UNIVERSIdade FEEVALE

NATAN PORTILHO DA SILVA

CLASSIFICAÇÃO AUTOMÁTICA DE LETRAS DE MÚSICAS NA LÍNGUA PORTUGUESA

Novo Hamburgo

2018

NATAN PORTILHO DA SILVA

CLASSIFICAÇÃO AUTOMÁTICA DE LETRAS DE MÚSICAS NA LÍNGUA PORTUGUESA

Trabalho de Conclusão de Curso

apresentado como requisito parcial

à obtenção do grau de Bacharel em

Ciência da Computação pela

Universidade Feevale

Orientador: Rodrigo Rafael V. Goulart

Novo Hamburgo

2018

Agradecimentos

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

À minha família por sempre estar ao meu lado e me apoiar em minhas escolhas.

Ao meu orientador por toda dedicação empenhada para tornar este trabalho possível.

Resumo

Com a crescente quantidade de músicas disponíveis online através da popularização de serviços de *streaming*, a possibilidade de pesquisar músicas e montar listas de execução que reflitam o estado de espírito de usuários tem se mostrado de grande importância. Técnicas de classificação podem ser utilizadas para classificar letras de músicas em categorias como ‘triste’ e ‘feliz’. Este trabalho propõem uma pesquisa a respeito da aplicação de técnicas de classificação para categorizar músicas em língua portuguesa nas categorias “positiva” e “negativa”. Os resultados obtidos pelo estudo mostram que o Algoritmo LibSVM foi aquele que apresentou os melhores resultados, com uma taxa de acerto total de 58,38%. Além disso, os resultados indicam que as músicas na classe “negativa” são mais fáceis de se classificar do que as músicas na classe “positiva”. Além do LibSVM, são apresentados os resultados obtidos com: Naive Bayes, SMO, KNN, J48 e K-means.

Palavras-chave: Processamento da Linguagem Natural, Classificação de Textos. Música. Análise de Sentimentos. Streaming. Língua Portuguesa.

Abstract

With the huge quantity of online music content and the success of music streaming services, the desire of making our own playlists based on our mood or state of mind is in high demand. Text classification methods can be used to classify music in categories like “sad” and “happy”. This research aims to apply text classification techniques to classify music in Portuguese language in classes that represent its polarity. The study shows us that the LibSVM algorithm performed the best, correctly classifying 58.38% of the documents. Besides that, the results show us that songs of the class “negative” are easier to classify than those of the class “positive”. We also present the result for the following algorithms: Naïve Bayes, SMO, KNN, J48 and K-means.

Key words: Natural Language Processing, Text Classification. Music. Sentiment Analysis. Streaming. Portuguese language.

**LISTA DE FIGURAS**

[Figura 1 – Modelo de Russel com 28 palavras 13](#_Toc501348982)

[Figura 2 – Fórmula do idf 21](#_Toc501348983)

[Figura 3 – Fórmula do TF-IDF 21](#_Toc501348984)

[Figura 4 – Algoritmo de Oregon 22](#_Toc501348985)

[Figura 5 – Processo do estudo 25](#_Toc501348985)

[Figura 6 – Filtro utilizando três bases de dados 28](#_Toc501348985)

[Figura 7 – Chamanda para API do Vagalume 29](#_Toc501348985)

[Figura 8 – Exemplo de resposta da API do Vagalume 29](#_Toc501348985)

[Figura 9 – Músicas classificadas 31](#_Toc501348985)

[Figura 10 – Porcentagem de músicas por classe 32](#_Toc501348985)

[Figura 11 – Porcentagem de letras positivas e negativas 33](#_Toc501348985)

LISTA DE quadros

[Quadro 1 – Classes Propostas 13](#_Toc500242172)

[Quadro 2 - Classes encontradas usando HWA, HSL e HSW 14](#_Toc500242173)

[Quadro 3 – Performance dos Sistemas 15](#_Toc500242174)

[Quadro 4 - Precisão de Classificação 16](#_Toc500242175)

Quadro 5 - Precisão de Classificação por Emoções (K-means)..................................................16 Quadro 6 - Taxa de acerto com LibSVM...................................................................................17 Quadro 7 - Resultados de taxa de acerto com os diferentes classificadores................................18 Quadro 8 - Impacto do idioma na classificação..........................................................................18 Quadro 9 - Palavras negativas e positivas..................................................................................27 Quadro 10 - Exemplos de valores de TF-IDF.............................................................................34 Quadro 11 - Palavras com *stemização*.......................................................................................35 Quadro 12 - TF-IDF após *stemização*........................................................................................35 Quadro 13 - Modelo final para aprendizado de máquina............................................................36 Quadro 14 - Matriz de confusão do LibSVM.............................................................................38 Quadro 15 - *F-measure* por classe no LibSVM..........................................................................39 Quadro 16 - Matriz de confusão do SMO...................................................................................39

Quadro 17- *F-measure* por classe no SMO................................................................................40

Quadro 18 - Matriz de confusão do Naive Bayes.......................................................................40

Quadro 19 - *F-measure* no Naive Bayes....................................................................................40

Quadro 20 - Matriz de confusão do KNN...................................................................................41

Quadro 21 - Resultados de *f-measure* com KNN.......................................................................41

Quadro 22 - Matriz de confusão do J48......................................................................................42

Quadro 23 -Resultados de *F-measure* com J48..........................................................................42 Quadro 24 - Matriz de confusão com K-means..........................................................................42 Quadro 25 - Comparação entre resultados.................................................................................43 Quadro 26 - Palavras mais comuns em casos de erro.................................................................45 Quadro 27 - Palavras mais comuns em casos de acerto..............................................................45

Lista de Abreviaturas e Siglas

|  |  |
| --- | --- |
| MRI | *Music Information Retrieval* |
| TF-iDF | *Term Frequency–inverse Document Frequency* |
| **ABPD** | **Associação Brasileira dos Produtores de Discos** |
| RID | *Regressive Imagery Dictionary* |
| HSL | *Hindi Subjective* |
| HSW | *Hindi SentiWordnet* |
| HWA  API | *Hindi Wordnet Affect*  *Application Programming Interface* |
| RI | Recuperação de Informação |
| IP | *Internet Protocol* |
| NLP | *Natural Language Processing* |
| NB | Naive Bayes |
| SMO | *Sequential minimal optimization* |
| SVM | *Support Vector Machine* |
| KNN | *K-neareast neighbor* |

Sumário

[Introdução 09](#_Toc500240122)

[1 CLASSIFICAÇÃO DE LETRAS MUSICAIS 12](#_Toc500240123)

[1.1 Mood Classification of Hindi Songs on Lyrics 12](#_Toc500240124)

[1.2 Music Classification Method based on Lyrics for Music Therapy 15](#_Toc500240125)

[1.3 A Study on Music Modd Detection in Online Digital Music Database 16](#_Toc500240125)

[1.4 Automatic Lyrics-based Music Genre Classification in a Multilingual Setting 17](#_Toc500240125)

[2 CONCEITOS DE PROCESSAMENTO DE LINGUAGEM NATURAL 20](#_Toc500240129)

[2.1 Tratamento de texto 20](#_Toc500240130)

[2.1.1 *Stop Words* 20](#_Toc500240131)

[2.1.2 *TF-IDF* 21](#_Toc500240131)

[2.1.3 *Stemização* 22](#_Toc500240130)

[2.2 Algoritmos de classificação 23](#_Toc500240130)

3 desenvolvimento da solução 25

[3.1 Proposta de Solução 25](#_Toc500240130)

[3.2 Desenvolvimento do *Dataset* 26](#_Toc500240131)

[3.2.1 Coleta de informações para a criação do *dataset* 26](#_Toc500240131)

[3.2.2 Coleta de letras musicais em português 28](#_Toc500240130)

[3.2.3 Classificação das músicas 30](#_Toc500240130)

[3.2.4 Remoção de letras duplicadas 31](#_Toc500240131)

[3.2.5 Remoção do número excessivo de letras positivas e letras neutras 32](#_Toc500240131)

[2.2.6 Desenvolvimento de um Modelo para Aprendizado de Máquina 33](#_Toc500240130)

[3.2.7 Remoção de *Stop Words* 33](#_Toc500240131)

[3.2.8 Aplicação de TF-IDF 34](#_Toc500240130)

[3.2.9 Aplicação de *Stemização* 35](#_Toc500240130)

[3.2.10 Remoção de valores com contador único 36](#_Toc500240131)

[3.3 Modelo final para aprendizado de máquina 36](#_Toc500240131)

4 [RESULTADOS E DISCUSSÕES 38](#_Toc500240155)

[4.1 Rresultados com LibSVM 38](#_Toc500240131)

[4.2 Resultados com SMO 39](#_Toc500240131)

[4.3 Resultados com Naive Bayes 40](#_Toc500240131)

[4.4 Resultados com KNN 41](#_Toc500240131)

[4.5 Resultados com J48 41](#_Toc500240131)

[4.6 Resultados com K-means 42](#_Toc500240131)

[4.7 Comparação e análise dos resultados 43](#_Toc500240131)

[4.8 Análise de erros e acertos 44](#_Toc500240131)

[CONCLUSÃO 47](#_Toc500240155)

[Referências Bibliográficas 49](#_Toc500240156)

Introdução

Desde os tempos mais remotos, a música está presente nas manifestações pessoais e sociais dos seres humanos (SCHEFER; DOMINGUES, 2012). As pessoas são atraídas por músicas porque elas expressam e induzem emoções (JUSLIN; LAUKKA, 2004). Além disso, músicas podem ser utilizadas para melhorar a concentração, reduzir o estresse e para treinar a mente (FURUYA; HUANG; KAWAGOE, 2015).

Duncan e Fox (2005) mostram que as pessoas são interessadas em criar coleções musicais baseadas em seu estado de espírito. Para suprir essa necessidade, os sistemas computacionais devem ser capazes de lembrar os gostos dos usuários baseado em músicas ouvidas anteriormente. Além disso, os autores enfatizam que as ferramentas devem ser capazes de indicar músicas adequadas para diferentes estados de espírito, atividades ou diferentes horários do dia.

A pesquisa de músicas com base em emoções tem se mostrado um critério importante para os ouvintes (VIGNOLI, 2004). Bischoff et al. (2009) suportam essa tese indicando que os padrões de pesquisa de usuários refletem a importância existente na associação entre músicas e emoções e que a forma que falamos e pensamos sobre música é fortemente influenciada por sentimentos e contexto.

Com a crescente presença de músicas disponíveis na internet, ferramentas inteligentes de busca se fazem necessárias para consumir os bancos de dados musicais. Através de uma análise automática de músicas, como por exemplo, classificação de gênero, conteúdo, artista ou similaridade entre músicas, sistemas de recomendação podem auxiliar usuários a encontrar aquelas que são mais relevantes para eles (FELL; SPORLEDER, 2014).

Sistemas de *Music Information Retrieval* (MIR) tem grande relevância na gerência de dados de músicas disponíveis *online*. Tais sistemas são amplamente baseados na extração de características sonoras de áudios gravados digitalmente (YING et al, 2015). Porém, Mayer e Rauber (2011) nos mostram que existem diversas características semânticas em letras de músicas que não podem ser detectadas utilizando apenas técnicas de extração de áudio e que, portanto, o conteúdo textual de uma letra pode auxiliar em sua classificação.

De acordo com Kim et al. (2010), todos os anos, sistemas de classificação de categorias musicais baseados em áudio tem uma evolução em sua performance, porém, quando se fala nas emoções contidas em uma música, apenas a utilização de áudio não é o suficiente. Nesses casos, se faz necessária a utilização de metadados como *tags* criadas por usuários ou letras de músicas.

Apesar de proporcionar uma rica quantidade de informação semântica e de ter um forte impacto na percepção humana quanto a emoções, poucos estudos foram realizados com foco na extração de características de letras para a classificação musical (XIONG et al., 2017). De acordo com Fell e Sporleader (2014), estudos de classificação baseados em letras estão ainda em seus primeiros dias e há, de fato, evidência empírica da necessidade de estudos mais aprofundados para verificar a relevância de letras na classificação musical.

Ying et al. (2015), aplicaram técnicas de TF-iDF (em inglês, *Term Frequency–inverse Document Frequency*, em português, frequência do termo–inverso da frequência nos documentos) em uma base de dados de 1.000 músicas de língua inglesa para verificar a existência de relação entre estilos musicais e palavras que expressam algum tipo de emoção. Os resultados mostram que alguns estilos musicais tendem a utilizar determinadas palavras com maior frequência, mostrando assim uma relação clara entre categorias musicais e categorias de palavras. Também há indícios de que técnicas baseadas em letras podem ser utilizadas para outros tipos de classificação, por exemplo, Fell e Sporleder (2014) utilizaram 400.000 músicas na língua inglesa para, além de efetuar uma classificação de categorias, também indicar quais músicas podem ser consideradas ‘melhores’ e quais podem ser consideradas ‘piores’ em termos de opiniões de usuários em relação as mesmas, além de indicar aproximadamente em que época as músicas foram lançadas.

Yang et al. (2017) aplicaram técnicas de mineração textual em 4.651 músicas da língua inglesa para descobrir qual seria sua classificação em classes de emoções. As classificações obtidas foram comparadas com *tags* que representam a concepção de seres humanos sobre as músicas, os resultados obtidos indicam que alguns sentimentos são mais fáceis de se identificar em letras musicais do que outros. Patra et al. (2015), introduzem um estudo para verificar a classificação de emoções em músicas na língua hindi, além disso, os autores utilizaram as letras de músicas para tentar descobrir a polaridade das mesmas. Similarmente, Foruya (2015) utiliza músicas da língua inglesa para dar prosseguimento em estudo anteriormente realizado pelos mesmos autores utilizando músicas em japonês. Os resultados apresentados indicam que o método proposto pelos autores é capaz de classificar músicas em inglês nas categorias de ‘raiva’ e ‘medo’, porém em outras categorias os resultados não foram satisfatórios, o que indica que a realização de novos estudos pode melhorar os resultados obtidos.

Este trabalho propõem a pesquisa e utilização de diferentes algoritmos de classificação para categorizar músicas em classes de polaridade, além de efetuar uma análise comparativa sobre a eficácia das técnicas aplicadas, porém, utilizando letras de músicas em português. Um estudo de classificação para músicas em língua portuguesa se mostra relevante tendo em vista que o relatório da Associação Brasileira dos Produtores de Discos (ABPD) divulgado em maio de 2017 (PRÓ-MÚSICA BRASIL, 2017), mostra que no Brasil o segmento de *streaming* musical cresceu 52,4% em 2016 comparado com 2015, e que, 21 das 40 músicas mais ouvidas no segmento de *streaming* no Brasil foram músicas em português.

O projeto aqui descrito é de natureza aplicada, tendo como objetivo a resolução de um problema. Foi utilizada uma abordagem qualitativa, validando assim se o problema é resolvido com a solução proposta. Os objetivos deste trabalho enquadram-no como uma pesquisa exploratória, sendo proposta uma solução que amplie conhecimentos na área de classificação musical. Quanto ao procedimento técnico, foi feito um estudo bibliográfico e experimental, sendo necessário o conhecimento do estado da arte para efetuar experimentos para avaliação dos resultados obtidos.

O principal objetivo deste trabalho é contribuir para a evolução dos sistemas de classificação e recomendação musical aplicando as técnicas estudadas de trabalhos existentes. Este estudo utiliza letras de músicas na língua portuguesa, classificando-as como “positivas” ou “negativas”. Nossos resultados demonstram que o algoritmo LibSVM foi o que teve maiores resultados de acerto de classificação. Além disso, mostramos que a classe “negativa” se mostra mais fácil de se classificar do que a classe “positiva”.

No capítulo 1 são apresentados trabalhos relacionados ao estudo aqui proposto, indicando os algoritmos e técnicas utilizadas pelos autores. No capítulo 2 são apresentados os principais conceitos e técnicas de processamento de linguagem natural utilizados neste estudo. No capítulo 3 é detalhado o desenvolvimento da solução proposta, contendo todos os passos para a criação da base de dados e modelo de aprendizado de máquina. No capítulo 4 são apresentados os resultados e discussões. Por fim, é apresentada a conclusão do estudo e são discutidas as possibilidades de trabalhos futuros.

# CLASSIFICAÇÃO DE LETRAS MUSICAIS

Para propor uma solução ao problema de classificação de letras de músicas na língua portuguesa em relação a sua polaridade, é necessário analisar o estado da arte em estudos de classificação textual. Neste capítulo são apresentados trabalhos relacionados ao estudo aqui proposto. São apresentadas as técnicas para criação de modelos de aprendizagem de máquina utilizadas pelos autores, bem como os algoritmos de classificação utilizados pelos mesmos.

Primeiramente, o estudo de Patra et al. (2015) é abordado, que tem como objetivo classificar músicas na língua hindi em classes que representem emoções e classifica-las em relação à polaridade. Em seguida, o estudo realizado por Furuya et al. (2014) é analisado. Os autores apresentam uma proposta de classificação de letras de músicas na língua japonesa em classes que representam sentimentos. Logo após, é relatado o estudo efetuado por Yang et al. (2017) sobre a classificação automática de letras de músicas em inglês em classes que representem emoções. Por fim, é apresentado estudo realizado por Howard et al. (2011), que tem como objetivo classificar músicas em português e espanhol em gêneros musicais.

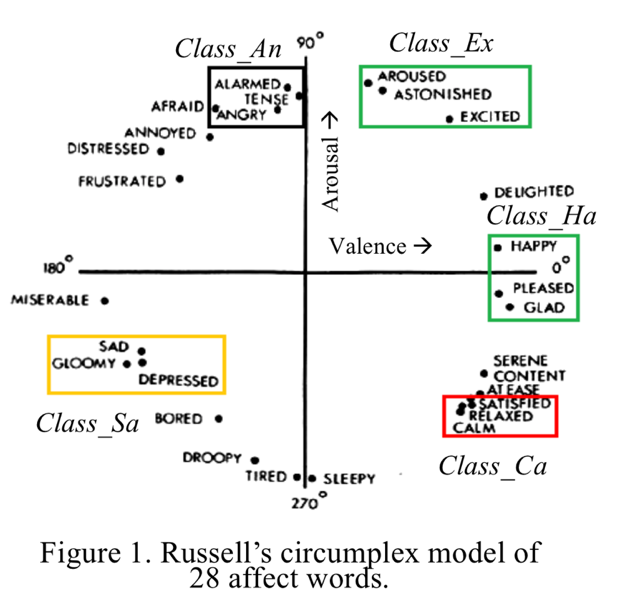
Os trabalhos relatados nesta seção têm relação com o estudo aqui apresentado pois tratam-se de estudos de classificação utilizando letras musicais em diversos idiomas, além disso, algumas das técnicas apresentadas pelos autores aqui citados também são aplicadas no presente estudo.

## 1.1 Mood Classification of Hindi Songs on Lyrics

Em estudo realizado em 2015, Patra et al. propõem um método para classificação de músicas na língua hindi em categorias que representam emoções além de classifica-las em uma das seguintes classes: postiva ou pegativa. Os autores utilizaram o modelo de Russel (RUSSEL, 1980) para fazer o agrupamento de palavras com sentimentos similares para formar as possíveis classes. O resultado do agrupamento com o modelo de Russel pode ser visto na Figura 1, contendo cinco classes com três subclasses cada. O motivo desse agrupamento é coletar músicas similares e agrupa-las dentro de uma única classe.

O Quadro 1 apresenta as cinco classes utilizadas no estudo. A classe “*Class\_Ex*” contém os sentimentos “Animado”, “Admirado” e “Excitado”, “*Class\_Ha*” contém os sentimentos “Satisfeito”, “Feliz” e “Contente”, a classe “*Class\_Ca*”, por sua vez, apresenta os sentimentos “Calmo”, “Descontraído” e “Satisfeito”, enquanto isso, a classe “*Class\_Sa*” apresenta os sentimentos “Triste”, “Sombrio” e “Depressivo”, por fim, a classe “*Class\_An*” traz os sentimentos “Bravo”, “Alarmado” e “Tenso”.

**Figura 1 - Modelo de Russel com 28 palavras Adaptado de Patra et al. (2015)**



Quadro 1 - Classes Propostas Adaptado de Patra et al. (2015)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class\_Ex | Class\_Ha | Class\_Ca | Class\_Sa | Class\_An |
| Animado | Satisfeito | Calmo | Triste | Bravo |
| Admirado | Feliz | Descontraído | Sombrio | Alarmado |
| Excitado | Contente | Satisfeito | Depressivo | Tenso |

O *dataset* utilizado para este estudo é uma replicação textual correspondente ao corpus de áudio criado por Patra at al. (2013 apud PATRA; DAS; BANDYOPADHYAY, 2015), a base de dados final contém as letras de 461 músicas. Além disso, alunos de graduação com idades entre 18 e 24 anos efetuaram a classificação das letras nas classes apresentadas no Quadro 1. Cada música do *dataset* foi anotada por cinco estudantes.

Durante o processo de anotação (classificação) das letras, os autores verificaram que existem algumas confusões entre as classes Class\_An e Class\_Ex”, “Class\_Ha e Class\_Ex” e “Class\_Sa e Class\_Ca”. Isso ocorre porque as mesmas têm características parecidas. Além disso, foi verificado que em alguns casos os anotadores tinham sentimentos diferentes quando ouviam a música e quando apenas liam as letras. Para evitar confusões, cada uma das músicas foi anotada com uma polaridade: positiva ou negativa. A utilização da polaridade aumentou os resultados de concordância entre anotadores de 88% para 95%.

Para a criação de um *framework* de classificação musical, primeiramente os autores utilizaram três analisadores léxicos de sentimentos para classificar os sentimentos das palavras presentes nas letras musicais. Os analisadores léxicos são: *Hindi Subjective Lexicon* (HSL) (BAKLIWAL; ARORA; VARMA, 2012 apud PATRA; DAS; BANDYOPADHYAY, 2015), *Hindi SentiWordnet* (HSW) (JOSHI; R; BHATTACHARYYA, 2010 apud PATRA; DAS; BANDYOPADHYAY, 2015) e *Hindi Wordnet Affect* (HWA) (DAS; PORIA; BANDYOPADHYAY, 2012 apud PATRA; DAS; BANDYOPADHYAY, 2015). O HSL contém duas listas, uma de adjetivos (3.909 positivos, 2.974 negativos e 1.225 neutros) e outra de advérbios (193 positivos, 178 negativos e 518 neutros.) O HSW consiste de 2.168 palavras positivas, 1.391 palavras negativas e 6.426 palavras neutras. O HWA, por sua vez, consiste de 2.986, 357, 500, 3.185, 801 e 431 palavras pertencentes das classes raiva, desgosto, medo, triste, feliz e surpresa, respectivamente (PATRA; DAS; BANDYOPADHYAY, 2015).

Os resultados das classes encontradas pelos três analisadores léxicos podem ser verificados no Quadro 2.

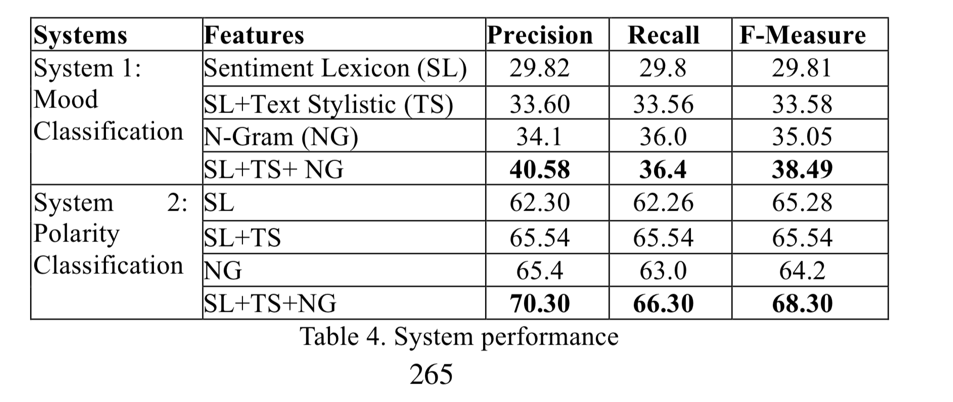
Quadro 2 - Classes encontradas usando HWA, HSL e HSW Adaptado de Patra et al. (2015)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classes | HWA | Classes | HSL | HSW |
| Bravo | 241 | Positivo | 1172 | 857 |
| Desgosto | 13 |
| Medo | 13 |
| Feliz | 349 | Negativo | 951 | 628 |
| Triste | 107 |
| Surpresa | 38 |

Além disso, foram consideradas características como, número de palavras únicas, número de palavras repetidas, número de linhas únicas e linhas terminadas com a mesma palavra. Os autores consideraram o TF- IDF (*Term Frequency-Inverse Document Frequency*, detalhes são apresentados no capítulo 2) (PATRA; DAS; BANDYOPADHYAY, 2015) para identificar o quão importante uma palavra é dentro de uma letra musical, além disso, *stop words* (detalhes sobre *stop words* são apresentados no capítulo 2) foram removidas.

Para desenvolver um Sistema automático para a classificação de letras, foram utilizados diversos algoritmos de *machine learning* (aprendizado de máquina, em português), porém o que teve maior sucesso foi o LibSVM implementado na ferramenta Weka. Foram desenvolvidos dois sistemas de anotações (classificação de cada letra), o primeiro sistema apresenta a classificação das músicas em uma das classes apresentadas no Quadro 1. O segundo sistema classifica as letras em “positiva” ou “negativa”. Os resultados obtidos para ambos os sistemas podem ser verificados no Quadro 3.

Quadro 3 - Performance dos Sistemas (PATRA; DAS; BANDYOPADHYAY, 2015)



Como pode ser verificado, os melhores resultados obtidos foram de *F-Measure* de 38.49 e 68.3 nas classificações de emoção e polaridade, respectivamente. Patra et al. (2015) indicam que a precisão de apenas de 38.49 se da ao fato de que músicas em Hindi podem ter muitas variações de sentimentos em suas letras.

## 1.2 Music Classification Method based on Lyrics for Music Therapy

Em 2014, Furuya et al. apresentam um método para a classificação de músicas em japonês em categorias que representem emoções. O método proposto pelos autores possui os seguintes passos:

* Criação de uma lista de palavras que expressam sentimentos. O conjunto de emoções usado é o seguinte: alegria, conforto, tristeza, raiva, animação, bondade, vergonha, ódio, surpresa e medo.
* Análise morfológica para extrair informações sobre cada palavra presente na música, como por exemplo, se a palavra é um substantivo, verbo, adjetivo ou substantivo adjetivo;
* Análise do número de vezes que palavras que incitam emoções aparecem na letra da música. O número de vezes em que a informação na letra “combina” com uma palavra presente na lista de palavras é contabilizado, criando assim um vetor palavras vs. frequência;
* O vetor é criado contendo o rateio do número de palavras na música e o número de vezes que uma palavra em particular é incluída em cada sentimento (na lista de sentimentos);
* Após este processo, a música é classificada em uma categoria de emoção utilizando-se o método de Ward (FURUYA; HUANG; KAWAGOE, 2014) e o método K-means (FURUYA; HUANG; KAWAGOE, 2014).

Ao aplicar os dois métodos propostos em um *dataset* de 50 músicas previamente classificadas por 10 participantes, é possível perceber que o método K-means apresentou um resultado melhor que o método Ward. Os resultados para ambos métodos podem ser verificados no Quadro 4. Outro ponto interessante é que para as classificações para cada classe utilizando o K-means, a classe “tristeza” foi a que obteve o maior resultado *F-measure*. Os resultados para todas as classes podem ser verificados no Quadro 5. Os autores utilizam como *baseline* para comparação os resultados encontrados por Yamada (2011 apud FURUYA; HUANG; KAWAGOE, 2014).

Quadro 4 - Precisão de Classificação Adaptado de Furuya et al. (2014)

|  |  |  |  |
| --- | --- | --- | --- |
| Método | k-means | ward | *baseline* |
| *Recall* | 27.7% | 24.3% | 45.0% |
| Precisão | 39.0% | 36.0% | 43.0% |
| *F-Measure* | 32.4% | 29.0% | 44.0% |

Quadro 5 - Precisão de Classificação por Emoções (K-means) Adaptado de Furuya et al. (2014)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sentimento | alegria | tristeza | animação | bondade | *baseline* |
| *Recall* | 27.3% | 36.7% | 20.0% | 22.2% | 45.0% |
| Precisão | 37.5*%* | 55.0% | 37.5% | 26.1% | 43.0% |
| *F-Measure* | 31.6% | 44.0% | 26.1% | 24.0% | 44.0% |

## 1.3 A Study on Music Mood Detection in Online Digital Music Database

O estudo de Yang et al. (2017) tem o objetivo de investigar como as letras de músicas afetam o sentimento das pessoas. Os autores mostram que a utilização de modelos que contenham informações de letras de músicas apresenta resultados aceitáveis para o problema de classificação em emoções. Neste estudo os autores utilizam *Support Vector Machines* (SVM) para classificar letras de 4.651 músicas em 66 categorias diferentes. Todas as letras foram coletadas do site Allmusic.com, sendo que 4/5 do corpus é para treinamento de máquina e 1/5 é utilizado para teste.

Os autores utilizaram a ferramenta Lingpipe (YANG et al., 2017) para separar as letras em palavras, remover *stop words* e relacionar palavras com classes gramaticais. Todas as letras de músicas utilizadas pelos autores já estavam classificadas na base de dados do site Allmusic.com nas seguintes classes “Irônico”, “Educado”, “Sarcástico”, “Bobo”, “Ardente”, “Feliz” e “Drama”. Os resultados obtidos com o emprego da implementação LibSVM e *cross validation* de 5 *folds* estão reunidos no Quadro 6.

Quadro 6 - Taxa de acerto com LibSVM Adaptado de Yang et al. (2017)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sentimento | Irônico | Educado | Sarcástico | Bobo | Ardente | Feliz | Drama |
| Precisão | 50% | 47.66% | 45.9% | 55.14% | 75.37% | 76.38% | 91.65% |
| F-measure | 9% | 11.76% | 12.59% | 25.82% | 57.1% | 62.89% | 81.06% |

Segundo os autores, os resultados obtidos indicam que as classes “Feliz”, “Ardente” e “Drama” são as mais simples de se classificar e que, isso pode indicar que as letras que pertencem a essas classes apresentam termos que contenham emoções mais claras. Por outro lado, as outras classes utilizadas no estudo se mostram difíceis de classificar.

## 1.4 Automatic Lyrics-based Music Genre Classification in a Multilingual Setting

Em estudo realizado em 2011, Howard et al. informam que a maioria dos estudos de classificação de gênero musical que utilizam letras de músicas são focados em bases de dados que utilizam apenas letras em uma língua. Neste estudo, os autores apresentam uma proposta de estudo para classificação de músicas em gêneros musicais usando letras de músicas em português e espanhol. Foi utilizada a base de dados *Latin Music Database* (LMD) (SILLA; KOERICH; KAESTNER, 2008) que contém mais de 3.000 músicas de mais de 10 gêneros diferentes. Como o LMD não provê as letras das músicas, as letras de 500 músicas foram coletadas manualmente, contendo 50 músicas para cada um dos seguintes gêneros musicais: Axé, Bachata, Bolero, Forro, Gaúcha, Merengue, Pagode, Salsa, Sertanejo e Tango.

Para classificar as letras de músicas em gêneros musicais, os autores utilizaram os seguintes algoritmos: Naive Bayes(NB), *Sequential minimal optimization* (SMO) e J48, todos implementados na ferramenta Weka com *cross validation* de 10 *folds*. O Quadro 7 apresenta os resultados obtidos pelos autores para os três classificadores utilizados.

Quadro 7 - Resultados de taxa de acerto com os diferentes classificadores Adaptado de Howard et al. (2011)

|  |  |  |  |
| --- | --- | --- | --- |
| Gênero | Naive Bayes | SMO | J48 |
| Axé | 66.0% | 68% | 56.0% |
| Bachata | 68.0% | 58.8% | 39.2% |
| Bolero | 09.8% | 39.2% | 11.8% |
| Forro | 41.2% | 45.1% | 33.3% |
| Gaúcha | 62.7% | 51.0% | 37.3% |
| Merengue | 68.6% | 72.5% | 39.2% |
| Pagode | 47.1% | 37.3% | 35.3% |
| Salsa | 72.5% | 62.7% | 45.1% |
| Sertaneja | 72.5% | 60.8% | 19.6% |
| Tango | 94.1% | 78.4% | 41.2% |

É possível verificar que o Naive Bayes apresentou um bom desempenho, classificando corretamente 60.3% das letras testadas. Porém, o classificador apresentou uma variação entre os gêneros musicais. Tango, por exemplo, foi corretamente classificado 94.1% das vezes, enquanto bolero foi corretamente classificado apenas 9.8% das vezes. SMO classificou corretamente 57.4% das letras do corpus, apesar de apresentar uma média um pouco mais baixa que o Naive Bayes, o classificador SMO conseguiu supera-lo nas classificações do gênero bolero. O classificador J48 foi o que apresentou pior rendimento, classificando corretamente apenas 35.8% das letras. Assim como o Naive Bayes, o J48 teve dificuldade para classificar músicas na categoria Bolero.

Quadro 8 - Impacto do idioma na classificação Adaptado de Howard et al. (2011)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Idioma | NB | SMO | J48 | Média |
| Espanhol | 75.0% | 68.0% | 40.0% | 61.0% |
| Português | 58.0% | 51.0% | 37.0% | 48.6% |
| Ambas | 10.0% | 39.0% | 12.0% | 20.3% |

Os autores ainda puderam verificar que o idioma parece ter grande importância nos resultados de classificação. O Quadro 8 mostra que os gêneros musicais que tem apenas músicas em espanhol tem vantagem sobre gêneros que tem músicas em português e gêneros que tem músicas em ambos idiomas. A razão para tanto é obvia, os autores utilizaram cinco gêneros de músicas com letras apenas em português e três com músicas apenas em espanhol. Segundo os autores, como o número de classes para escolher é menor após filtrar classes da mesma língua, músicas em espanhol tem mais facilidade para serem classificadas corretamente. Os pesquisadores ainda indicam que a classe “bolero” é a mais difícil de se classificar corretamente porque a mesma tem uma quantidade equilibrada de letras em espanhol e em português.

# CONCEITOS E TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL

Processamento de Linguagem Natural (do inglês, *Natural Processing Language* ou NLP) é uma área de conhecimento que explora métodos computacionais para interpretar e processar a linguagem humana em sua forma textual ou em sua forma falada (ALLEN, 2006). Como o objetivo do Processamento de Linguagem Natural é interpretar e processar a linguagem humana com precisão comparável a de seres humanos, NLP é considerada uma disciplina que faz parte da Inteligência Artificial (LIDDY, 2001). Neste estudo são aplicadas técnicas de Processamento de Linguagem Natural para resolver o problema de classificação de documentos de texto. Para Soucy et al. (2001), a classificação de textos é o processo utilizado para identificar em que classe um determinado documento de texto pertence. Como o objetivo deste estudo é a classificação de letras em português nas classes “positiva” e “negativa, utilizaremos algumas técnicas de classificação presentes nos trabalhos citados no capítulo 1. A seguir detalhamos as técnicas e algoritmos aplicados neste estudo.

2.1 **Tratamento de texto**

Este estudo utiliza um modelo de aprendizado de máquina juntamente com a execução de algoritmos de classificação para resolver o problema de classificar músicas na língua portuguesa. Para que o modelo possa ser criado, faz-se necessário o tratamento prévio das letras musicais. Para tanto, utilizamos algumas técnicas de tratamento de texto que são apresentadas nesta seção. Primeiramente, é apresentada a técnica de remoção de *stop words*. Em seguida é detalhado o algoritmo TF-IDF (do inglês, *Term Frequency – inverse document frequency*) e, por fim, é apresentada a técnica de *stemização (*do inglês, *stemming)*.

2.1.1 ***Stop Words***

Uma *stop word* (em português, palavra de parada) é uma palavra que tem a mesma probabilidade de aparecer em documentos que não são relevantes quanto em documentos que são relevantes para determinada pesquisa (WILBUR; SIROTKIN, 1992). Como essas palavras aparecem em todos os documentos, elas não apresentam relevância para a classificação, por este motivo, *stop words* são removidas dos *datasets* de classificação.

Como exemplo de remoção de palavras de parada, podemos analisar a frase ‘Um teste de remoção de palavras de parada’ que se torna ‘teste remoção palavras parada’ após a remoção das *stop words*. Podemos analisar que as palavras de parada ‘um’ e ‘de’ foram removidas. Na seção 3.3.1, é apresentado exemplo de remoção de *stop words* em uma das letras utilizadas neste estudo.

2.1.2 **TF-IDF**

*Term Frequency – inverse document frequency* (TF-IDF), em português “frequência do termo–inverso da frequência nos documentos”, é uma das medidas estatística mais comuns nos sistemas de Recuperação de informação (RI) (AIZAWA, 2003). Segundo Joachims (1996), o TF-IDF representa um documento *d* como um vetor para que documentos similares contenham vetores similares. Os valores do vetor são calculados através de uma combinação das estatísticas TF(w,d) e DF(w). Onde:

TF = *Term Frequency* (Frequência do termo)

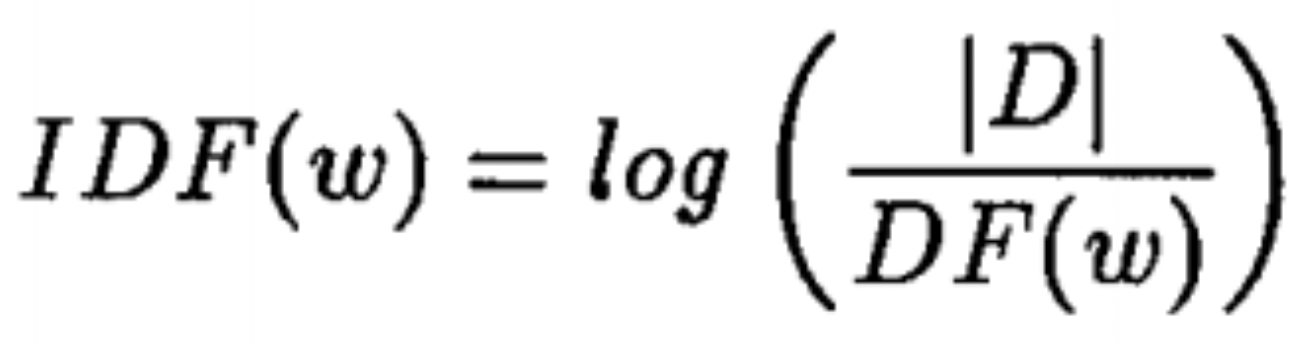
w = *word* (palavra ou termo)

DF = *Document Frequency* (Frequência no documento)

|D| = Número total de documentos

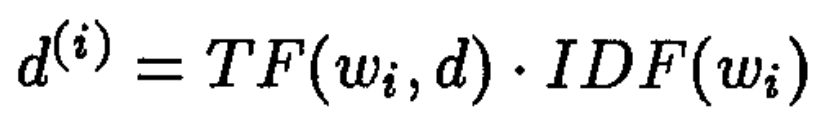
A frequência do termo (*term frequency*) TF(w,d) é o número de vezes que a palavra *w* ocorre em um documento *d*. A frequência no documento (*document frequency*) DF(w) é o número de documentos em que a palavra *w* aparece ao menos uma vez. A frequência inversa do documento (*inverse document frequency*) IDF(w) pode ser calculada a partir da frequência do documento (JOACHIMS, 1996). A fórmula para este cálculo é apresentada na Figura 2.

Figura 2 - Fórmula do idf Fonte: (JOACHIMS, 1996)



Ainda segundo Joachims (1996), a frequência inversa do documento de uma palavra é baixa quando a palavra ocorre em muitos documentos e alta quando a palavra está presente em um número baixo de documentos. O valor do TF-IDF é então calculado com a seguinte fórmula, apresentada na Figura 3:

Figura 3 - Fórmula do TF-IDF (JOACHIMS, 1996)

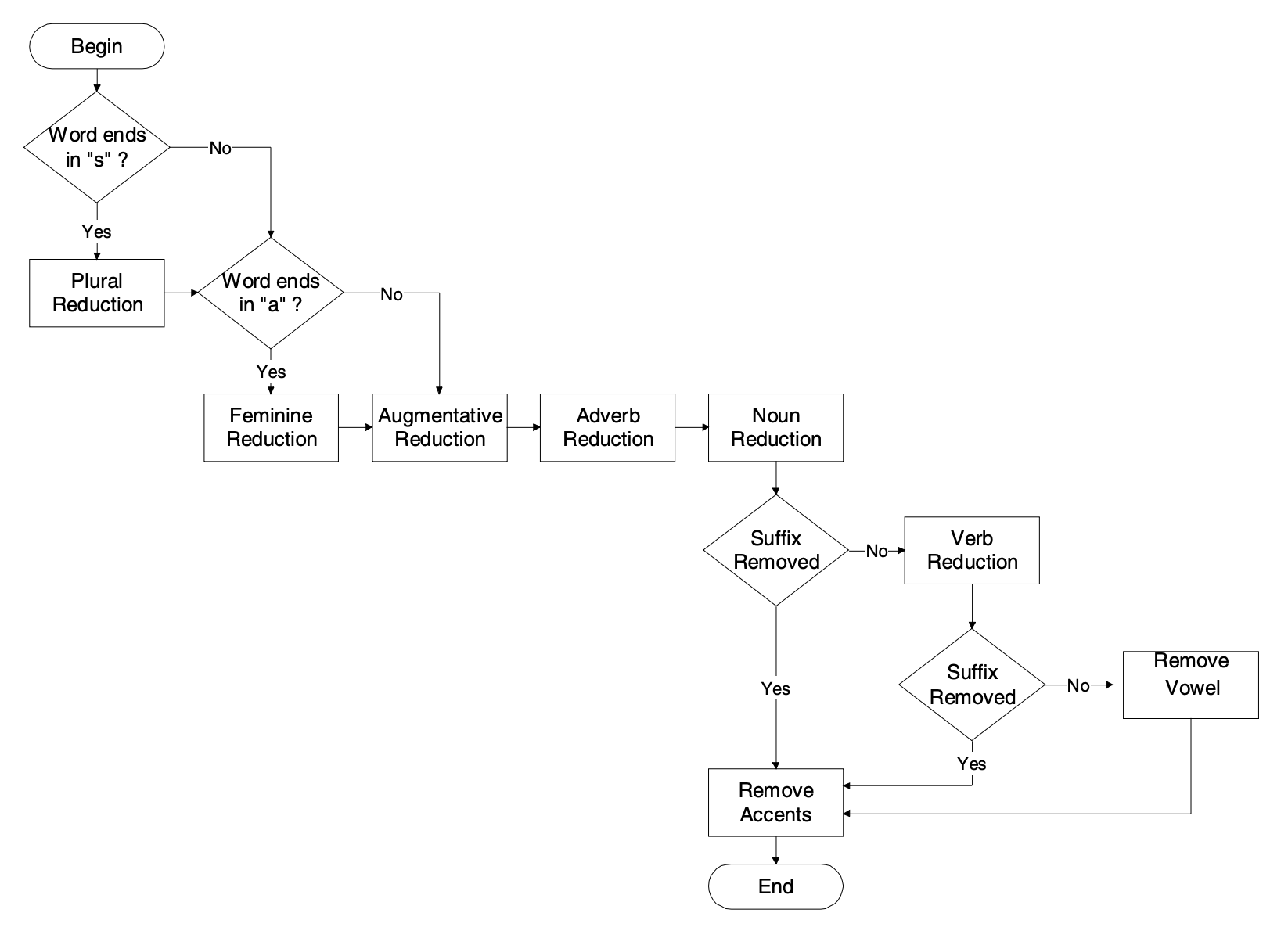


O resultado da fórmula é representado por *d^(i)*, este é o “peso” do termo (palavra) dentro do documento, este peso nos informa o quão relevante é determinada palavra dentro do documento. Exemplos de valores de TF-IDF utilizados neste estudo são apresentados na seção 3.3.2.

2.1.3 ***Stemização***

Algoritmos de *stemização* são utilizados para reduzir palavras ao seu radical, de forma que elas se tornem iguais (LOVINS, 1968). De acordo com Porter (2011), o motivo da aplicação de algoritmos de *stemização* é melhorar a performance de sistemas de IR (do inglês, *Information Retrieval*), fazendo com que palavras com formas similares sejam identificadas como sendo a mesma “palavra”. Por exemplo, na língua portuguesa poderíamos representar o sentimento de amizade com diversas palavras como “amigo”, “amiga”, “amigão”, “amigona”, “amigos”, “amigas”, etc. Ao aplicar um algoritmo de *stemização* nas palavras citadas, todas poderiam ser reduzidas para a mesma “palavra”, que seria a raiz comum de todas as palavras, neste caso o resultado seria “amig”. Dessa forma, é possível verificar se existe alguma relação entre os documentos que apresentam palavras que estejam agrupadas neste *cluster*.

Figura 4 - Algoritmo de Oregon Fonte: (ORENGO; HUYCK, 2001)



Como a eficiência de algoritmos de *stemização* está fortemente ligada com a língua para a qual o algoritmo foi desenvolvido (ORENGO; HUYCK, 2001), precisamos utilizar um algoritmo feito especificamente para a língua portuguesa. Neste estudo utilizamos o Algoritmo de Orengo (ORENGO; HUYCK, 2001) para fazer a *stemização* das palavras das músicas coletadas. Na Figura 4 é apresentado um diagrama explicando os passos seguidos pelo algoritmo de Oregon. Pode-se analisar que primeiramente o algoritmo verifica se a palavra termina com a letra “s”, se a resposta for sim, é efetuada a “redução de plural”. Em seguida, é verificado se a palavra termina com a letra “a” e, caso termine, é efetuada a “redução de gênero feminino da palavra”, os próximos passos são “redução de aumentativo”, “redução de advérbio” e “redução de substantivo”, respectivamente. Então, é analisado se o sufixo da palavra foi removido, caso tenha sido removido, é efetuada a remoção de acentos e o algoritmo é finalizado. Caso o sufixo ainda não tenha sido removido, é realizada a “redução de verbo” e verificado novamente se o sufixo foi removido. Desta vez, se o sufixo foi removido, é feita a remoção de acentos e o algoritmo é finalizado. Porém, caso o sufixo ainda não tenha sido removido, é feita a “remoção de vogais”, “remoção de acentos” e então, o algoritmo é finalizado.

2.2 **Algoritmos de classificação**

A seguir, são brevemente apresentados os algoritmos de aprendizado de máquina utilizados neste estudo. Os algoritmos de classificação aqui utilizados são: Naive Bayes, *Support Vector Machine*, J48, KNN e K-means.

O Naive Bayes é um classificador probabilístico que assume que todos os atributos dos modelos de entrada são independentes uns dos outros (MCCALLUM; NIGAM, 1998). Em estudo publicado em 2002, Pang et al. mostram que em certos casos o Naive Bayes apresenta resultados próximos a algoritmos mais complexos no problema de polarização de documentos de texto.

Máquinas de vetores de suporte (SVM, do inglês: *support vector machine*) é um popular método de aprendizado de máquina para resolver problemas de classificação, regressão e outros problemas de aprendizado de máquina (CHANG; LIN, 2011). O algoritmo de máquinas de vetores de suporte foi utilizado neste estudo por apresentar alta performance nos problemas de classificação textual (HU; DOWNIE; EHMANN, 2009). Neste estudo, utilizamos duas implementações de SVM: LibSVM (CHANG; LIN, 2011) e *Sequential minimal optimization* (SMO) (PLATT, 1998).

O algoritmo J48 é uma implementação do algoritmo de árvore de decisão C4.5. O J48 cria uma árvore de decisão a partir dos dados do modelo utilizado. Essa árvore é então aplicada em cada tupla da base de dados, resultando em uma classificação para a respectiva tupla (PATIL, 2013).

K vizinhos mais próximos (do inglês, *K Nearest Neighbour Classifier* ou KNN), é um algoritmo que cria classes através do modelo de dados aplicado ao algoritmo. Um objeto é classificado de acordo com os vizinhos mais próximos do objeto em questão (YONG; YOUWEN; SHIXIONG, 2009). Apesar de ser um dos algoritmos de classificação mais simples que existe (CUNNINGHAM; DELANY, 2007), quando o *dataset* de treino é muito grande, o algoritmo sofre com a necessidade de grande poder computacional (GAYATHRI, 2013).

K-means é um método de *clustering* que automaticamente separa instâncias de uma base de dados em um número *k* de grupos (WAGSTAF et al., 2001). Por ser um algoritmo relativamente simples e que não requer muito poder de processamento, o K-means pode ser utilizado no processamento de grandes bases de dados (MACQUEEN, 1967).

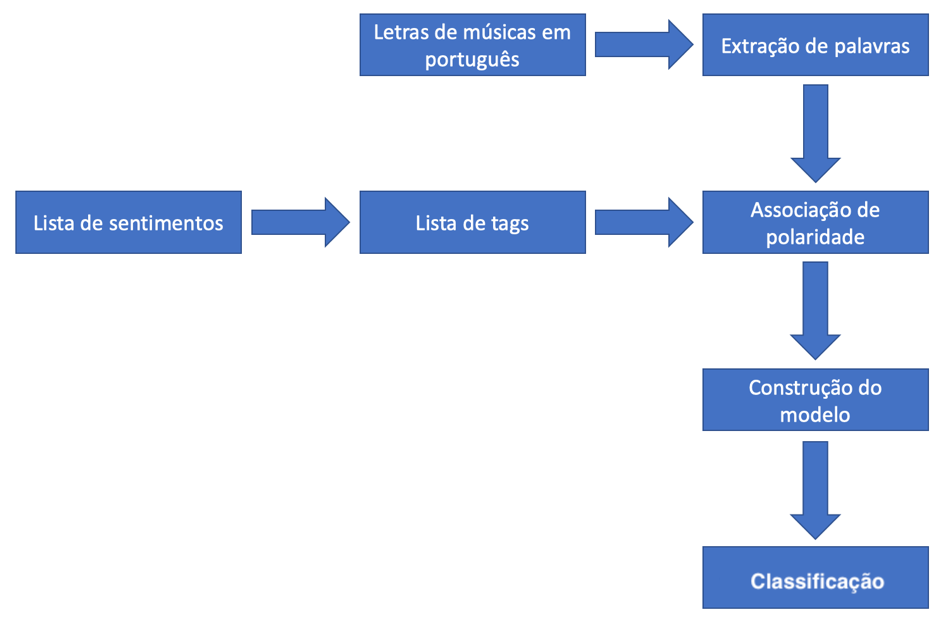
# 3 desenvolvimento da solução

A seguir, é abordado o desenvolvimento da solução proposta para resolver o problema de classificação de letras de músicas na língua portuguesa. Primeiramente, uma explicação geral da solução é apresentada, em seguida, são detalhados os passos tomados para a criação de um corpus contendo informações musicais. É explicado como as letras das músicas foram coletadas e como as músicas foram inicialmente classificadas. Por fim, é abordada a criação de um modelo para posterior utilização de algoritmos de aprendizado de máquina.

## 3.1 Proposta de Solução

Para solucionar o problema de classificar músicas na língua portuguesa quanto à polaridade, propomos uma adaptação de método utilizado por Furuya et al. (2014) para a coleta e tratamento dos dados. A figura 5 apresenta o processo de classificação de músicas na língua portuguesa.

Figura 5 - Processo do estudo



Primeiramente, foram coletadas as traduções para o português de diversas letras musicais juntamente com suas classificações, de forma que os dados foram tratados para formar um modelo contendo diferentes atributos. Os nomes de músicas e dos artistas foram coletados da base de dados *Million Song Dataset* (MSD) (BERTIN-MAHIEUX et al., 2011), as *tags* para classificação foram coletadas do *The Last.fm dataset* (CANTADOR; BELLOGIN; VALLET, 2010) em uma combinação com palavras do *Opinion-lexicon-English* (HU; LIU, 2004). As letras em português foram coletadas a partir do site www.vagalume.com.br. O trabalho desenvolveu a construção de modelo para a utilização de classificação supervisionada. Foram utilizados os algoritmos disponíveis no Weka, uma ferramenta que proporciona acesso a diversos algoritmos de aprendizado de máquina através de uma *interface* gráfica simples (HOLMES; DONKIN; WITTEN, 1994). Os resultados gerados pelos algoritmos são então apresentados e comparados com aqueles encontrados por outros autores.

## 3.2 Desenvolvimento do *Dataset*

Para criar um modelo que contenha informações de letras de músicas em português, é necessária a utilização de uma base de dados que contenha as letras musicais. Não foi encontrado até o momento um corpus que pudesse ser utilizado neste estudo. Desta forma, um corpus foi desenvolvido a partir de outras diferentes bases de dados disponíveis *online*. A seguir, descrevemos o processo de criação do *dataset* utilizado como corpus neste estudo.

### 3.2.1 Coleta de informações para a criação do *dataset*

Como ponto de partida, utilizamos a base de dados *Million Song Dataset* (MSD), um corpus de larga escala desenvolvido para ajudar no desenvolvimento de pesquisas científicas (BERTIN-MAHIEUX et al., 2011). O MSD contém informações de 1 milhão de músicas que pertencem a 44.745 artistas diferentes. O site do *Million Song Dataset* (https://labrosa.ee.columbia.edu/millionsong/) também provê um grupo de corpus criados pela comunidade de pesquisadores, essas bases de dados são complementares ao MSD. Um destes corpus é o *The Last.fm*, que se trata de um corpus que contém *tags* criadas por usuários do site www.last.fm para 505.216 músicas do MSD, ao total, o *dataset* apresenta 522.366 *tags* distintas.

Utilizando o MSD e o *The Last.fm*, conseguimos fazer uma relação entre músicas e percepções de usuários através das *tags* criadas. Como os usuários podem criar qualquer *tag* desejada, as 522.366 *tags* presentes não necessariamente representam algum tipo de sentimento positivo ou negativo. Algumas das *tags* mais populares, por exemplo, apresentam o estilo musical da música que contém a *tag*, diante disso, se apresenta a necessidade de filtrar músicas que contenham *tags* que representem algum sentimento positivo ou negativo e deixar de lado aquelas que somente contenham outros tipos de informações. Para filtrar as músicas que contêm *tags* que representam algum tipo de sentimento, precisamos primeiramente identificar estes sentimentos nas *tags* criadas pelos usuários. Para tanto, utilizamos o *Opinion-lexicon-English*, um *dataset* contendo 2.007 palavras positivas e 4.783 palavras negativas que foi criado por Hu e Liu (2004) ao longo de vários anos. Com o *Opinion-lexicon-English* fizemos um mapeamento das músicas do MSD que contenham *tags* no *Last.fm* e que estejam também presentes no *Opinion-lexicon-English*. Portanto, *tags* do *The Last.fm* que são palavras listadas como negativas no *Opinion-lexicon-English* são entendidas como *tags* negativas, por sua vez, *tags* que representam palavras que foram listadas como positivas são *tags* positivas. O Quadro 9 apresenta exemplos de palavras que foram classificadas como positivas ou negativas.

Quadro 9 - Palavras negativas e positivas

|  |  |
| --- | --- |
| **PALAVRA** | **POLARIDADE** |
| abnormal | NEGATIVA |
| abomination | NEGATIVA |
| aborted | NEGATIVA |
| abrasive | NEGATIVA |
| absence | NEGATIVA |
| absurd | NEGATIVA |
| absurdity | NEGATIVA |
| abuse | NEGATIVA |
| accidental | NEGATIVA |
| accusation | NEGATIVA |
| ﻿accessible | POSITIVA |
| accomplished | POSITIVA |
| accurate | POSITIVA |
| adequate | POSITIVA |
| admiration | POSITIVA |
| adorable | POSITIVA |
| adore | POSITIVA |
| adored | POSITIVA |
| adoring | POSITIVA |
| advanced | POSITIVA |

Inicialmente, tínhamos o MSD com informações de um milhão de músicas diferentes, o *The Last.fm* com *tags* para 505.216 dessas músicas e o *Opinion-lexicon-English* contendo 6.970 palavras. Após efetuar o mapeando das músicas do MSD com as músicas do *The Last.fm* e das *tags* do *The Last.fm* com as palavras presentes no *Opinion-lexicon-English*, temos um total de 121.382 músicas com pelo menos uma *tag* positiva e 88.544 músicas com pelo menos uma *tag* negativa e 155.994 músicas com ao menos uma *tag* positiva ou negativa. Portanto, 346.222 músicas que não possuem *tags* de sentimento foram descartadas. A Figura 6 exemplifica as informações contidas nas três bases de dados utilizadas até o momento, nela é apresentada a intersecção das três bases de dados citadas nos parágrafos anteriores.

Figura 6 - Filtro utilizando três bases de dados

**155.994**

### 3.2.2 Coleta de letras musicais em português

Nenhuma das bases de dados citadas até o momento provê informações sobre as letras das músicas. Além disso, não encontramos até o momento um corpus que contenha as letras das músicas do MSD em sua versão em português ou algum outro banco de dados de músicas em português que já apresente polarização ou algum atributo que indique emoções. Portanto, para a realização desde estudo fez-se necessária a coleta das letras das músicas citadas na seção anterior.

O site brasileiro Vagalume (<https://www.vagalume.com.br/>), lançado em 2002, possui as letras e traduções de milhões de músicas populares de diversos artistas. O Vagalume possui uma API (do inglês, *Application programming Interface*) que pode ser utilizada por desenvolvedores para coletar informações das músicas disponíveis no Vagalume e apresenta-las em outros sites ou sistemas. Neste estudo a API do Vagalume foi utilizada para coletar as letras em português das músicas do MSD que tem *tags* no *The Last.fm* que correspondem a palavras do *Opinion-lexicon-English*. A Figura 7 apresenta um exemplo de como enviar uma requisição para o Vagalume com o objetivo de buscar a letra da música *One* do artista U2.

Figura 7 - Chamada para API do Vagalume

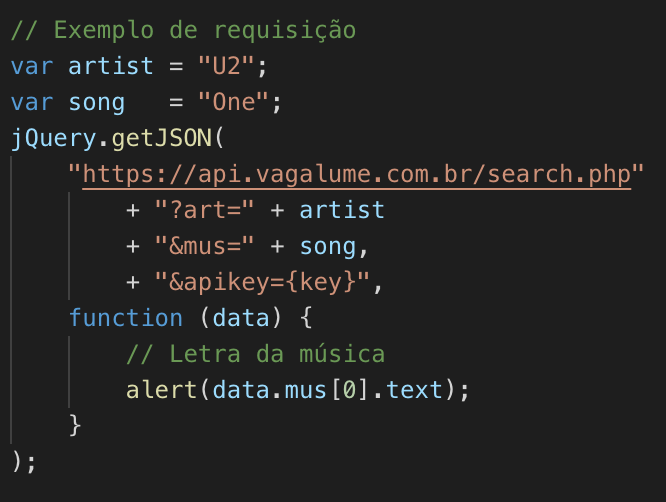
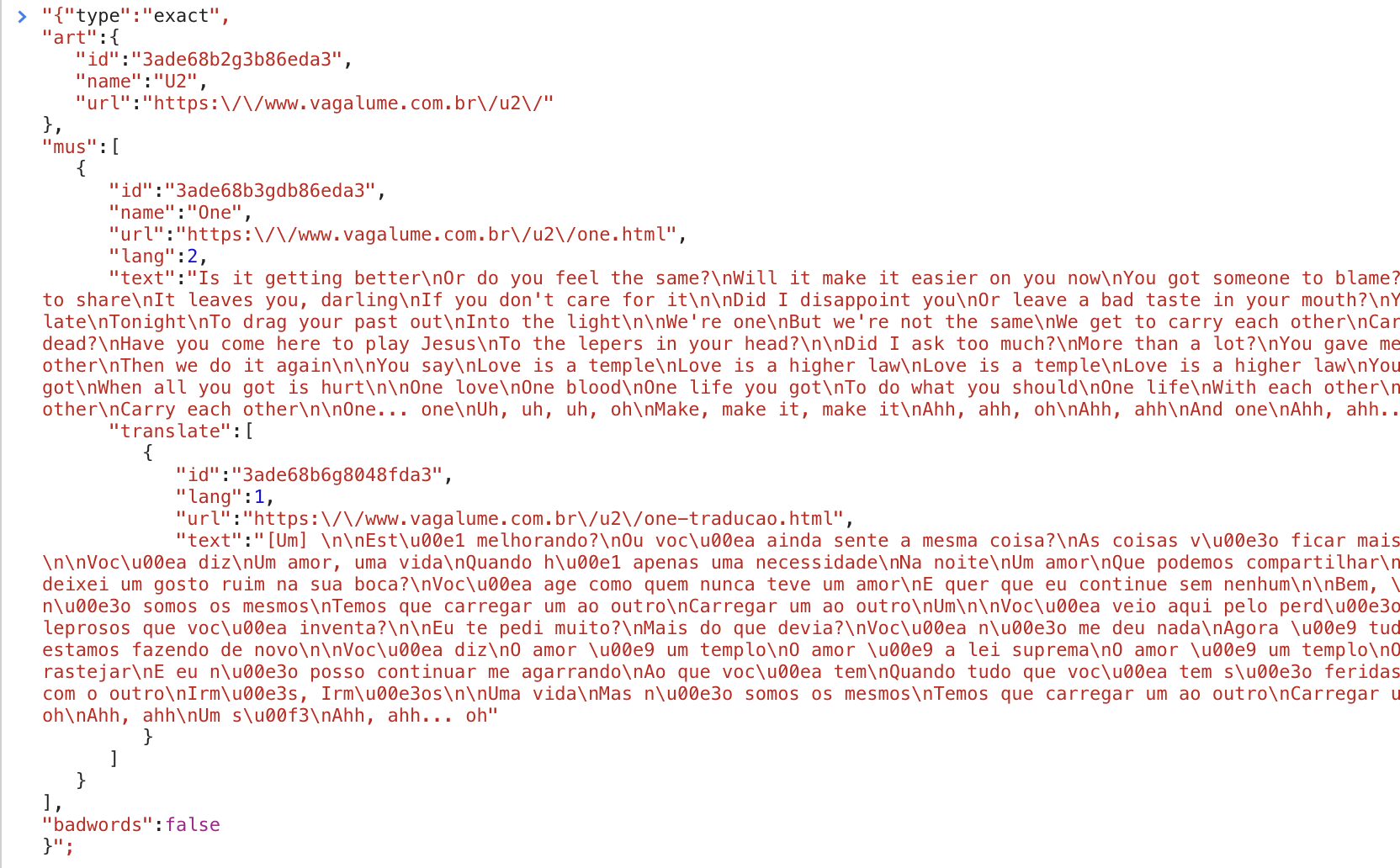


Figura 8 - Exemplo de resposta da API do Vagalume



Através de requisições ao site do Vagalume, podemos coletar algumas informações das músicas, entre elas a letra original e uma tradução para o português, caso a tradução exista no Vagalume. A Figura 8 apresenta a resposta da requisição, efetuada através do navegador Google Chrome.

Para os fins deste estudo, utilizamos as traduções das músicas para o português. Enviamos diversas requisições para a API do Vagalume durante a coleta das letras das músicas e, como o Vagalume possui um limite de requisições que pode ser efetuado a partir de um IP (*Internet Protocol*) durante determinado período de tempo, não foi possível coletar todas as letras em tempo hábil. Executamos o algoritmo de requisições diversas vezes e coletamos letras em português para 23.457 músicas.

### 3.2.3 Classificação das músicas

Utilizamos as *tags* das músicas cadastradas pelos usuários para determinar se uma música é “positiva” ou “negativa”. Músicas foram classificadas como positivas se elas contêm mais *tags* positivas do que negativas. Músicas foram classificadas como negativas se elas contêm mais *tags* negativas do que positivas, por sua vez, músicas com o mesmo número de *tags* positivas e negativas foram classificadas como neutras.

Para exemplificar como o filtro foi feito, digamos que determinada música contenha apenas 3 *tags*, sendo elas: *adored*, *admiration* e *abomination*. Como podemos visualizar no Quadro 9, *adored* (adorado, em português) foi classificada como positiva, *admiration* (admiração, em português) também foi classificada como positiva e *abomination* (abominação, em português) foi classificada como uma palavra negativa pelo *Opinion-lexicon-English*. Isso significa que a música em questão seria classificada como positiva por apresentar um número maior de *tags* positivas do que de *tags* negativas.

Foram classificadas apenas as músicas em que foi possível coletar a letra da API do Valume, destas, os resultados foram os seguintes:

* Total: 23.457;
* Positivas: 16.588;
* Negativas: 4.537;
* Neutras 2.332;

Na Figura 9, apresentamos um gráfico contendo as quantidades de músicas classificadas para cada categoria. Podemos ver que as músicas positivas representam 71% de todas as letras coletadas da API do vagalume. Este número é consideravelmente maior do que a quantidade de letras em português coletadas para a classe negativa, que representa 19% do total. Por fim, as músicas que foram classificadas como neutras representam apenas 10% da base de dados.

Figura 9 - Músicas classificadas

### 3.2.4 Remoção de letras duplicadas

O *Million Song Dataset* apresenta informações duplicadas para algumas músicas. Mesmo que as músicas tenham artistas e letras diferentes entre si, é possível que a música esteja duplicada, pois artistas diferentes podem fazer versões da mesma música, além disso, existem dados de músicas cantadas ao vivo e sua versão gravada em estúdio. Realizamos a remoção das letras duplicadas. Com a remoção de letras duplicadas, ficamos com um total de 12.573 letras classificadas. Os dados são os seguintes:

* Total:12.573;
* Positivas: 8.363;
* Negativas: 2.740;
* Neutras 1.470;

Na Figura 10, apresentamos um gráfico contendo as quantidades de músicas classificadas para cada categoria. Podemos ver que as músicas positivas agora representam 66% de todas as letras coletadas da API do vagalume. Novamente, este número é consideravelmente maior em relação a quantidade de letras em português coletadas para a classe negativa, que representa 22% do total. Por fim, as músicas que foram classificadas como neutras representam 12% da base de dados.

Figura 10 - Porcentagem de músicas por classe

### 3.2.5 Remoção de número excessivo de letras positivas e letras neutras

Ao analisar os dados das letras de músicas coletadas, podemos verificar a existência de um número excessivo de músicas com polaridade positiva em relação as músicas com polaridade negativa. É possível analisar, inclusive, que a soma da quantidade de letras negativas com a quantidade de letras neutras ainda não atinge a quantidade de letras positivas na base de dados. Para evitar que os algoritmos de aprendizado de máquina possam dar alguma preferência para a polaridade positiva (*bias*), foram removidas de forma aleatória grande quantidade das músicas positivas para que possamos ter uma base de dados mais equilibrada. As letras neutras também foram removidas, pois o objetivo deste estudo é de classificar letras como positivas ou negativas. Após a remoção das músicas duplicadas e das músicas neutras, temos os seguintes números:

* Total: 5.000;
* Positivas: 2.500;
* Negativas: 2.500;

Na Figura 11 é apresentada a versão final do *dataset* balanceado, contendo 50% de letras classificadas como “positivas” e 50% de letras musicais classificadas como “negativas”.

Figura 11 - Porcentagem de letras positivas e negativas

### 3.3 Desenvolvimento de um Modelo para Aprendizado de Máquina

Com os dados coletados, criamos um corpus contendo informações de 12.573 músicas diferentes, sendo que neste estudo serão utilizadas informações de 5.000 músicas, 2.500 positivas e 2.500 negativas. Para que seja possível a aplicação de algoritmos de aprendizado de máquina nos dados coletados, criamos um modelo no formato *csv* contendo, além da polaridade, as 10 palavras mais relevantes para cada letra musical e um contador informando quantas vezes essas palavras aparecem na letra da música. Por fim, o modelo possui um contador de palavras positivas e um contador de palavras negativas em cada letra. Esta seção detalha a criação do modelo.

### 3.3.1 Remoção de *Stop words*

Como *stop words* aparecem em todos os documentos, elas não apresentam relevância para a classificação, por este motivo efetuamos a remoção das mesmas. A seguir, é apresentado exemplo de remoção de *stop words* da música de id ﻿TRWDFUY128E0784BD1.

Música com *stop words*:

“Passo a passo, de coração para coração,

Esquerda direita esquerda

Todos nós caímos

Passo a passo, de coração para coração,

Esquerda direita esquerda

Nós todos caímos, como soldados de brinquedo

Pouco a pouco, separados, nós nunca vencemos

Mas a batalha continua, para os soldados de brinquedo”

Música sem *stop words*:

“Passo passo, coração coração,

Esquerda direita esquerda

Todos caímos

Passo passo, coração coração,

Esquerda direita esquerda

todos caímos, soldados brinquedo

Pouco pouco, separados, nunca vencemos

Mas batalha continua, soldados brinquedo”

Com a remoção de *stop words*, pode-se analisar que palavras como ‘a’, ‘de’, ‘os’, ‘para’ e ‘nós’ foram removidas.

### 3.3.2 Aplicação de TF-IDF

Aplicamos o algoritmo TF-IDF nas letras coletadas. O Quadro 10 apresenta alguns dos resultados obtidos através do algoritmo.

Quadro 10 - Exemplos de valores de TF-IDF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID NO MSD | P1 | P2 | P3 | TFIDFP1 | TFIDFP2 | TFIDFP3 |
| TRDEZPM128E078F1F8 | ensinará | filosofia | majestade | 17.4816... | 12. 459... | 12.034... |
| TRMHAJE128F429B800 | afasto | desmorona | desmorono | 37.073... | 31.149... | 19.466… |

Os resultados nos mostram, por exemplo, que as palavras mais importantes no documento (música) de id TRDEZPM128E078F1F8 são: ensinará, filosofia e majestade, respectivamente, pois estas foram as palavras com o maior “peso” calculado através do TF-IDF. Podemos analisar também que para a música TRMHAJE128F429B800 temos duas palavras muito similares com valores diferentes de TF-IDF, sendo essas palavras “desmorona” e “desmorono”, que possuem os valores “31.149...” e “19.466...”, respectivamente. Isso nos leva a crer que a palavra “desmorona” tem mais importância do que a palavra “desmorono” neste documento. Porém, como as duas palavras possuem a mesma raiz, podemos aplicar um algoritmo de normalização textual e torna-las uma só.

### 3.3.3 Aplicação de *Stemização*

Conforme apresentado no capítulo 2, neste estudo utilizamos o Algoritmo de Oregon (2001) para *stemização*. No Quadro 11, podemos conferir alguns exemplos de palavras após a aplicação do Algoritmo de Orengo.

Quadro 11 - Palavras com *stemização*

|  |  |
| --- | --- |
| Palavra original | Palavra após *stemização* |
| justificado | justific |
| vire-se | vire-s |
| vida | vid |
| desmorona | desmoron |
| desmorono | desmoron |
| afasto | afast |

Após a aplicação do Algoritmo de Orengo nas letras coletadas, o TF-IDF foi calculado novamente. O cálculo de TF-IDF deve ser refeito pelo fato de que a *stemização* irá combinar algumas palavras da base de dados. No Quadro 12, são apresentados novamente os resultados para as músicas TRDEZPM128E078F1F8 e TRMHAJE128F429B800, porém, agora com *stemização*.

Quadro 12 - TF-IDF após *stemização*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID NO MSD | P1 | P2 | P3 | TFIDFP1 | TFIDFP2 | TFIDFP3 |
| TRDEZPM128E078F1F8 | filosof | majestad | pois | 12.277... | 12.034... | 11.639... |
| TRMHAJE128F429B800 | desmoron | afast | tud | 33.719... | 23.947... | 11.183… |

Fazendo uma comparação entre o Quadro 10 e o Quadro 12, podemos verificar as mudanças de pesos em alguns termos após a *stemização*. Além disso, anteriormente a música TRMHAJE128F429B800 tinha duas palavras muito similares: “desmorona” e “desmorono”. Após a aplicação do algoritmo de Orengo as duas palavras foram combinadas e tornaram-se “desmoron” com um peso de “33.719…”.

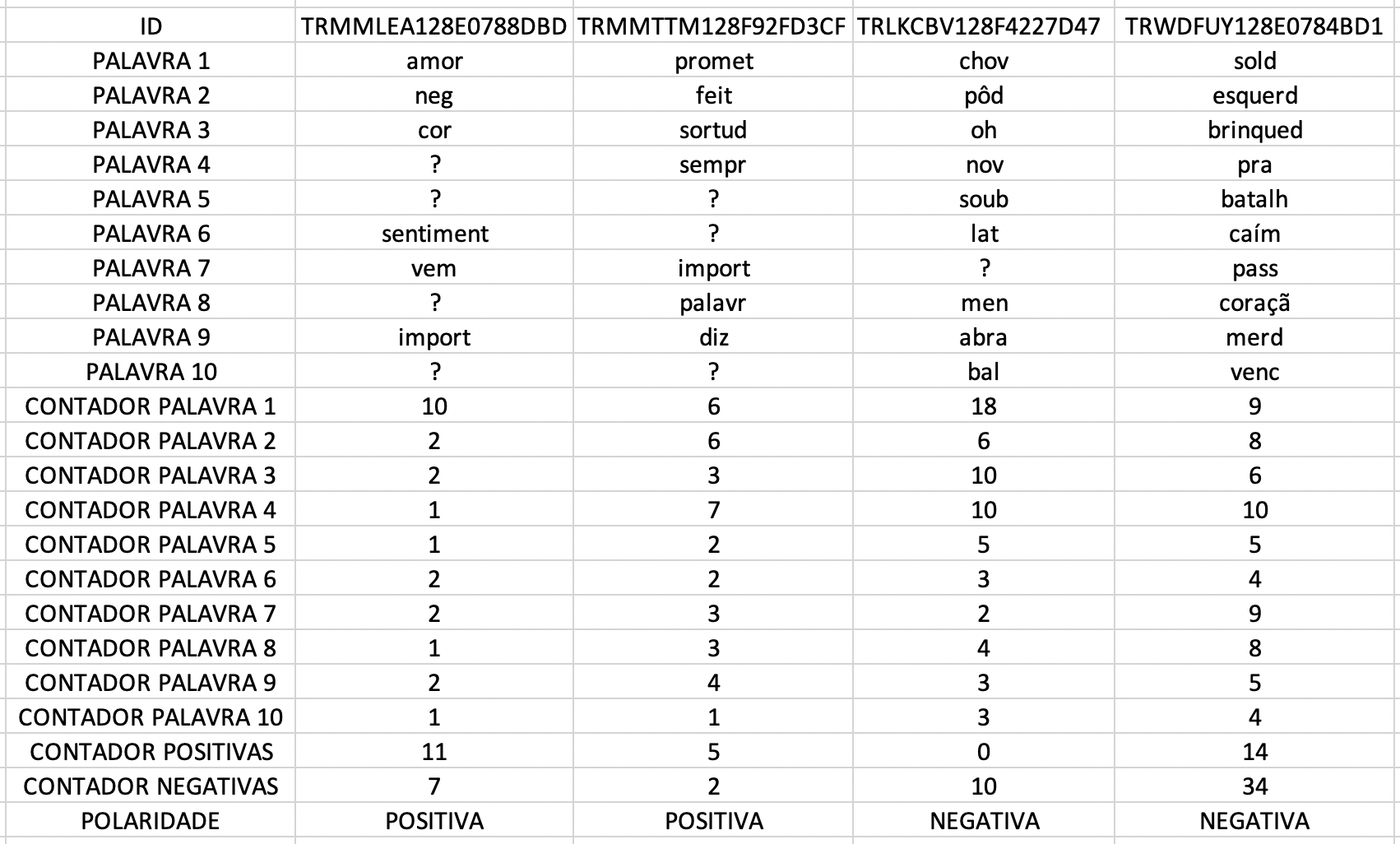
### 3.3.4 Remoção de valores com contador único

Os valores que aparecem uma única vez por atributo foram removidos. Por exemplo, a palavra “pequim” aparece uma única vez no atributo “palavra1” no documento TRMMCTJ128F933AE36. Isso indica que a palavra “pequim” tem bastante relevância para o documento, mas não necessariamente uma grande relevância para a polaridade positiva, pois este trata-se do único documento em que a palavra aparece. As palavras removidas são representadas com o valor “?” no modelo para aprendizado de máquina.

### 3.3.5 Modelo final para aprendizado de máquina

O modelo criado para ser aplicado em algoritmos de aprendizado de máquina conta com 24 atributos. Os atributos do modelo em questão são apresentados no Quadro 13, o MSD ID é apresentado por questões informativas, porém o ID não é utilizado no aprendizado de máquina. Os dez seguintes atributos são as palavras mais importantes na letra de cada música, em seguida são apresentadas as frequências em que as dez palavras mais importantes aparecem no documento. Então, temos o contador de palavras positivas no documento, contador de palavras negativas do documento e, por fim, a polaridade associada a música.

Quadro 13 - Modelo final para aprendizado de máquina



As dez palavras mais importantes de cada música foram calculadas com a utilização do TF-IDF após *stemização* e remoção de *stop words*. Os contadores para cada uma das dez palavras mais importantes seguem a mesma ordenação das palavras. Os contadores de palavras positivas e negativas contabilizam valores de todas as palavras presentes nas músicas após aplicação do algoritmo de *stemização*, e não somente as dez palavras consideradas mais importantes através do algoritmo TF-IDF.

# resultados e discussões

Nesta seção são apresentados os resultados obtidos pelo presente estudo. Foram utilizadas as implementações na ferramenta Weka para os seguintes algoritmos: LibSVM, SMO, Naive Bayes, KNN, J48 e k-means. Para cada um dos algoritmos são apresentadas suas porcentagens totais de acertos de classificação, bem como suas porcentagens individuais para cada uma das classes de classificação. Além disso, para a avaliação dos resultados utilizamos as medidas de precisão, *recall* (MANNING;SCHÜTZE, 1999) e *F-measure*. Ao fim dessa seção são comparados os resultados dos algoritmos citados para verificar qual apresenta os melhores resultados. Todos os algoritmos foram executados com *cross-validation* de 10 *folds*.

4.1 **Resultados com LibSVM**

A matriz de confusão do LibSVM apresentada no Quadro 14 nos mostra que 1.150 músicas foram corretamente classificadas como positivas e que 1.350 foram incorretamente classificadas como negativas. Por sua vez, 1.769 músicas foram corretamente classificadas como negativas e 731 foram incorretamente classificadas como positivas. Como podemos analisar, 46% das músicas positivas foram corretamente classificadas enquanto 70,76% das músicas negativas foram corretamente classificadas pelo algoritmo. Com a junção de resultados das duas classes, temos um total de 58,38% instâncias corretamente classificadas.

Quadro 14 - Matriz de confusão do LibSVM

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 1.150 | 1.350 |
| Negativa | 731 | 1.769 |

No Quadro 15 são apresentados os resultados de *F-measure* para cada uma das duas classes e para o total de instâncias. É possível analisar que obtivemos um valor total de *F- measure* de 57,70 em comparação aos 68,30 apresentados por Patra et al. (2015). É importante lembrar, no entanto, que o estudo de Patra et al. (2015) utilizava letras de músicas em hindi, enquanto os resultados aqui apresentados tratam-se de letras de músicas coletadas na língua portuguesa.

Quadro 15 - *F-measure* por classe no LibSVM

|  |  |
| --- | --- |
| Classe | *F-measure* |
| Postiva | 52,50 |
| Negativa | 63,30 |
| Total | 57,70 |

O estudo de Yang et al. (2017) não classifica músicas em classes de polaridade, mas sim em classes mais específicas de sentimentos. Enquanto o melhor resultado encontrado pelos autores foi o de *F-measure* 81,06 para a classe “Drama”, nosso melhor resultado com o LibSVM foi o de 63,30 para a classe “Negativa”.

Os resultados encontrados com o LibSVM nos levam a acreditar que as músicas que foram previamente classificadas como negativas são mais fáceis de se classificar do que as letras que tem a classificação positiva. Podemos associar essa conclusão com os resultados de Furuya et al. (2015), que indicam que músicas em classes que representam “raiva” e “medo” são as mais fáceis de classificar.

4.2 **Resultados com SMO**

Com a implementação do *Sequential minimal optimization* (SMO) disponível na ferramenta Weka, obtivemos 55,30% das instâncias do modelo corretamente classificadas. O melhor resultado apresentado foi em relação as classificações da classe negativa. A matriz de confusão presente no Quadro 16 nos mostra que 1.342 músicas positivas foram corretamente classificadas e 1.158 músicas foram incorretamente classificadas como negativas. Por sua vez, 1.077 músicas foram incorretamente classificadas como positivas e 1.423 músicas foram corretamente classificadas como negativas.

Quadro 16 - Matriz de confusão do SMO

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 1.342 | 1.158 |
| Negativa | 1.077 | 1.423 |

No estudo realizado por Howard et al. (2011), os autores apresentam uma classificação com acerto de 51% para músicas em português em classes de gênero musical. Aqui conseguimos um acerto de 53,68% na classificação de músicas positivas e 56,92% na classificação de músicas negativas.

Quadro 17 - *F-Measure* por classe no SMO

|  |  |
| --- | --- |
| Classe | *F-measure* |
| Postiva | 54,60 |
| Negativa | 56,00 |
| Total | 55.30 |

Como pode ser visto no Quadro 17, o SMO apresentou um *F-measure* de 55,30 para o total de instâncias, 56 para a classe negativa e 54.60 para a classe positiva. Mais uma vez, é possível analisar que os resultados foram melhores na classificação de músicas com sentimentos negativos.

4.3 **Resultados com Naive Bayes**

A matriz de confusão apresentada no Quadro 18 mostra que 1.150 letras de músicas foram corretamente classificadas como positivas, 1.350 foram incorretamente classificadas como negativas, 776 incorretamente classificadas como positivas e 1.724 corretamente classificadas como negativas. Portanto, 57,48% das letras da base de dados foram corretamente classificadas pelo Naive Bayes.

Quadro 18 - Matriz de confusão do Naive Bayes

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 1.150 | 1.350 |
| Negativa | 776 | 1.724 |

No Quadro 19 são apresentados os resultados de *F-Measure* para cada uma das classes com a utilização do Naive Bayes. Obtives 52,00 e 61,90 para as classes “positiva” e “negativa”, respectivamente. Com a junção das duas classes, o resultado total de *F-measure* é de 56,90.

Quadro 19 - *F-Measure* com Naive Bayes

|  |  |
| --- | --- |
| Classe | *F-measure* |
| Postiva | 52,00 |
| Negativa | 61,90 |
| Total | 56,90 |

Mais uma vez, é possível verificar que as músicas da classe negativa foram mais fáceis de classificar do que as músicas da classe positiva. Enquanto 68,96% das músicas negativas foram corretamente classificadas, apenas 46% das músicas positivas foram corretamente classificadas.

4.4 **Resultados com KNN**

O KNN classificou corretamente 1.334 músicas positivas e 1.290 músicas negativas. Por outro lado, 1.156 músicas foram incorretamente classificadas como negativas e 1.210 músicas foram incorretamente classificadas como positivas. A matriz de confusão do KNN é apresentada no Quadro 20.

Quadro 20 - Matriz de confusão do KNN

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 1.334 | 1.156 |
| Negativa | 1.210 | 1.290 |

Desta vez, podemos verificar que a classe “positiva” obteve resultados melhores do que a classe “negativa”. Porém, os números apresentam uma diferença de apenas 1,76%. São 53.36% de acertos na classe “positiva” em comparação aos os 51.60% de acertos da classe “negativa”. Os números de *F-measure* são apresentados no Quadro 21. Pode ser verificado que a classe “positiva” apresenta *F-measure* de 53,20, a classe “negativa” apresenta o valor de 52,20 e o resultado total é de 52,70.

Quadro 21 - Resultados de F-measure com KNN

|  |  |
| --- | --- |
| Classe | *F-measure* |
| Positiva | 53,20 |
| Negativa | 52,20 |
| Total | 52,70 |

4.5 **Resultados com J48**

O modelo criado neste estudo não foi o bastante para que o algoritmo de árvore de decisão J48 pudesse fazer a classificação dos dados. Como pode ser verificado no Quadro 22, 2.500 letras aparecem corretamente como letras positivas, 0 aparecem erroneamente como letras negativas, 0 são corretamente apresentadas como letras negativas e 2.500 são incorretamente classificadas como positivas. Esses números apresentam 50% das letras corretamente classificadas. Porém, o que ocorreu é que o J48 não conseguiu encontrar distinção entre as duas classes e criou uma árvore de decisão com apenas um nodo, este sendo positivo. Portando, para todas as instâncias do modelo o algoritmo “chutou” o valor “positivo”.

Quadro 22 - Matriz de confusão do J48

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 2.500 | 0 |
| Negativa | 2.500 | 0 |

No Quadro 23 são apresentados os resultados obtidos de *F-measure* para o J48. É possível analisar que o Weka apenas apresenta *F-measure* para a classe “positiva” com o valor de 66,70. Os valores não disponíveis são representados com “?”.

Quadro 23 – Resultados de *F-measure* com J48

|  |  |
| --- | --- |
| Classe | *F-measure* |
| Positiva | 66,70 |
| Negativa | ? |
| Total | ? |

4.6 **Resultados com K-means**

O algoritmo de *clustering* K-means classificou corretamente 56,10% dos dados da base de dados, classificando corretamente 34,52% das instâncias positivas e 77.68% das negativas.

Quadro 24 - Matriz de confusão com k-means

|  |  |  |
| --- | --- | --- |
| Classe correta | Positiva | Negativa |
| Positiva | 863 | 1.637 |
| Negativa | 558 | 1.942 |

No Quadro 24 podemos verificar que 863 instâncias foram corretamente classificadas como positivas e 1.942 foram corretamente classificadas como negativas. Por outro lado, 1.637 foram incorretamente classificadas como negativas e 558 incorretamente classificadas como positivas. Assim como a maioria dos outros algoritmos utilizados neste estudo, o K-means obteve maior porcentagem de acerto com a classe “negativa”. É interessante analisar, porém, que 1.421 (28,42%) das letras foram classificadas como positivas e 3.579 (71,58%) das letras foram classificadas como negativas, o que indica que o K-means resultou em uma alta tendência de “chutar” na classe “negativa”. O Weka não apresenta resultado de *F-measure* para o algoritmo K-means.

4.7 **Comparação e análise dos resultados**

Após aplicar os seis algoritmos de aprendizado de máquina no modelo criado, podemos comparar os resultados de cada um dos métodos para ver qual apresenta o melhor desempenho para resolver o problema de classificar músicas na língua portuguesa nas classes de polaridade. No Quadro 25, são apresentadas as seguintes informações sobre a execução de cada um dos algoritmos: acertos na classificação da classe “positiva”, acertos na classificação da classe “negativa”, acertos totais, precisão, *recall* e *F-measure*. Como o Weka não apresenta resultados de *F-measure* para o K-means, o resultado não é apresentado no quadro.

O algoritmo J48 apresenta 100% de acertos para a classe “positiva” e 0% de acertos para a classe “negativa” por ter criado uma árvore de decisão com apenas um nodo, este sendo positivo, portanto, seus resultados são descartáveis e não há motivos para compara-los com os demais algoritmos.

Quadro 25 - Comparação entre resultados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algoritmo | Acertos Positiva | Acertos Negativa | Acertos totais | Precisão | *Recall* | *F-measure* |
| LibSVM | 46% | 70,76% | **58,38%** | **58,90** | **58,40** | **57,70** |
| SMO | 53,68% | 56,92 | 55,30 | 55,30 | 55,30 | 55,30 |
| Naive Bayes | 46% | 68,96% | 57,48% | 57,90 | 57,50 | 56,90 |
| KNN | **53,76%** | 51,60% | 52,68% | 52,70 | 52,70 | 52,70 |
| K-means | 34,52% | **77,68%** | 56,10% | x | x | x |

Podemos analisar que o algoritmo KNN foi o que obteve o melhor resultado na classificação das letras na classe “positiva”, ficando mais de 7% acima dos resultados obtidos com o algoritmo LibSVM e Naive Bayes. O SMO aparece como o segundo melhore resultado, apenas 0,08% atrás do resultado obtido pelo KNN. Já para as classificações de letras na classe “negativa”, o algoritmo K-means foi o que apresentou o melhor resultado com 77,68% de acertos. É interessante ressaltar, porém, que os algoritmos LibSVM e Naive Bayes apresentaram resultados de classificação com mais de 68% de instâncias corretamente classificadas como negativas. Quanto ao número total de acertos, LibSVM foi o método que obteve o maior número de letras corretamente classificadas, apresentando um rendimento de 58,38%, ligeiramente maior do que o apresentado pelos demais algoritmos, com o Naive Bayes logo em seguida com 57,48% de acertos. Os resultados similares entre Naives Bayes e LibSVM corroboram com os resultados encontrados por Pang et al. (2002) que mostram que o Naive Bayes pode apresentar resultados comparáveis com os de algoritmos mais complexos em problemas de polarização de documentos de texto.

Em estudo realizado em 2015, Patral et al. apresentam um *F-measure* de 68.30 na classificação de polaridades de músicas em hindi utilizando o algoritmo LibSVM. Neste estudo obtivemos *F-measure* de 57,70 com o mesmo algoritmo, porém, utilizando músicas em português. Analisando os resultados de todos os algoritmos utilizados neste estudo, podemos verificar que somente o J48 não obteve uma porcentagem de acerto total maior que 50%.

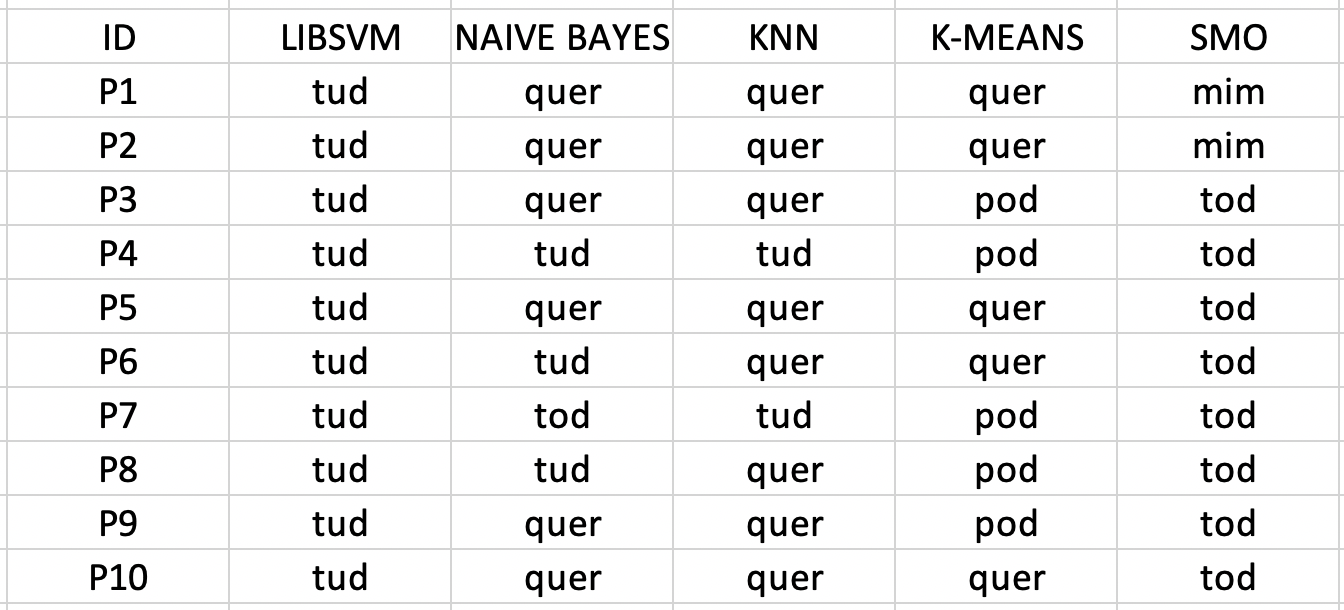
Podemos verificar números relativamente baixos nos acertos de classificação para a classe “positiva”. Por outro lado, K-means, LibSVM e Naive Bayes apresentam 77,68%, 70,76% e 68,69% de acerto na classe “negativa”, respectivamente. Esses números sugerem que há mais facilidade de se classificar músicas que foram previamente classificadas como “negativas”, assim como é apresentado no estudo de Furuya et al. (2015).

4.8 **Análise de erros e acertos**

Após executar os algoritmos na ferramenta Weka, é possível analisar os erros e acertos gerados pelos mesmos. No Quadro 26, são apresentadas as palavras mais frequentes que não são “?” (conforme seção 3.3.4) para cada atributo quando ocorrem erros de classificação para cada algoritmo.

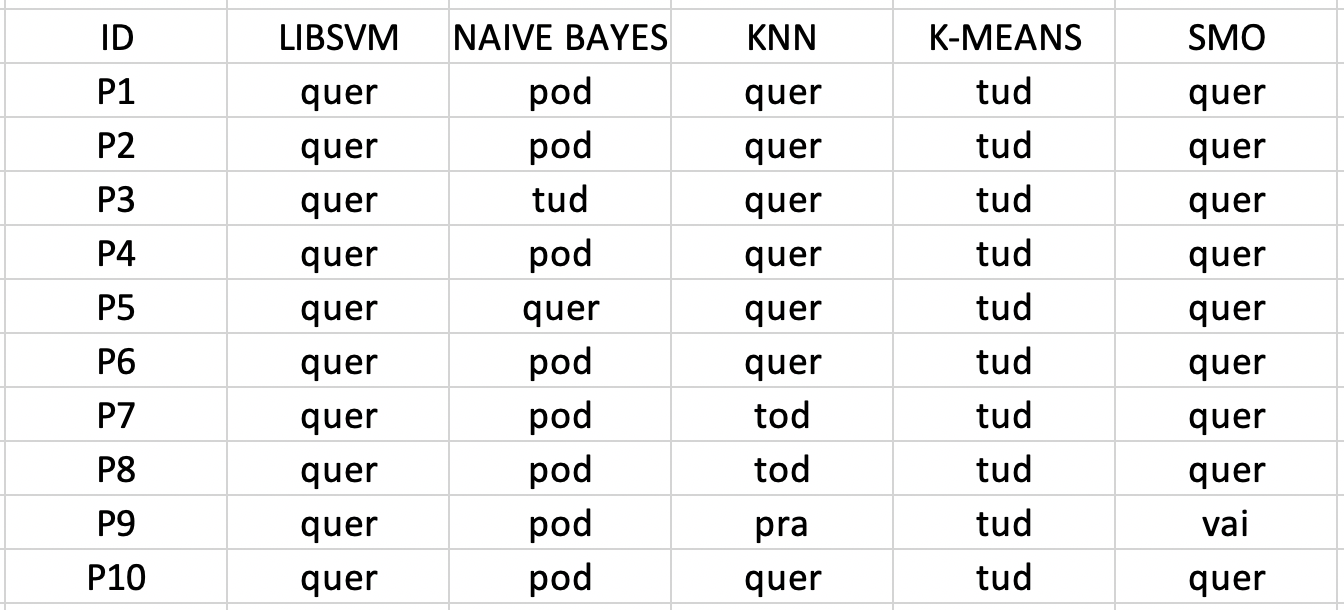
É possível analisar que para o algoritmo LibSVM, a palavra “tud” aparece como a mais frequente para todos os atributos nos casos de erros de classificação. Já para o algoritmo Naives Bayes, podemos verificar que três palavras aparecem como as mais frequentes: “quer” para os atributos “palavra1”, “palavra2”, “palavra3”, “palavra5”, “palavra9” e “palavra10”, “tud” para os atributos “palavra4”, “palavra6” e “palavra8” e “tod” para “palavra7”. Quanto ao KNN, “quer” aparece como palavra mais frequente em casos de erro para os atributos “palavra1”, “palavra2”, “palavra3”, “palavra5”, “palavra6”, “palavra8”, “palavra9” e “palavra10”. Já a palavra “tud” aparece como a mais frequente para os atributos “palavra4” e “palavra7”. Quanto ao algoritmo K-means, pode ser analisado que “quer” aparece como palavra mais frequente para erros nos atributos “palavra1”, “palavra2”, “palavra5”, “palavra6” e “palavra10”. Enquanto isso, “pod” aparece como palavra mais frequente para os atributos “palavra3”, “palavra4”, “palavra7”, “palavra8” e “palavra9”. Quanto ao algoritmo SMO, pode ser analisado que a palavra “mim” é a mais comum para os atributos “palavra1” e “palavra2”. Para os atributos restantes, a palavra mais frequente é “tod”.

Quadro 26 - Palavras mais comuns em casos de erro



Pode ser analisado que somente cinco palavras estão presentes no Quadro 26, o que indica que as mesmas são muito frequentes no *dataset* criado.

Quadro 27 - Palavras mais comuns em casos de acerto



No Quadro 27 são apresentadas as palavras mais frequentes que não são “?” (conforme seção 3.3.4) em cada atributo para os casos de acertos de todos os algoritmos. Podemos analisar que para o LibSVM, a palavra “quer” aparece como a mais frequente de todos os atributos. Para o algoritmo Naive Bayes a palavra “pod” aparece como a mais frequente para os atributos “palavra1”, “palavra2”, “palavra4”, “palavra6”, “palavra7”, “palavra8”, “palavra9” e “palavra10”. Já a palavra “tud” aparece como mais frequente para o atributo “palavra3”. A palavra “quer” aparece como mais frequente para o atributo “palavra5”. Quanto ao KNN, a palavra “quer” aparece como mais frequente para os atributos “palavra1”, “palavra2”, “palavra3”, “palavra4”, “palavra5”, “palavra6” e “palavra10”. A palavra “tod” aparece como mais frequente para os atributos ““palavra7” e “palavra8”. Já para o atributo “palavra9”, a palavra “pra” é a mais frequente. Para os casos de acerto do algoritmo K-means, podemos analisar que a palavra “tud” é a mais frequente para todos os atributos. Para o algoritmo SMO, “vai” aparece como a palavra mais frequente para o atributo “palavra9”. Para os restantes atributos, “quer” é a palavra mais recente.

Assim como no Quadro 26, podemos analisar que poucas palavras aparecem no Quadro 27: apenas seis palavras distintas.

CONCLUSÃO

Este trabalho foi realizado através de uma pesquisa exploratória com a leitura de diversos artigos científicos. Foram estudadas as técnicas de classificação disponíveis no estado da arte e as mesmas foram utilizadas para a classificação de letras de músicas na língua portuguesa em duas classes que representam sentimentos negativos ou positivos. Apresentamos estudos relacionados ao trabalho aqui apresentado, bem como algoritmos de classificação também utilizados pelos estudos antecessores a este. Este estudo apresentou de forma detalhada a criação de uma base de dados com informações de letras musicais na língua portuguesa, mostramos como foram coletadas e mescladas informações de outras bases de dados presentes na internet, bem como os passos tomados na criação dos atributos para o modelo de aprendizado de máquina. Apresentamos os resultados obtidos na aplicação dos algoritmos LibSVM, SMO, Naive Bayes, KNN, J48 e K-means ao modelo proposto. Nosso estudo mostrou que dos seis algoritmos utilizados, o LibSVM foi aquele que apresentou os melhores resultados de classificação. Também foi possível verificar maior facilidade na classificação de letras em português na classe negativa do que na classe positiva, o que pode indicar que músicas com sentimentos negativos realmente apresentam características mais fortes do que músicas com sentimentos positivos. Por fim, apresentamos propostas para futuros trabalhos que venham a utilizar os dados aqui coletados.

Como herança do presente estudo, a base de dados aqui utilizada fica disponível para outros autores. São, no total, 12.573 letras de músicas traduzidas para a língua portuguesa, 5.000 destas são as mesmas utilizadas neste estudo. A base de dados que fica disponível para futuros estudos consta com 8.363 músicas classificadas como positivas 2.740 músicas classificadas como negativas e 1.470 músicas classificadas como neutras. Além da base de dados coletada, também fica disponível para futuros estudos o modelo em formato *csv* que aqui utilizamos para a aplicação de diferentes técnicas de aprendizado de msáquina na ferramenta Weka. Conforme exemplo apresentado no Quadro 13, nosso modelo possui informações de 5.000 letras de músicas traduzidas para a língua portuguesa, sendo que 2.500 estão classificadas como positivas e 2.500 estão classificadas como negativas. O modelo e a base de dados podem ser solicitados através do *e-mail*: natanportilho@outlook.com.

Apresentando uma taxa de acerto total de 58,38% utilizando o algoritmo LibSVM, mostramos que é possível criar um modelo de aprendizado de máquina para letras de músicas na língua portuguesa que apresente resultados comparáveis com aqueles apresentados por estudos que utilizam letras em outros idiomas. Acreditamos que o presente estudo se trata de um passo importante para os sistemas de classificação de textos na língua portuguesa e que, a base de dados resultante deste estudo poderá ser de grande auxílio para futuros estudos que venham utilizar aprendizado de máquina para classificações na língua portuguesa.

Aqui utilizamos um modelo de aprendizado de máquina proposto no capítulo 3, futuros estudos podem utilizar variações deste modelo para verificar se melhores resultados serão apresentados, especialmente com o algoritmo J48, que não se mostrou compatível ao modelo de aprendizado de máquina aqui proposto. Um novo modelo de aprendizado de máquina poderia ser criado a partir da base de dados coletada por este estudo. Além disso, futuros estudos podem propor a utilização de novas técnicas ou algoritmos de classificação textual para documentos na língua portuguesa.

Referências Bibliográficas

AIZAWA, A. An information-theoretic perspective of tf-idf measures. Information Processing and Management, v. 39, n. 1, p. 45–65, 2003.

ALLEN, J. F. Natural Language ProcessingThe Cambridge Handbook of Second Language AcquisitionRochester, New York, USAUniversity of Rochester, 2006. Disponível em: <http://ebooks.cambridge.org/ref/id/CBO9781139051729A033>.

BERTIN-MAHIEUX, T.; ELLIS, D. P. W.; WHITMAN, B.; LAMERE, P. the Million Song Dataset. 12th International Society for Music Information Retrieval Conference (ISMIR 2011), n. Ismir, p. 591–596, 2011. Disponível em: <http://ismir2011.ismir.net/papers/OS6-1.pdf>.

BISCHOFF, K.; FIRAN, C. S.; PAIU, R.; NEJDL, W.; DE, L. S.; LAURIER, C.; SORDO, M. Music Mood and Theme Classification - a Hybrid Approach. Information Retrieval, n. Ismir, p. 657–662, 2009. Disponível em: <http://mtg.upf.edu/files/publications/music\_mood\_and\_theme\_classification-L3S\_MTG.pdf>.

CANTADOR, I.; BELLOGÍN, A.; VALLET, D. Content-based recommendation in social tagging systems. Proceedings of the fourth ACM conference on Recommender systems - RecSys ’10, p. 237, 2010. Disponível em: <http://portal.acm.org/citation.cfm?doid=1864708.1864756>.

CHANG, C.-C.; LIN, C.-J. Libsvm. ACM Transactions on Intelligent Systems and Technology, v. 2, n. 3, p. 1–27, 2011. Disponível em: <http://dl.acm.org/citation.cfm?doid=1961189.1961199>.

CUNNINGHAM, P.; DELANY, S. J. K -Nearest Neighbour Classifiers. Multiple Classifier Systems, n. April 2007, p. 1–17, 2007.

DUNCAN; FOX; Computer–aided music distribution: The future of selection, retrieval and transmission. First Monday, 2005. Disponível em: <http://journals.uic.edu/ojs/index.php/fm/article/view/1220/1140>. Acesso em: 25 de março de 2018.

FELL, M.; SPORLEDER, C. Lyrics-based Analysis and Classification of Music. International Conference on Computational Linguistics, v. 25, n. 23–29, p. 620–631, 2014.

FURUYA, M.; HUANG, H.-H.; KAWAGOE, K. Evaluation of Music Classification Method based on Lyrics of English Songs, 2015.

FURUYA, M.; HUANG, H.-H.; KAWAGOE, K. Music classification method based on lyrics for music therapy. Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS ’14, p. 382–383, 2014. Disponível em: <http://dl.acm.org/citation.cfm?doid=2628194.2628203>.

GAYATHRI, K. Text document preprocessing with the Bayes formula for classification using the support vector machine. [s.d.]

HOLMES, G.; DONKIN, A.; WITTEN, I. H. WEKA: a machine learning workbench. Proceedings of ANZIIS ’94 - Australian New Zealnd Intelligent Information Systems Conference, p. 357–361, [s.d.]Disponível em: <http://ieeexplore.ieee.org/document/396988/>.

HOWARD, S.; SILLA, C. N.; JOHNSON, C. G. "Automatic lyricsbased music genre classification in a multilingual setting", Proceedings of the Thirteenth Brazilian Symposium on Computer Music, 2011.

HU, M.; LIU, B. Mining and summarizing customer reviews. Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04, p. 168, 2004. Disponível em: <http://portal.acm.org/citation.cfm?doid=1014052.10ß14073>.

HU, X.; DOWNIE, J. S.; EHMANN, A. F. Lyric text mining in music mood classification. American Music, v. 183, n. Ismir, p. 411–416, 2009.

JOACHIMS, T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. the 14th International Conference on Machine Learning (ICML ’97), p. 143–151, 1997. Disponível em: <papers2://publication/uuid/9FC2122D-6D49-4DC5-AC03-E353D5B3D1D1>.

JUSLIN, P. N.; LAUKKA, P. Expression, Perception, and Induction of Musical Emotions: A Review and a Questionnaire Study of Everyday Listening. Journal of New Music Research, v. 33, n. 3, p. 217–238, 2004. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/0929821042000317813>.

KIM, Y. E.; SCHMIDT, E. M.; MIGNECO, R.; MORTON, B. G.; RICHARDSON, P.; SCOTT, J.; SPECK, J. A.; TURNBULL, D. Music Emotion Recognition: a State of the Art Review. Information Retrieval, n. Ismir, p. 255–266, 2010. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.231.7740&rep=rep1&type=pdf%5Cnhttp://ismir2010.ismir.net/proceedings/ismir2010-45.pdf>.

LIDDY, E. D. Natural Language Processing. In: Center for Natural Language Processing, Anais...2001.

MACQUEEN, J. SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS J. In: Proceedings of the Fifth Symposium on Math, Statistics, and Probability, Anais...1967.

MANNING, Christopher D.; SCHÜTZE, Hinrich. Foundations of Statistical Natural Language Processing. 1. ed. [s.l.]: MIT Press, 1999.

MAYER, R.; RAUBER, A. Musical genre classification by ensembles of audio and lyrics features. Proc. Int. Society for Music Information Retrieval Conf. (ISMIR), n. Ismir, p. 675–680, 2011.

MCCALLUM, A.; NIGAM, K. A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization, p. 41–48, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9324&rep=rep1&type=pdf>.

ORENGO, V. M.; HUYCK, C. A stemming algorthm for the portuguese language. Proceedings of String Processing and Information Retrieval, 2001. SPIRE 2001., p. 186–193, 2001. Disponível em: <http://homes.dcc.ufba.br/~dclaro/download/mate04/Artigo Erick.pdf>.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP ’02, v. 10, n. July, p. 79–86, 2002. Disponível em: <http://dl.acm.org/citation.cfm?id=1118693.1118704%5Cnhttp://portal.acm.org/citation.cfm?doid=1118693.1118704>.

PATIL, T. R. Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. International Journal Of Computer Science And Applications, ISSN: 0974-1011, v. 6, n. 2, p. 256–261, 2013.

PATRA, B. G.; DAS, D.; BANDYOPADHYAY, S. Mood Classification of Hindi Songs based on Lyrics12th International Conference on Natural Language Processing, 2015.

PLATT, John C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, v. 73, n. 3, p. 356–358, 1998.

PORTER, M. Snowball: A language for stemming algorithms. Snowball, p. 1–15, 2001. Disponível em: <http://snowball.tartarus.org/texts/introduction.html>.

PRÓ-MÚSICA BRASIL. Mercado Fonográfico Mundial e Brasileiro em 2016. p. 14, 2017. Disponível em: <http://www.pro-musicabr.org.br/wp-content/uploads/2017/05/Mercado-de-Música-Global-e-Brasileiro-em-2016-FINAL.pdf>. Acesso em: 01 de abril de 2018.

RUSSELL, J. A. A circumplex model of affect. Journal of Personality and Social Psychology, v. 39, n. 6, p. 1161–1178, 1980.

SCHERER, CLEUDET DE ASSIS; DOMINGUES, A. Música e desenvolvimento infantil: reflexões sobre a formação do professor. IX ANPED SUL - Seminário de Pesquisa em Educação da Região Sul, p. 15, 2012.

SOUCY, P.; MINEAU, G. W. A simple KNN algorithm for text categorization. Proceedings 2001 IEEE International Conference on Data Mining, p. 647–648, 2001. Disponível em: <http://ieeexplore.ieee.org/document/989592/>.

VAGALUME. Documentação: Letras de Músicas, Disponível em: api.vagalume.com.br Acesso em: 07 de outubro de 2018.

VIGNOLI, F. Digital music interaction concepts: A user study. Proceedings of the International Society for Music Information Retrieval Conference (ISMIR’04), n. April, p. 415–420, 2004.

WAGSTAF, K.; CARDIE, C.; ROGERS, S.; SCHROEDL, S. Constrained K-means Clustering with Background Knowledge. Eighteenth International Conference on Machine Learning, p. 577–584, 2001.

WILBUR, W. J.; SIROTKIN, K. The automatic identification of stop words. Journal of Information Science, v. 18, n. 1, p. 45–55, 1992.

XIONG, Yu; SU, Feng; WANG, Qianqian. CONFERENCE, I. I. AUTOMATIC MUSIC MOOD CLASSIFICATION BY LEARNING CROSS-MEDIA RELEVANCE BETWEEN AUDIO AND LYRICS State Key Laboratory for Novel Software Technology Nanjing University, Nanjing 210023, China. n. July, p. 961–966, 2017.

YANG, K.-C.; HUANG, C.-H.; YANG, C.; LIN, Y.-S. A Study on Music Mood Detection in Online Digital Music Database. PACIS 2017 Proceedings, 2017. Disponível em: <http://aisel.aisnet.org/pacis2017/144>.

YING T. C; DORAISAMY Shyamala; ABDULLA N. L. Lyrics-Based Genre Classification Using Variant tf-idfWeighting Schemes. Journal of Applied Sciences. [S.L], 2015. Disponível em: <https://scialert.net/abstract/?doi=jas.2015.289.294 >. Acesso em: 01 de abril de 2018.

YONG, Z.; YOUWEN, L.; SHIXIONG, X. An Improved KNN Text Classification Algorithm Based on Clustering. Journal of Computers, v. 4, n. 3, p. 230–237, 2009. Disponível em: <http://academypublisher.com/ojs/index.php/jcp/article/view/669>.