

Editorial: Para Monteiro (2012), informações editoriais são aquelas referentes a execução da obra, marcações de ênfase e dinâmica de execução. Por exemplo, sinais na partitura que definem um momento onde a nota deverá soar mais forte do que as outras ou ainda, que determinado compasso crescerá e depois diminuirá de ênfase. A autora também ressalta que é possível realizar a execução de uma música sem tais notações.

Textual: Informações textuais são letras das músicas, hinos, corais, óperas, etc. Para Downie (2003), a informação textual não é suficiente para o reconhecimento de uma obra musical uma vez que diversas músicas compartilham de letras por vezes muito parecidas, quando não idênticas. Em outros casos, existem obras que mantêm a mesma letra, porém diferenciam-se musicalmente pelas outras facetas completamente distintas da obra original.

Bibliográfica: Informações extras, não exatamente atreladas ao conteúdo musical, quais sejam: informações relacionadas à criação intelectual e ao registro da obra como título, gravadora, compositor, intérprete, autor da letra, entre outras.

Um outro conceito abordado neste trabalho foi o de pitch, que, segundo Klapuri (2017) é uma propriedade perceptual dos sons que permite a sua ordenação numa escala relacionada com a frequência, ou mais frequentemente, o pitch é a qualidade que permite julgar os sons como "alto" e "baixo" no sentido associado a melodias. O pitch só pode ser determinado em sons que tenham uma frequência clara e estável o suficiente para distinguir-se de ruído.

2.2 - INTELIGÊNCIA ARTIFICIAL

Para BUENO (1991), existem muitas definições para Inteligência Artificial, algumas filosóficas, outras mais pragmáticas, dependendo muito do campo de atuação dos autores das mesmas. Para E. Rich (1983), inteligência artificial é o estudo de como fazer computadores realizarem tarefas para as quais, até o momento, "o homem faz melhor". Para Shannon (1985) a IA atua em dois grandes campos: o primeiro deles, limitação das habilidades humanas (visão, fala, gestos, etc.) e o segundo como sendo a duplicação de resultados estabelecidos pelo homem por meio de sua habilidade e/ou experiência. Ainda segundo Shannon, as técnicas de IA e suas aplicações podem ser subdivididas nas seguintes áreas:

1. Soluções de Problemas: abordagem heurística que, sem analisar todas as alternativas, mostra um caminho com boa chance de êxito;
2. Raciocínio Lógico (dedução);
3. Processamento de Linguagem Natural (tradução e compreensão de textos);
4. Robótica e Visão (manipulação de objetos, sequenciamento de tarefas, reconhecimento de padrões);
5. Programação Automática (geradores automáticos de programas computacionais);

6. Aprendizagem (o sistema aprendendo com sua própria experiência, melhorando sua performance);
7. Sistemas Especialistas (armazenam o conhecimento de uma área específica de atuação, utilizando-o como suporte à tomada de decisão).

A busca por sistemas que fossem capaz de imitar o processamento cerebral humano é antiga. Rosenblatt (1958) propôs uma técnica chamada perceptron, que por sua vez é, segundo Ferneda (2006), é um classificador e gerador de funções lógicas binárias em uma rede neural. Especificamente neste trabalho, utilizou-se uma abordagem ainda mais específica, chamada de Redes Neurais com Multicamadas de Perceptron (MLP) e Backpropagation. Para Borsato (2008), as redes neurais do tipo perceptron de múltiplas camadas (MLP) são recomendadas para resolver diversos tipos de problemas de aproximação, classificação, categorização e predição.

2.3 - COMPUTAÇÃO MUSICAL E RECUPERAÇÃO DA INFORMAÇÃO MUSICAL

O conceito de computação musical tem sido amplamente discutido por diversos autores de Inteligência Artificial. Para Miletto (2004), computação musical é um ramo multidisciplinar da área das ciências da computação, dedicado ao estudo dos problemas musicais. Segundo o autor, a computação musical faz estudos aprofundados de métodos, técnicas e algoritmos para processamento e geração de som e música, representações digitais e armazenamento de informação sonora e musical, trazendo sempre como premissa de que a informação tratada é arte - o que diferencia a computação musical de outras áreas similares, tais como computação gráfica.

Ao fazer uma análise do histórico da computação musical, observa-se que estudos da área remetem ao ano de 1876. Nesse ano, Alexander Graham Bell trazia ao mundo o telefone. Miletto (2004), afirma que Graham Bell é “o pai da música eletrônica” e, por conseguinte, o precursor de seu estudo.

Para um melhor embasamento, os principais pontos em que a computação musical é baseada são listados a seguir.

Som: Caracterizado pelas variações na pressão do ar captado pelo ouvido humano. A forma como a pressão varia define seu formato, ou seja, se a pressão varia de acordo com um padrão repetitivo, ela tem um formato de onda periódica, se não há um padrão, então a pressão é chamada de ruído. Para o autor, o som possui três elementos básicos: altura tonal, volume e timbre, especificados da seguinte forma: **Altura Tonal:** A variação sonora das notas musicais pode ser caracterizada como altura tonal. Num piano, por exemplo, a disposição das teclas se dá, respectivamente, da esquerda para a direita, das notas mais graves para as notas mais agudas. Esta "altura", ou seja, se uma nota é mais grave ou mais aguda, é chamada de altura tonal. Uma repetição de uma onda periódica é chamada de ciclo. Cada nota vibra numa frequência medida em hertz. Ao analisar os ciclos das notas por unidade de tempo, observa-se que o número de ciclos diminui ou aumenta de acordo com a altura tonal, sendo que as notas mais graves têm menos ciclos por unidade de tempo e as mais agudas tem mais ciclos. Por exemplo, 100 hz indica que a variação de um determinado som será de 100 ciclos por segundo. Há um intervalo audível pelo ouvido humano, ou seja, nós só podemos ouvir sons que variam de 20 hz (mais grave) até 20.000 Hz (mais agudo). Existe uma razão logarítmica entre a frequência e a altura tonal.

Volume: É a emissão audível do som. Por exemplo, ao tocar com mais força uma tecla de piano, o som emitido será mais forte; caso a tecla seja tocada de maneira mais leve, o som emitido será mais fraco. Ao analisar o som com um osciloscópio (aparelho próprio para este tipo de medição), observa-se que a amplitude ou altura da onda produzida pelo toque será maior ou menor, de acordo com o impulso inicial. Esta altura ou ainda, a distância máxima entre o máximo e o eixo central é chamada de amplitude. Quanto mais forte é o som, maior será sua amplitude.

Timbre: É o formato da onda, é o que diferencia os sons dos mais diversos instrumentos. O timbre permite diferenciar sons na natureza e, de um modo geral, formas de ondas arredondadas produzem um timbre mais suave enquanto que as formas de ondas pontiagudas dão um timbre mais estridente. Além da informação teórica e mensurável a respeito da música, a computação musical também estuda maneiras computacionais de representar as informações obtidas nos estudos. Este tema recebe o nome de RIM - Representação da Informação Musical. Santini e Souza (2007) abordam RIM como uma interação complexa entre as diversas facetas da música (tempo, harmonia, timbre, frequência, editoria, entre outras), evidenciando um dos problemas recorrentes desta técnica, a questão da diversidade representacional, uma vez que a música pode ser originária de um arquivo de áudio, partitura ou ainda, informações presentes em arquivos de metadados, posteriormente abordados como MIDI, que, ainda segundo os autores, é um formato simbólico de representação musical, semelhante a um manual de instruções. As autoras afirmam que há uma (aparente) possibilidade infinita de experiências de acordo com cada recepção do sinal emitido, introduzindo um novo desafio aos pesquisadores de RIM e demonstrando que não basta apenas decodificar ou medir os sinais musicais, mas também fazer análise da percepção daqueles que

recebem os sinais. Em outras palavras, afirmam que aplicar métricas para análise e representação da informação musical não seria possível sem medir junto a uma audiência. Numa abordagem física, músicas são sinais analógicos com uma determinada variação temporal e rítmica, não sendo, portanto, facilmente entendidas por um computador, torna-se necessário discretizar tal sinal.

Existem diversas técnicas para transformar o sinal analógico em digital resultando em uma amostragem com perda de informação inerente à natureza discreta do computador. Esta perda é causada pela compressão que, por sua vez, é um mecanismo de conversão que gera arquivos em diversos formatos, dependendo da técnica, por exemplo, *mp3*, *wav*, *flac*, etc.

Todas estas conversões e formatos, contudo, ainda não são suficientes para definir uma música de acordo com seu estilo ou outra característica qualquer. Para tanto, é necessário uma vez mais converter o arquivo para um formato entendível a nível lógico e para este fim, existem outras técnicas e formatos, por exemplo, MIDI, que para Correa (2012), é um formato para representação em que cada nota é descrita em termos de altura, duração e intensidade, muito utilizados em sistemas digitais de composição musical. O formato MIDI é um dos padrões estabelecidos pelos diversos estudos na área, sendo base para diversos meios, desde composições musicais a representação da informação musical.

2.4 - TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL E CLASSIFICAÇÃO

Classificação, segundo Domingos (2007), é uma técnica de inteligência artificial caracterizada por ter entre seus arquivos de treinamento e teste um atributo chamado de classe. Este atributo é, normalmente, o objetivo esperado quando utiliza-se esta técnica. Existem diversos classificadores na literatura. As abordagens utilizadas neste trabalho são as redes neurais e árvores de decisão (puras e derivadas), utilizadas para classificar o gênero presente no conjunto de informações extraídas de cada música (utilizando técnicas de MIR), valendo-se de informações de outros arquivos de áudio previamente extraídas e inseridas nos conjuntos de treinamento. As características dos algoritmos escolhidos para este trabalho são demonstradas nas seções seguintes. A escolha de tais técnicas deu-se pela praticidade e afinidade do autor com as mesmas, além de uma extensa pesquisa sobre algoritmos de classificação voltados a classificação de gênero musical.

2.4.1 - ÁRVORES DE DECISÃO: ALGORITMO J48

Árvores de decisão são dispositivos capazes de classificar um determinado conjunto de atributos de acordo com um treinamento prévio. Neste trabalho, aborda-se o classificador J48. Tal algoritmo surgiu da necessidade de recodificar seu precursor, o algoritmo C4.5, que, originalmente, é escrito na linguagem C, para a linguagem Java (WITTEN, et al., 2005). Ele tem a finalidade de gerar uma árvore de decisão baseada em um conjunto de dados de treinamento, sendo este modelo utilizado para classificar as instâncias em um outro conjunto de testes. Um dos aspectos para a grande utilização do algoritmo J48 pelos especialistas em mineração de dados é o fato de que o algoritmo mostra-se adequado para os procedimentos envolvendo as variáveis qualitativas contínuas e também as discretas, presentes nas bases de dados. O algoritmo J48 foi proposto por QUINLAN (1993) e apresenta um bom resultado para a montagem de árvores de decisão, a partir de um conjunto de dados de treinamento. Para a montagem da árvore, o algoritmo J48 utiliza a abordagem de dividir-para-conquistar, onde um problema complexo é decomposto em subproblemas mais simples, aplicando recursivamente a mesma estratégia a cada subproblema, dividindo o espaço definido pelos atributos em subespaços, associando-se a eles uma classe (WITTEN E FRANK, 2005).

2.4.2 - ÁRVORES DE DECISÃO: ALGORITMO RANDOM FOREST

Random Forest (Floresta Randômica) é um modelo de classificação criado por Leo Breiman, descrito por (BREIMAN, 2001) como uma combinação de árvores de decisão, sendo que cada árvore é dependente de valores aleatórios amostrados independentemente e com a mesma distribuição para todas as árvores na “floresta”. O algoritmo Random Forest se mostra eficiente convergindo efetivamente para vários problemas, inclusive em problemas envolvendo grandes conjuntos de dados (ZAKARIAH, 2014) em diferentes áreas como: Detecção de fraudes de cartão de crédito, classificação genética, detecção de e-mail Spam, entre outros. Ao fim deste trabalho, percebeu-se que o Random Forest obteve o melhor resultado para classificação de gêneros de músicas.

2.4.1 - REDES NEURASIS

Segundo Ferneda (2006), as redes neurais artificiais diferenciam-se entre si pela sua arquitetura e pela forma como os pesos associados às conexões são ajustados durante o processo de aprendizado. A arquitetura de uma rede neural restringe o tipo de problema no qual a rede poderá ser utilizada e é definida pelo número de camadas (camada única ou múltiplas camadas), pelo número de nós em cada camada, pelo tipo de conexão entre os nós (*feedforward* ou *feedback*) e por sua topologia.

Uma das propriedades mais importantes de uma rede neural artificial é a capacidade de aprender por intermédio de exemplos e fazer inferências sobre o que aprendeu, melhorando gradativamente o seu desempenho. As redes neurais utilizam um algoritmo de aprendizagem cuja tarefa é ajustar os pesos de suas conexões.

Existem duas formas básicas de aprendizado de redes neurais: aprendizado supervisionado e aprendizado não supervisionado. No aprendizado supervisionado, um agente externo apresenta à rede neural alguns conjuntos de padrões de entrada e seus correspondentes padrões de saída.

Conforme afirma TELVIO (2005), redes neurais podem ser utilizadas para classificação de padrões de entrada em categorias ou classes. Uma rede neural pode receber uma coleção de padrões de entrada com a categoria à qual cada padrão particular pertence. Depois, segundo o autor, pode-se apresentar um novo padrão ainda desconhecido e apresentá-lo à rede, e a tarefa da mesma será classificar este novo padrão corretamente. Esse tipo de classificação pode ser resolvido por um aprendizado supervisionado. Uma das vantagens do uso de uma rede neural para realizar classificação de padrões é a construção de fronteiras de decisões não lineares entre as diferentes classes e oferecer um método prático para resolver problemas altamente complexos de classificação de padrões, ou seja, é possível definir não apenas um conjunto resposta no qual a entrada da rede é classificado, como também um grau de pertinência distinto para os diversos grupos previamente inseridos na rede. Há também uma importante função para o aprendizado não supervisionado na classificação de padrões, especialmente quando não há um conhecimento inicial das categorias em que os padrões serão classificados. Para que o processo de classificação de padrões tenha sucesso é necessário que exista o mapeamento dos padrões de entrada em um espaço de classificação. Este espaço é definido por regiões de decisão que separam os padrões, atribuindo cada padrão em uma classe. Neste trabalho, foram utilizadas redes neurais de aprendizagem supervisionada com multicamadas (MLP).

2.5 – TRABALHOS RELACIONADOS

A metodologia abordada no presente trabalho já se provou válida por meio de outros trabalhos presentes na literatura e disponíveis na internet. Diversos trabalhos utilizaram métricas MFCCs e inteligência artificial para classificar gêneros de músicas.

SINGH (2014) utilizou técnicas de IA e RIM para extração e reconhecimento de gêneros musicais valendo-se de uma estrutura muito semelhante ao que foi proposto no presente trabalho, com a diferença de que o autor utilizou um conjunto de gêneros fixos nos quais seriam classificadas o conjunto de músicas das quais as informações foram extraídas. O autor apresentou os resultados por meio de grau de pertinência.

LACHMISH (2016) apresentou uma abordagem semelhante, todavia, valeu-se de uma biblioteca de amostras de música, as quais foram classificadas também por meio de técnicas de IA em um conjunto fixo de gêneros, obtendo até 63% de classificações corretas.

Por fim, SHRINIVASAN (2016) abordou o tema de forma semelhante aos autores anteriores, adotando não apenas as métricas MFCCs como parâmetro, mas também outras frequências presentes nas informações presentes nos arquivos. O autor, assim como os demais, também utilizou um conjunto de gêneros fixo para classificação, obtendo até 70% de acurácia.

O presente trabalho, assim como os anteriores citados, utiliza métricas MFCC para classificação, porém difere dos demais por não fixar um conjunto de gêneros para classificação, mas obter esta informação dos arquivos e utiliza-la através de algumas normalizações, como a generalização dos gêneros que será discutida nas sessões posteriores.

3 - EXTRAÇÃO DE ATRIBUTOS E CLASSIFICAÇÃO DE GÊNEROS MUSICAIS

Neste capítulo, são abordadas as estratégias de extração de informação das músicas e as técnicas de classificação utilizadas.

De maneira geral, este trabalho consistiu-se em obter determinadas características das músicas analisadas a fim de utilizá-las como entrada para um algoritmo de classificação. Todo este processo de extração de informações é, como descrito nos capítulos anteriores, parte da ciência chamada Computação Musical, já os processos de classificação são parte de uma das áreas de Inteligência Artificial.

Conforme descrito nos objetivos, tornou-se necessário escolher quais abordagens de extração e classificação seriam utilizadas e também a maneira como estes procedimentos seriam executados. A figura a seguir demonstra, em termos gerais, o fluxo de extração de informações e classificação planejados para o este trabalho.

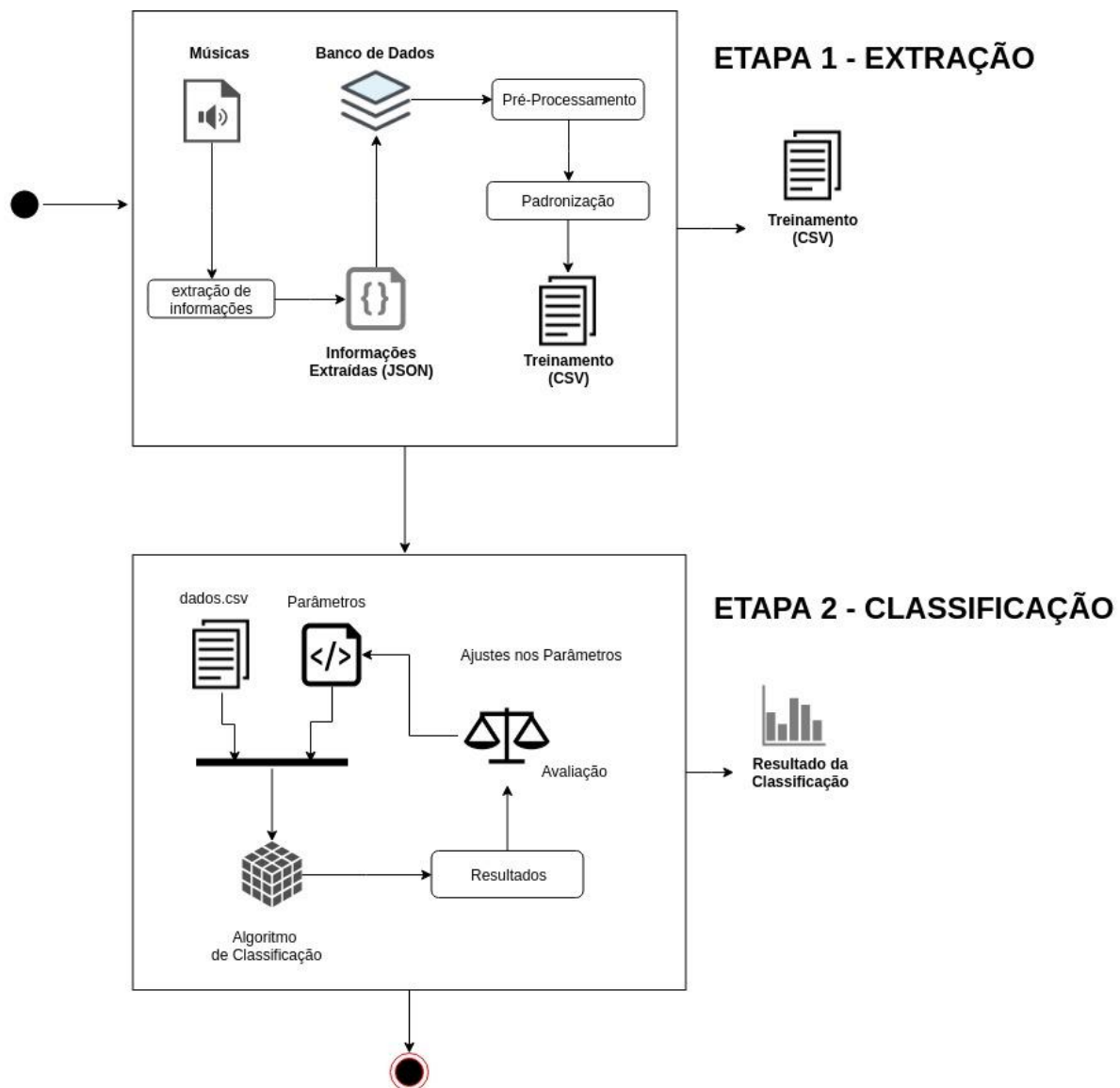


Figura 3 - Etapas de Extração e Classificação

Como é possível observar na figura acima, diversas tarefas foram necessárias para a obtenção dos insumos para cada etapa. Para dar continuidade ao processo, realizou-se duas pesquisas de ferramentas para extração e classificação. Durante a pesquisa sobre qual ferramenta de extração seria utilizada no projeto, diversas bibliotecas e ferramentas foram encontradas. As que mais se adequaram às

características do projeto foram as seguintes: ffmpeg², openSMILE³, librosa⁴ e essentia⁵.

Optou-se por fazer uso da biblioteca essentia (doravante, denotada apenas por essentia) por conta da facilidade que a mesma proporciona para configurar a saída esperada e também para extrair informações dos arquivos de áudio.

Segundo Bogdanov (2013), essentia é uma ferramenta capaz de extrair diversas informações de um arquivo de áudio, baseado num arquivo de configuração. Tal ferramenta foi escrita na linguagem de programação C++ e posteriormente portada como um módulo para a linguagem python. A essentia também é capaz de recuperar as informações presentes no arquivo de maneira organizada e parametrizável, sendo que, uma vez instalada, pode ser acessada diretamente do sistema operacional hospedeiro.

Para analisar os resultados de cada algoritmo de classificação, novamente diversas opções surgiram. Inicialmente optou-se por construir os classificadores utilizando bibliotecas da linguagem python, mas esta ideia não foi levada adiante. Outras ferramentas de IA encontradas foram: rapidMiner⁶, Apache Mahout⁷, Matlab⁸ e WEKA⁹.

Neste trabalho, optou-se por utilizar ferramenta WEKA, que é uma biblioteca com diversos algoritmos de inteligência artificial, por conta da familiaridade do autor para com a mesma e também por tratar-se de uma ferramenta livre.

² FFMPEG Disponível em <<https://ffmpeg.org/ffmpeg.html>>

³ OpenSMILE Disponível em <<https://mediatum.ub.tum.de/doc/1082431/1082431.pdf>>

⁴ Librosa Disponível em <<https://librosa.github.io/librosa/>>

⁵ Essentia Disponível em <<http://essentia.upf.edu/>>

⁶ RapidMiner Disponível em <<https://rapidminer.com>>

⁷ Apache Mahout Disponível em <<https://mahout.apache.org/>>

⁸ Matlab Disponível em <<https://www.mathworks.com/products/matlab.html>>

⁹ WEKA Disponível em <<https://www.cs.waikato.ac.nz/ml/weka/>>

Para a execução dos algoritmos de classificação descritas neste trabalho (treinamento, testes e avaliações), utilizou-se um acervo pessoal contendo 120 gigabytes de arquivos de músicas (aproximadamente 20 mil arquivos) no formato mp3.

3.1 - ETAPA 1 - EXTRAÇÃO DAS INFORMAÇÕES

Esta seção compreende a primeira etapa descrita na figura 3. Toda a extração das informações presentes nos arquivos musicais neste trabalho se deu por meio da utilização da *essentia*. A convenção de notas com acidente adotada pela *essentia* se dá pelo sustenido, ou seja, em seus resultados de extração, nunca haverá um bemol, mas sustenidos em cada saída. Por exemplo, sabe-se que Lá Sustenido é, muitas vezes, chamado também de Si Bemol, ao utilizar a *essentia*, a saída de seus extratores sempre conterà o acidente em formato de sustenido. Se a afinação do arquivo em que se está executando a análise for Si bemol, no arquivo de saída aparecerá Lá Sustenido.

Das diversas características presentes no arquivo, as informações mais interessantes para a modificação do resultado são a quantidade de samplers usada para extração, que é definida pelo tempo total em segundos desde o início da música (ou de um determinado tempo previamente especificado) até um determinado tempo final especificado.

Neste projeto, optou-se por utilizar 60 segundos de cada música para fins de extração por dois motivos. O primeiro deles é fruto da observação das características dos arquivos de entrada. Percebeu-se que em boa parte das músicas, os primeiros 15 segundos de cada música são normalmente utilizados para introdução, seguida por verso e refrão. O segundo fator crucial foi a capacidade de processamento dos equipamentos disponíveis no momento da extração. Durante a etapa de extração, para os primeiros 60 segundos dos 17 mil arquivos foram necessárias aproximadamente 72 horas para que o algoritmo de extração terminasse sua execução. Esse tempo é diretamente proporcional aos parâmetros estabelecidos

como entrada para a essentia. Observou-se que ao aumentar a quantidade de tempo, a essentia demorava mais para extrair e, por esta razão, optou-se por estabelecer 60 segundos como parâmetro para extração.

A descrição completa dos parâmetros utilizados para a extração é mostrada a seguir:

```
    default
    startTime: 0
    endTime: 60
    outputFrames: 0
    outputFormat: json
    requireMbid: FALSE
    indent: 4

average_loudness:
  frameSize: 88200
  hopSize: 44100
  windowType: hann
  silentFrames: noise

  lowlevel:
    frameSize: 2048
    hopSize: 1024
    zeroPadding: 0
    windowType: blackmanharris62
    silentFrames: noise
    stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]

    tonal:
      frameSize: 4096
      hopSize: 2048
      zeroPadding: 0
      windowType: blackmanharris62
      silentFrames: noise
      stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]

      rhythm:
        method: degara
        minTempo: 40
        maxTempo: 208
        stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]
```

Figura 4 - Parâmetros para extração

3.1.1 - ESCOLHA DOS ATRIBUTOS E GÊNEROS

Esta seção aborda quais atributos extraídos das músicas foram utilizados para os processos de classificação, conforme etapa 2 descrita na figura 3.

Após o primeiro esforço para extração, tornou-se necessário definir quais informações seriam analisadas pelos algoritmos propostos, uma vez que a saída da essentia é composta por diversos atributos. Segundo Rajani (2009), para classificação de gênero musical baseado em MIR, os dados mais interessantes seriam os referentes a escala mel, pitch, afinação e outros. Para este trabalho, optou-se, inicialmente, por utilizar apenas a métrica MFCC, extraída dos arquivos de áudio utilizando a essentia. A métrica MFCC, segundo CUADROS (2007) é uma medida de extração de características que tem por objetivo aproximar o espectro do som ao que é percebido pelo ouvido humano, baseado na escala Mel. Ela codifica o espectro da potência de um som, sendo calculado por meio de uma transformada de Fourier do logaritmo do espectro do sinal.

Métrica	Relação	Definição
average_loudness	Geral	Intensidade Média dos picos
bpm	Ritmo	Batidas por minuto
beats_loudness_mean	Ritmo	Intensidade média de cada batida
danceability	Ritmo	Métrica de danceabilidade
chords_changes_rate	Tom	Taxa de Mudança de Acordes
mfcc (13 valores)	Melodia	Coefficientes MEL
gênero	Gênero	O gênero da música (classe)

Tabela 1 - Atributos utilizados

A descrição de cada atributo a seguir é baseada na documentação da essentia¹⁰.

¹⁰ Documentação da Essentia Disponível em <<http://essentia.upf.edu/documentation/reference/>>

Características de cada atributo escolhido para a classificação são demonstrado a nas sessões seguintes.

3.1.1.1 - INTENSIDADE MÉDIA (average_loudness)

Segundo consta na documentação da essentia, esta métrica é a computação da altura média do sinal. A figura a seguir demonstra uma contagem simples do atributo average_loudness.

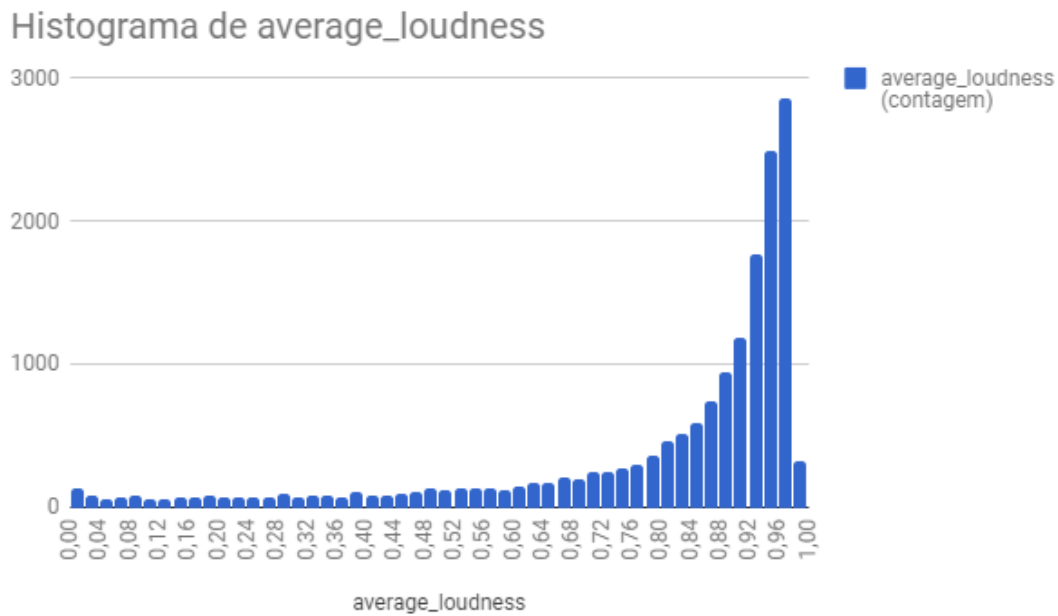


Figura 5 - average_loudness (Altura Média)

3.1.1.2 - BATIDAS POR MINUTO (bpm)

Esta métrica é a extração das posições (picos) das batidas da música, estimando seu valor no tempo. Em outras palavras, a quantidade de picos de uma música em um determinado espaço de tempo. A figura a seguir demonstra a distribuição das informações de bpm.

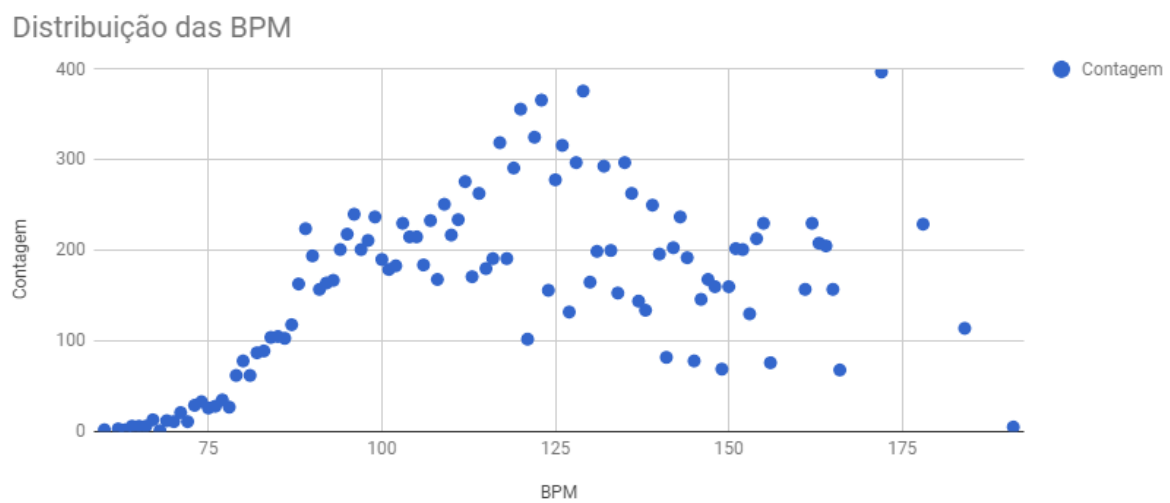


Figura 6 - BPM (Batidas por Minuto)

3.1.1.3 - INTENSIDADE MÉDIA DAS BATIDAS (beats_loudness_mean)

O espectro da energia de cada batida. A energia é calculada com a frequência e o intervalo especificado para cada banda. A figura a seguir demonstra a distribuição das medidas beats_loudness_mean.

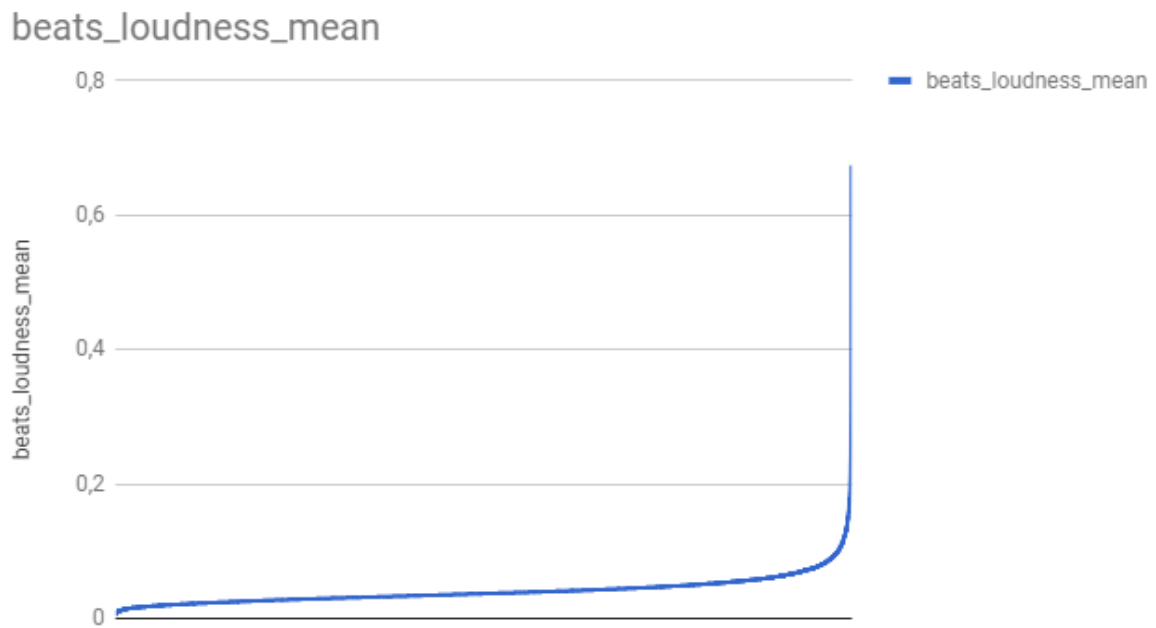


Figura 7- beats loudness mean (Altura média das batidas)

3.1.1.4 - DANCEABILIDADE (danceability)

Cálculo da danceabilidade estimada do sinal, derivado do cálculo de DFA.



Figura 8 - danceability (Danceabilidade)

3.1.1.5 - TAXA DE ALTERAÇÃO DE ACORDES POR INTERVALO

(chords_change_rate)

Determina a progressão de acordes descrita pela nota que dá tom a música.

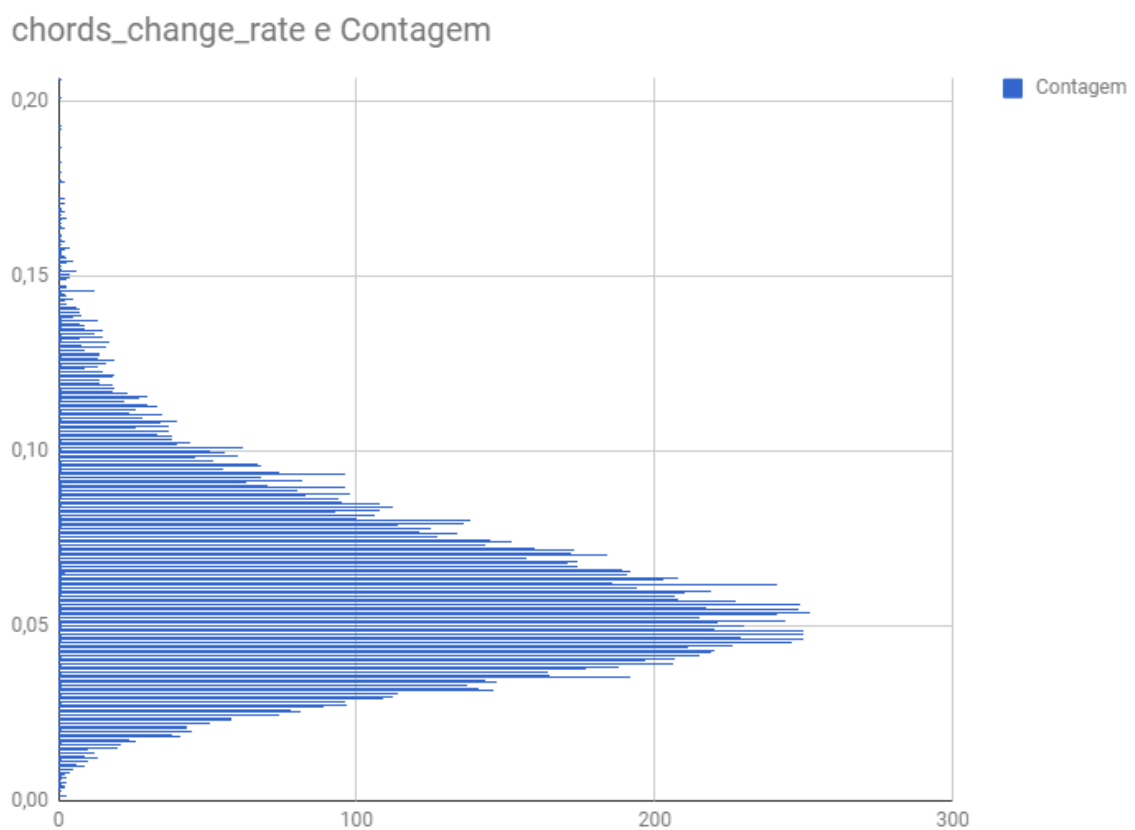


Figura 9 - chords_change_rate (Taxa de alteração dos acordes)

3.1.1.6 - 13 PRIMEIROS CENTRÓIDES MFCC

São o resultado do cálculo dos 13 primeiros coeficientes cepstrum da frequência MEL, que por sua vez é de um espectro. Os MFCCs apresentam um bom desempenho em extração de características presente em sinais da fala humana uma vez que conseguem representar os aspectos percentualmente relevantes do espectro do som de curta duração. As métricas MEL são uma escala perceptual de pitch, que é a frequência fundamental do som percebida. O nome mel originou-se da palavra melodia para indicar que a escala é baseada em comparação de pitch.

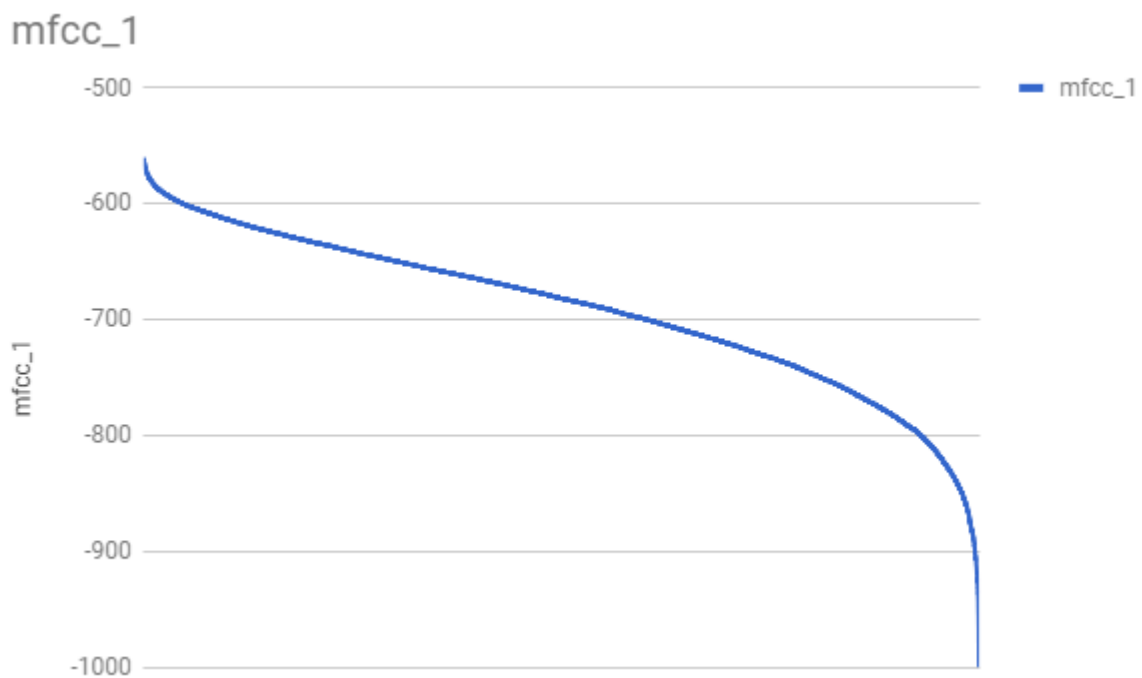


Figura 10 - Distribuição média MFCC 1

3.1.1.6 - GÊNERO

Durante o desenvolvimento do projeto, duas técnicas para obtenção de gênero foram abordadas.

A primeira delas consistiu em buscar individualmente o gênero de cada arquivo na internet, por meio de conexões com bibliotecas de terceiros, baseando-se no nome do arquivo e no nome do diretório do sistema operacional onde o mesmo se encontrava. Esta abordagem demonstrou-se problemática e pouco eficiente, uma vez que as buscas tomavam um tempo extremamente grande para serem concluídas e adicionavam uma lógica que fugia do escopo definido para este trabalho.

A segunda abordagem foi a utilização das tags ID3 presentes em cada arquivo. Nestas tags, encontram-se diversas meta-informações sobre o arquivo, como autor, intérprete, gênero, álbum do qual a música faz parte, entre outras. Esta técnica foi a escolhida para o projeto, dando origem a lista de gêneros abaixo.

Os gêneros extraídos das músicas eram, por vezes, dispersos, ou seja, não havia como fazer contagem significativa dos mesmos para um agrupamento cujo peso faria sentido durante a etapa de classificação. Neste sentido, foi executada uma contagem para definir quais gêneros entraram nos arquivos de treinamento e teste. Gêneros cuja contagem foi maior do que 500 arquivos entraram na classificação. Para aqueles arquivos cuja contagem do gênero não atingiu o limite inferior, foi definido por uma inferência simples, executada da seguinte maneira: Se o gênero presente no arquivo é “metal progressivo”, seu gênero “pai” é “metal”, portanto, para fins de classificação, o gênero deste arquivo será “metal”. Músicas cujo gênero não se adequava ao gênero pai, ou seja, aquelas cuja derivação e contagem eram respectivamente desconhecidas e pouco significantes foram desconsideradas.

Um gráfico com a distribuição de músicas por gênero é mostrado a seguir:

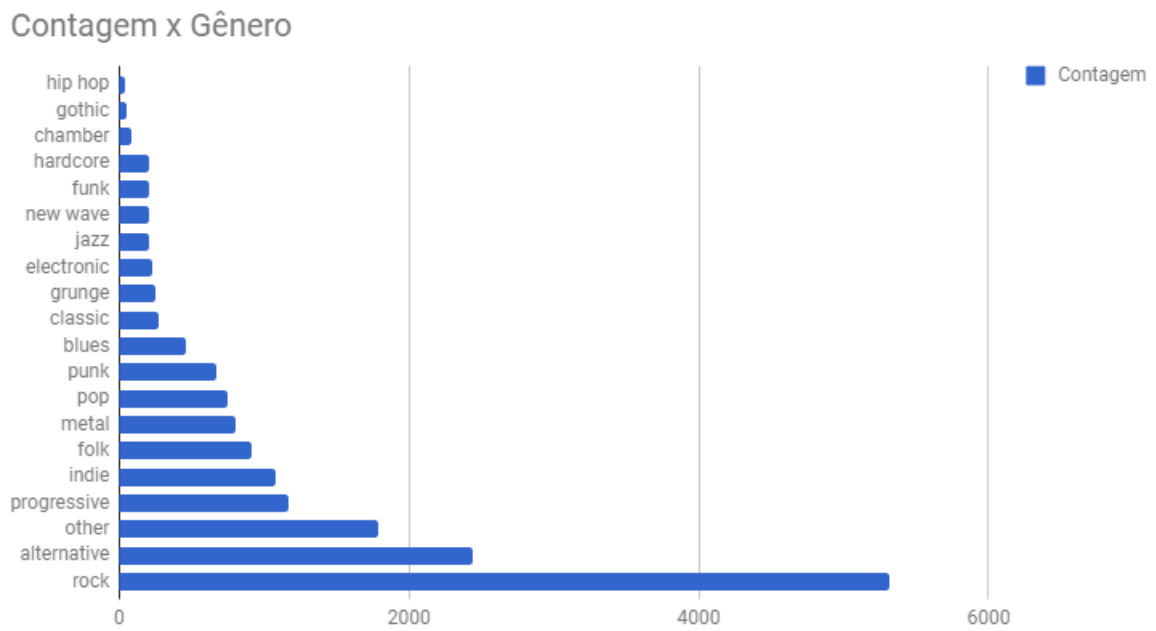


Figura 11 – Contagem de Gêneros

3.1.2 - PRÉ-PROCESSAMENTO

A fim de criar entradas que fossem entendidos pela ferramenta WEKA, criou-se uma série de programas para leitura, armazenamento e conversão de informação. Estes programas foram escritos nas linguagens Python e Javascript.

Após a instalação de ambas as ferramentas WEKA e essentia, criou-se um arquivo chamado profile.yml que define como se dará a extração dos dados por parte da ferramenta essentia. Neste arquivo, encontram-se as métricas e intervalos com as quais a essentia operou. O arquivo encontra-se citar diretamente qual apêndice, anexo é uma informação que não foi você quem desenvolveu nos anexos deste documento.

Em seguida, após a criação do arquivo, foi necessário criar uma ferramenta capaz de ler iterar sobre as músicas do acervo, a fim de utilizar cada uma delas como entrada para a extração executada pela essentia. Após a extração dos atributos, os dados foram armazenados num banco. Criou-se também um programa capaz de ler estas informações guardadas no banco que, por fim, criava um arquivo no formato csv. Este arquivo contém as informações extraídas para fins de classificação e o atributo classe, que é gênero de cada música. Tal programa também cria arquivos com uma determinada quantidade de músicas por gênero, facilitando treinamento e avaliação dos resultados das classificações executadas por cada algoritmo.

Uma vez que as informações foram extraídas dos arquivos e armazenadas no banco de dados, tornou-se simples alterar os atributos cuja saída primária é textual (string) para valores inteiros ou decimais. Bastou fazer uma conversão para cada

atributo, durante a etapa da construção dos arquivos de treino (csv). Esta conversão textual para numérica é comumente chamada de *parseInt*, conversão de valores textuais para valores inteiros e *parseFloat*, conversão de valores textuais para valores decimais.

A principal conversão efetuada neste trabalho foi a métrica bpm, cujo valor fornecido pela essentia é decimal e observa-se o uso de valores inteiros para tal atributo. O layout final dos arquivos é demonstrado no apêndice 2 deste documento e a distribuição das informações ficou da seguinte maneira:

- Quantidade total de arquivos: 17.000
- Quantidade total de gêneros: 19
- Gêneros: 'alternative', 'blues', 'chamber', 'classic', 'electronic', 'folk', 'funk', 'gothic', 'grunge', 'hardcore', 'hip hop', 'indie', 'jazz', 'metal', 'new wave', 'pop', 'progressive', 'punk' e 'rock'.

3.6 - ETAPA 2 - CLASSIFICAÇÃO

Em posse das informações obtidas nas etapas descritas anteriormente (arquivos no formato csv para treinamento e teste), tornou-se possível avançar no sentido de aplicar as técnicas de classificação. É importante ressaltar que as técnicas utilizadas no trabalho diferem um pouco daquelas apresentadas na literatura, principalmente em relação aos parâmetros disponíveis. Em alguns momentos, parâmetros descritos nos artigos não aparecem ou não estão disponíveis via interface na ferramenta WEKA.

Toda a validação foi feita utilizando a técnica de cross validation, que é caracterizado pelo distribuição dos arquivos entre conjuntos de tamanho N para

treinamento e teste. O número de folds (a quantidade de arquivos por conjunto) utilizado para cada execução também foi considerado um parâmetro.

Para todos os casos, foi utilizado um arquivo para treinamento e teste no formato csv, contendo variações em quantidade de linhas. Observou-se que ao aumentar o número de linhas no arquivo, obtinha-se um melhor resultado para qualquer classificador. Os melhores resultados obtidos para cada algoritmo se deu quando utilizou-se um arquivo com 17 mil linhas para treinamento e teste. Os atributos e suas configurações, bem como os resultados obtidos são apresentados em cada tópico a seguir.

3.6.1 - ALGORITMO J48

3.6.1.1 - ESPECIFICAÇÃO DOS PARÂMETROS DISPONÍVEIS PARA O ALGORITMO J48 NA FERRAMENTA WEKA

Seed: Um número randômico usado para reduzir os erros gerados pela poda.

Unpruned: Define se a poda (eliminação de alguma ramificação de menor significância) será utilizada ou não.

ConfidenceFactor: Fator de confiança para poda.

NumFolds: Determina a quantidade de dados que será usada para a redução de erros durante o processo de poda.

NumDecimalPlaces: Quantidade de números após a vírgula.

BatchSize: Quantidade de instâncias para processamento em cada momento.

ReducedErrorPruning: Determina se haverá uma redução de erros usando o algoritmo de poda do J48 ou o do C 4.5¹¹.

UseLaplace: Determina se a contagens de folhas será suavizadas com base no teorema de Laplace¹².

DoNotMakeSplitPointActualValue: Determina se o ponto de divisão não será realocado para um valor de dados real. Isso pode resultar em acelerações substanciais de grandes conjuntos de dados com atributos numéricos.

SubtreeRaising: Determina se deverão ser considerados as sub árvores, aumentando a operação durante a poda.

BinarySplits: Determina se deverão ser usadas divisões binárias em atributos nominais ao construir as árvores.

¹¹ Algoritmo C4.5 Disponível em <http://saiconference.com/Downloads/SpecialIssueNo10/Paper_3-A_comparative_study_of_decision_tree_ID3_and_C4.5.pdf>

¹² Teorema de Laplace Disponível em <http://tutorial.math.lamar.edu/pdf/Laplace_Table.pdf>

DoNotCheckCapabilities: Determina se os recursos do classificador não serão verificados antes de o classificador ser construído.

MinNumObj: O número mínimo de instâncias por folha.

UseMDLcorrection: Determina se a correlação MDL é usada ao encontrar divisões em atributos numéricos.

CollapseTree: Determina se partes da árvore serão removidas.

3.6.1.2 - RESULTADOS

A tabela a seguir demonstra a alteração de parâmetros de entrada para o algoritmo J48, durante os testes executados. Percebeu-se que os atributos que mais impactaram no momento da classificação foram os relativos à taxa de distribuição dos arquivos para treinamento e teste, bem como utilizar ou não a poda. Para o algoritmo J48, diversos testes auxiliares foram executados. A seguir, são demonstrados os 9 que melhor classificam os dados.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8	Teste 9
folds	10	10	10	10	10	10	10	10	10
batchSize	100	100	100	100	100	100	100	100	100
binarySplits	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
collapseTree	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
confidenceFactor	0.25	0.25	0.25	0.21	0.21	0.21	0.23	0.23	0.24
debug	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
doNotCheckCapabilities	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
doNotMakeSplitPointActualValue	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
minNumObj	15	15	66	66	66	60	60	60	60
numDecimalPlaces	2	2	2	2	2	2	2	2	2
numFolds	3	3	3	3	3	3	3	3	3
reducedErrorPruning	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
seed	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
subtreeRaising	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
unpruned	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
useLaplace	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
useMDLcorrection	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
Classificados Corretamente	16,17%	16,45%	17,13%	17,20%	31,18%	31,27%	31,3249 %	31,3881 %	31,4406 %

Tabela 2 - Testes executados com o J48

A utilização pura e simples de um algoritmo de árvore de decisão não se mostrou tão adequada para classificação de gêneros musicais, uma vez que os resultados obtidos com o J48 ficaram na média, abaixo daqueles obtidos com as demais técnicas, todavia, ao aplicar a técnica de poda, percebeu-se uma melhora

significativa nos resultados. Um gráfico com o resultado da classificação realizada pelo algoritmo é mostrado a seguir:

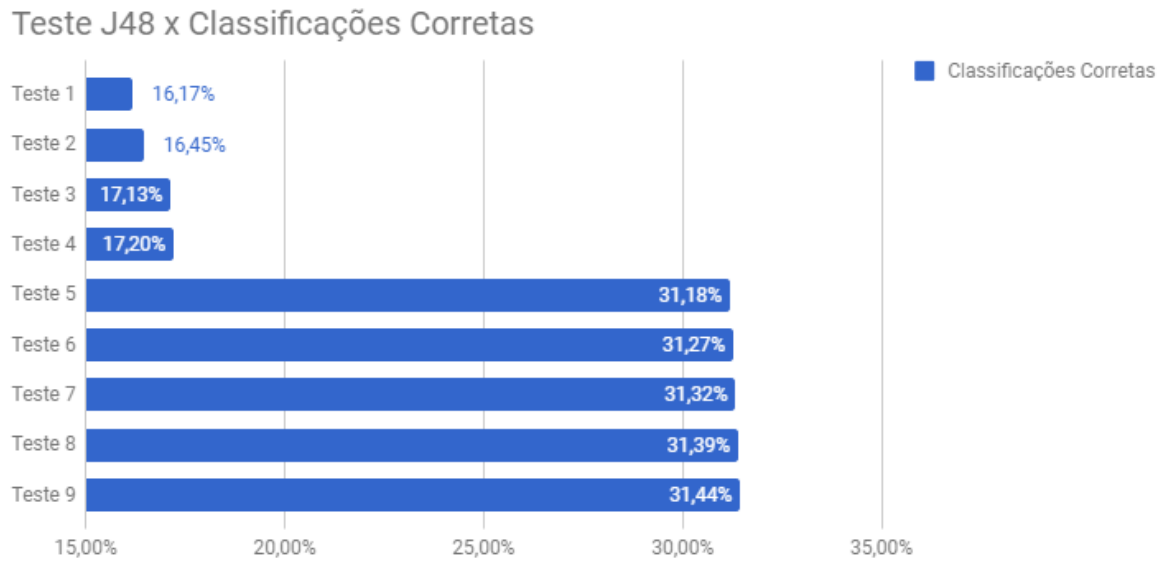


Figura 12 - Melhoria dos resultados do J48

3.6.2 - RANDOM FOREST

3.6.2.1 - ESPECIFICAÇÃO DOS PARÂMETROS DISPONÍVEIS PARA O ALGORITMO RANDOM FOREST NA FERRAMENTA WEKA

Os principais conceitos do Random Forest são o de “bagging” e o método de “subespaço aleatório”. Bagging consiste em escolher uma amostra das observações (aleatoriamente) ao invés de todas elas. O método do subespaço aleatório significa escolher uma amostra das características ao invés de todas elas para agregar à árvore. Se o número de observações é grande, ou seja, o conjunto de dados possui inúmeras amostras e o número de árvores é pequeno, isso significa que algumas observações vão ficar fora da classificação ou vão ser classificadas apenas uma vez. Se o número de características é grande e o número de árvores é pequeno, algumas características serão perdidas.

No WEKA, os atributos disponíveis são listados a seguir.

Seed: Número aleatório a ser utilizado pelo algoritmo

RepresentCopiesUsingWeights: Determina se serão apresentadas cópias das instâncias ao invés de explicitar seus pesos.

StoreOutOfBagPredictions: Determina se o algoritmo irá armazenar as classificações fora das “bags”.

NumExecutionSlots: O número de slots de execução (threads) a serem usados na construção do conjunto.

BagSizePercent: Tamanho de cada bag, como porcentagem do tamanho do conjunto de treinamento.

NumDecimalPlaces: O número de casas decimais a serem usadas.

BatchSize: O número de instâncias a serem processadas se a previsão em lote estiver sendo executada. Mais ou menos instâncias podem ser fornecidas, mas isso dá às implementações a chance de especificar um tamanho de lote preferencial.

NumIterations: O número de iterações a serem executadas.

OutputOutOfBagComplexityStatistics: Determina se o Random Forest deve gerar estatísticas baseadas em complexidade quando a avaliação *calcOutOfBag* for executada.

BreakTiesRandomly: Determina se os loops serão randomicamente quebrados quando muitos atributos trazem o mesmo resultado.

DoNotCheckCapabilities: Determina se os recursos do classificador não serão verificados antes de o classificador ser construído.

MaxDepth: Profundidade máxima para checagem de atributos.

CalcOutOfBag: Determina se o cálculo de erros “fora dos bags” estará presente.

NumFeatures: Determina o randomicamente o número de atributos que serão utilizados pelo classificador.

3.6.2.2 - RESULTADOS

A tabela a seguir demonstra a alteração de parâmetros de entrada para o algoritmo Random Forest, durante os testes executados. Percebeu-se que para o Random Forest, os atributos que mais impactaram no momento da classificação foram os relativos à quantidade de tempo para treinamento, bem como o tamanho da

bag. Também foram executados diversos testes auxiliares e a seguir são apresentados os 9 que melhor classificam os dados.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8	Teste 9
folds	15	20	20	20	10	15	15	20	15
bagSizePercent	80	80	80	80	100	100	80	80	80
batchSize	250	250	250	300	100	200	250	250	250
breakTiesRandomly	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
calcOutOfBag	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
computeAttributeImportance	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
debug	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
doNotCheckCapabilities	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
maxDepth	0	0	0	0	0	0	0	0	0
numDecimalPlaces	2	2	2	2	2	2	2	2	2
numExecutionSlots	1	1	2	1	1	1	1	1	1
numFeatures	0	0	0	0	0	0	0	0	0
numIterations	250	250	250	300	100	200	250	250	250
outputOutOfBagComplexityStatistics	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
printClassifiers	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
seed	1	1	1	1	1	1	1	1	1
storeOutOfBagPredictions	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Classificados Corretamente	27,74%	27,56 %	27,57 %	27,53 %	36,36%	36,25%	36,42 %	36,24 %	37,45%

Tabela 3 - Testes com o Random Forest

O classificador Random Forest demonstrou resultados superiores aos demais, mesmo para as tentativas com arquivos com menos dados. Um gráfico com o avanço da classificação realizada pelo algoritmo é mostrado a seguir:

Testes Random Forest x Classificações Corretas

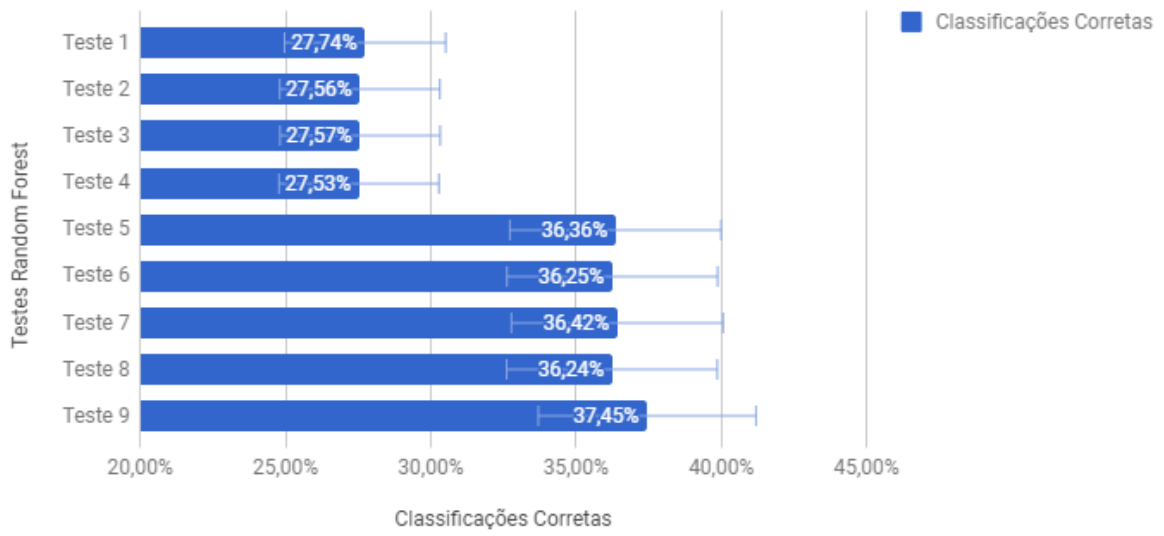


Figura 13 - Resultados do Random Forest

3.6.3 - REDES NEURAIAS MLP - MULTILAYER PERCEPTRON

3.6.3.1 - ESPECIFICAÇÃO DOS PARÂMETROS DISPONÍVEIS PARA O ALGORITMO DE REDES NEURAIAS MLP NA FERRAMENTA WEKA

Para classificação das entradas dos usuários, optou-se por utilizar uma técnica de inteligência artificial chamada de Rede Neural Artificial. Esta técnica, segundo Ferneda (2006), aborda uma metodologia de aprendizagem supervisionada cuja saída depende das entradas e mais elementos dentro do sistema, como uma função de ativação e correções. Segundo o autor, as redes neurais artificiais tentam emular o funcionamento do cérebro humano, que sabidamente faz ligações cognitivas por meio de células do sistema nervoso chamadas de neurônios. Os neurônios captam os estímulos de outras partes do sistema nervoso de tempos em tempos e transmite para o restante de seu corpo, onde são processados.

Neste trabalho, a abordagem de redes neurais utilizada foi, como descrito nos textos acima, multicamada de perceptron com Backpropagation (MLP). Conforme mencionado, redes neurais são um sistema de aprendizagem supervisionada, o que significa que haverá interação com os resultados e avaliação dos mesmos.

Os atributos da rede neural e seus valores de classificação são demonstrados na tabela a seguir:

Seed: Determina o número randômico que será utilizado para os pesos iniciais da rede durante o treinamento

Momentum: “Impulso” aplicado aos pesos durante as atualizações.

HiddenLayers: Define a quantidade e o comportamento das camadas “escondidas” na rede.

ValidationThreshold: Determina a validação do teste. Verifica quantas vezes a classificação um conjunto de atributos piorou antes de treinar.

NormalizeAttributes: Determina se os atributos serão normalizados.

NumDecimalPlaces: Determina o número de casas decimais.

Decay: Determina se a taxa de aprendizagem deverá se ajustar (diminuir) de acordo com os dados.

ValidationSetSize: Determina o tamanho do conjunto de validação

TrainingTime: O número de períodos (épocas) em que a rede treinará.

AutoBuild: Determina se as conexões da rede serão feitas automaticamente.

NormalizeNumericClass: Determina se as classes serão normalizadas nos conjuntos.

LearningRate: Taxa de aprendizagem ou peso

DoNotCheckCapabilities: Determina se os recursos do classificador não serão verificados antes de o classificador ser construído.

Reset: Determina se a rede será reconstruída de acordo com a taxa de aprendizagem atingida pelo conjunto atual.

3.6.3.2 - RESULTADOS

A tabela a seguir demonstra a alteração de parâmetros de entrada para o algoritmo de Redes Neurais MLP. Também foram executados diversos testes auxiliares e a seguir são apresentados os 7 que melhor classificam os dados.

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7
folds	10	15	20	20	20	20	20
autoBuild	1	1	1	1	1	1	1
batchSize	100	100	100	100	100	100	100
debug	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
decay	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
doNotCheckCapabilities	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
hiddenLayers	a	t	t	t	t	a	t
learningRate	0,3	0,4	0,4	0,5	0,4	0,4	0,4
momentum	0,2	0,3	0,3	0,3	0,4	0,4	0,3
normalizeAttributes	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
normalizeNumericClass	1	1	1	1	1	1	1
numDecimalPlaces	2	2	2	2	2	2	2
reset	1	1	1	1	1	1	1
seed	0	0	0	0	0	0	0
trainingTime	500	500	500	500	500	500	500
validationSetSize	0	0	0	0	0	0	0
validationThreshold	20	20	20	20	20	20	20
Classificados Corretamente	16,98 %	18,52 %	18,68 %	17,64 %	17,75 %	17,42 %	33,93 %

Tabela 4 - Testes com MLP

É possível observar que o atributo que mais impacta na classificação da MLP é o tipo de hiddenLayers. No WEKA, Este atributo recebe diferentes valores textuais, cuja implicação será a quantidade e a maneira como essas camadas serão criadas. No WEKA, a letra ‘a’ representa “atributos” para a rede MLP.

Outros atributos, como a taxa de aprendizagem se mostrou determinante para uma melhor classificação, todavia, melhores resultados só foram obtidos ao aumentar

a quantidade de arquivos para treino. Um gráfico com o avanço da classificação executada pelo algoritmo é mostrado a seguir:

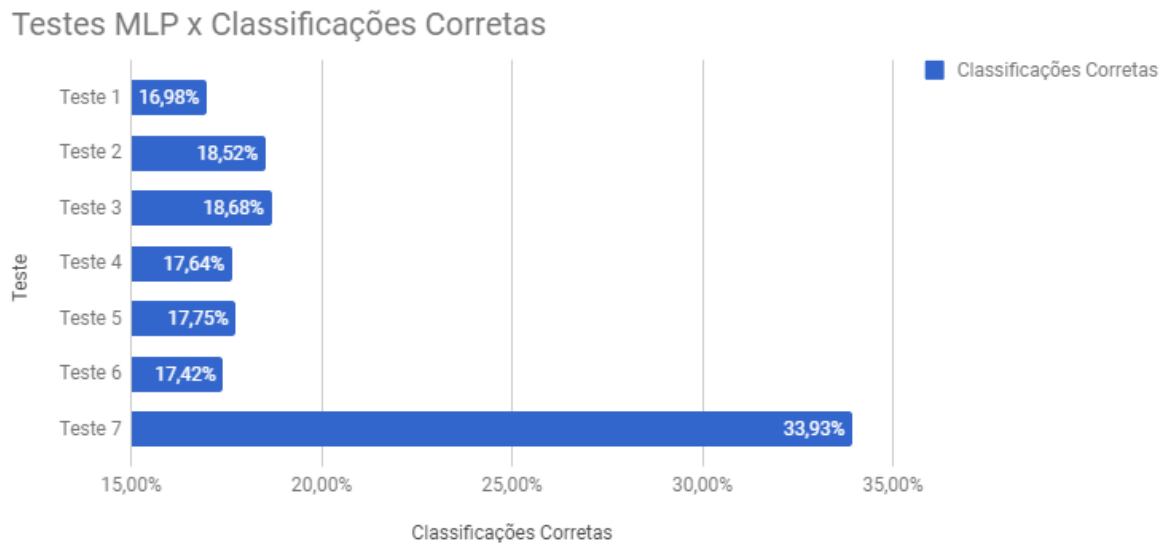


Figura 14 - Resultados MLP

3.7 - AVALIAÇÃO DOS RESULTADOS

O gráfico a seguir mostra os resultados dos testes com cada algoritmo, alinhados em ordem crescente de classificações corretas.

Os resultados ficaram todos abaixo de 50% de classificação correta, o que demonstra que os atributos escolhidos para a execução das técnicas podem não ter sido os melhores. Todavia, é importante lembrar que, para todos os algoritmos, houve uma melhora significativa nos resultados obtidos a partir do momento em que adicionou-se as métricas MFCC ao conjunto de treino. Os resultados que antes permaneciam próximos aos 15% de acerto passaram a registrar de aproximadamente 38% ao adicionar tais métricas.

Percebe-se também que os melhores resultados foram atingidos com o algoritmo Random Forest.

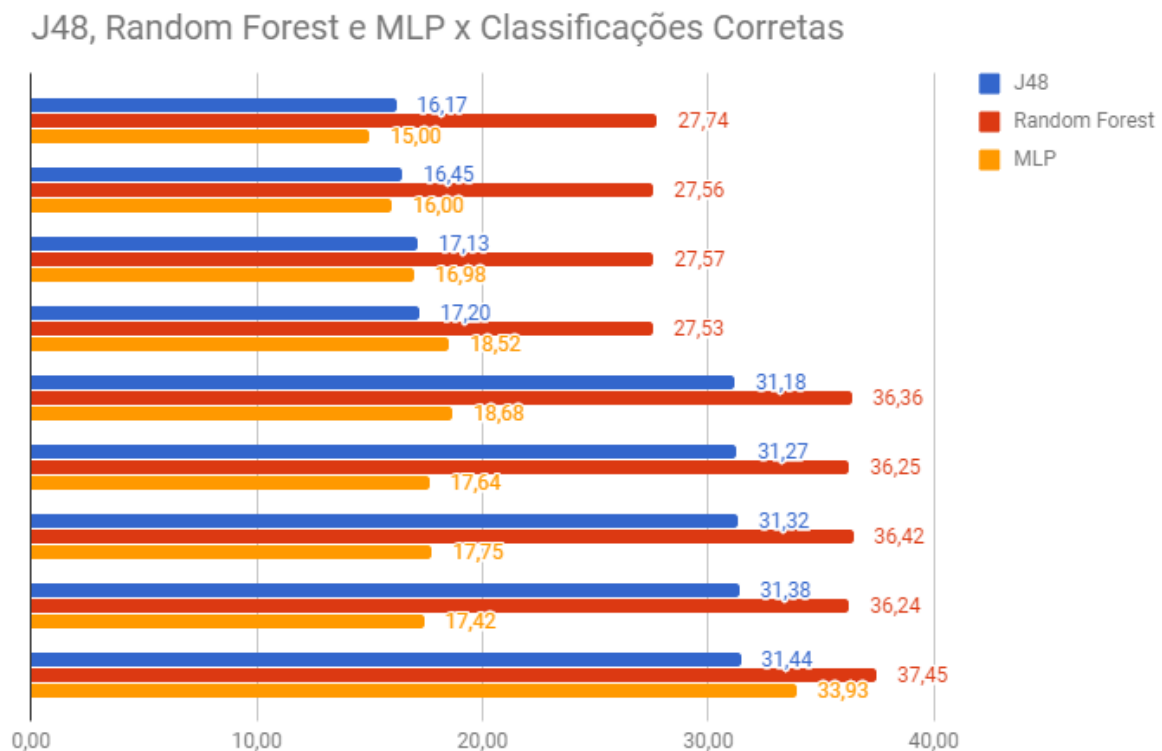


Figura 15 - Comparação entre os algoritmos J48, RF e MLP

4 - CONCLUSÃO

Neste relatório, foram apresentadas a arquitetura e aspectos gerais da ferramenta proposta e seu funcionamento esperado.

Ao término deste trabalho, conclui-se que a principal dificuldade ao abordar temas como computação musical é o desafio de transformar um dado musical em informação.

Dovicchi (2004) afirmou que em que tanto para um indivíduo de determinada cultura como para um músico, a música tem significado bem diverso do significado para um indivíduo não pertencente à esta cultura ou que conheça a ambientação musical. Observando tais apontamentos, é possível inferir que a tarefa de classificação de gêneros, por ser passível de interpretações diferentes por pessoas diferentes torna-se uma tarefa completa. O processo de descoberta de conhecimento em músicas por si só é uma ciência e valendo-se desta área da tecnologia, foi possível classificar dados de entrada musical em informação de gênero. Percebeu-se na literatura, que as métricas ideais para reconhecimento de padrões de gênero não são facilmente aplicáveis na prática, necessitando haver um processamento prévio das saídas de bibliotecas como por exemplo as da essentia.

Observa-se, por exemplo, que os coeficientes MFC são intervalos numéricos de magnitude muito distinta e não são facilmente utilizáveis como entrada para uma técnica de classificação. Utilizar apenas as demais métricas, todavia, não foi uma tentativa interessante. Essas métricas são, medidas primárias e nem tão relacionadas

ao gênero musical, mas ao ritmo da música e, por este motivo, as saídas do programa criado não satisfizeram os objetivos propostos em termos de classificação.

Percebeu-se que ritmo e danceabilidade são características muitas vezes comuns em arquivos de áudio distintos, por este motivo, as saídas observadas no programa foram pouco conclusivas, mas de grande valia para o estudo de semelhanças de características entre gêneros musicais. Ao utilizar ambas as métricas (mfcc e demais), percebeu-se que os resultados melhoraram significativamente, porém não ultrapassando 40% de acerto.

Finalmente, conforme descrito nas seções anteriores, devido a uma questão de limitações de processamento, não foi possível utilizar mais do que 60 segundos como amostra para o momento da extração das informações. Aumentar esse número poderia melhorar consideravelmente os resultados, tendo em vista que as informações obtidas neste cenário seriam potencialmente mais abrangentes do que aquelas utilizadas neste trabalho.

Outro fator decisivo seria aumentar o número de atributos para classificação, uma vez que a essentia disponibiliza diversas outras informações sobre as músicas. Utilizar outros atributos para classificação poderia trazer melhores resultados na etapa de classificação.

4.1 - TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se utilizar mais atributos extraídos por ferramentas de MIR como entrada para os algoritmos de classificação, além de utilizar outras ferramentas para MIR.

Um possível algoritmo de classificação com potencial tão grande quanto os utilizados é SVM (Support Vector Machines). Este algoritmo demonstra grande capacidade de classificar dados nem tão correlacionados.

Recomendam-se também abordagens probabilísticas, que talvez tragam resultados complementares ao tema proposto seriam redes bayesianas, cujos resultados mostrariam o grau de pertencimento de cada música a cada gênero e sistemas *fuzzy*.

Por fim, recomenda-se também extrair as informações dos arquivos utilizando outros parâmetros, como a quantidade de tempo. Conforme descrito nas seções anteriores, neste trabalho, utilizou-se apenas 60 segundos como parâmetro para MIR, todavia, esta foi uma limitação imposta pelo hardware.

REFERÊNCIAS BIBLIOGRÁFICAS

SANTINI, R. M.; SOUZA, R. F. **Classificação colaborativa de conteúdos não-textuais na internet: as novas formas de mediação e organização da informação da música através da folksonomia.** Encontro Nacional de Pesquisa em Ciência da Informação, v. 11, 2010.

ALMEIDA, Maria Christina Barbosa de. **A informação em museus de arte: de unidades isoladas a sistema integrado.** *Musas: Revista Brasileira de Museus e Museologia*, Rio de Janeiro, v.2, n.2, p.140-154, 2006.

HU, XIAO and J. Stephen Downie. **Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata.** ISMIR, 2007.

SOARES, Isaque Sales. **Computação musical: O Reflexo do Avanço Tecnológico no Mundo Fonográfico.** Minas Gerais. UNIPAC, 2016.

BARROS, Camila Monteiro de. **Representação da informação musical: subsídios para recuperação da informação em registros sonoros e partituras no contexto educacional e de pesquisa.** *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, Florianópolis, v. 17, n. 35, dez. 2012. ISSN 1518-2924, 2018.

GROVE, George; SADIE, Stanley (Ed.). **Dicionário Grove de música: edição concisa.** Rio de Janeiro: Jorge Zahar, 1994.

MILETTO, L. L. Costalonga, L. V. Flores, E. F. Fritsch, M. S. Pimenta e R. M. Vicari. **Introdução à Computação Musical.** Rio Grande do Sul: UFRGS, 2004.

CORRÊA, Débora Cristina. **Inteligência artificial aplicada à análise de gêneros musicais.** 2012. Tese (Doutorado em Física Aplicada) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2012.

DOMINGOS, J. P. Silva, **Algoritmos de Classificação baseados em Análise Formal de Conceitos,** Minas Gerais, UFMG, 2007.

Carlos D. R. Cuadros, **Comparação entre as técnicas de MFCC e ZCPA para Reconhecimento robusto de locutor em ambientes ruidosos,** Rio de Janeiro: UFF, 2007.

Rajani, Mohit **Supervised Learning in Genre Classification**, Stanford University, and CS229: Machine Learning, 2009.

A. P. Klapuri, **Multiple fundamental frequency estimation based on harmonicity and spectral smoothness**, IEEE Trans. Speech and Audio Proc., 11(6), 804–816, 2003.

FERNEDA, Edberto. **Redes neurais e sua aplicação em sistemas de recuperação de informação**. Ci. Inf., Brasília, v. 35, n. 1, p. 25-30, 2006.

BORSATO, Dionísio et al. **Aplicação de redes neurais artificiais na identificação de gasolinas adulteradas comercializadas na região de Londrina - Paraná**. Quím. Nova, São Paulo, v. 32, n. 9, p. 2328-2332, 2009.

COSTA, Miguel Antonio Bueno da. **Uma abordagem sobre Inteligência Artificial e simulação, com uma aplicação na pecuária de corte nacional**. Prod., São Paulo, v. 2, n. 1, p. 51-59, 1992.

Santini, Rosie, **RECUPERAÇÃO DA INFORMAÇÃO DE MÚSICA E A CIÊNCIA DA INFORMAÇÃO**, VIII ENANCIB: Bahia, 2007.

Dmitry Bogdanov et al, **ESSENTIA: AN AUDIO ANALYSIS LIBRARY FOR MUSIC INFORMATION RETRIEVAL**, Paraná, ISMIR 2013.

Robert E. Shannon, Richard Mayer, Heimo H. Adelsberger, **Expert systems and simulation Expert systems and simulation**, Volume: 44 issue: 6, page(s): 275-284 Texas A&M University, Texas, 1985.

F. ROSENBLATT, **THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION**, Psychol. Review. 65. 386 - 408. 10.1037/h0042519, 1958.

WITTEN, I. H.; FRANK, Eibe. **Data Mining: Practical Machine Learning Tools and Techniques**. San Francisco: Morgan Kaufmann p 525, 2005.

Abdollahi, Yadollah & Zakaria, Azmi & Abbasiyannejad, Mina & Fard Masoumi, Hamid Reza & Ghaffari Moghaddam, Mansour & Matori, Khamirul & Jahangirian, Hossein (kamran & Keshavarzi, Ashkan. **Artificial neural network modeling of p-cresol photodegradation**. Chemistry Central journal. 7. 96. 10.1186/1752-153X-7-96, 2014

BREIMAN, L. Machine Learning. 45: 5. <https://doi.org/10.1023/A:1010933404324>, 2001.

C4.5: **Programs for Machine Learning** by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993.

Télvio Orrú, **Redes Neurais Artificiais para Reconhecimento e Classificação de Padrões**, São Paulo. UNICAMP, 2005

RICH, E., **Artificial Intelligence**, McGraw-Hill, N.Y., 1983.

SHRINIVASAN, Suhas, **Python scripts to do music genre classification**, Mysore, Karnataka, 2016. Disponível em <<https://github.com/suhasshrinivasan/music-genre-classification>>. Acesso em 30 de junho de 2018

LACHMISH, Matan, **Classify music genre from a 10-second sound stream using a Neural Network**, Israel, 2016. Disponível em <<https://github.com/mlachmish/MusicGenreClassification>>. Acesso em 30 de junho de 2018

SINGH, Jasdev, **GenreXpose Automated music genre classification using machine learning**, New Delhi, India, 2014. Disponível em <<https://github.com/suhasshrinivasan/music-genre-classification>>. Acesso em 30 de junho de 2018

Abraham, G. **The Tradition of Western Music**, Oxford University Press, Ely House, London: Ebener Baylis & Son Ltd., 1974

APÊNDICES

APÊNDICE 1 - PERFIL UTILIZADO PARA EXTRAÇÃO

startTime: 0

endTime: 60

outputFrames: 0

outputFormat: json

requireMbid: false

indent: 4

lowlevel:

frameSize: 2048

hopSize: 1024

zeroPadding: 0

windowType: blackmanharris62

silentFrames: noise

stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]

average_loudness:

frameSize: 88200

hopSize: 44100

windowType: hann

silentFrames: noise

rhythm:

method: degara

minTempo: 40

maxTempo: 208

stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]

tonal:

frameSize: 4096

hopSize: 2048

zeroPadding: 0

windowType: blackmanharris62

silentFrames: noise

stats: ["mean", "var", "median", "min", "max", "dmean", "dmean2", "dvar", "dvar2"]

APÊNDICE 2 - EXEMPLO DE ARQUIVO USADO PARA TREINAMENTOS

"average_loudness";"bpm";"beats_loudness_mean";"danceability";"chords_changes_rate";"chords_number_rate";"mfcc_1";"mfcc_2";"mfcc_3";"mfcc_4";"mfcc_5";"mfcc_6";"mfcc_7";"mfcc_8";"mfcc_9";"mfcc_10";"mfcc_11";"mfcc_12";"mfcc_13";"genre"
"0.925303399563";"91";"0.0208109132946";"0.348432";"0.0618716180325";"0.0100541375577";"-628.80480957";"103.486824036";"-36.6466217041";"16.6600952148";"-2.41294765472";"0.960262715816";"4.67348384857";"1.46282374859";"0.336569160223";"0.36273470521";"-3.41569423676";"-0.695533156395";"-4.82358360291";"hardcore"
"0.965685427189";"92";"0.0336442142725";"0.421603";"0.0920340269804";"0.0100541375577";"-594.906677246";"91.1509399414";"-17.8723564148";"29.9439411163";"7.62147665024";"4.96801519394";"3.45501685143";"5.35736179352";"1.89638650417";"3.75619101524";"2.65643143654";"1.34139060974";"0.558651745319";"hardcore"
"0.968143761158";"172";"0.0280781947076";"0.383275";"0.088940449059";"0.00618716143072";"-595.464416504";"86.6218414307";"-1.79106676579";"31.604675293";"18.8286933899";"8.31268787384";"-3.67880654335";"2.67287826538";"0.75805491209";"-1.69551610947";"-3.61278796196";"-4.06345129013";"-2.43477249146";"hardcore"
"0.941324293613";"119";"0.0279283281416";"0.397213";"0.0572312436998";"0.00850734766573";"-674.634277344";"142.71836853";"-20.1995353699";"43.7065963745";"19.6010704041";"-5.82082462311";"13.6708459854";"9.37569713593";"1.63172149658";"6.78490018845";"2.77630496025";"1.79057753086";"-3.91092681885";"hardcore"
"0.941201806068";"155";"0.0301310382783";"0.473868";"0.0966743975878";"0.00773395225406";"-616.728210449";"78.365814209";"0.513355016708";"33.4414215088";"16.5326480865";"8.96598911285";"3.88464736938";"7.57724475861";"0.826829135418";"3.23369431496";"2.34550857544";"2.71785950661";"-0.965415894985";"hardcore"

APÊNDICE 3 - EXTRATOR DE INFORMAÇÕES

```
# extractor.py

from __future__ import print_function

import os

import argparse

import json

import sys

import tcc_essentia

OUTPUT_FOLDER = os.path.abspath("./results")

def main():

    """main"""

    desc = "This module reads the information from given folder."

    parser = argparse.ArgumentParser(description=desc)

    print("\n[ TCC - TRACK EXTRACTOR ]\n%s\n" % desc)

    parser.add_argument('-i', '--input_folder')

    parser.add_argument('-o', '--output_folder', default=OUTPUT_FOLDER)

    parser.add_argument('-p', '--profile', default="profile.yml")

    parsed = parser.parse_args()

    walk_in_folders(parsed.input_folder, parsed.output_folder)

def get_path(a, b):

    return os.path.abspath(os.path.join(a, b))

def walk_in_folders(input_folder, output_folder):

    for root, _dirs, files in os.walk(input_folder):

        for file_name in files:

            if ".mp3" in file_name:

                path = get_path(root, file_name)

                tcc_essentia.run_esem(

                    path,

                    "profile.yml",
```

```
output_folder
```

```
)
```

```
if __name__ == "__main__":
```

```
    try:
```

```
        main()
```

```
    except KeyboardInterrupt:
```

```
        print("done.")
```

APÊNDICE 4 - INTERFACE ENTRE O SIS. OPERACIONAL E ESSENTIA

```
# tcc_essentia.py

from __future__ import print_function
import os
import subprocess
import hashlib

def log_message(msg, level=-1):
    """Log message preceded by the log level"""

    header = "INFO" if level < 0 else "WARNING" if level == 0 else "ERROR"
    print("\n[ {} ]\n{}".format(header, msg))

def get_hash(a_string, concat_after=""):
    """Return MD5 of string and concat something after"""
    return "{}{}".format(
        hashlib.md5(a_string.encode('utf-8')).hexdigest(),
        concat_after
    )

def run_rim(command_name, music, output_folder, output_format, profile=""):
    """Run ESEM with args"""

    log_message("\nExtracting '{}' From: '{}' \n".format(command_name, music))

    output_file = os.path.join(output_folder, get_hash(music, output_format))

    if not os.path.exists(output_file):
```

```

    if profile != "":
        command = [command_name, music, output_file, profile]
    else:
        command = [command_name, music, output_file]

    subprocess.call(command)
    print("\nDone.")
    return output_file
def run_mfcc(music, output_folder):
    """run essentia_streaming_mfcc"""
    return run_rim(
        "essentia_streaming_mfcc",
        music,
        output_folder,
        ".yml")
def run_esem(music, profile, output_folder):
    """run essentia_streaming_mfcc"""
    return run_rim(
        "essentia_streaming_extractor_music",
        music,
        output_folder,
        ".json",
        profile
    )

if __name__ == "__main__":
    try:
        print("Import this file.")
    except KeyboardInterrupt:
        print("Bye.")

```


APÊNDICE 5 - CRIADOR DE ARQUIVOS DE TREINO E TESTE

```
// train-test.js

const fs = require('fs');

const process = require('process');

const resolvePath = require('path').resolve

const mysql = require('mysql');

const connection = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: 'YOUR_PASSWORD',

  database: 'mir'

});

connection.connect();

const MAX = process.argv[2] ? parseInt(process.argv[2]) : 1000;

const OUTPUT_FILE = resolvePath("./results", `train_${MAX}.csv`);

const HEADERS = [

  "average_loudness",

  "bpm",

  "beats_loudness_mean",

  "danceability",

  "chords_changes_rate",

  "chords_number_rate",

  ...[...Array(13).keys()].map(n => `mfcc_${n + 1}`),

  "genre"

];

const HEADER = `${HEADERS.map(he => `${he}`).join(";")}\n`;

const appendGenreSetToTrain = (super_genre) => {

  const QUERY = `
```

```

SELECT
    average_loudness,
    bpm_int AS bpm,
    beats_loudness_mean,
    (SELECT REPLACE(danceability_d, ',', '.')) AS danceability,
    chords_changes_rate,
    chords_number_rate,
    mfcc,
    super_genre AS genre
FROM
    music
WHERE
    super_genre = '${super_genre}'
LIMIT ${MAX};`;

```

```

connection.query(QUERY, (error, results, fields) => {
    if (error) { console.log(error.message); }

    let out = '';
    results.forEach(result => {
        const parsedMFCC = result.mfcc.split(";");
        const resultList = [
            result.average_loudness,
            result.bpm,
            result.beats_loudness_mean,
            result.danceability,
            result.chords_changes_rate,
            result.chords_number_rate,
            ...parsedMFCC,
            result.genre,

```

```

        ];

        out += `${resultList.map(el => `"${el}"`).join(";")}\n`;
    });

    fs.appendFileSync(OUTPUT_FILE, out);
    console.log(super_genre);
});
}

const init = (id, parsedFile, index) => {
    const QUERY = `SELECT DISTINCT super_genre from music`;
    connection.query(QUERY, (error, results, fields) => {
        if (error) { console.log(error.message); }
        results.forEach((result) => {
            appendGenreSetToTrain(result.super_genre);
        });
    });
};

fs.writeFile(OUTPUT_FILE, '', () => {
    fs.appendFileSync(OUTPUT_FILE, HEADER);
    init();
});

```

Classificação Automática de Gêneros Musicais Utilizando Técnicas de Inteligência Artificial

Emerson Demetrio Plácido

Departamento de Informática e Estatística –

Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil

emer.demetrio@gmail.com

Abstract. *This article describes the processes of Music Information Retrieval (MIR) and Artificial Intelligence (AI) algorithms applied in order to classify musical genres. Using information extracted from music files for model construction and training, as well as for tests. Also shown are the results of the classifications and which algorithms were most suitable for the defined objective.*

Resumo. *O presente artigo descreve os processos de Recuperação da Informação Musical (RIM) e algoritmos de Inteligência Artificial (IA) aplicados a fim de classificar gêneros musicais. Utilizando as informações extraídas dos arquivos de música para construção e treinamento do modelo, bem como para testes. São mostrados também os resultados das classificações e quais algoritmos se mostraram mais adequados para o objetivo definido.*

1. Introdução

A tarefa de classificação de gêneros musicais é um tema discutido no meio acadêmico e na indústria. Novos gêneros surgem todos os dias, sejam gêneros originais ou derivados de outros gêneros. No presente artigo, discute-se uma análise sobre algoritmos de extração da informação musical e de IA, a fim de classificar o gênero de um arquivo de áudio (música). O principal desafio abordado foi o de obter as informações sobre faixas de áudio, de maneira que fosse possível comparar arquivos distintos em busca de suas semelhanças.

Uma vez obtidas tais informações, os próximos processos foram: classificar os gêneros dos arquivos utilizando IA, demonstrar os resultados obtidos e, por fim, melhorar o modelo construído com novos padrões reconhecidos - observando que algoritmos de classificação são capazes de aperfeiçoar suas próximas saídas através de resultados prévios, reutilizados como entrada nas classificações. A figura a seguir demonstra cada etapa executada neste trabalho.

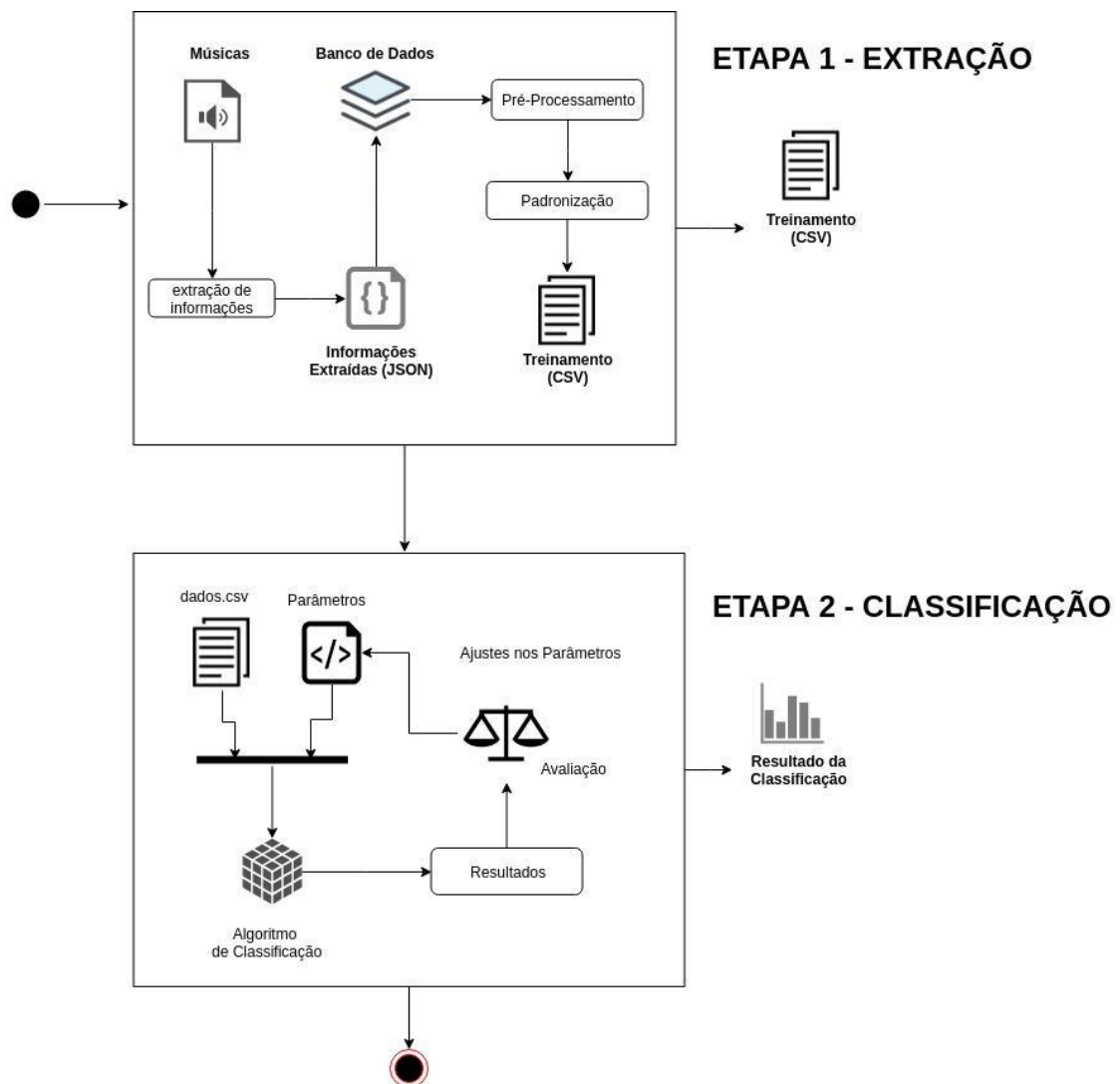


Figura 1 - etapas do trabalho abordado neste artigo

2. Recuperação da Informação Musical – RIM

O conceito de recuperação da informação musical tem sido amplamente discutido por diversos autores de Inteligência Artificial. Para Miletto (2004) RIM faz estudos aprofundados de métodos, técnicas e algoritmos para processamento e geração de som e música, bem como representações digitais e armazenamento de informação sonora e musical, trazendo sempre como premissa de que a informação tratada é arte - o que diferencia a computação musical de outras áreas similares, tais como computação gráfica.

Santini e Souza (2007) abordam RIM como uma interação complexa entre as diversas facetas da música (tempo, harmonia, timbre, frequência, editoria, entre outras), evidenciando um dos problemas recorrentes desta técnica, a questão da diversidade representacional, uma vez que a música pode ser originária de um arquivo de áudio, partitura ou ainda, informações presentes em metadados.

3. Inteligência Artificial – IA

Para BUENO (1991), existem muitas definições para Inteligência Artificial, algumas filosóficas, outras mais pragmáticas, dependendo muito do campo de atuação dos autores das mesmas. Para E. Rich (1983) inteligência artificial é o estudo de como fazer computadores realizarem tarefas

para as quais, até o momento, “o homem faz melhor”. Para Shannon (1985) a IA atua em dois grandes campos: o primeiro deles, limitação das habilidades humanas (visão, fala, gestos, etc.) e o segundo como sendo a duplicação de resultados estabelecidos pelo homem através de sua habilidade e/ou experiência.

4. Classificação de Gêneros

Classificação, segundo Domingos (2007), é uma técnica de IA caracterizada por ter entre seus arquivos de treinamento e teste um atributo chamado de classe. Este atributo é, normalmente, o objetivo esperado quando utiliza-se esta técnica. Existem diversos classificadores na literatura. As abordagens utilizadas neste trabalho foram as redes neurais e árvores de decisão (puras e derivadas), utilizadas para classificar o gênero presente no conjunto de informações extraídas de cada música, valendo-se de informações de outros arquivos de áudio previamente extraídas e inseridas nos conjuntos de treinamento.

É sabido que cada algoritmo tem suas peculiaridades, bem como diferentes parâmetros de entrada. Neste projeto, utilizaram-se os algoritmos descritos a seguir, em diversos testes. Nestes testes, alteraram-se parâmetros de cada algoritmo a fim de obter um melhor resultado de classificação.

4.1. Algoritmo J48

O J48 surgiu da necessidade de recodificar seu precursor, o algoritmo C4.5 (que, originalmente é escrito na linguagem C) para a linguagem Java, tendo a finalidade de gerar uma árvore de decisão, baseada em um conjunto de dados de treinamento, sendo este modelo utilizado para classificar as instâncias em um outro conjunto de testes. Proposto por QUINLAN (1993), o J48 apresenta um bom resultado para a montagem de árvores de decisão, a partir de um conjunto de dados de treinamento (WITTEN, et al., 2005).

4.2. Algoritmo Random Forest

Random Forest (Floresta Randômica) é um modelo de classificação criado por Leo Breiman, descrito por (BREIMAN, 2001) como uma combinação de árvores de decisão, sendo que cada árvore é dependente de valores aleatórios amostrados independentemente e com a mesma distribuição para todas as árvores na “floresta”.

4.3. Redes Neurais – MLP

Segundo Ferneda (2006), as redes neurais artificiais diferenciam-se entre si pela sua arquitetura e pela forma como os pesos associados às conexões são ajustados durante o processo de aprendizado. Para que o processo de classificação de padrões tenha sucesso, é necessário que exista o mapeamento dos padrões de entrada em um espaço de classificação. Este espaço é definido por regiões de decisão que separam os padrões, atribuindo cada padrão em uma classe. Neste trabalho, foram utilizadas redes neurais de aprendizagem supervisionada com multicamadas (MLP).

5. Resultados

De maneira geral, este trabalho consistiu-se em obter determinadas características das músicas analisadas a fim de utilizá-las como entrada para um algoritmo de classificação. Optou-se por fazer uso da biblioteca *essentia*¹³ por conta da facilidade que a mesma proporciona para configurar a saída esperada e também para extrair informações dos arquivos de áudio.

Para analisar os resultados de cada algoritmo de classificação, optou-se por utilizar ferramenta *WEKA*¹⁴, que é uma biblioteca com diversos algoritmos de inteligência artificial, por conta da familiaridade do autor para com a mesma e também por tratar-se de uma ferramenta livre.

5.1. J48

Com o algoritmo J48, percebeu-se que os atributos que mais impactaram no momento da classificação foram os relativos à taxa de distribuição dos arquivos para treinamento e teste, bem como utilizar ou não a poda. Para o algoritmo J48, diversos testes auxiliares foram executados.

A utilização pura e simples de um algoritmo de árvore de decisão não se mostrou tão adequada para classificação de gêneros musicais, uma vez que os resultados obtidos com o J48 ficaram na média, abaixo daqueles obtidos com as demais técnicas, todavia, ao aplicar a técnica de poda, percebeu-se uma melhora significativa nos resultados.

5.2. Random Forest

Percebeu-se que para o Random Forest, os atributos que mais impactaram no momento da classificação foram os relativos à quantidade de tempo para treinamento, bem como o tamanho da bag. Também foram executados diversos testes auxiliares com este algoritmo. O classificador Random Forest demonstrou resultados superiores aos demais, mesmo para as tentativas com arquivos com menos dados.

5.3. Redes Neurais MLP

Neste trabalho, a abordagem de redes neurais utilizada foi, como descrito acima, multicamada de perceptron com Backpropagation. Redes neurais são um sistema de aprendizagem supervisionada, o que significa que haverá interação com os resultados e avaliação dos mesmos. Foi possível observar que o atributo que mais impacta na classificação da MLP é o tipo de hiddenLayers. No *WEKA*, Este atributo recebe diferentes valores textuais, cuja implicação será a quantidade e a maneira como essas camadas serão criadas. No *WEKA*, a letra ‘a’ representa “atributos” para a rede MLP.

6. Avaliações, Conclusões e Trabalhos Futuros

A imagem a seguir demonstra os resultados das classificações obtidas com os 3 algoritmos durante os testes com melhores resultados. Conforme mencionado, os diversos testes foram executados com a alteração de cada parâmetro em cada algoritmo.

¹³ *ESSENTIA* - Disponível em <http://essentia.upf.edu>

¹⁴ *WEKA* - Disponível em <https://www.cs.waikato.ac.nz/ml/weka>

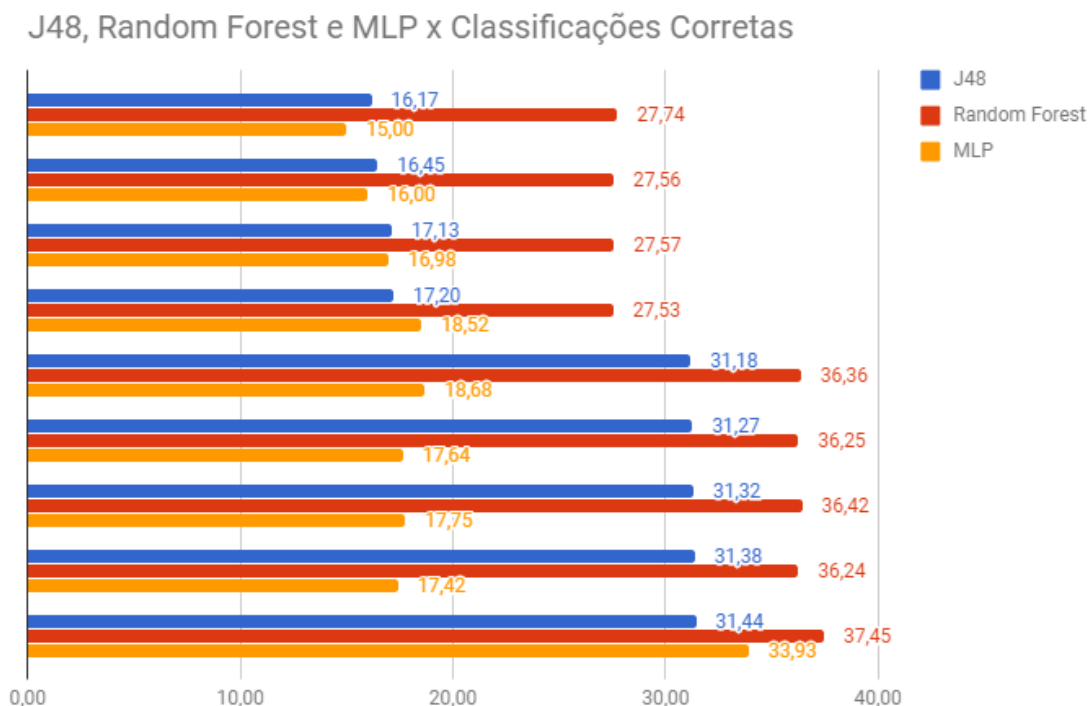


Figura 2 - Resultados de 9 testes com os 3 algoritmos

O gráfico acima mostra os resultados dos testes com cada algoritmo, alinhados em ordem crescente de classificações corretas. Os resultados ficaram todos abaixo de 50% de classificação correta, o que demonstra que os atributos escolhidos para a execução das técnicas podem não ter sido os melhores. Todavia, é importante lembrar que, para todos os algoritmos, houve uma melhora significativa nos resultados obtidos a partir do momento em que adicionou-se as métricas MFCC ao conjunto de treino.

Os resultados que antes permaneciam próximos aos 15% de acerto passaram a registrar de aproximadamente 38% ao adicionar tais métricas. Percebe-se também que os melhores resultados foram atingidos com o algoritmo Random Forest. Conclui-se que a principal dificuldade ao abordar temas como computação musical é o desafio de transformar um dado musical em informação.

O processo de descoberta de conhecimento em músicas por si só é uma ciência e valendo-se desta área da tecnologia, foi possível classificar dados de entrada musical em informação de gênero. Percebeu-se através da literatura que as métricas ideias para reconhecimento de padrões de gênero não são facilmente aplicáveis na prática, necessitando haver um processamento prévio das saídas de bibliotecas como por exemplo as da essentia.

Finalmente, denota-se que, devido a uma questão de limitações de processamento, não foi possível utilizar mais do que 60 segundos como amostra para o momento da extração das informações. Aumentar esse número poderia melhorar consideravelmente os resultados, tendo em vista que as informações obtidas neste cenário seriam potencialmente mais abrangentes do que aquelas utilizadas neste trabalho. Outro fator decisivo seria aumentar o número de atributos para classificação, uma vez que a essentia disponibiliza diversas outras informações sobre as músicas. Utilizar outros atributos para classificação poderia trazer melhores resultados na etapa de classificação.

Como trabalhos futuros, sugere-se utilizar mais atributos extraídos por ferramentas de MIR como entrada para os algoritmos de classificação, além de utilizar outras ferramentas para MIR. Um possível algoritmo de classificação com potencial tão grande quanto os utilizados é SVM (Support Vector Machines). Este algoritmo demonstra grande capacidade de classificar dados nem tão

correlacionados. Por fim, recomenda-se também extrair as informações dos arquivos utilizando outros parâmetros, como a quantidade de tempo.

7. Referências

MILETTO, L. L. Costalonga, L. V. Flores, E. F. Fritsch, M. S. Pimenta e R. M. Vicari. Introdução à Computação Musical. Rio Grande do Sul: UFRGS, 2004.

SANTINI, R. M.; SOUZA, R. F. Classificação colaborativa de conteúdos não-textuais na internet: as novas formas de mediação e organização da informação da música através da folksonomia. Encontro Nacional de Pesquisa em Ciência da Informação, v. 11, 2010.

COSTA, Miguel Antonio Bueno da. “Uma abordagem sobre Inteligência Artificial e simulação, com uma aplicação na pecuária de corte nacional.” Prod., São Paulo, v. 2, n. 1, p. 51-59, 1992.

RICH, E., “Artificial Intelligence”, McGraw-Hill, N.Y., 1983.

Robert E. Shannon, Richard Mayer, Heimo H. Adelsberger, Expert systems and simulation Expert systems and simulation, Volume: 44 issue: 6, page(s): 275-284
Texas A&M University, Texas, 1985.

DOMINGOS, J. P. Silva, Algoritmos de Classificação baseados em Análise Formal de Conceitos, Minas Gerais, UFMG, 2007.

C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993.

WITTEN, I. H.; FRANK, Eibe. Data Mining: Practical Machine Learning Tools and Techniques. San Francisco: Morgan Kaufmann p 525, 2005.

BREIMAN, L. Machine Learning. 45: 5. 2001.