

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ  
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Antonio Leandro Martins Candido

**Moderação inteligente de mensagens em ambientes virtuais de  
aprendizagem utilizados por alunos privados de liberdade**

Fortaleza – CE

20 de abril de 2021

Antonio Leandro Martins Candido

**Moderação inteligente de mensagens em ambientes  
virtuais de aprendizagem utilizados por alunos privados  
de liberdade**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Computação Aplicada

Orientador: Prof. Dr. Corneli Gomes Furtado Júnior

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

20 de abril de 2021

Dados Internacionais de Catalogação na Publicação  
Instituto Federal do Ceará - IFCE  
Sistema de Bibliotecas - SIBI  
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

---

C217m Candido, Antonio Leandro Martins.

Moderação inteligente de mensagens em ambientes virtuais de aprendizagem utilizados por alunos privados de liberdade / Antonio Leandro Martins Candido. - 2021.  
74 f. : il. color.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus Fortaleza, 2021.

Orientação: Prof. Dr. Corneli Gomes Furtado Júnior.

1. Classificação de Texto. 2. Ambiente Virtual de Aprendizagem. 3. Aluno Privado de Liberdade. I. Título.

CDD 004

---



Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)  
Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)  
Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Antonio Leandro Martins Candido

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

---

**Orientador: Prof. Dr. Corneli Gomes Furtado  
Júnior**

Instituto Federal de Educação, Ciência e Tecnologia do  
Ceará (IFCE)

---

**Prof. Dr. Amauri Holanda de Souza Júnior**  
Instituto Federal de Educação, Ciência e Tecnologia do  
Ceará (IFCE)

---

**Prof. Dr. Otávio Alcântara de Lima Júnior**  
Instituto Federal de Educação, Ciência e Tecnologia do  
Ceará (IFCE)

---

**Prof. Dr. Thiago Medeiros Barros**  
Instituto Federal de Educação, Ciência e Tecnologia do  
Rio Grande do Norte (IFRN)

Fortaleza – CE  
20 de abril de 2021

*Este trabalho é dedicado à minha família, em especial minha esposa Cleia e filhos, Sarah,  
Uziel e Mizael.*

# Agradecimentos

A Deus, por ter me abençoado e por estar sempre presente nesta nova etapa da minha vida.

Ao meu pai Antonio Candido do Carmo, por ser um exemplo para minha vida e por sempre me auxiliar da forma que pôde, nunca me desamparando ou deixando sozinho nesta jornada chamada vida.

À minha mãe Francisca Liduina Martins Candido, por sempre se preocupar com o meu bem estar e me manter diariamente em suas orações.

Carinhosamente, à minha esposa Cleia Candido e aos meus filhos Sarah Miryam, Antony Uziel e Leandro Mizael, por me ajudarem sempre que puderam, na maioria das vezes se esforçando para manter um ambiente familiar propício aos estudos.

À minha irmã Lidiane Adelaide, meu cunhado Luiz Carlos e meus sogros Maria da Conceição e José Dias, por sempre torcerem e me motivarem a ir adiante.

Especialmente, ao meu orientador Prof. Dr. Corneli Gomes Furtado Júnior, pela competência, paciência e contribuições para a realização da pesquisa proposta nesta dissertação. E, também, aos professores Dr. Otávio Alcântara, Dr. Amauri Holanda e Dr. Thiago Medeiros, pelas contribuições.

A todos os meus familiares pelo apoio e incentivo em todos os momentos e aos meus amigos e colegas, pelas trocas de conhecimentos, conselhos e incentivos, com destaque aos amigos Francisco Aislan, Vitor Guimarães, Carlos Estevão, Leonardo Brendo, Elias Paulino, Fabrício Freitas, Juliana Garça, Luan Sousa, Vitor Prata e Wesley Nojosa.

A todas as pessoas que, de alguma forma, colaboraram para que este trabalho fosse realizado.

*“Bem sei eu que tudo podes, e que nenhum dos teus propósitos podem ser impedidos.”*  
*(Jó 42:2)*

# Resumo

Ambientes virtuais de aprendizagem (AVAs) utilizados por alunos privados de liberdade devem impedir a comunicação indevida entre os envolvidos no processo de ensino e aprendizagem. As técnicas de aprendizagem de máquina podem ser empregadas para detectar padrões entre vários textos. E, devido ao número potencialmente elevado de mensagens em AVAs, faz-se necessária moderação inteligente dessa comunicação, com o objetivo de classificar os diálogos dos fóruns e verificar se são adequados ou inadequados. Com o intuito de solucionar esse problema, apresenta-se, nesta pesquisa, um modelo para moderação inteligente de mensagens em AVAs para alunos com restrição de liberdade. Para estabelecer as principais técnicas, algoritmos e métricas, realizou-se levantamento do estado da arte sobre classificação de textos e relacionaram-se os trabalhos com aspectos semelhantes ao deste trabalho. Além disso, descrevem-se, de forma sucinta, todas as técnicas utilizadas durante o processo de desenvolvimento do modelo. Para validar o modelo, é necessário possuir mensagens que representam os diálogos, tanto adequados, como inadequados. Logo, construiu-se um *dataset*, formado pelas mensagens obtidas de uma base de dados, de um curso à distância do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pelas mensagens encontradas em glossários, definidos por entidades governamentais, que descrevem as palavras, frases e gírias utilizadas por pessoas reclusas em sistemas prisionais. Com o propósito de avaliar as técnicas que devem ser utilizadas para implementação do modelo, analisaram-se os desempenhos dos classificadores Árvore de Decisão, Naive Bayes, K-NN, MLP, Regressão Logística e SVM, combinados com as técnicas de pré-processamento de remoção de *stopwords*, *stemming*, e as técnicas de extração de características *Count Vectorizer* e *Term Frequency – Inverse Document Frequency* (TF-IDF). Indica-se, nos resultados, que o melhor desempenho, relativo à classificação dos diálogos, alcança valores de até 96,58% para a métrica *F1-score*, com o uso do classificador MLP. Porém, ressalta-se que a melhor configuração do modelo refere-se à utilização do classificador SVM, que alcança um valor inferior de 93,76% para *F1-score*. Entretanto, seu resultado de tempo de execução é treze vezes mais rápido, um resultado importante, pois as mensagens precisam ser classificadas o mais breve possível, para que não haja atraso no uso do AVA. Por fim, conclui-se que é possível detectar os diálogos inadequados e tomar medidas para garantir o uso correto dos ambientes virtuais de aprendizagem.

**Palavras-chave:** Classificação de Texto. Ambiente Virtual de Aprendizagem. Aluno Privado de Liberdade.

# Abstract

Virtual learning environments (VLEs) used by students deprived of their liberty must prevent undue communication between those involved in the teaching and learning process. Machine learning techniques can be employed to detect patterns between various texts and due to the potentially large number of messages in VLEs, an intelligent moderation of this communication is necessary in order to classify the dialogues of the forums and verify if they are appropriate or inappropriate. In order to solve this problem, this research presents a model for intelligent moderation of messages in VLEs for students with restricted freedom. In order to establish the main techniques, algorithms and metrics, a survey of the state of the art on text classification was carried out and works with similar aspects to this research were related, besides, all the techniques used during the process of developing the model are briefly described. To validate the model, it is necessary to have messages that represent the dialogs, both adequate and inadequate. Thus, a dataset was built, formed by the messages obtained from a database of a distance learning course at the Federal Institute of Education, Science and Technology of Ceará and by the messages found in glossaries, defined by governmental entities, which describe the words, phrases and slang used by inmates in prison systems. In order to evaluate the techniques that should be used in the implementation of the model, the performances of the Decision Tree, Naive Bayes, K-NN, MLP, Logistic Regression and SVM classifiers, combined with the preprocessing techniques of removing keywords, stemming, and the techniques of feature extraction - Count Vectorizere and Term Frequency - Inverse Document Frequency (TF-IDF) were analyzed. The results indicate that the best performance in the classification of the dialogs reaches values up to 96.58% for the F1-score metric, with the use of the MLP classifier. However, the best model configuration is with the use of the SVM classifier, which reaches a lower value of 93.76% for the F1-score, but its execution time is thirteen times faster, an important result, because messages need to be classified as soon as possible, so that there is no delay in using the VLE. Finally, it is concluded that it is possible to detect the inappropriate dialogs and take measures to ensure the correct use of virtual learning environments.

**Keywords:** Text Classification. Virtual Learning Environment. Students deprived of their liberty.

# Lista de ilustrações

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – exemplo de uma matriz construída com o Count Vectorizer . . . . .                                | 23 |
| Figura 2 – gráfico de dispersão sem divisões . . . . .                                                      | 26 |
| Figura 3 – gráfico de dispersão com divisões . . . . .                                                      | 26 |
| Figura 4 – exemplo de uma árvore de decisão . . . . .                                                       | 27 |
| Figura 5 – classificação com K-NN . . . . .                                                                 | 30 |
| Figura 6 – modelo do neurônio . . . . .                                                                     | 33 |
| Figura 7 – perceptron multicamadas . . . . .                                                                | 34 |
| Figura 8 – curva da regressão logística . . . . .                                                           | 36 |
| Figura 9 – hiperplanos do SVM . . . . .                                                                     | 37 |
| Figura 10 – matriz de confusão . . . . .                                                                    | 39 |
| Figura 11 – etapas da pesquisa . . . . .                                                                    | 48 |
| Figura 12 – construção do <i>dataset</i> . . . . .                                                          | 49 |
| Figura 13 – processo de tokenização da mensagem . . . . .                                                   | 51 |
| Figura 14 – processo de remoção de <i>stopwords</i> da mensagem . . . . .                                   | 51 |
| Figura 15 – visualização de predominância das palavras . . . . .                                            | 52 |
| Figura 16 – processo de <i>stemming</i> na mensagem . . . . .                                               | 52 |
| Figura 17 – modelo de moderação inteligente de mensagens em ambientes virtuais<br>de aprendizagem . . . . . | 56 |
| Figura 18 – comparação de F1-score entre os classificadores . . . . .                                       | 63 |
| Figura 19 – comparação de tempo(ms) entre os classificadores . . . . .                                      | 63 |
| Figura 20 – métricas do MLP . . . . .                                                                       | 64 |
| Figura 21 – métricas do SVM . . . . .                                                                       | 65 |
| Figura 22 – matrizes de confusão da melhor iteração . . . . .                                               | 66 |

# Lista de tabelas

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Tabela 1 – resumo dos trabalhos de classificações de textos em geral. . . . .        | 44 |
| Tabela 2 – resumo dos trabalhos de classificações de textos em AVAs. . . . .         | 47 |
| Tabela 3 – distribuição dos cenários . . . . .                                       | 53 |
| Tabela 4 – resultado dos classificadores no cenário 1 com count vectorizer . . . . . | 58 |
| Tabela 5 – resultado dos classificadores no cenário 2 com count vectorizer . . . . . | 59 |
| Tabela 6 – resultado dos classificadores no cenário 3 com count vectorizer . . . . . | 60 |
| Tabela 7 – resultado dos classificadores no cenário 4 com count vectorizer . . . . . | 60 |
| Tabela 8 – resultado dos classificadores no cenário 1 com TF-IDF . . . . .           | 60 |
| Tabela 9 – resultado dos classificadores no cenário 2 com TF-IDF . . . . .           | 61 |
| Tabela 10 – resultado dos classificadores no Cenário 3 com TF-IDF . . . . .          | 61 |
| Tabela 11 – resultado dos classificadores no cenário 4 com TF-IDF . . . . .          | 61 |
| Tabela 12 – resultados do classificador MLP . . . . .                                | 62 |
| Tabela 13 – resultados do classificador SVM . . . . .                                | 62 |

# Lista de abreviaturas e siglas

|        |                                                              |
|--------|--------------------------------------------------------------|
| AVA    | Ambiente Virtual de Aprendizagem                             |
| CV     | Count Vectorizer                                             |
| EAD    | Educação à Distância                                         |
| GNU    | General Public License                                       |
| IA     | Inteligência Artificial                                      |
| IC     | Inteligência Computacional                                   |
| IFCE   | Instituto Federal de Educação, Ciência e Tecnologia do Ceará |
| KDT    | Knowledge Discovery from Text                                |
| KNN    | K-Nearest Neighbor                                           |
| LSA    | Latent Semantic Analysis                                     |
| ML     | Machine Learning                                             |
| MLP    | Multi Layer Perceptron                                       |
| MT     | Mineração de Textos                                          |
| NLP    | Natural Language Processing                                  |
| PSVM   | Proximal Suporte Vector Machine                              |
| RF     | Random Forest                                                |
| RI     | Recuperação da Informação                                    |
| SVM    | Support Vector Machine                                       |
| TF-IDF | Term Frequency - Inverse Document Frequency                  |
| WEKA   | Waikato Environment for Knowledge Analysis                   |

# Sumário

|            |                                          |           |
|------------|------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                        | <b>14</b> |
| <b>1.1</b> | <b>Motivação para a dissertação</b>      | <b>16</b> |
| <b>1.2</b> | <b>Descrição do problema</b>             | <b>16</b> |
| <b>1.3</b> | <b>Objetivos geral e específicos</b>     | <b>17</b> |
| <b>1.4</b> | <b>Produção científica</b>               | <b>18</b> |
| <b>1.5</b> | <b>Estrutura da dissertação</b>          | <b>19</b> |
| <b>2</b>   | <b>FUNDAMENTAÇÃO TEÓRICA</b>             | <b>20</b> |
| <b>2.1</b> | <b>Mineração de textos</b>               | <b>20</b> |
| <b>2.2</b> | <b>Pré-processamento</b>                 | <b>21</b> |
| 2.2.1      | Tokenização                              | 21        |
| 2.2.2      | Remoção de stopwords                     | 21        |
| 2.2.3      | Stemming                                 | 22        |
| <b>2.3</b> | <b>Extração de características</b>       | <b>22</b> |
| 2.3.1      | Count vectorizer                         | 22        |
| 2.3.2      | TF-IDF                                   | 23        |
| <b>2.4</b> | <b>Aprendizagem de máquina</b>           | <b>24</b> |
| 2.4.1      | Árvores de decisão                       | 25        |
| 2.4.2      | K-Nearest neighbors                      | 28        |
| 2.4.3      | Naive bayes                              | 30        |
| 2.4.4      | Redes neurais artificiais                | 32        |
| 2.4.4.1    | Multi layer perceptron                   | 33        |
| 2.4.5      | Regressão logística                      | 35        |
| 2.4.6      | Support vector machine                   | 36        |
| <b>2.5</b> | <b>Validação cruzada</b>                 | <b>38</b> |
| <b>2.6</b> | <b>Métricas</b>                          | <b>39</b> |
| <b>3</b>   | <b>TRABALHOS CORRELATOS</b>              | <b>42</b> |
| <b>3.1</b> | <b>Classificações de textos em geral</b> | <b>42</b> |
| <b>3.2</b> | <b>Classificações de textos em AVAs</b>  | <b>45</b> |
| <b>4</b>   | <b>METODOLOGIA</b>                       | <b>48</b> |
| <b>4.1</b> | <b>Universo da pesquisa</b>              | <b>48</b> |
| <b>4.2</b> | <b>Extração dos dados</b>                | <b>49</b> |
| <b>4.3</b> | <b>Pré-processamento</b>                 | <b>50</b> |
| <b>4.4</b> | <b>Extração de características</b>       | <b>53</b> |

|     |                                   |    |
|-----|-----------------------------------|----|
| 4.5 | Aprendizagem de máquina . . . . . | 54 |
| 4.6 | Conclusão . . . . .               | 56 |
| 5   | RESULTADOS . . . . .              | 58 |
| 6   | CONSIDERAÇÕES FINAIS . . . . .    | 67 |
|     | REFERÊNCIAS . . . . .             | 69 |

# 1 Introdução

O sistema penitenciário brasileiro é assunto de preocupação nacional, principalmente ao considerar que, com uma população de 209 milhões de habitantes, o Brasil tem mais de 800 mil pessoas vivendo em prisões (BRASIL, 2016). A situação da maioria dos presídios é de superlotação, de pouca verba e infraestrutura insuficiente.

Intuitivamente, é dedutível que a população prisional no Brasil tenha menor grau de escolaridade e os números confirmam essa impressão. Segundo o relatório do Ministério da Justiça (BRASIL, 2016), manter os jovens na escola, pelo menos até o término do ensino fundamental, pode ser uma das políticas de prevenção mais eficientes para a redução da criminalidade e, por conseguinte, da população prisional. Ainda segundo esse relatório, cerca de 53% da população prisional possui ensino fundamental incompleto.

A Lei n.º 7.210 de 11 de julho de 1984, Lei de Execução Penal, garante direitos aos presos do sistema penitenciário brasileiro, proporcionando ao infrator medidas assistenciais e de reabilitação. Uma dessas medidas pode ser o acesso à educação (BRASIL, 1984).

É notório o poder da educação como instrumento de desenvolvimento social e de garantia da qualidade de vida para membros de uma comunidade. O conhecimento torna-se ferramenta primordial para alavancar a economia e contribuir, positivamente, com a minimização das mazelas sociais. Além disso, a educação possui importante papel para a formação e o resgate do cidadão, por meio de sua inserção socioprofissional no mundo do trabalho e emprego, uma vez que oferece as ferramentas indispensáveis ao desenvolvimento de competências e habilidades, ampliando a sua chance de reinserção no mundo do trabalho e renda, e consequente resgate social e econômico.

A Educação a Distância (EaD), modalidade na qual alunos e professores estão separados, física ou temporalmente, surgiu em decorrência da necessidade de proporcionar educação aos segmentos da população, não adequadamente servidos pelo sistema tradicional de ensino. Essa modalidade de ensino, mediada por tecnologias digitais, pode ser utilizadas para a formação da população prisional como medida de ressocialização (FERREIRA, 2016).

A modalidade de EaD no sistema penitenciário é utilizada em todos os níveis e nas modalidades educacionais formais e profissionais, com possibilidade de acesso às tecnologias e à informação, proporcionando oportunidade educacional apropriada, considerando as características dos educandos, seus interesses, as especificidades, as situações do encarceramento, os perfis dos estudantes, as faixas etárias, suas diversidades e culturas, e, ainda, o Princípio da Equidade, da Diferença e da Proporcionalidade. O que pode garantir, entre tantos aspectos na sociedade prisional, a oportunidade de se construir conhecimento

significativo e tangível para a realidade dos sujeitos educandos presos nas escolas prisionais por meio da EaD (FERREIRA, 2016).

O emprego da EaD em unidades prisionais deve ser realizado com cautela, por conta das ferramentas assíncronas que podem ser utilizadas de forma inadequada para a realização de comunicações indevidas, por parte dos alunos reclusos, pois não é possível o acompanhamento contínuo e ininterrupto por parte dos professores. Nesse caso, se faz necessária a implementação de técnicas de Inteligência Artificial (IA) capazes de auxiliar, indiretamente, os professores no controle da utilização dessas ferramentas (AKYUZ; KURT, 2010).

As técnicas de IA estão sendo cada vez mais aplicadas à EaD. Além disso, também houve a evolução dos AVAs (JAIN, 2015). Os atuais AVAs possuem ferramentas capazes de facilitar o desenvolvimento do conhecimento e, também, outras ferramentas que podem auxiliar os professores na avaliação da interação e do desempenho dos alunos. Os AVAs possuem espaços reservados para discussões acerca das disciplinas ministradas nessa modalidade, como fóruns e *chats* (DAVID et al., 2014).

Os *chats* e fóruns de discussões possuem papéis importantes no processo de ensino e aprendizagem à distância, por proporcionarem o aumento de interação nas disciplinas ministradas pelos professores.

Entretanto, é necessário que haja um efetivo acompanhamento, tanto direto como indireto, pois se tratam de ferramentas síncronas, como os *chats*, em que a comunicação é feita em tempo real, e ferramentas assíncronas, como os fóruns, em que as mensagens são registradas e podem ser respondidas em momentos oportunos, porém permanecem visíveis para outros alunos no ambiente virtual (LOBATO, 2012).

O acompanhamento indireto, sem a participação contínua e ativa do professor, pode ocasionar a sobrecarga de informações. Uma alternativa para essa problemática gerada nos AVAs constitui as automatizações dos sistemas para auxiliar os professores nessas atividades (RIGO et al., 2013).

No fim da década de 90, surgiu a técnica de Recuperação da Informação (RI). Baeza e Ribeiro Neto (1999) relatam, desde então, que esta tem sido foco de diversas pesquisas, buscando técnicas automatizadas capazes de organizar e pesquisar documentos de textos em linguagem natural (não estruturados), de maneira a obter, com base na consulta de um usuário, a informação desejada para fins diferenciados, tais como bibliotecas, armazenamento e disponibilidade de informações para setores específicos, disseminação de informações, seleção de documentos etc. Com a evolução da área da RI, surgiu a Mineração de Textos (do inglês *Text Mining*). Segundo Tan (1999), a área de Mineração de Textos trata da extração de padrões ou conhecimentos interessantes com base em um conjunto de documentos textuais.

A Mineração de Textos permite a transformação de dados textuais não estruturados em atributos estruturados, após a aplicação de algoritmos de Mineração de Dados. O seu uso permite a extração de conhecimento novo, a partir de dados brutos não estruturados (REZENDE; MARCACINI; MOURA, 2011). Nos últimos quinze anos, as tarefas de manipulação de textos baseados em seus conteúdos ganharam destaque na área de Sistemas de Informação, devido ao aumento da disponibilidade de documentos em formato digital e a consequente necessidade de acessá-los e organizá-los de maneiras mais flexíveis (SOARES, 2004).

O processo de classificar conteúdos textuais é a atividade de rotular textos em linguagem natural com categorias temáticas a partir de um conjunto pré-definido. A abordagem dominante para esse problema é baseada em técnicas de Aprendizagem de Máquina (do inglês *Machine Learning*), em que se implementa um classificador que, a partir de um conjunto de textos categorizados, ou seja, rotulados de acordo com a categoria, por exemplo, economia, artes ou tecnologia, consegue aprender as características de cada categoria (SOARES, 2004).

Uma das tarefas dos tutores dos AVAs é verificar se a interação dos discentes, com base no tema de discussão proposto, está sendo adequada. Como ferramenta auxiliar, é possível utilizar as técnicas de aprendizagem de máquina para classificar os textos e, sugerir aos tutores, quais discentes estão com os seus diálogos adequados ou inadequados. Como benefícios da utilização dessas técnicas, obtém-se a celeridade das avaliações e o suporte aos tutores, com resultados reais, obtidos por meio de um classificador, amparando suas decisões finais às avaliações.

## 1.1 Motivação para a dissertação

Os diversos AVAs disponíveis, Moodle (2021), TelEduc (2021), Amadeus (2021), Dokeos (2021), não ofertam ferramentas que permitam a moderação inteligente das mensagens, existentes em *chats* e fóruns, por exemplo, realizadas entre alunos, tutores e professores. Assim, o uso do AVA pode ser desvirtuado para realização de comunicação indevida (comunicação externa ao presídio ou entre presos, por exemplo), facilitando a prática de delitos, mesmo que o aluno esteja sob a tutela do Estado.

## 1.2 Descrição do problema

Em um sistema de moderação inteligente, toda a comunicação relacionada à aula ou à disciplina deve ocorrer normalmente, enquanto as não relacionadas, ou consideradas inadequadas, devem ser impedidas de chegar ao destinatário. Ao tutor deve ser possível o

tratamento de falsos-positivos<sup>1</sup>.

Incorporado ao AVA, um modelo de moderação inteligente pode contribuir com a formação e desenvolvimento de competências, com a ressocialização de alunos com restrição de liberdade e, de maneira indireta, com a redução do número de reincidências em crimes.

O modelo deve utilizar técnicas de aprendizagem de máquina, capazes de classificar as mensagens enviadas pelos alunos privados de liberdade, identificar as que são inadequadas em relação ao que está sendo discutido nos fóruns e impedir o diálogo indevido entre os alunos do sistema prisional. Destaca-se que o modelo deve ser capaz de classificar em um breve período de tempo, não interferindo no fluxo do uso do fórum de um AVA.

Os algoritmos de aprendizagem de máquina precisam ser utilizados para a classificação de um conjunto de diálogos, formado pelas mensagens obtidas de uma base de dados, de um curso à distância do Instituto Federal de Educação, Ciência e Tecnologia do Ceará; de glossários, elaborados por entidades governamentais, Ministério Público do estado do Ceará, Polícia Militar do estado do Ceará e de Minas Gerais; e de publicações de trabalhos científicos e notícias (OLIVEIRA, 2006; LIMA, 2007; BIONDI, 2009; CEARÁ, 2018; ADORNO, 2019). Os glossários mencionados anteriormente definem as palavras, frases e gírias utilizadas por pessoas reclusas em sistemas prisionais.

Diante do cenário apresentado para esta dissertação, busca-se responder as seguintes questões: é possível moderar os diálogos realizados por alunos privados de liberdade, incentivar o ensino à distância em ambientes prisionais, com utilização deste modelo e, garantir a segurança, evitando o uso incorreto dessas ferramentas educacionais?

### 1.3 Objetivos geral e específicos

O objetivo deste trabalho é propor um modelo inteligente para a moderação das trocas de mensagens, entre os atores que compõem o ensino a distância para pessoas restritas de liberdade.

Os objetivos específicos são:

- realizar um levantamento do estado da arte sobre classificação de textos;
- relacionar os trabalhos com aspectos semelhantes ao da pesquisa proposta nesta dissertação;
- elencar as ferramentas e técnicas que devem ser utilizadas para o desenvolvimento do modelo;
- construir um *dataset* contendo os diálogos que serão classificados;

<sup>1</sup> Falsos-positivos são mensagens adequadas classificadas como inadequadas. Neste trabalho a classe de mensagens inadequadas são a classe 1, com a prioridade.

- aplicar as técnicas de pré-processamento e extração de características;
- aplicar os algoritmos de aprendizagem de máquina,
- avaliar o desempenho do modelo com a utilização das técnicas de pré-processamento e extração de características;
- avaliar o desempenho dos diferentes classificadores;
- avaliar os resultados em relação ao balanceamento da base de dados;
- definir a melhor configuração para o modelo.

## 1.4 Produção científica

Durante o período de desenvolvimento deste projeto de mestrado, os seguintes trabalhos científicos foram aceitos e publicados em capítulos de livros, eventos e jornais científicos.

### Capítulos de livro

- CANDIDO, Antonio Leandro Martins; FURTADO JÚNIOR, Corneli Gomes; FREITAS, Francisco Aislan da Silva. Moderação inteligente de mensagens em ambientes virtuais de aprendizagem para alunos privados de liberdade. In: ARAÚJO FILHO, Patrício Moreira de; NERES, Raimundo Luna; MARTINS, Ernane Rosa; BRANDÃO, Raimundo José Barbosa (org.). **Educação 4.0: tecnologias educacionais**. São Luis: Pascal, 2021. v. 5, cap. 14, p. 172, ISBN 978-65-86707-42-7.

### Eventos

- CANDIDO, A. L. M.; FURTADO JÚNIOR, C. G.; FREITAS, F. A. S., 2019. **Moderação inteligente de mensagens em ambientes virtuais de aprendizagem para alunos privados de liberdade**. CBIE 2019. XXX Simpósio Brasileiro de Informática na Educação, p. 100-110.
- FREITAS, Aislan; ALBUQUERQUE, Monck; SALES, Gilvandenys Leite; MEDEIROS, Cláudio; CANDIDO, Antonio Leandro Martins; FILHO, Pedro Rebouças, 2018. **O Uso de Redes Neurais Artificiais para Análise da Presencialidade em um Fórum LV**. CBIE 2018. XXIX Simpósio Brasileiro de Informática na Educação, p. 1903.

### Jornais Científicos

- CANDIDO, Antonio Leandro Martins; FURTADO JÚNIOR, Corneli Gomes; FREITAS, Francisco Aislan da Silva, 2019. **Moderação inteligente de mensagens em ambientes virtuais de aprendizagem para alunos privados de liberdade**. Brazilian Journal of Development, v. 5, n. 12, p. 31397-31410, ISSN 2525-8761.

## 1.5 Estrutura da dissertação

Este trabalho está organizado da seguinte maneira: após a Introdução realizada no Capítulo I, a Fundamentação Teórica é apresentada no Capítulo II, os Trabalhos Correlatos são apresentados no Capítulo III, a Metodologia é apresentada no Capítulo IV, os Resultados são apresentados no Capítulo V e, por fim, as Considerações Finais são apresentadas no Capítulo VI.

## 2 Fundamentação teórica

Neste capítulo é apresentado o estado da arte sobre o tema classificação de textos e são descritas as técnicas que podem ser utilizadas para essa tarefa.

### 2.1 Mineração de textos

Alguns autores divergem quanto à origem das técnicas de mineração de textos (MT), uns consideram que as técnicas de mineração de textos derivaram das técnicas de mineração de dados e procuram extrair informações que sejam úteis de dados que não estão previamente estruturados e que a princípio são difíceis de tratar, recuperando informações importantes contidas em bases de dados textuais (ARANHA; VELLASCO, 2006) (FELDMAN; SANGER, 2007).

Já outros autores afirmam que a mineração de textos teve sua origem relacionada à área de Descoberta de conhecimento em Textos (do inglês *Knowledge Discovery from Text* - *KDT*), área que analisa grandes volumes de dados de maneira inteligente e automatizada com a intenção de encontrar algum conhecimento útil. A análise qualitativa e quantitativa desses grandes volumes de dados textuais, a busca por informações específicas em documentos e a melhor compreensão de textos, constituem elementos essenciais da KDT (AMBRÓSIO; MORAIS, 2007).

Lopes (2004) define, para a mineração de texto, a finalidade de extração de padrões relevantes e não triviais ou a descoberta de conhecimento com base em elementos textuais não estruturados. Aranha e Passos (2006) descrevem o estudo da mineração de textos como a extração de regularidades, padrões e tendências de conteúdos textuais, geralmente para objetivos específicos. A aplicação da mineração de textos compreende a utilização de algoritmos computacionais que analisam e identificam informações, em textos, frases ou palavras, que, normalmente, não seriam recuperadas por meio da utilização das técnicas comuns de consulta, considerando-se o formato não estruturado em que estes estão descritos (AMBRÓSIO; MORAIS, 2007).

De acordo com Silva (2010), dados estruturados são caracterizados por conterem uma estrutura lógica de armazenamento, como um banco de dados relacional. Os dados não estruturados não contêm nenhum tipo de estrutura lógica de armazenamento, aspecto comum entre conteúdos textuais. A mineração de textos pode ser considerada um campo multidisciplinar por envolver várias áreas de conhecimento, entre elas a informática, estatística, linguística e ciência cognitiva (ARANHA, 2007). Estima-se que cerca de 80% das informações disponíveis nas organizações encontram-se no formato de textos (HE;

ZHA; LI, 2013).

Mineração de Textos são técnicas baseadas em uma série de passos bem definidos, mas que variam de acordo com o estilo de abordagem do algoritmo, segundo Moraes e Ambrósio (2007). Basicamente esses passos são o pré-processamento, que engloba a abordagem e preparação dos dados; logo em seguida, a indexação e normalização, que identificam similaridades nas palavras; a remoção das palavras vazias (*stopwords*); e, então, o cálculo de relevância das palavras e a seleção de termos. Após esses passos, é possível classificar os textos e analisar os resultados (FELDMAN; SANGER, 2007).

O Processamento de Linguagem Natural (PLN), a aprendizagem de máquina, Recuperação da informação (RI), Inteligência Computacional (IC) e a ciência cognitiva são outras técnicas, além das que a mineração de textos adquiriu da mineração de dados, que também contribui para esse campo de pesquisas em bases textuais (CHOWDHURY, 2003).

## 2.2 Pré-processamento

### 2.2.1 Tokenização

A tokenização realiza a divisão das mensagens em pedaços. Cada termo ou palavra se torna um *token*, que são as unidades individuais e elas podem ser palavras, fonemas ou até frases completas, porém, neste trabalho é escolhido que sejam, apenas, palavras individuais.

Essa escolha da divisão dos *tokens* se baseia em outros trabalhos da literatura científica, pois, na análise de textos, os *tokens* são, na maioria das vezes, apenas palavras (AMBRÓSIO; MORAIS, 2007; ARANHA; VELLASCO, 2006; AZEVEDO; BEHAR; REATEGUI, 2011; CATAE, 2012; ISLAM; ELAHI; AHMED, 2017; QU et al., 2018).

### 2.2.2 Remoção de stopwords

Em pesquisas de mineração de textos, frequentemente ouvimos o termo “*stopwords*”, que são as palavras irrelevantes, por serem um conjunto de palavras comumente usadas em qualquer idioma.

O motivo pelo qual as palavras irrelevantes são críticas para a classificação de textos é que, se removermos as palavras que são comumente usadas em um determinado idioma, podemos nos concentrar nas palavras importantes.

### 2.2.3 Stemming

O *stemming* é importante na compreensão e no processamento de linguagem natural. Com a aplicação da técnica de *stemming*, as palavras flexionadas ou derivadas são reduzidas à raiz, para facilitar a identificação de similaridade entre as palavras no processo de classificação.

## 2.3 Extração de características

As máquinas, por mais avançadas que sejam, não são capazes de entender palavras e frases da mesma maneira que os humanos. Para tornar os documentos de textos mais entendíveis para os computadores, eles devem, primeiramente, ser convertidos em alguma estrutura numérica. Existem algumas técnicas usadas para conseguir isso e, neste trabalho, utilizam-se as técnicas de modelos de espaço vetorial, também conhecidos como modelos de saco de palavras (do inglês *Bag of Words*). No processamento de linguagem natural, como todos os preditores são extraídos do texto, a limpeza de dados, o pré-processamento e a engenharia de recursos têm um impacto ainda mais significativo para o desempenho do modelo.

*Bag of Words* é uma abordagem intuitiva, a sua construção segue os seguintes passos:

- dividir os documentos em tokens (palavras / termos) seguindo algum tipo de padrão;
- atribuir um peso a cada token proporcional à frequência com que é exibido no documento;
- criar uma matriz de termos do documento com cada linha representando um documento (frase / mensagem) e cada coluna representando um token.

Os algoritmos vetorizadores fornecidos pela biblioteca de aprendizagem de máquina de código aberto para a linguagem de programação *Python*<sup>1</sup>, *Scikit-Learn*, são bastante confiáveis, eles permitem executar todas as etapas acima de uma só vez de maneira eficiente. Neste trabalho, utilizam-se dois algoritmos de vetorização de textos, Count Vectorizer (CV) e TF-IDF.

### 2.3.1 Count vectorizer

Count Vectorizer (CV) conta o número de vezes que um token aparece no documento e usa esse valor como peso. Com a sua utilização, a base de dados é convertida em uma matriz esparsa com a frequência de cada termo (palavra) representada por tokens. São utilizados parâmetros para definir um limite máximo e mínimo de frequência para os

---

<sup>1</sup> (BURIOL; ARGENTA, 2009)

termos a serem selecionados (BAKER, 2019). Um exemplo de matriz construída pelo CV pode ser vista na Figura 1, em que cada termo recebe a sua quantidade de frequência em cada uma das mensagens.

Figura 1 – exemplo de uma matriz construída com o Count Vectorizer

|           | Termo1 | Termo2 | Termo3 | ... | TermoN |
|-----------|--------|--------|--------|-----|--------|
| mensagem1 | 5      | 6      | 1      |     | 0      |
| mensagem2 | 0      | 0      | 0      |     | 1      |
| :         |        |        |        |     |        |
| mensagemN | 1      | 2      | 2      |     | 0      |

Fonte: autor.

O limite máximo de frequência dos termos nas mensagens é de até 70%, para não serem selecionados termos comuns em ambas as classes e o limite mínimo é de, pelo menos, duas frequências, com o propósito de remover termos com pouca relevância para a classificação. Esses limites foram definidos após vários testes, quando se alcançaram melhores resultados na classificação.

### 2.3.2 TF-IDF

TF-IDF é a combinação de duas técnicas estatísticas, o TF (do inglês *Term Frequency*) e o IDF (do inglês *Inverse Document Frequency*). Essa medida é frequentemente utilizada na mineração de textos. O TF mede o quão frequente um termo ocorre em um texto. Pode ser representado pela divisão entre a quantidade de vezes que um termo aparece no texto e o total de termos que o texto possui.

O IDF mede quanto um termo é importante, visto que o TF pode dar importância para termos como "de", "para", "como", caso não seja utilizada a técnica de remoção das *stopwords* e por serem termos bastante comuns, mas com pouca importância. O IDF procura dar um peso maior para termos que ocorrem mais raramente (BAKER, 2019).

O TF-IDF atribui um peso a cada token que demonstra o quão recorrente esse termo é em todo o conjunto do texto. O valor atribuído pelo TF-IDF ao termo aumenta, proporcionalmente, à medida que o número de ocorrências desse termo em um documento ou texto também aumenta. No entanto, esse valor é equilibrado pela frequência do termo no conjunto de todos os documentos ou textos. Isso auxilia a distinguir o fato da ocorrência de alguns termos serem, geralmente, mais comuns que outros, facilitando a classificação, como demonstrado na Equação 2.1.

$$W_{ij} = tf_{ij} \times \log \frac{N}{df_i} \quad (2.1)$$

Vale ressaltar que  $tf_{i,j}$  é número de frequência do termo  $i$  na mensagem  $j$ ,  $df_i$  é o número de mensagens contendo o termo  $i$  e  $N$  o número total de mensagens.

## 2.4 Aprendizagem de máquina

Os estudos sobre Aprendizagem de Máquina ganham cada vez mais destaque na comunidade da computação. O aumento contínuo da quantidade de dados disponíveis em todas as áreas organizacionais e sociais fornecem razões capazes de justificar o conceito de que a análise inteligente de dados deve se tornar ainda mais difundida e uma ferramenta chave para o desenvolvimento tecnológico (SMOLA; VISHAWNATHAN, 2008).

Aprendizado de máquina pode ser definido como uma área de estudo da IA que produz algoritmos capazes de proporcionar conhecimento a computadores utilizando-se de dados de eventos passados. O aprendizado de máquina é capaz de evidenciar características e padrões que dificilmente seriam perceptíveis de maneira clara para um humano, utilizando técnicas triviais de análise de dados (AMARAL, 2016). O estudo dos algoritmos de aprendizado de máquina pode ser dividido em três grupos: algoritmos de aprendizado supervisionado, não supervisionado e por reforço.

A técnica de aprendizagem de máquina adquire informações de um meio externo e junta-as em um modelo desenvolvido. Por outro lado, técnicas com regras fixas podem possuir maior poder de escala e automação, entretanto possuem manutenções custosas e manuais.

Classificar e ordenar um grande volume de dados pode ser uma tarefa inviável para ser feita manualmente. É possível mapear um grande volume de dados em casos futuros após uma observação prévia com a utilização de dispositivos inteligentes, desenvolvidos com a capacidade de inferir propriedades e atributos a informações antes desconhecidas (MITCHELL, 1997).

As técnicas de recuperação da informação e aprendizagem de máquina possuem semelhanças. Tratando-se, especificamente, dos classificadores<sup>2</sup> de textos, os algoritmos mais comuns, que são utilizados nesses dois campos de pesquisas, são:

- Árvores de decisão;
- K-vizinhos mais próximos (do inglês *K Nearest Neighbor* - *KNN*);
- *naive bayes*<sup>3</sup>;

<sup>2</sup> Classificadores são dispositivos inteligentes utilizados para inferir propriedades e atributos sobre informações até então desconhecidas (MITCHELL, 1997).

<sup>3</sup> (MOONEY; BENNETT; ROY, 1998)

- máquina de vetores de suporte<sup>4</sup> (do inglês, *support Vector Machine* - *SVM*);
- rede Neural Artificial Perceptron Multi Camadas (do inglês *multiLayer Perceptron* - *MLP*);
- regressão logística.

Em diversas pesquisas, se faz necessário realizar comparações entre diferentes métodos de classificação para determinar qual é o melhor em casos específicos (DUMAIS, 1998; HUANG; LU; LING, 2003; YANG; ZHANG; KISIEL, 2003). Os algoritmos possuem características diferentes, custos diferentes, dificuldades de implementação distintas, que devem ser consideradas no momento da escolha. Muitas vezes essas características sobressaem a importância da precisão nos resultados dos algoritmos (SEBASTIANI, 2002; MANNING; RAGHAVAN; SCHUTZE, 2008).

### 2.4.1 Árvores de decisão

As árvores de decisão, também conhecidas por indução de regras, são um dos modelos de aprendizado de máquina mais comuns e utilizados. São capazes de fornecer uma fácil interpretação e conseguem demonstrar o caminho que deve ser percorrido até o objetivo desejado. Elas pertencem ao grupo de aprendizagem de máquina supervisionado e podem ser utilizadas para regressões e para classificações (GAMA, 2004).

As árvores de decisão são compostas por nós e folhas que se relacionam aos atributos dos dados. No seu processo de aprendizado, a árvore assimila novas informações e realiza inferências sobre a classificação dos dados (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007). São utilizadas como ferramentas de suporte à tomada de decisões e usam um gráfico no formato de árvore, demonstrando, visivelmente, as condições e as probabilidades para se chegar ao objetivo esperado.

Nos modelos de árvores de decisão, os resultados são fáceis de explicar e de serem convertidos em decisões práticas, diferentemente de outros algoritmos de aprendizagem de máquina e também redes neurais artificiais, em que modelos podem ser complexos e difíceis de explicar.

As árvores de decisão fazem divisões dos dados em subconjuntos até conseguir que os subconjuntos fiquem mais puros. O objetivo é fazer com que os subconjuntos fiquem com menos classes possíveis para escolhas, o ideal é atingir o estágio em que apenas é possível induzir uma classe ao subconjunto. Porém, todo esse processo de construção da árvore pode exigir um poder computacional maior do que em outros algoritmos.

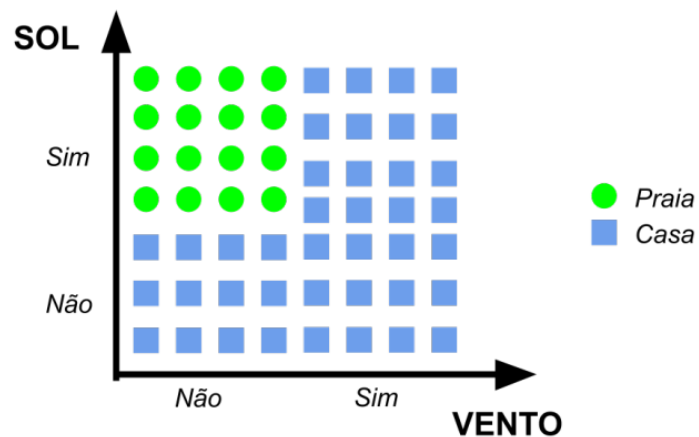
Uma forma de trabalhar matematicamente com a pureza é por meio da análise da entropia e do ganho de informação. Vamos imaginar um exemplo para abordar o conceito de árvores de decisão. Suponhamos que alguém possui uma base de dados com pessoas

---

<sup>4</sup> (MA; GUO, 2014)

que vão à praia ou decidem ficar em casa. O que determina se as pessoas vão à praia são duas características: sol e vento. Quando há sol e não há vento, as pessoas vão à praia. Caso contrário, as pessoas ficam em casa. Na Figura 2, ilustra-se essa representação em um gráfico de dispersão.

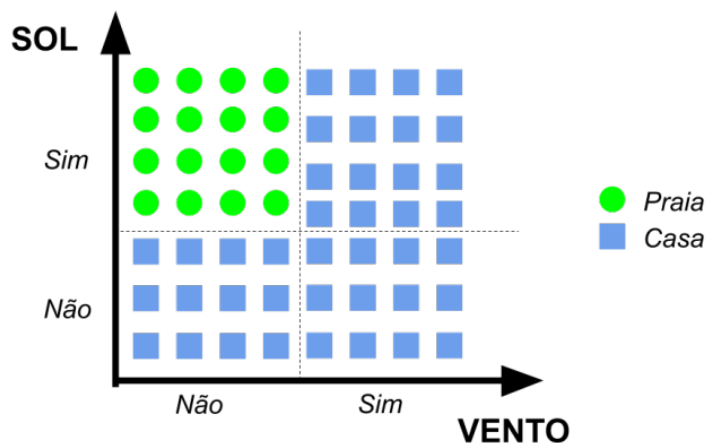
Figura 2 – gráfico de dispersão sem divisões



Fonte: autor.

Para as características mencionadas, somente é possível valores correspondentes a alguma das duas classes. Para a variável sol, nesse cenário, expõe-se se há sol ou não sol, o mesmo se aplica à variável vento (se há vento ou não). Já as classes são duas, praia ou casa. As árvores de decisão, ao final de sua aplicação, criam diversas linhas para tentar separar os dados. Nesse cenário se torna fácil aplicar duas retas capaz de separar as pessoas que vão à praia e as que ficam em casa. Na Figura 3, ilustra-se essa representação.

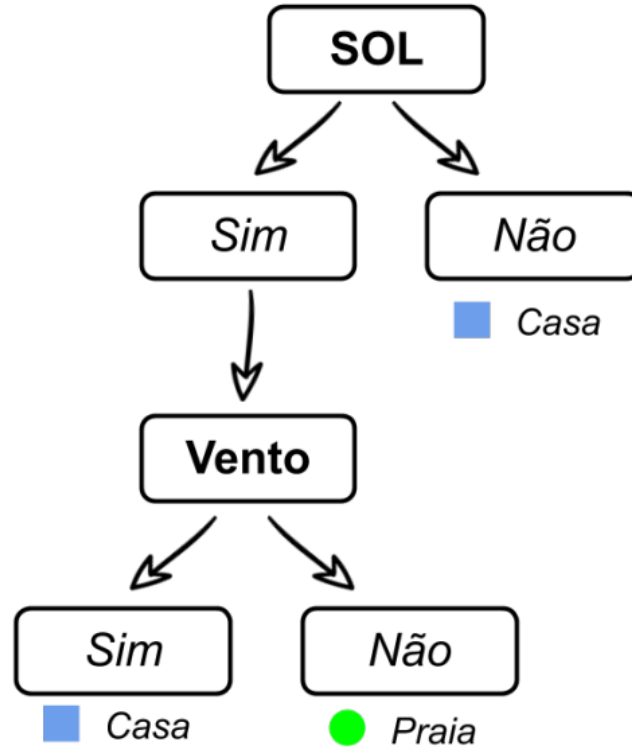
Figura 3 – gráfico de dispersão com divisões



Fonte: autor.

Com base nessas retas, é gerada uma árvore, similar a um fluxograma, em que se torna mais fácil a compreensão do caminho percorrido pelos dados até a classificação de novas pessoas. Na Figura 4, é ilustrada a árvore de decisão final.

Figura 4 – exemplo de uma árvore de decisão



Fonte: autor.

Quando é gerada uma árvore de decisão, os novos registros de dados, nesse caso, quando novas pessoas pensarem em ir à praia, passarão por esse fluxo, apresentado na árvore de decisão criada. A geração da árvore de decisão constitui processo recursivo, em que será calculado a entropia, os pesos e o ganho da informação. Na Equação 2.2, está representado o cálculo da entropia:

$$Entropia(S) = -\rho_{(+)} \log_2 \rho_{(+)} - \rho_{(-)} \log_2 \rho_{(-)} \quad (2.2)$$

Em que  $S$  é uma amostra,  $\rho_{(+)}$  é a proporção da classe positiva e  $\rho_{(-)}$  é a proporção da classe negativa.

Com a entropia calculada, o próximo passo é obter o valor do peso. Na Equação 2.3, ilustram-se os cálculos:

$$Peso = \frac{\text{numero de amostras do filho}}{\text{numero de amostras do pai}} \quad (2.3)$$

Com o valor da entropia e dos pesos, pode-se obter o ganho da informação. Na Equação 2.4, ilustra-se essa representação:

$$\text{Ganho da informação} = Entropia_{pai} - \sum Peso_{filho} * Entropia_{filho} \quad (2.4)$$

A respeito de um problema de classificação, inicialmente calcula-se a entropia das classes. Posteriormente, a entropia e o peso de uma característica e, assim, o ganho da informação. Ao obter o ganho de informação de cada característica, o maior ganho se torna "filho" do nó principal e a recursividade é iniciada para as demais características, seguindo as mesmas etapas já citadas.

Ao contrário dos demais classificadores que usam a representação vetorial de documentos no espaço contínuo, a modelagem de árvores de decisão depende do uso de atributos discretos. Uma generalização do algoritmo é considerar cada palavra como um atributo com dois possíveis valores: positivo e negativo. Um atributo é considerado negativo quando o termo não está presente no documento. E positivo quando o documento faz referência a ele (CATAE, 2012).

A dificuldade em classificar documentos está relacionada à grande quantidade de atributos. Em geral, o número de palavra é da ordem de dezenas de milhares. Por isso, um passo importante do algoritmo é a seleção dos atributos a serem utilizados na árvore (CATAE, 2012).

Existem diferentes métodos para determinar a forma de crescimento da árvore. O problema de aprendizado ótimo é conhecido como da classe NP-completo (HYAFIL; RIVEST, 1976) e, por isso, os algoritmos são heurísticos. O objetivo é encontrar pontos de mínimos locais, variando entre cálculo de entropia, chi-square ou diferentes medidas. Os métodos ID3 e C4.5 são os mais utilizados (QUINLAN, 1993).

### 2.4.2 K-Nearest neighbors

No reconhecimento de padrões, o algoritmo K-Nearest Neighbors (K-NN) é um método não paramétrico usado para classificação e regressão. Segundo Altman (1992), nos dois casos, a entrada consiste nos k exemplos de treinamento mais próximos no espaço de recursos. A saída depende se o K-NN é usado para classificação ou regressão:

- Na classificação com K-NN, a saída é uma associação de classe. Um objeto é classificado pelo voto de pluralidade de seus vizinhos, sendo o objeto atribuído à

classe mais comum entre os  $k$  vizinhos mais próximos ( $k$  é um número inteiro positivo, geralmente pequeno). Se  $k = 1$ , o objeto é simplesmente atribuído à classe do único vizinho mais próximo.

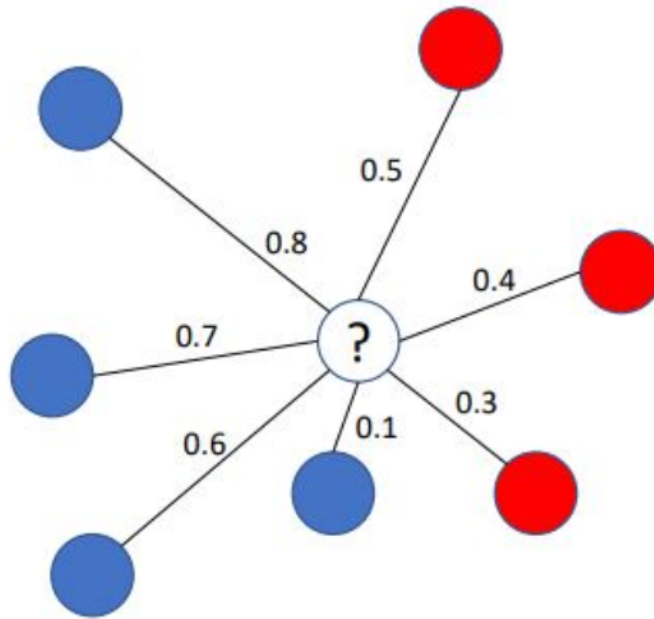
- Na regressão por K-NN, a saída é o valor da propriedade para o objeto. Este valor é a média dos valores de  $k$  vizinhos mais próximos.

K-NN é um tipo de aprendizado baseado em instância, ou aprendizado lento, em que a função é aproximada, apenas, localmente e todo o cálculo é adiado até a classificação. Tanto para classificação quanto para regressão, uma técnica útil pode se atribuir pesos às contribuições dos vizinhos, para que os vizinhos mais próximos contribuam mais com a média do que os mais distantes. Por exemplo, um esquema de ponderação comum consiste em atribuir a cada vizinho um peso de  $\frac{1}{d}$ , em que  $d$  é a distância do vizinho.

Os vizinhos são obtidos de um conjunto de objetos para os quais a classe (para a classificação K-NN) ou o valor da propriedade do objeto (para a regressão K-NN) é conhecida. Isso pode ser pensado como o conjunto de treinamento para o algoritmo, embora nenhuma etapa explícita do treinamento seja necessária. Uma peculiaridade do algoritmo K-NN é que ele é sensível à estrutura local dos dados (ALTMAN, 1992).

Ao encontrar uma nova amostra, é realizado o cálculo de distância dessa amostra para as demais amostras em seu espectro; se o valor de  $k$  for definido com o valor um, essa nova amostra seria classificada como azul, já que o "vizinho" mais próximo dela possui distância de 0,1, consoante Figura 5. Porém, se o valor de  $k$  for igual a dois, existem duas amostras que são próximas, uma amostra na cor azul (0,1) e uma amostra na cor vermelha (0,3). Se houver empate, é possível classificar tanto de uma cor com outra, mas pode-se utilizar um critério de desempate com base na menor distância.

Figura 5 – classificação com K-NN



Fonte: autor.

Normalmente se recomenda utilizar o número de vizinhos com um inteiro ímpar, para evitar situações de empate e para que haja uma classificação real mais próxima dos dados. Vamos imaginar que o valor de  $k$  é definido como 3, como menor distância, é apresentado um azul (0,1), dois vermelhos com distâncias 0,3 e 0,4 respectivamente. Nesse cenário a nova amostra seria classificada como vermelha, já que são apresentados dois vermelhos como menor distância (AHA; KIBLER; ALBERT, 1991).

O K-NN implementado utiliza a distância euclidiana, logo:

$$DistânciaEuclidiana(x, y) = \sqrt{\sum_i^n (x_i - y_i)^2} \quad (2.5)$$

A menor distância entre as amostras selecionadas pelo valor de  $k$  e entre o novo registro é o que determinará a qual classe ele fará parte.

Além da distância euclidiana existem outros cálculos de distâncias que podem ser utilizados no K-NN, como a Minkowsky, Chebyshev e outras.

### 2.4.3 Naive bayes

O Naive Bayes é um classificador probabilístico e bastante utilizado para a classificação de textos, sua implementação tem por base o Teorema de Bayes e é considerada uma

das mais fáceis e didáticas (RENNIE et al., 2003). O Naive Bayes é considerado um dos mais eficientes a respeito de processamento e precisão para classificação de novas amostras (GOMES; CASTRO NETO; HENRIQUES, 2013). Eles classificam um objeto em uma determinada classe baseando-se na probabilidade de esse objeto pertencer a esta classe.

Um classificador bayesiano pode utilizar uma das quatro funções discriminantes diferentes dadas por:

1. **Quadrática:** matriz de covariância diferentes e probabilidades a priori diferentes.
2. **Linear:** matriz de covariância diagonal com atributos da mesma variância e probabilidades priori diferentes.
3. **Distância Euclidiana:** matriz de covariância diagonal com atributos da mesma variância e probabilidades a priori iguais.
4. **Mahalanobis:** matriz de covariância diferentes e probabilidades a priori iguais.

Foi utilizado um classificador bayesiano com um modelo de evento multinomial, em que as amostras (vetores de recursos) representam as frequências com as quais determinados eventos foram gerados por um multinomial  $(p_1, \dots, p_n)$  em que  $p_i$  é a probabilidade do evento  $i$  ocorrer. Um vetor de características  $X = (x_1, \dots, x_n)$  é um histograma, quando  $x_i$  contém o número de vezes que o evento  $i$  é observado em um evento particular. O modelo de evento multinomial é usado para classificação de documentos (textos), com eventos representando a ocorrência de uma palavra em um único documento. A probabilidade de observar um histograma  $x$  é dada por:

$$p(x|C_k) = \frac{(\sum_i x_i)!}{(\prod_i x_i)!} \prod_i (p_{ki})^{x_i} \quad (2.6)$$

O classificador multinomial Naive de Bayes se torna um classificador linear quando é expresso da seguinte forma:

$$\log p(C_k|x) \propto \log(p(C_k) \prod_{i=1}^n (p_{ki})^{x_i}) = \log p(C_k) + \sum_{i=1}^n x_i * \log p_{ki} = b + w_k^T x \quad (2.7)$$

Em que  $b = \log p(C_k)$  e  $w_{ki} = \log p_{ki}$ . Se uma determinada classe e valor de recurso nunca ocorrerem juntos nos dados de treinamento, a estimativa de probabilidade baseada em frequência será zero, porque a estimativa de probabilidade é diretamente proporcional ao número de ocorrências do valor de um recurso. Isso é problemático porque eliminará todas as informações nas outras probabilidades quando elas forem multiplicadas.

Portanto, muitas vezes é desejável incorporar uma correção de amostra pequena, em todas as estimativas de probabilidade, de modo que nenhuma probabilidade seja definida para ser exatamente zero.

Rennie et al. (2003) discutem problemas com a suposição multinomial no contexto da classificação de documentos e possíveis maneiras de atenuar esses problemas, incluindo o uso do TF-IDF em vez de frequências simples de termos e a normalização do comprimento do documento, para produzir um classificador Naive Bayes que seja competitivo com uma Máquina de Vetores de Suporte.

#### 2.4.4 Redes neurais artificiais

O cérebro consiste em  $10^{11}$  células nervosas (também chamadas de neurônios) interconectadas por meio de  $10^{15}$  junções sinápticas para formar milhões de redes neurais. Centenas de áreas corticais especializadas são formadas com base nessas redes para diferentes tarefas de processamento de informações.

O Neurônio se divide em:

- dendritos: que conduz impulsos (entradas) de outras células em direção ao corpo celular;
- axônios: que conduz impulsos (saídas) para fora do corpo celular para outras células;
- sinapses: o ponto em que os impulsos passam de uma célula para outra.

Se a excitação ou inibição de um neurônio depende das estimulações sinápticas que recebe de outros neurônios, a intensidade de cada estimulação sináptica é, por sua vez, determinada pelo nível de excitação (ou inibição) do neurônio que transmite o sinal e pela conectividade sináptica entre os neurônios transmissores e receptores (HERTZ; PALMER; KROGH, 1990).

Esse processo pode ser modelado, matematicamente, por:

$$a_i = \sum_{j=1}^n x_j w_{ij} \quad (2.8)$$

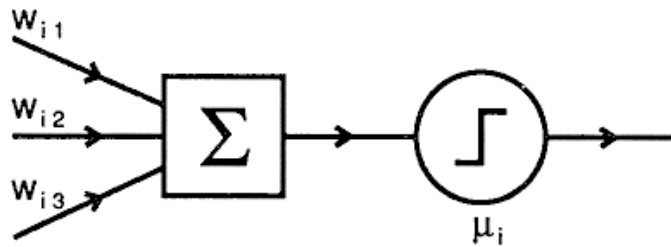
Em que  $a_i$  representa a ativação do neurônio de ordem  $i$  em uma população de  $n$  neurônios,  $x_j$  representa a intensidade do sinal transmitido pelo neurônio de ordem  $j$  ( $j = 1, 2, 3, \dots, n$ ) e  $w_{ij}$  representa a conectividade sináptica do neurônio de ordem  $j$  para o neurônio de ordem  $i$ .

Se o nível de ativação  $a_i$  for maior que o limiar representado por  $\mu_i$ , o neurônio receptor é excitado para gerar potenciais de ação que viajam ao longo de seu axônio para outros neurônios no fluxo. A intensidade desse sinal de saída é uma função de ativação:

$$y_i = \begin{cases} 0 & \text{if } a < \mu_i \\ f(a_i) & \text{if } a > \mu_i \end{cases} \quad (2.9)$$

Consoante Figura 6, um dos modelo de um neurônio, porém existem inúmeros outros:

Figura 6 – modelo do neurônio



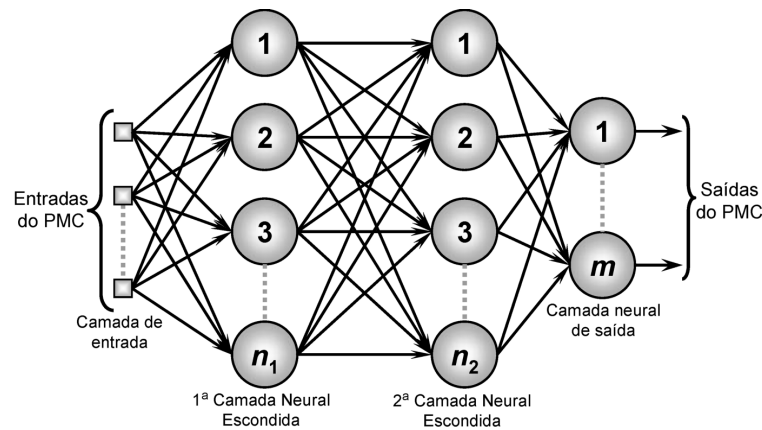
Fonte: autor.

#### 2.4.4.1 Multi layer perceptron

Perceptron Multicamadas (do inglês *Multi Layer Perceptron - MLP*) é uma rede neural que possui uma ou mais camadas ocultas, com uma quantidade indeterminada de neurônios. Para treinar a rede MLP, o algoritmo comumente utilizado é o de retropropagação (*Backpropagation*), não é possível prever a saída desejada nas camadas intermediárias. Por isso, a camada oculta possui esse nome.

Diferentemente de outras redes neurais artificiais, como a Perceptron Simples e a Adaline, em que há, apenas, um único neurônio de saída y, na Figura 7, é possível observar que a MLP pode relacionar o conhecimento a vários neurônios de saída.

Figura 7 – perceptron multicamadas



Fonte: (MOREIRA, 2018)

Moreira (2018) descreve os quatro passos que são executados pelo algoritmo de aprendizado da MLP.

1º Passo: inicialização

- Atribuir valores aleatórios para os pesos e limites;
- A escolha dos valores iniciais influencia o comportamento da rede;
- Na ausência de conhecimento prévio, os pesos e limites devem ter valores iniciais aleatórios e pequenos uniformemente distribuídos.

2º Passo: ativação

- Calcular os valores dos neurônios da camada oculta;
- calcular os valores dos neurônios da camada de saída.

3º Passo: treinamento dos Pesos

- Calcular os erros dos neurônios das camadas de saída e oculta;
- calcular a correção dos pesos;
- Atualizar os pesos dos neurônios das camadas de saída e oculta.

4º Passo: iteração

- Repetir o processo a partir do passo 2 até que satisfaça o critério de erro.

Uma das limitações da rede MLP é que, ao se aumentar muito o número de camadas e neurônios, ela tende a ficar com um número de parâmetros muito grande e, com isso, tão pesada ao ponto de o hardware não conseguir processar e ela não convergir (chegar a um resultado), talvez, por essa razão, até a evolução do hardware, ela tenha ficado um pouco estagnada (MOREIRA, 2018).

### 2.4.5 Regressão logística

A regressão logística é a análise de regressão apropriada a ser realizada quando a variável dependente é dicotômica (binária). Como todas as análises de regressão, a regressão logística é uma análise preditiva. A regressão logística é usada para descrever dados e explicar a relação entre uma variável binária dependente e uma ou mais variáveis independentes nominais, ordinais e de intervalo.

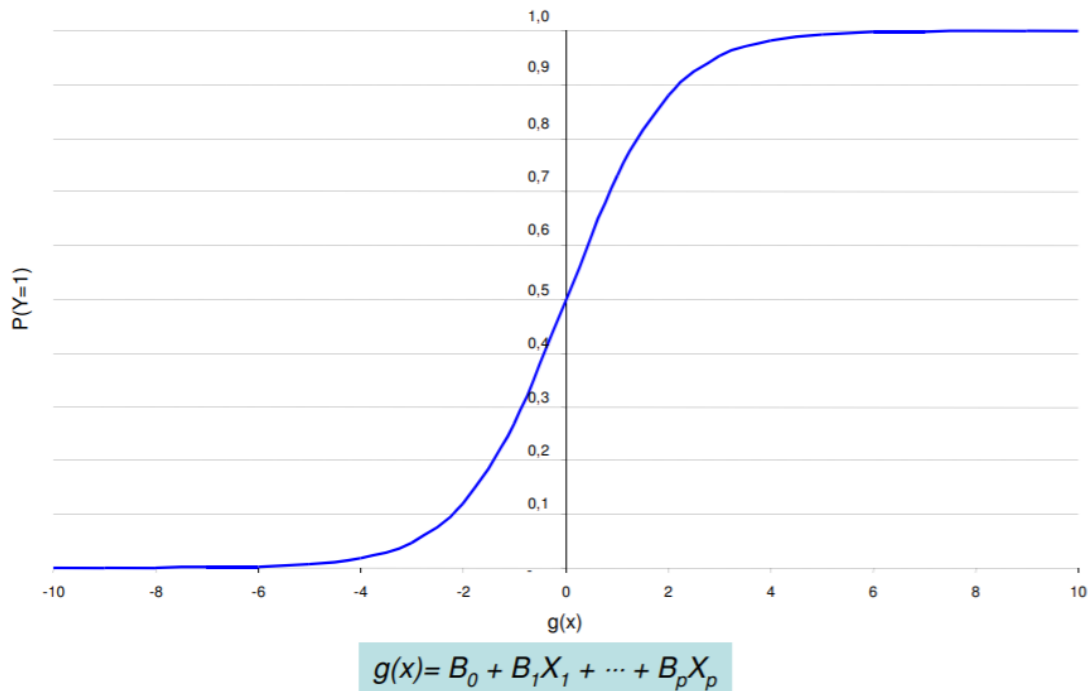
Na regressão logística, a possibilidade de ocorrência de um evento pode ser estimada diretamente. No caso de a variável dependente  $Y$  assumir, apenas, dois possíveis estados (1 ou 0) e haver um conjunto de  $p$  variáveis independentes  $X_1, X_2, \dots, X_p$ , o modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \quad (2.10)$$

em que,  $g(x) = B_0 + B_1X_1 + \dots + B_pX_p$

Os coeficientes  $B_0, B_1, \dots, B_p$  são estimados a partir do conjunto dados, pelo método da máxima verossimilhança, em que encontra uma combinação de coeficientes que maximiza a possibilidade da amostra ter sido observada. Considerando-se uma certa combinação de coeficientes  $B_0, B_1, \dots, B_p$  e variando os valores de  $X$ , observa-se que a curva logística tem comportamento probabilístico no formato da letra  $S$ , o que é uma característica da regressão logística, consoante Figura 8 (HOSMER; LEMESHOW, 1985).

Figura 8 – curva da regressão logística



Fonte: autor.

- a) quando  $g(x) \rightarrow +\infty$ , então  $P(Y = 1) \rightarrow 1$
- b) quando  $g(x) \rightarrow -\infty$ , então  $P(Y = 1) \rightarrow 0$

Para utilizar o modelo de regressão logística para discriminação de dois grupos, a regra de classificação é a seguinte:

- se  $P(Y = 1) > 0,5$  então classifica-se  $Y = 1$
- se  $P(Y = 1) < 0,5$  então classifica-se  $Y = 0$

Ao selecionar o modelo para a análise de regressão logística, outra consideração importante é o ajuste do modelo. Adicionar variáveis independentes a um modelo de regressão logística sempre aumentará a quantidade de variância explicada nas probabilidades de *log* (normalmente expressa como  $R^2$ ). No entanto, adicionar mais e mais variáveis ao modelo pode resultar em sobreajuste (do Inglês *Overfitting*), o que reduz a generalização do modelo além dos dados nos quais o modelo é ajustado.

#### 2.4.6 Support vector machine

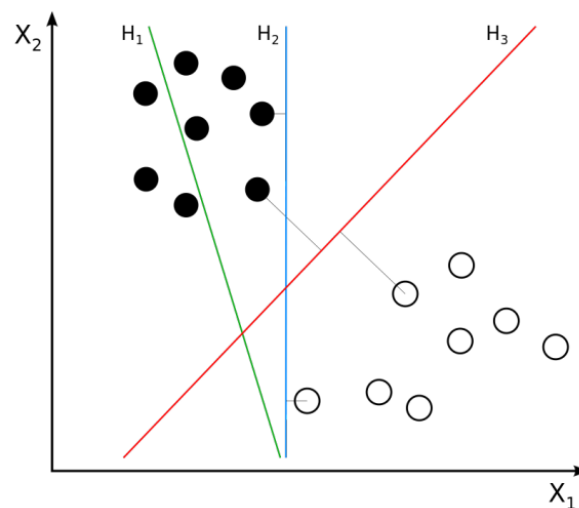
Na aprendizagem de máquina, o Support Vector Machine (SVM, também conhecida por Máquina de Vetores de Suporte) são modelos de aprendizagem supervisionada. Nesse algoritmo, sua aprendizagem é associada à análise dos dados, o que pode influenciar a classificação e regressão (CORTES C.; VAPNIK, 1995).

Dado um conjunto de exemplos de treinamento, cada reta está pertencente a uma ou a outra duas categorias, um algoritmo de treinamento SVM constrói um modelo que atribui novos exemplos a uma categoria ou a outra, tornando-o um classificador linear binário não probabilístico (embora os métodos como escala de Platt existem para usar SVM em uma configuração de classificação probabilística). Um modelo SVM é uma representação dos exemplos como pontos no espaço, mapeados para que os exemplos das categorias separadas sejam divididos por uma lacuna clara o mais ampla possível. Novos exemplos são mapeados no mesmo espaço e prevê-se que pertençam a uma categoria com base no lado da lacuna em que caem.

Além de executar a classificação linear, os SVMs podem executar, com eficiência, uma classificação não linear usando o que é chamado de truque do kernel, mapeando, implicitamente, suas entradas em espaços de recursos de alta dimensão (CORTES C.; VAPNIK, 1995).

Mais formalmente, uma máquina de vetor de suporte constrói um hiperplano ou conjunto de hiperplanos em um espaço de dimensão alta ou infinita, que pode ser usada para classificação, regressão ou outras tarefas como detecção de outliers. Intuitivamente, é obtida uma boa separação pelo hiperplano que tem a maior distância do ponto de dados de treinamento mais próximo de qualquer classe (a chamada margem funcional), pois, em geral, quanto maior a margem menor o erro de generalização do classificador (BEN-HUR, 2001). Na Figura 9, é ilustrado um exemplo de aplicação do SVM.

Figura 9 – hiperplanos do SVM



Fonte: autor.

$H_1$  não separa as classes,  $H_2$  faz essa separação, mas apenas com uma pequena margem, já o  $H_3$  separa as classe com margem máxima. Para descobrir o hiperplano ótimo, é calculado um produto escalar entre os pesos  $w$  com as características  $x$ , dado

essa informação logo:

$$\text{Hiperplano} = w * x + b \quad (2.11)$$

O ajuste do hiperplano é baseado nos erros e no custo, logo:

$$\frac{1}{2} |w|^2 + c \sum_{i=1}^n a_i \quad (2.12)$$

Considerando-se  $c$  o elemento de punição por classificação incorreta, quando há valor de  $c$  elevado, pretende-se encontrar 100% de separação. Quando o valor de  $c$  é baixo, permitem-se mais erros. No SVM é necessário selecionar um kernel, cada um possui uma forma de atualizar seus valores. Nesse exemplo é exposto o kernel linear, logo:

$$k(x, y) = (x * y) \quad (2.13)$$

## 2.5 Validação cruzada

A validação cruzada é um método estatístico utilizado para estimar a capacidade dos modelos de aprendizagem de máquinas.

É normalmente utilizado na aprendizagem aplicada à máquina para comparar e selecionar um modelo para um dado problema de modelação preditiva, porque é fácil de compreender, fácil de implementar e resulta em estimativas de competências que, geralmente, têm um enviesamento menor do que outros métodos. A validação cruzada k-fold é um procedimento utilizado para estimar a perícia do modelo em novos dados (BROWNLEE, 2018).

A validação cruzada com k-folds é um procedimento de reamostragem utilizado para avaliar modelos de aprendizagem de máquinas numa amostra de dados limitada (BROWNLEE, 2018).

O procedimento tem um único parâmetro chamado k que se refere ao número de grupos em que uma determinada amostra de dados deve ser dividida. Como tal, o procedimento é muitas vezes chamado de validação cruzada de k por dobra. Quando se escolhe um valor específico para k, este pode ser utilizado em vez de k na referência ao modelo, tal como k=10, tornando-se uma validação cruzada de 10 vezes.

A validação cruzada é utilizada, principalmente, na aprendizagem de máquina para estimar a perícia de um modelo de aprendizagem da máquina sobre dados não vistos.

Utiliza uma amostra limitada para estimar o desempenho geral do modelo quando utilizado para fazer previsões sobre dados não utilizados durante a formação do modelo.

É um método popular porque é simples de compreender e porque, geralmente, resulta numa estimativa menos enviesada ou menos otimista da perícia do modelo do que outros métodos, tais como uma simples divisão de treino e teste.

## 2.6 Métricas

A acurácia da classificação é o número total de previsões corretas dividido pelo número total de previsões feitas para um conjunto de dados. Como medida de desempenho, a acurácia é inadequada para problemas de classificação com classes desbalanceadas.

A principal razão é que o grande número de exemplos da classe majoritária (ou classes) superará o número de exemplos na classe minoritária, o que significa que, mesmo modelos inábeis, podem obter pontuações de acurácia entre 90% ou 99%, dependendo da gravidade.

Uma alternativa ao uso da acurácia nas classificações é usar as métricas de Precisão (do inglês *Precision*) e Revocação (do inglês *Recall*), em que a *Precision* quantificará o número de previsões de classe positiva que realmente pertencem à classe positiva e a *Recall* quantificará o número de previsões de classe positivas feitas com todos os exemplos positivos no conjunto de dados. Em sequência, a métrica *F1-score* fornecerá uma pontuação única que equilibrará os valores da *Precision* e da *Recall* em um novo e único número (SHUNG, 2018).

Uma matriz de confusão é uma tabela frequentemente usada para descrever o desempenho de um modelo de classificação em um conjunto de dados de teste pelos quais os valores verdadeiros são conhecidos. Todas as métricas podem ser calculadas usando os quatro parâmetros, verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo, apresentados na Figura 10 a seguir.

Figura 10 – matriz de confusão

|                     | Predição de Classes |                     |
|---------------------|---------------------|---------------------|
|                     | Classe = ADEQUADA   | Classe = INADEQUADA |
| Classe = ADEQUADA   | Verdadeiro Positivo | Falso Negativo      |
| Classe = INADEQUADA | Falso Positivo      | Verdadeiro Negativo |

Fonte: autor.

Verdadeiros positivos e verdadeiros negativos são as observações previstas corretamente e, portanto, mostradas em verde. As observações incorretas são os falsos positivos e os falsos negativos, que são mostrados em vermelho.

Verdadeiros positivos (VP): esses são os valores positivos previstos corretamente, o que significa que o valor da classe real é SIM e o valor da classe prevista também é SIM. Por exemplo, o valor real da classe indica que a mensagem é ADEQUADA e a previsão diz a mesma coisa.

Verdadeiros negativos (VN): esses são os valores negativos previstos corretamente, o que significa que o valor da classe real é NÃO e o valor da classe prevista também é NÃO. Por exemplo, se o valor real da classe indica que a mensagem é INADEQUADA e a previsão diz a mesma coisa.

Falsos positivos (FP): quando a classe real é ADEQUADA e a classe prevista é INADEQUADA. Por exemplo, a classe real diz que a mensagem é ADEQUADA, mas a classe prevista diz que a mensagem é INADEQUADA.

Falsos Negativos (FN): quando a classe real é INADEQUADA, mas a classe prevista é ADEQUADA. Por exemplo, o valor real da classe indica que uma mensagem é INADEQUADA e a classe prevista informa que a mensagem é ADEQUADA.

Com esses quatro parâmetros, podemos calcular a Acurácia, a *Precision*, a *Recall* e a *F1-measure*.

**Acurácia:** é a medida de desempenho mais intuitiva e é simplesmente uma proporção da observação prevista corretamente com o total de observações. Pode-se pensar que, se tivermos alta acurácia, nosso modelo será o melhor. Sim, a acurácia é uma ótima medida, mas somente quando você tem conjuntos de dados simétricos e/ou balanceados, em que os valores de falsos positivos e falsos negativos são quase os mesmos. Portanto, deve-se observar outros parâmetros para avaliar o desempenho de um modelo.

$$\text{Acurácia} = \text{VP} + \text{VN} / \text{VP} + \text{FP} + \text{VN} + \text{FN}$$

***Precision*:** é a razão entre as observações positivas previstas corretamente e o total de observações positivas previstas. Essa métrica responde a seguinte questão, de todas as mensagens que rotularam como adequadas, quantas realmente são adequadas? A alta precisão está relacionada à baixa taxa de falsos positivos.

$$\text{Precision} = \text{VP} / \text{VP} + \text{FP}$$

***Recall*:** é a razão entre as observações verdadeiras positivas previstas corretamente e todas as observações da classe real ADEQUADAS. Essa métrica, também, responde uma questão, de todas as mensagens que realmente são adequadas, quantas foram rotuladas?

$$\text{Recall} = \text{VP} / \text{VP} + \text{FN}$$

**F1-score:** é a média ponderada da *Precision* e *Recall*. Portanto, essa métrica considera tanto os falsos positivos quanto os falsos negativos. Intuitivamente, não é tão fácil entender como a acurácia, mas a *F1-score* é geralmente mais útil que a acurácia, especialmente se você tiver uma distribuição de classe **desigual** e/ou **desbalanceadas**. A acurácia funciona melhor se os falsos positivos e falsos negativos tiverem um custo ou quantidade semelhante. Se o custo ou quantidade de falsos positivos e falsos negativos forem muito diferentes, é melhor observar os resultados obtidos na *Precision* e *Recall*, e, posteriormente, obter o resultado da *F1-score*.

$$\text{F1-score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

## 3 Trabalhos correlatos

Neste capítulo descrevem-se, de forma sucinta, os trabalhos que serviram de apoio, para entender as tecnologias utilizadas na classificação de textos, bem como para entender a evolução das pesquisas na área de mineração de textos em geral e, especificamente, na área educacional.

### 3.1 Classificações de textos em geral

Zhou, Wang e Wu (2005) utilizaram uma técnica para analisar textos chineses baseando-se na similaridade das palavras. Após conseguirem obter os vetores com as características dos textos, incluindo o subconjunto com as similaridades, aplicaram uma SVM que foi capaz de categorizar os textos de entrada com precisão mais alta que a utilização apenas de uma SVM sem analisar a similaridade. Os autores utilizaram como base de dados 6451 textos que pertenciam a cinco categorias diferentes, meio ambiente, computação, economia, esporte e política. Essa variação do algoritmo de SVM se chama Máquina de Vetores de Suporte Proximal (do inglês *Proximal Suporte Vector Machine - PSVM*).

Wang e Qian (2008) utilizaram SVM em seu trabalho, com o objetivo de realizar a classificação de documentos de textos. Essa técnica é bastante empregada com o intuito de melhorar as buscas em documentos de textos através da internet. Zhang, Huang e Zhang (2009) realizaram uma pesquisa similar, porém utilizaram o classificador K-NN e seu objetivo foi realizar agrupamentos de documentos de acordo com similaridades, podendo, assim, classificá-los.

Hao, Ying e Longyuan (2009) também utilizaram SVM para classificar textos. Uma variação importante observada em seu trabalho foi a utilização do algoritmo Naive Bayes para realizar o treinamento da SVM. Na etapa de teste, utilizaram, apenas, a SVM e concluíram que essa adaptação tornou a técnica melhor e mais precisa, em comparação à utilização unicamente da SVM em todo o processo.

Em uma pesquisa de comparações, Valero, Gómez e Pineda (2009) utilizaram três técnicas de aprendizagem de máquina, para detectar qual a que possuía melhor eficácia na categorização de textos sobre desastres naturais. Concluíram que, mesmo variando o tamanho da janela de classificação, o algoritmo SVM sempre possuía melhores acurácias em comparação aos demais algoritmos, Naive Bayes e Árvore de Decisão com o C4.5.

Mubeen, Qaseem e Govardhan (2011) utilizaram a ferramenta de simulação WEKA (do inglês *Waikato Environment for Knowledge Analysis*), que é uma coleção de algoritmos

de aprendizagem de máquina para tarefas de mineração de dados. Em seu trabalho, analisaram a influência da distribuição de uma palavra no texto, a probabilidade de encontrar a palavra em outros locais do texto e atribuíram valores para as palavras de acordo com a sua distribuição. Esses valores foram denominados como recursos de distribuição. Essas características diferentes foram calculadas para a passagem da janela, usando classificadores distintos. O classificador que fornecia o resultado mais preciso era selecionado para a classificação do texto e, experimentos realizados revelaram que os recursos de distribuição foram úteis para a classificação do texto.

Um novo método de classificação de textos foi apresentado por Gao e Guan (2012), que classificaram textos chineses com a utilização do algoritmo de Rocchio, que é um algoritmo clássico para implementar feedback de relevância que foi desenvolvido usando o modelo de espaço vetorial e tem o propósito de classificar textos como relevantes ou não relevantes. Utilizaram esses vetores para gerar livros de códigos como modelos de classificação usando o algoritmo Rocchio, o livro de códigos foi usado para categorizar os documentos de destino usando pontuações vetoriais. No pré-processamento do texto, utilizaram TF-IDF, a medida estatística que tem o intuito de indicar a importância de uma palavra de um documento ou texto em relação a uma coleção de documentos ou textos. Ela é frequentemente utilizada como fator de ponderação na recuperação de informações e na mineração de dados. Após realizarem os experimentos, concluíram que sua técnica pode alcançar melhores desempenhos.

Xue e Li (2015) realizaram uma pesquisa para avaliar o desempenho da classificação de textos utilizando o algoritmo de Florestas Aleatórias (do inglês *Random Forest - RF*). O algoritmo RF é um famoso algoritmo de aprendizado integrado e tem sido amplamente aplicado em muitos campos. Além disso, devido ao rápido desenvolvimento da Internet, a classificação de textos tornou-se a principal tecnologia para processar e organizar documentos em grande escala. Em sua pesquisa, apresentaram os detalhes do algoritmo RF e avaliaram o modelo de classificação de documentos de texto usando o algoritmo RF, e concluíram que o erro de generalização tem um limite superior baseado no número de árvores da floresta aleatória, ou seja, tanto a dimensão dos recursos quanto o número de árvores de decisão têm influência direta no desempenho do modelo de classificação proposto.

Em um trabalho de classificação de textos em bengalês, Islam, Elahi e Ahmed (2017) utilizaram SVM com TF-IDF. Além disso, utilizaram técnicas de tratamentos de n-gramas, que são as junções de duas ou mais palavras que possam expressar melhor o sentido ou o contexto da categoria a ser classificada, porém concluíram que o classificador alcançava melhores resultados com a utilização de unigramas, ou seja, palavras isoladas. Também identificaram que o aumento do número de treinamentos fazia com que o classificador conseguisse melhores resultados na categorização dos textos que eram divididos entre doze

categorias.

Jo (2017) conseguiu classificar textos utilizando o classificador K-NN. Uma variação interessante em seu trabalho foi utilizar a vetorização do texto em strings e não de forma numérica. O desempenho para o classificador se tornou melhor, evitando problemas em codificação de textos em vetores numéricos, também conseguiu propor um sistema mais transparente, pois vetores de string representam os textos de forma mais simbólica que os convertidos numericamente.

Utilizando a técnica TF-IDF associada com o algoritmo de Bayes, Qu et al. (2018) definiram essa associação como Algoritmo de Bayes Melhorado com base em TF-IDF. Em sua pesquisa, classificaram textos chineses em, apenas, duas categorias, positivos ou negativos, técnica utilizada para detectar o sentimento expresso nos textos em relação a um tema em discussão. Apresentaram melhores resultados com a utilização do seu modelo adaptado em relação aos algoritmos clássicos de Bayes e Naive Bayes.

Observa-se, na Tabela 1, que, durante a evolução das pesquisas de classificação de textos, a utilização de algoritmos de aprendizagem de máquina associados a outros algoritmos de pré-processamento de textos alcançaram melhores resultados nas classificações.

Tabela 1 – resumo dos trabalhos de classificações de textos em geral.

| <b>Autores</b>      | <b>Algoritmos</b>        | <b>Pré-processamento e Extrações de Características</b> |
|---------------------|--------------------------|---------------------------------------------------------|
| Zhou et al., 2005   | SVM                      | -                                                       |
| Wang e Qian., 2008  | SVM                      | -                                                       |
| Zhang et al., 2009  | K-NN                     | -                                                       |
| Hao et al., 2009    | SVM com Naive Bayes      | -                                                       |
| Valero et al., 2009 | SVM, Naive Bayes e C4.5  | -                                                       |
| Mubeen et al., 2011 | Naive Bayes, Bayes e SVM | Distribuição                                            |
| Gao e Guan, 2012    | Rocchio                  | TF-IDF                                                  |
| Xue e Li, 2015      | Random Forest            | -                                                       |
| Islam et al., 2017  | SVM e n-Gramas           | TF-IDF                                                  |
| Jo, 2017            | K-NN                     | Count Vectorizer                                        |
| Qu et al., 2018     | Bayes e Naive Bayes      | TF-IDF                                                  |

Após o estudo desses trabalhos, foi possível compreender quais técnicas podem ser utilizadas para o pré-processamento, extração de características e classificação de textos. Porém, foi necessário realizar levantamento mais aprofundado, em busca de pesquisas que aplicassem essas técnicas em dados educacionais. As pesquisas encontradas são apresentadas a seguir.

## 3.2 Classificações de textos em AVAs

Diversos são os trabalhos relacionados ao problema da classificação de textos educacionais, eles se propõem a analisar dúvidas em textos, classificar os conteúdos em gêneros textuais e detectar os níveis de colaboração dos alunos, porém, até o presente momento, não foram encontrados trabalhos que abordassem a moderação inteligente de mensagens em AVA para alunos privados de liberdade. O modelo desta pesquisa foi idealizado e implementado, inicialmente, no ano de 2019 (CANDIDO; FURTADO JÚNIOR; FREITAS, 2019).

Gabrielli (2016) reforça a preocupação que se deve ter com os fóruns em AVAs, uma vez que, na maioria das vezes, uma única pessoa, geralmente o professor da disciplina, assume o papel de administrador e moderador de um fórum. Ela, também, destaca que, nos fóruns, os alunos não precisam pedir autorização para enviar mensagens e que as mensagens ficam expostas a todos os participantes do curso. Sendo assim, é possível que, em determinados momentos, os estudantes interajam entre si sem a participação do professor para moderar as mensagens caso seja necessário. Logo, identifica-se um risco se essa configuração de ensino a distância for aplicada a alunos do sistema prisional.

Com a utilização do classificador Naive Bayes e Análise Semântica Latente (do inglês *Latent Semantic Analysis - LSA*) em um fórum de um AVA, Lui, Li e Choy (2007) avaliam o nível de participação dos alunos, atribuem uma nota às mensagens consideradas relevantes e identificam de quais categorias as mensagens pertencem, porém realizam seus experimentos em, apenas, um cenário de testes.

Ravi e Kim (2007) utilizam SVM para classificar as mensagens dos alunos em dois grupos diferentes, o grupo das perguntas e o grupo das respostas. O classificador alcança precisões de 88% para as categorias de perguntas e 73% para a categoria de respostas. É possível alcançar valores mais altos de precisão se utilizar TF-IDF no processo de extração de características.

Scheuer e McLaren (2008) afirmam que o SVM, Árvores de Decisão Aprimorada e Listas de Decisão são os mais efetivos dentre os seis algoritmos de aprendizado de máquina que utilizam para a classificação das mensagens de fóruns em duas categorias, Boas Discussões ou Más Discussões. Comparam o desempenho dos classificadores com os julgamentos dos professores e conseguem valores  $Kappa^1$  variando entre 60% e 71%.

Lin, Hsieh e Chuang (2009) propõem um Sistema Classificador de Gêneros utilizando Árvores de Decisão, cujo propósito é identificar a quais dos gêneros as mensagens enviadas pelos alunos pertencem. As mensagens podem ser classificadas como anúncios, perguntas, interpretações, conflitos, afirmações ou outros. Um aspecto interessante é que a ferramenta é integrada em um AVA, porém a classificação não acontece em tempo real.

---

<sup>1</sup> (LANDIS; KOCH, 1977)

Azevedo, Behar e Reategui (2011) realizam uma análise qualitativa das contribuições textuais registradas por alunos em fóruns de discussão. Utilizam o software MineraFórum aplicado em dez diferentes fóruns e realizam as comparações das análises feitas pelo software e pelos professores. Obtêm bons valores de precisão alcançando até 96,43%.

O trabalho de Oliveira Junior e Esmín (2011) é desenvolver uma ferramenta para classificar as mensagens entre positivas ou negativas. Fazem uma comparação entre os algoritmos SVM e Naive Bayes, o SVM se destaca com precisão de 71% em relação aos 64% do Naive Bayes. Propõem uma ferramenta de monitoramento automático de mensagens. Entretanto, a classificação não acontece em tempo real, é acionada uma vez por dia por meio de um agendamento no sistema operacional.

Rolim, Ferreira e Costa (2016) propõem uma solução computacional para classificar as mensagens de um fórum em três categorias, dúvidas, postagens neutras ou respostas. Utilizam três técnicas de classificação, Naive Bayes, Árvores de Decisão e a rede neural MLP<sup>2</sup>, alcançam valores altos de precisão chegando até 97% com a utilização da MLP.

Ferreira et al. (2017) realizam um trabalho para identificar o nível de colaboração dos alunos nos fóruns, utilizam o classificador SVM e alcançam precisões de 80%, 68%, 60% e 84% para as categorias presença de grupo, elogio/apreciação, introdução de novas perspectivas e compartilhamento de informações, respectivamente.

Existem técnicas comuns utilizadas no pré-processamento nestes trabalhos, a remoção das *stopwords* e, a realização de *stemming*, que reduzem as palavras ao seu tronco, base ou raiz. Alguns utilizaram algoritmos para a segmentação dos termos em ngramas, contadores de frequência de termos, como o CV e o parâmetro indicador TF-IDF. Na Tabela 2, é possível ver um resumo de quais técnicas foram utilizadas nos trabalhos relacionados.

---

<sup>2</sup> (KRUSE et al., 2013)

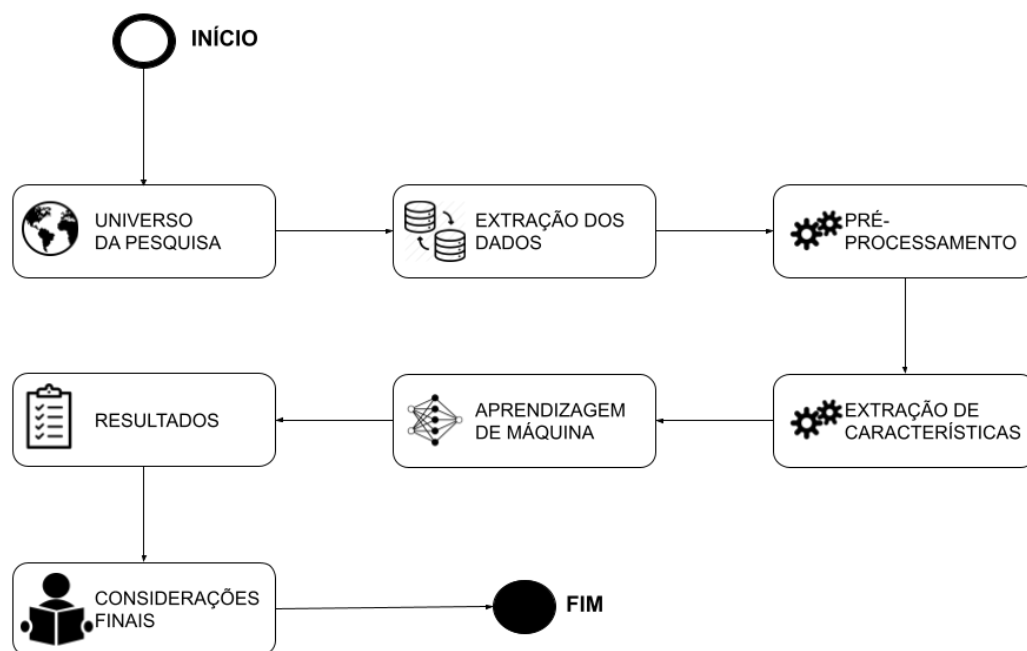
Tabela 2 – resumo dos trabalhos de classificações de textos em AVAs.

| <b>Autores</b>                | <b>Algoritmos</b>                           | <b>Pré-processamento e Extrações de Características</b> |
|-------------------------------|---------------------------------------------|---------------------------------------------------------|
| Lui, Li e Choy (2007)         | LSA e Naive Bayes                           | Stopwords e Stemming                                    |
| Ravi e Kim (2007)             | SVM                                         | Stopwords, Stemming e N-gramas                          |
| Scheuer e McLaren (2008)      | Árvores de Decisão, Listas de Decisão e SVM | Stopwords, Stemming e N-gramas                          |
| Lin et al. (2009)             | Árvores de Decisão                          | Stopwords, N-gramas e TF-IDF                            |
| Azevedo et al. (2011)         | Grafos                                      | Stopwords e Stemming                                    |
| Oliveira Junior e Esmin, 2011 | Naive Bayes e SVM                           | Stopwords                                               |
| Rolim et al. (2016)           | Árvores de Decisão, Naive Bayes e MLP       | Stopwords, Stemming, CV e TF-IDF                        |
| Ferreira et al. (2017)        | SVM                                         | TF-ISF                                                  |

## 4 Metodologia

Este capítulo tem por objetivo descrever a metodologia empregada para a realização da pesquisa proposta nesta dissertação e abordar as seguintes etapas: universo da pesquisa, extração dos dados, pré-processamento, extração de características e a aprendizagem de máquina (consoante Figura 11).

Figura 11 – etapas da pesquisa



Fonte: autor.

### 4.1 Universo da pesquisa

De acordo com Furtado Junior (2016), o aumento da complexidade dos sistemas computacionais fez crescer o número de técnicas e métodos para desenvolvê-los de forma correta e segura. As simulações realizadas nos projetos de sistemas permitem o estudo do problema modelado em diferentes níveis de abrangência e múltiplos cenários.

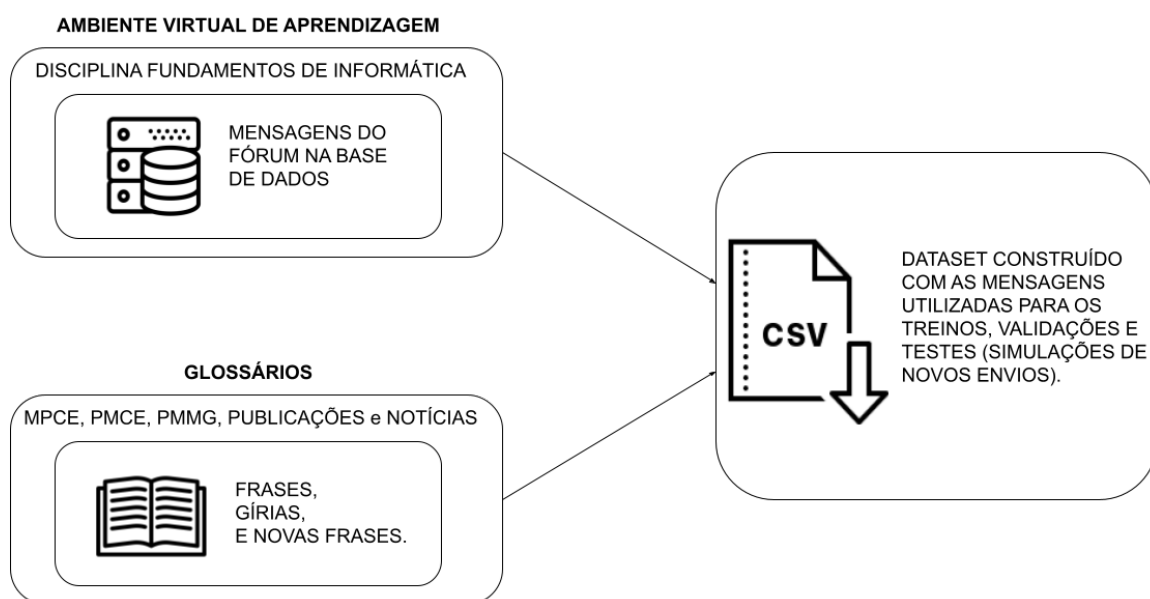
Nesta pesquisa são realizadas simulações de trocas de mensagens entre alunos privados de liberdade em educação à distância, são aplicadas técnicas de pré-processamento e extração de características nas mensagens e, posteriormente, são utilizados algoritmos de aprendizagem de máquina para realização da classificação dessas mensagens.

## 4.2 Extração dos dados

Para conseguir classificar automaticamente as mensagens que participam da simulação de envio ao fórum, é necessário construir um *dataset*<sup>1</sup> contendo as mensagens consideradas adequadas e inadequadas à disciplina, para ser utilizado no processo de aprendizagem dos algoritmos de classificação.

As mensagens consideradas adequadas são coletadas de uma base de dados do AVA da disciplina de Fundamentos de Informática do curso à distância de Técnico em Informática do IFCE. Já as mensagens consideradas inadequadas, são coletadas dos glossários, de publicações de trabalhos científicos e notícias. Nesses glossários existem algumas frases utilizadas por presos em unidades prisionais. Além dessas, existem também várias gírias e termos com significados relativos à essa comunidade prisional. A construção desse *dataset* é ilustrada na Figura 12.

Figura 12 – construção do *dataset*



Fonte: autor.

Após a construção desse *dataset*, são totalizadas 1930 mensagens, distribuídas da seguinte forma: 1216 mensagens previamente classificadas e rotuladas como adequadas e 714 rotuladas como inadequadas.

Dessas 1930 mensagens, 1544 são selecionadas aleatoriamente para o processo de treinamento e validação. Outras 386 mensagens, também selecionadas aleatoriamente, são reservadas para serem utilizadas no processo de teste do modelo, simulando, então, a

<sup>1</sup> Disponível em: <https://drive.google.com/file/d/1YWEzdzW5weYeGU5tvs74sxr2WTrTplE/view?usp=sharing>

utilização do modelo em um ambiente de produção, como o incorporado como um *plugin* em um ambiente virtual de aprendizagem de uma plataforma de ensino à distância.

Não foram estabelecidos critérios para o tamanho das mensagens, porém observa-se que as mensagens obtidas nos glossários e consideradas inadequadas são menores, contendo menos termos em relação as mensagens das disciplinas.

Exemplos de mensagens consideradas adequadas:

- a tecnologia de circuitos impressos também foi criada;
- conclusão, houve grandes mudanças;
- desde aquela época, vários processadores da Intel já foram lançados;
- já existem *smartphones* bastante poderosos;
- isso tornou a informática muito mais acessível.

Exemplos de mensagens consideradas inadequadas:

- estou com a maior larica;
- não se preocupa, ele é zureta;
- vamos fazer a parada hoje;
- vou botar o terror;
- se liga que ele é xis nove.

### 4.3 Pré-processamento

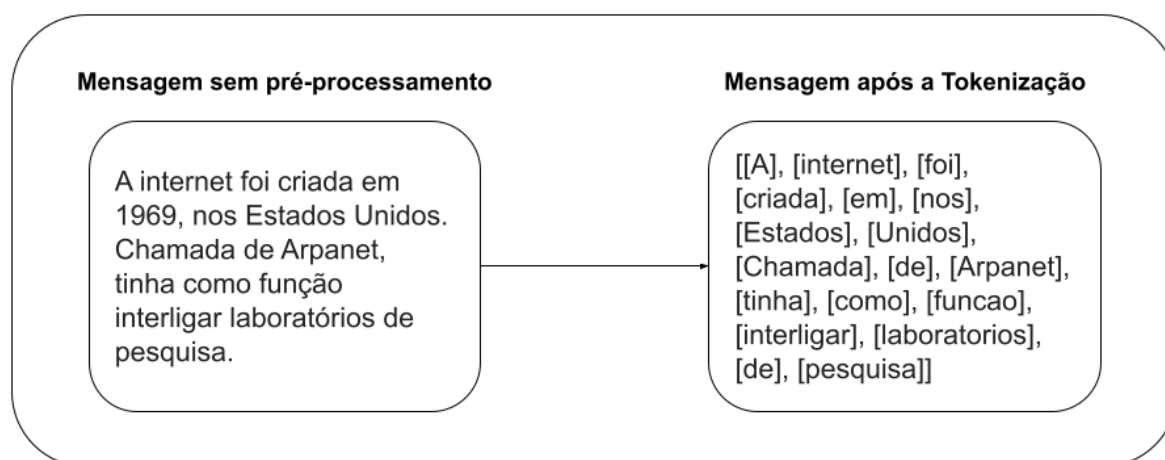
No pré-processamento são aplicadas quatro técnicas: remoção de símbolos, pontuações e acentuações em geral; tokenização; remoção das *stopwords* e o uso de *stemming*. Os símbolos, pontuações e acentuações não interferem nos resultados das classificações, porém são removidos para que se diminuam os números de *tokens*, tornando o processo de classificação mais rápido e relevante.

A remoção das acentuações gráficas é realizada com o intuito de garantir a classificação correta de todas as mensagens, inclusive as que os alunos possam escrever acentuadas incorretamente.

Em um ambiente real da aplicação do modelo, não se espera que todas as mensagens sejam escritas 100% da forma correta, de acordo com as regras da Língua Portuguesa. Então, esse tratamento foi aplicado para que o classificador consiga comparar todas as mensagens, uma vez que as mensagens contendo palavras são acentuadas corretamente e, incorretamente, são apresentadas da mesma forma.

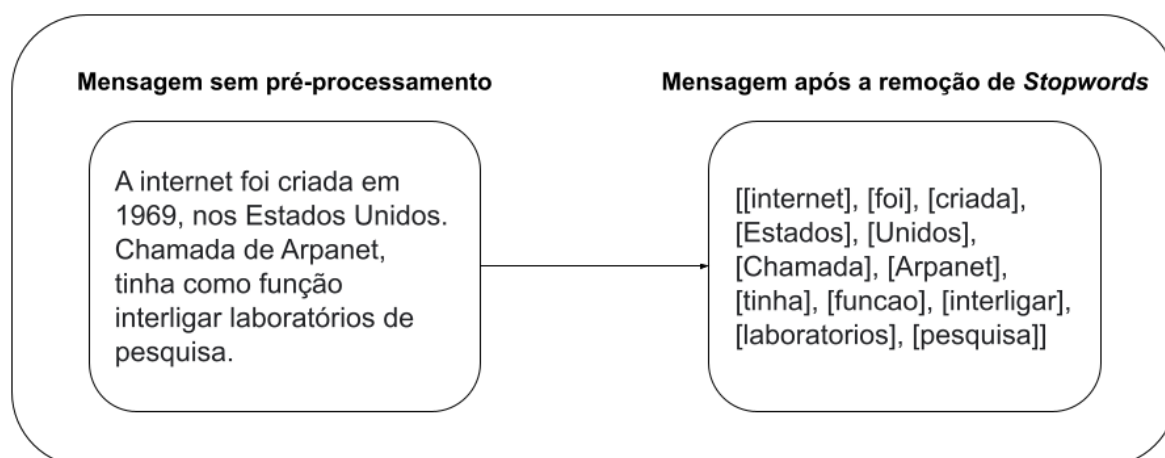
Entende-se, então, que uma frase com 18 palavras, sem considerar os símbolos, pontuações e acentuações gráficas, contém 18 *tokens* após a realização da tokenização na mensagem, consoante Figura 13.

Figura 13 – processo de tokenização da mensagem



Fonte: autor.

Na Figura 14, apresenta-se mensagem após a remoção dos símbolos, realização da tokenização e remoção das *stopwords*. Dependendo do tamanho de uma base de dados textual, após a remoção desses termos, a quantidade final de termos pode diminuir consideravelmente.

Figura 14 – processo de remoção de *stopwords* da mensagem

Fonte: autor.

Na Figura 15(a), percebe-se a predominância das palavras na base de dados sem a aplicação da técnica de remoção de *stopwords*, a predominância dos artigos e pronomes interfere na classificação. Já na Figura 15(b), apresenta-se a predominância das palavras quando utilizada a técnica de remoção de *stopwords*.

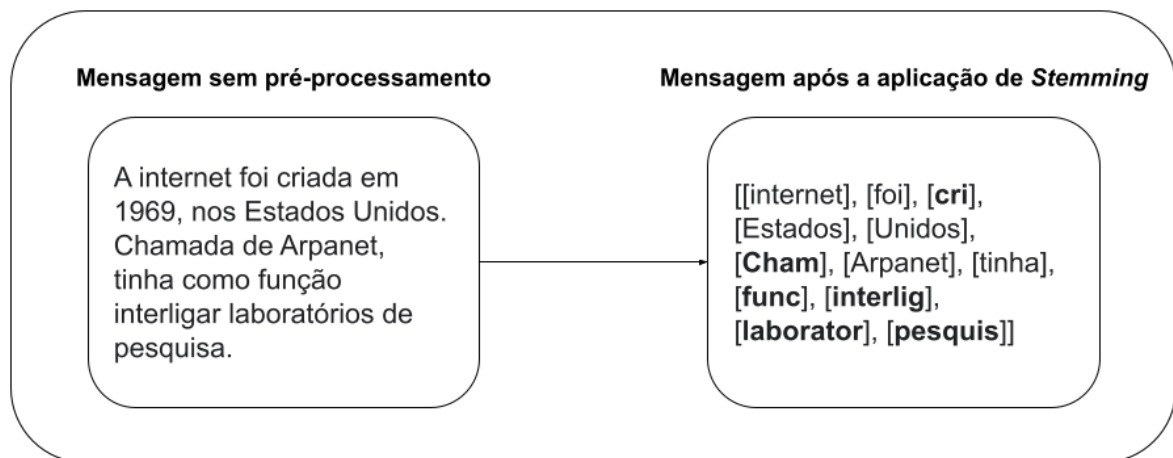
Figura 15 – visualização de predominância das palavras



(a) Predominância das palavras com as *stopwords* (b) Predominância das palavras sem as *stopwords*

Fonte: autor.

Na Figura 16, mostra-se qual deve ser o estado da mensagem após remoção dos símbolos e pontuações. Realiza-SE a tokenização, remoção das *stopwords* e utilização da técnica de *stemming*.

Figura 16 – processo de *stemming* na mensagem

Fonte: autor.

Para executar as técnicas de pré-processamento, é utilizado o Kit de Ferramentas de Linguagem Natural (do inglês *Natural Language Toolkit* - NLTK) desenvolvido em linguagem de programação Python e com documentação disponível (NLTK, 2018).

Após os processos de remoção de símbolos, números, acentuações e pontuações; tokenização; a remoção das *stopwords* e o uso de *stemming*, é feita a extração das características utilizando técnicas de vetorização, que converteram as mensagens no formato de *strings* para matrizes numéricas.

Na literatura existem afirmações de alguns autores que a técnica de extração de

características TF-IDF é capaz de remover as *stopwords* e é possível realizar a classificação de textos sem utilizar uma técnica de remoção de *stopwords*.

Para verificar se isso realmente acontece, e se há uma melhora ou não no desempenho dos algoritmos de classificação, são criados quatro cenários para os testes. Um cenário em que nenhuma das técnicas de pré-processamento é aplicada e cenários em que as técnicas são aplicadas de forma individual ou combinadas, conforme Tabela 3.

Tabela 3 – distribuição dos cenários

| Cenários  | Técnicas de Pré-processamento |          |
|-----------|-------------------------------|----------|
|           | Stopwords                     | Stemming |
| Cenário 1 |                               |          |
| Cenário 2 | x                             |          |
| Cenário 3 |                               | x        |
| Cenário 4 | x                             | x        |

A remoção dos símbolos e pontuações e a tokenização são aplicadas em todos os quatro cenários. No cenário 1, nenhuma técnica de pré-processamento é aplicada às mensagens, como a remoção de *stopwords* e o uso de *stemming*. No cenário 2, utiliza-se, apenas, a técnica de remoção das *stopwords*. No cenário 3, apenas a técnica de *stemming* é aplicada às mensagens. No cenário 4, as mensagens são pré-processadas com as duas técnicas, remoção das *stopwords* e aplicação de *stemming*.

As disciplinas em EAD têm, em seu planejamento, a disponibilidade do uso de chats e/ou fóruns, em que os alunos devem participar. Há casos em que se deve discutir acerca de temas específicos e outros em que a discussão é livre, apenas para esclarecimento de dúvidas.

Antes de realizar a próxima etapa de vetorização das mensagens, ainda são realizadas outras tarefas no pré-processamento das mensagens, a remoção das palavras abreviadas, siglas e termos usados na comunicação informal da internet, por exemplo: vc (você), tbm (também), vdc (vai dar certo), etc.

## 4.4 Extração de características

O contador de palavras *Count Vectorizer* é utilizado para codificar as mensagens de forma simples, além de construir um vocabulário com as palavras mais conhecidas, atribuindo a elas um índice que corresponde ao número de vezes que elas surgem em todo o conjunto.

Uma matriz esparsa é retornada com o número de colunas correspondendo ao total das palavras que existem em todo o conjunto e o número de linhas correspondendo ao total das mensagens. A matriz é esparsa por conter muitos índices com valor zero e

alguns valores inteiros, apenas na contagem de vezes em que cada palavra aparece naquela mensagem.

Por ser uma representação simplificada no processo de linguagem natural e recuperação de informação, utiliza-se para comparar os resultados em relação ao TF-IDF, além, do seu desempenho e contribuição a classificação em relação ao uso do TF-IDF.

O TF-IDF é utilizado para pontuar a importância de cada palavra em relação a todo o conjunto de palavras, com base na frequência em que elas aparecem nesse conjunto. Se uma palavra aparece com muita frequência em todo o conjunto, ela não é considerada importante para o TF-IDF, logo, é atribuída para ela uma pontuação baixa.

As palavras que aparecem com menos frequência em relação a todo o conjunto de palavras recebem notas maiores. Quando o TF-IDF realiza esse procedimento, destaca as palavras mais distintas em relação às outras.

Pelo menos, uma dessas técnicas precisam ser utilizadas para transformar os dados de texto em um formato que se possa aplicar os algoritmos de aprendizagem de máquina.

O *Count Vectorizer* tem uma interpretação e implementação mais simples, porém o TF-IDF é mencionado na literatura científica com um melhor desempenho, principalmente nesses casos, como o apresentado nesta dissertação, em que se aplicam técnicas de aprendizagem de máquina nessas matrizes que são retornadas desses processos.

Ambos *Count Vectorizer* e TD-IDF são utilizados da seguinte forma: primeiramente, utiliza-se nos quatro cenários de testes o *Count Vectorizer*, com cada um dos classificadores. Posteriormente, o mesmo é feito com a utilização do TF-IDF.

## 4.5 Aprendizagem de máquina

Para garantir que o modelo aprenda a classificar as mensagens, é necessário realizar o treinamento, a validação cruzada e o teste, com a utilização de um *holdout* dividido em 80% para treino e validação, com os 20% restantes dos dados destinados aos testes.

É realizado um treinamento com cada um dos classificadores analisados. Esse treinamento ocorre com base em um conjunto de mensagens pré-existentes, rotuladas enquanto adequadas ou inadequadas, ao assunto abordado na disciplina.

Na validação cruzada, é utilizado o método *k-fold*, com a subdivisão do conjunto de mensagens em 5-*folds*, com o objetivo de avaliar a capacidade de generalização do modelo. Um subconjunto é separado para a realização da validação e os demais quatro subconjuntos são utilizados para o treinamento. Esse procedimento é realizado 5 vezes, correspondendo ao número de *folds* e alternando o subconjunto que será separado para realização da validação.

Por fim, são avaliados os desempenhos de cada um dos classificadores, com a realização dos testes com as mensagens que são reservadas para esse momento.

Para a avaliação dos desempenhos dos classificadores, são utilizadas as métricas de *Precision*, *Recall* e *F1-score* [Hripcsak e Rothschild 2005]. A *Precision* é utilizada para avaliar a quantidade de instâncias que são classificadas corretamente, enquanto o *Recall* para avaliar a frequência em que o classificador encontra os exemplos de uma classe. A *F1-score* é utilizada como uma média harmônica da *Precision* e do *Recall*.

É importante ressaltar o porquê da escolha dessas métricas para a avaliação do desempenho desse modelo. A métrica *F1-score* é escolhida para representar o resultado, por ser a métrica ideal para avaliar o desempenho de classificadores em bases de dados desbalanceados.

Ressaltamos que foram realizados testes com o balanceamento da base de dados, porém não foram obtidos melhores resultados, então optou-se por manter a utilização dessas métricas em vez da utilização da acurácia como métrica de representação.

Além disso, vale ressaltar que, em um ambiente real de aplicação desse modelo, encontrar-se-á, também, uma situação em que a base de dados será desbalanceada.

São realizadas 50 iterações, para todo esse processo de classificação, que compreende os treinamentos, validações cruzadas e os testes. Além disso, são calculadas as médias e desvios padrões de cada métrica e o tempo médio de execução dessas iterações, em milissegundos.

Quando se pretende avaliar o tempo de execução de um algoritmo, é importante ressaltar as configurações de *hardware* utilizadas no computador em que são realizados os testes do modelo.

O computador utilizado possui a seguinte configuração: processador *Intel Core I7-9750H* de 2.60GHz com 6 núcleos e 12 processadores lógicos, memória RAM de 16 gigas e GPU *NVidia GeForce GTX 1050*.

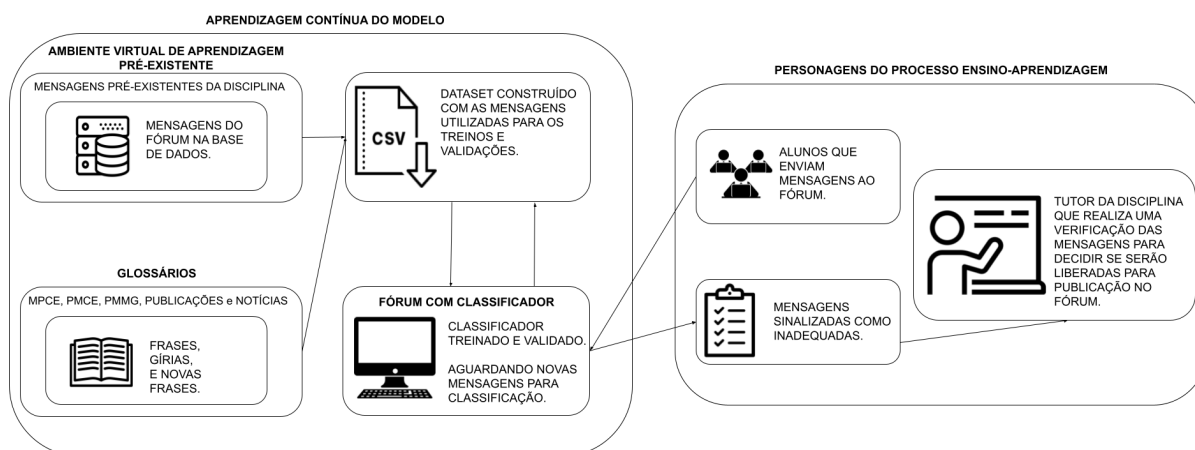
Para a escolha dos parâmetros dos classificadores, são realizados, previamente, alguns testes e avaliações de resultados, até obtenção do melhor resultado e escolha do melhor parâmetro.

Para a Árvore de Decisão, é utilizada a *entropy* e *gini*. No classificador k-NN, é utilizado o valor 3 para *k*. O classificador MLP é parametrizado com a função de ativação *relu* e a função de otimização utilizada é a *Adam*, com a taxa de aprendizagem variando de 0,1 a 0,5, com 150 neurônios de entrada, 100 neurônios na camada oculta e 50 neurônios na saída. Estabelece-se o limite de 300 iterações para a convergência. O SVM é parametrizado com o *kernel* linear e a regularização varia entre 0,1 e 10.

## 4.6 Conclusão

Este capítulo apresentou a metodologia utilizada para a construção do modelo de moderação inteligente das mensagens. Na Figura 17, apresenta-se uma generalização de como o modelo pode ser empregado em outros ambientes virtuais de aprendizagem, independentemente da disciplina. Porém, são necessárias mensagens prévias em bases de dados dos AVAs, para que seja possível construir o *dataset*, contendo as mensagens consideradas adequadas, juntamente às mensagens retiradas dos glossários e consideradas inadequadas, para que, então, o classificador possa aprender a classificar as mensagens referentes a outras disciplinas.

Figura 17 – modelo de moderação inteligente de mensagens em ambientes virtuais de aprendizagem



Fonte: autor.

Após os treinamentos, validações e testes dos classificadores, é possível receber as novas mensagens dos alunos, classificá-las e, se for necessário, sinalizar as mensagens que são inadequadas, para que os tutores possam avaliar as mensagens e decidirem se estas serão disponibilizadas no fórum. Além disso, podem-se também acrescentar, no *dataset*, todas as novas mensagens que forem enviadas pelos alunos e já classificadas e rotuladas, para que seja possível auxiliar o processo de reaprendizagem do classificador escolhido para o modelo.

No capítulo seguinte, são apresentados todos os resultados que são alcançados, com a utilização de cada um dos classificadores, em combinação com as técnicas de pré-processamento e as extrações de características. Esses resultados servem para orientar quanto à escolha do melhor classificador e se será útil a utilização das técnicas de pré-processamento e/ou extração de características. Além disso são apresentados, também, os valores alcançados no tempos de execução, fator importante para a consideração da escolha de qual classificador será incorporado definitivamente no modelo, uma vez que, por se

tratar de um ambiente web e aplicações executadas em servidor, o tempo de processamento e de resposta das aplicações deve ser o mais breve possível, para não haja interferência no processo de ensino-aprendizagem.

## 5 Resultados

Em 2019, período relativo ao início da pesquisa proposta nesta dissertação, foi possível classificar a base de dados apenas com dois classificadores, o Naive Bayes e o SVM. Com o avanço dos estudos sobre as técnicas e outros classificadores, foi possível analisar o comportamento do modelo com a utilização de mais quatro classificadores, Árvore de Decisão, K-NN, Regressão Logística e MLP.

No primeiro momento da pesquisa, o classificador SVM apresentava o melhor desempenho. Já, o classificador MLP apresentou os melhores resultados. Neste capítulo serão apresentados todos os resultados dos classificadores e, ao final, será feita comparação entre os classificadores MLP e SVM, com o intuito de analisar qual a melhor escolha para a utilização no modelo proposto.

Ao utilizar o Count Vectorizer ao invés do TF-IDF, os classificadores recebem uma matriz com uma dimensão maior para classificarem, pois, de acordo com um dos parâmetros, termos com, pelo menos, duas ocorrências nos diálogos são mantidos no conjunto de dados para a classificação. Por conta disso, o classificador SVM teve uma maior desvantagem em relação ao MLP no cenário 1 e teve um desempenho inferior ao Naive Bayes, por não possuir melhor desempenho para conjunto de dados maiores.

Na Tabela 4, expõem-se as médias das métricas com seus respectivos desvios padrões dos seis classificadores no cenário de teste 1, em que nenhuma técnica de pré-processamento foi aplicada.

Tabela 4 – resultado dos classificadores no cenário 1 com count vectorizer

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 84,99 $\pm$ 0,023 | 83,09 $\pm$ 0,026 | 85,05 $\pm$ 0,021 | 83,59 $\pm$ 0,023                   | 0,2920     |
| KNN (3)             | 68,19 $\pm$ 0,091 | 73,24 $\pm$ 0,033 | 73,51 $\pm$ 0,049 | 67,20 $\pm$ 0,084                   | 0,2867     |
| Naive Bayes         | 95,30 $\pm$ 0,010 | 94,44 $\pm$ 0,125 | 94,85 $\pm$ 0,113 | 94,63 $\pm$ 0,114                   | 0,2297     |
| MLP                 | 96,13 $\pm$ 0,010 | 95,72 $\pm$ 0,012 | 95,60 $\pm$ 0,012 | <b>95,63 <math>\pm</math> 0,011</b> | 3,5517     |
| Regressão Logística | 93,33 $\pm$ 0,010 | 93,81 $\pm$ 0,009 | 90,97 $\pm$ 0,014 | 92,17 $\pm$ 0,011                   | 0,2585     |
| SVM (Linear)        | 93,70 $\pm$ 0,009 | 93,20 $\pm$ 0,011 | 92,60 $\pm$ 0,011 | 92,86 $\pm$ 0,010                   | 0,2985     |

A comparação dos desempenhos considera a métrica *F1-score* e, analisando os resultados, é possível observar, também, que, para o cenário 1, o classificador K-NN obteve o melhor desempenho em relação aos outros 3 cenários que utilizam Count Vectorizer na extração das características.

Acredita-se que ele teve esse desempenho por não existirem muitas palavras distintas, pois, sem a remoção das *stopwords*, foram mantidas diversas palavras que eram iguais ou que possuíam proximidades nas suas construções. Além disso, em virtude de K-NN classificar uma entrada de acordo com a proximidade com algum K, em que é supervisionado

e rotulado com sua classe específica, houve maior facilidade nesse cenário para classificar as mensagens, porém o seu resultado não é satisfatório em comparação a outros algoritmos como SVM e MLP.

É importante observar, também, os resultados do tempo de execução dos algoritmos, embora a MLP tenha alcançado nesse cenário, e também alcançará em todos os outros cenários, o melhor desempenho. Tanto com a utilização do CV como com a utilização do TF-IDF, seu tempo de execução é bastante superior a qualquer um dos outros classificadores.

Após a remoção das *stopwords*, um aspecto interessante ao usar o CV é que ele foi capaz de remover outros termos que, ainda, possuíam bastante frequência. Entretanto, não representavam claramente uma classe, como a palavra “hoje” foi um termo que não foi removido no pré-processamento, porém foi bastante comum encontrá-la em mensagens tanto adequadas como inadequadas.

Quando aplicamos o CV, geramos uma matriz esparsa (com muitos valores iguais a zero). À medida em que o texto aumenta, a quantidade de *features* (características representadas nas colunas da matriz) também aumenta. Logo, temos uma matriz ainda mais esparsa, o que é um problema, pois matrizes ou vetores esparsos requerem mais memória e recursos computacionais ao modelar. Esse é um fator negativo a se considerar caso pretenda-se escolher o CV para a utilização no modelo.

Na Tabela 5, apresentam-se resultados dos testes no cenário 2. Em concordância com o que foi explicado anteriormente, percebe-se que, com a remoção das *stopwords*, o K-NN perdeu desempenho, pelo fato de que nesse cenário existem mais palavras distintas do que palavras repetidas.

Tabela 5 – resultado dos classificadores no cenário 2 com count vectorizer

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 77,72 $\pm$ 0,016 | 78,82 $\pm$ 0,014 | 82,68 $\pm$ 0,013 | 77,21 $\pm$ 0,016                   | 0,3049     |
| KNN (3)             | 58,15 $\pm$ 0,159 | 73,51 $\pm$ 0,071 | 64,75 $\pm$ 0,073 | 54,66 $\pm$ 0,147                   | 0,2702     |
| Naive Bayes         | 91,92 $\pm$ 0,011 | 91,08 $\pm$ 0,014 | 90,35 $\pm$ 0,013 | 90,67 $\pm$ 0,013                   | 0,2310     |
| MLP                 | 94,55 $\pm$ 0,010 | 93,59 $\pm$ 0,011 | 94,17 $\pm$ 0,013 | <b>93,85 <math>\pm</math> 0,011</b> | 4,0908     |
| Regressão Logística | 90,10 $\pm$ 0,015 | 92,37 $\pm$ 0,011 | 85,52 $\pm$ 0,022 | 87,84 $\pm$ 0,019                   | 0,2621     |
| SVM (Linear)        | 91,09 $\pm$ 0,014 | 90,46 $\pm$ 0,014 | 89,86 $\pm$ 0,028 | 89,85 $\pm$ 0,017                   | 0,2933     |

Ao utilizar, apenas, a técnica de pré-processamento de *stemming* em junção com o Count Vectorizer no cenário 3, todos os algoritmos tiveram uma melhora em seus desempenhos conforme Tabela 6. Isso evidencia que o cenário 2 pode ser considerado como a pior configuração para o modelo em que se utiliza, apenas, a técnica de remoção de *stopwords* e o CV.

Tabela 6 – resultado dos classificadores no cenário 3 com count vectorizer

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 82,42 $\pm$ 0,020 | 80,89 $\pm$ 0,020 | 84,15 $\pm$ 0,019 | 81,34 $\pm$ 0,021                   | 0,2822     |
| KNN (3)             | 58,78 $\pm$ 0,119 | 71,97 $\pm$ 0,039 | 68,42 $\pm$ 0,076 | 57,67 $\pm$ 0,123                   | 0,2750     |
| Naive Bayes         | 94,28 $\pm$ 0,010 | 93,27 $\pm$ 0,012 | 93,88 $\pm$ 0,013 | 93,55 $\pm$ 0,012                   | 0,2404     |
| MLP                 | 96,25 $\pm$ 0,007 | 95,61 $\pm$ 0,010 | 95,94 $\pm$ 0,009 | <b>95,75 <math>\pm</math> 0,009</b> | 3,7505     |
| Regressão Logística | 93,44 $\pm$ 0,014 | 93,80 $\pm$ 0,013 | 91,20 $\pm$ 0,019 | 92,30 $\pm$ 0,016                   | 0,2633     |
| SVM (Linear)        | 93,15 $\pm$ 0,012 | 92,17 $\pm$ 0,015 | 92,27 $\pm$ 0,016 | 92,18 $\pm$ 0,014                   | 0,3006     |

No último cenário, utilizando Count Vectorizer, utilizaram-se as duas técnicas de pré-processamento, remoção de *stopwords* e a *stemming*. Os classificadores SVM e MLP mantiveram bons desempenhos com a utilização do CV. Entretanto, consoante Tabela 7, o classificador K-NN obteve o seu pior desempenho nesse cenário.

Tabela 7 – resultado dos classificadores no cenário 4 com count vectorizer

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 78,78 $\pm$ 0,021 | 79,51 $\pm$ 0,017 | 83,29 $\pm$ 0,016 | 78,26 $\pm$ 0,021                   | 0,3021     |
| KNN (3)             | 55,39 $\pm$ 0,141 | 71,90 $\pm$ 0,053 | 64,60 $\pm$ 0,074 | 52,81 $\pm$ 0,140                   | 0,2683     |
| Naive Bayes         | 90,80 $\pm$ 0,012 | 89,82 $\pm$ 0,016 | 89,33 $\pm$ 0,016 | 89,51 $\pm$ 0,014                   | 0,2333     |
| MLP                 | 94,37 $\pm$ 0,011 | 93,42 $\pm$ 0,013 | 93,82 $\pm$ 0,012 | <b>93,59 <math>\pm</math> 0,012</b> | 4,2588     |
| Regressão Logística | 89,81 $\pm$ 0,013 | 91,58 $\pm$ 0,011 | 85,59 $\pm$ 0,020 | 87,68 $\pm$ 0,017                   | 0,2590     |
| SVM (Linear)        | 91,35 $\pm$ 0,011 | 89,50 $\pm$ 0,015 | 91,69 $\pm$ 0,011 | 90,38 $\pm$ 0,013                   | 0,2916     |

Dependendo do tamanho da base de dados em que são aplicados esses classificadores, o uso do algoritmo de extração de características TF-IDF pode interferir no critério tempo de execução, porém, no cenário 1, em que os resultados são apresentados na Tabela 8, os classificadores obtiveram seus melhores desempenhos.

Tabela 8 – resultado dos classificadores no cenário 1 com TF-IDF

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 88,43 $\pm$ 0,014 | 88,00 $\pm$ 0,016 | 85,28 $\pm$ 0,019 | 86,38 $\pm$ 0,017                   | 0,3088     |
| KNN (3)             | 94,31 $\pm$ 0,009 | 93,71 $\pm$ 0,011 | 93,43 $\pm$ 0,011 | 93,55 $\pm$ 0,010                   | 0,2607     |
| Naive Bayes         | 90,12 $\pm$ 0,015 | 93,10 $\pm$ 0,010 | 85,26 $\pm$ 0,016 | 87,82 $\pm$ 0,016                   | 0,2585     |
| MLP                 | 96,98 $\pm$ 0,006 | 96,70 $\pm$ 0,007 | 96,49 $\pm$ 0,008 | <b>96,58 <math>\pm</math> 0,006</b> | 5,4144     |
| Regressão Logística | 86,63 $\pm$ 0,016 | 90,98 $\pm$ 0,009 | 80,28 $\pm$ 0,019 | 83,05 $\pm$ 0,019                   | 0,3128     |
| SVM (Linear)        | 94,64 $\pm$ 0,010 | 95,17 $\pm$ 0,009 | 92,68 $\pm$ 0,013 | 93,76 $\pm$ 0,011                   | 0,4116     |

Além de passar para os classificadores um conjunto de dados menor, o TF-IDF consegue extrair os termos com maior relevância na base de dados sem a necessidade das funções de pré-processamento. Contudo, ele também foi aplicado em conjunto com as funções nos outros cenários para averiguar qual alteração no desempenho dos classificadores ocorre com essa combinação.

Na Tabela 9, os resultados apresentados são do cenário em que se aplica, somente, a remoção de *stopwords*.

Tabela 9 – resultado dos classificadores no cenário 2 com TF-IDF

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 86,19 $\pm$ 0,017 | 87,71 $\pm$ 0,018 | 80,65 $\pm$ 0,021 | 82,85 $\pm$ 0,021                   | 0,3525     |
| KNN (3)             | 91,08 $\pm$ 0,015 | 91,61 $\pm$ 0,013 | 88,06 $\pm$ 0,020 | 89,44 $\pm$ 0,017                   | 0,2892     |
| Naive Bayes         | 87,71 $\pm$ 0,019 | 91,49 $\pm$ 0,012 | 81,80 $\pm$ 0,021 | 84,53 $\pm$ 0,022                   | 0,2651     |
| MLP                 | 94,89 $\pm$ 0,009 | 93,86 $\pm$ 0,012 | 94,76 $\pm$ 0,010 | <b>94,26 <math>\pm</math> 0,010</b> | 6,0361     |
| Regressão Logística | 78,54 $\pm$ 0,020 | 87,37 $\pm$ 0,011 | 67,48 $\pm$ 0,019 | 68,97 $\pm$ 0,027                   | 0,3229     |
| SVM (Linear)        | 90,94 $\pm$ 0,013 | 92,90 $\pm$ 0,010 | 86,78 $\pm$ 0,016 | 88,96 $\pm$ 0,015                   | 0,3811     |

Assim como existiram cenários com a utilização do CV em que os classificadores tiveram desempenho menos favoráveis em relação aos outros cenários, a utilização do TF-IDF não apresentou diferença. Os resultados do cenário 3, apresentados na Tabela 10, revelam que a MLP teve um desempenho baixo nesse cenário, embora, ainda, se mantenha superior aos outros classificadores.

Tabela 10 – resultado dos classificadores no Cenário 3 com TF-IDF

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 87,41 $\pm$ 0,015 | 86,57 $\pm$ 0,016 | 84,40 $\pm$ 0,020 | 85,28 $\pm$ 0,018                   | 0,3227     |
| KNN (3)             | 92,53 $\pm$ 0,015 | 92,93 $\pm$ 0,014 | 90,01 $\pm$ 0,023 | 91,17 $\pm$ 0,018                   | 0,3060     |
| Naive Bayes         | 91,51 $\pm$ 0,017 | 94,14 $\pm$ 0,011 | 87,13 $\pm$ 0,021 | 89,56 $\pm$ 0,020                   | 0,2459     |
| MLP                 | 94,75 $\pm$ 0,012 | 95,44 $\pm$ 0,009 | 92,68 $\pm$ 0,023 | <b>93,77 <math>\pm</math> 0,015</b> | 5,8778     |
| Regressão Logística | 86,93 $\pm$ 0,019 | 91,45 $\pm$ 0,011 | 80,31 $\pm$ 0,023 | 83,21 $\pm$ 0,024                   | 0,2845     |
| SVM (Linear)        | 93,89 $\pm$ 0,013 | 94,54 $\pm$ 0,011 | 91,54 $\pm$ 0,019 | 92,80 $\pm$ 0,016                   | 0,3827     |

É importante observar, também, os desempenhos do classificador de Regressão Logística, que constituem o único classificador que apresentou bons resultados em relação ao momento de uso de CV ao invés de TF-IDF.

Na Tabela 11, mostram-se os resultados do quarto e último cenário utilizando TF-IDF. Esse cenário não apresentou resultados relevantes, logo, evidenciou-se que, no cenário 1, não há nenhuma técnica de pré-processamento e o uso, apenas, do TF-IDF é a melhor configuração para esse modelo proposto em que os classificadores conseguiram obter os melhores desempenhos, com destaque para os classificadores MLP e SVM.

Tabela 11 – resultado dos classificadores no cenário 4 com TF-IDF

| Classificador       | Acc               | Precision         | Recall            | F1-score                            | Tempo (ms) |
|---------------------|-------------------|-------------------|-------------------|-------------------------------------|------------|
| Árvore de Decisão   | 87,79 $\pm$ 0,012 | 86,83 $\pm$ 0,015 | 85,05 $\pm$ 0,014 | 85,79 $\pm$ 0,013                   | 0,3076     |
| KNN (3)             | 92,27 $\pm$ 0,016 | 92,75 $\pm$ 0,013 | 89,82 $\pm$ 0,025 | 90,95 $\pm$ 0,019                   | 0,2618     |
| Naive Bayes         | 91,55 $\pm$ 0,018 | 94,19 $\pm$ 0,011 | 87,28 $\pm$ 0,023 | 89,67 $\pm$ 0,021                   | 0,2365     |
| MLP                 | 94,70 $\pm$ 0,013 | 95,30 $\pm$ 0,010 | 92,81 $\pm$ 0,022 | <b>93,80 <math>\pm</math> 0,015</b> | 5,5128     |
| Regressão Logística | 86,47 $\pm$ 0,021 | 91,05 $\pm$ 0,012 | 79,44 $\pm$ 0,024 | 82,38 $\pm$ 0,026                   | 0,2730     |
| SVM (Linear)        | 94,13 $\pm$ 0,010 | 94,78 $\pm$ 0,010 | 91,78 $\pm$ 0,014 | 93,06 $\pm$ 0,012                   | 0,3732     |

Na Tabela 12, detalham-se todos os resultados alcançados pelo classificador MLP em todos os cenários, tanto com a utilização do Count Vectorizer como com a utilização do TF-IDF.

Tabela 12 – resultados do classificador MLP

| Técnica de Extração | Cenário | F1-score                            | Tempo (ms) |
|---------------------|---------|-------------------------------------|------------|
| TF-IDF              | 1       | <b>96,58 <math>\pm</math> 0,006</b> | 5,4144     |
| Count Vectorizer    | 3       | 95,75 $\pm$ 0,009                   | 3,7505     |
| Count Vectorizer    | 1       | 95,63 $\pm$ 0,011                   | 3,5517     |
| TF-IDF              | 2       | 94,26 $\pm$ 0,010                   | 6,0361     |
| Count Vectorizer    | 2       | 93,85 $\pm$ 0,011                   | 4,0908     |
| TF-IDF              | 4       | 93,80 $\pm$ 0,015                   | 5,5128     |
| TF-IDF              | 3       | 93,77 $\pm$ 0,015                   | 5,8778     |
| Count Vectorizer    | 4       | 93,59 $\pm$ 0,012                   | 4,2588     |

Observa-se que o seu tempo de execução sempre é bem superior em relação aos outros algoritmos, embora o seu resultado de desempenho seja satisfatório. É preciso analisar, cuidadosamente, a escolha desse algoritmo, pois deve-se considerar que o seu tempo de execução pode interferir no bom funcionamento de uma plataforma de educação à distância, principalmente quando se considerada, também, a classificação de muitas mensagens simultaneamente, em que haverá demanda de mais recursos de hardware, como capacidade de memória e processamento.

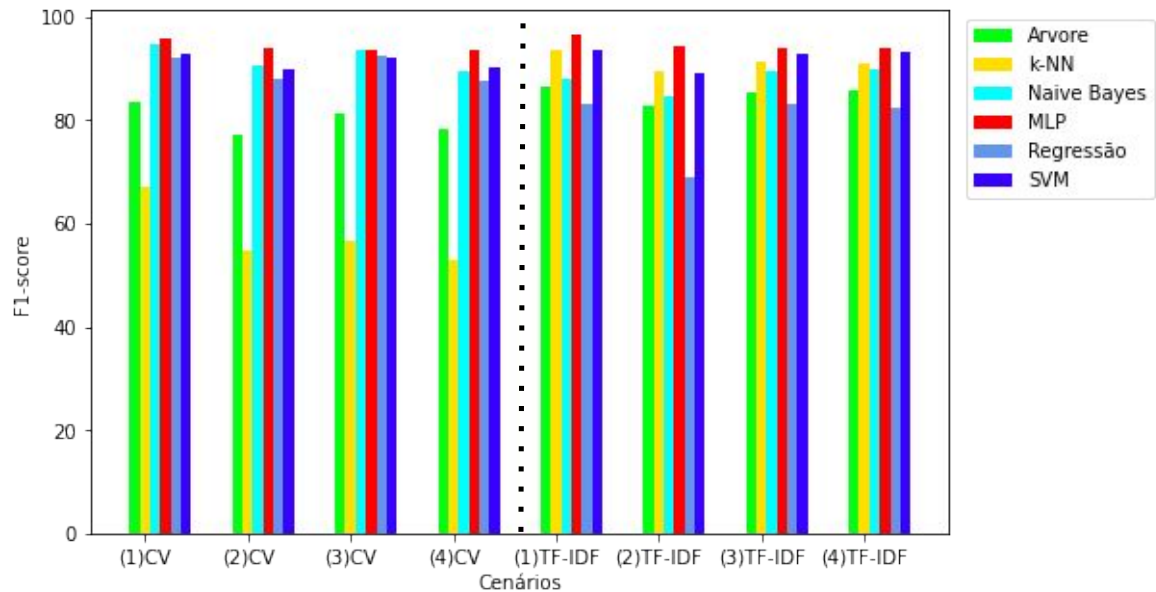
O classificador SVM não superou o MLP em nenhum cenário e se manteve como o segundo no *ranking* dos classificadores. Por essa razão, ele foi escolhido para ser comparado com o MLP, pois o seu tempo de execução é inferior. Seus resultados são apresentados na Tabela 13.

Tabela 13 – resultados do classificador SVM

| Técnica de Extração | Cenário | F1-score                            | Tempo (ms) |
|---------------------|---------|-------------------------------------|------------|
| TF-IDF              | 1       | <b>93,76 <math>\pm</math> 0,011</b> | 0,4116     |
| TF-IDF              | 4       | 93,06 $\pm$ 0,012                   | 0,3732     |
| Count Vectorizer    | 1       | 92,86 $\pm$ 0,010                   | 0,2985     |
| TF-IDF              | 3       | 92,80 $\pm$ 0,016                   | 0,3827     |
| Count Vectorizer    | 3       | 92,18 $\pm$ 0,014                   | 0,3006     |
| Count Vectorizer    | 4       | 90,38 $\pm$ 0,013                   | 0,2916     |
| Count Vectorizer    | 2       | 89,85 $\pm$ 0,017                   | 0,2933     |
| TF-IDF              | 2       | 88,96 $\pm$ 0,015                   | 0,3811     |

Conforme Figura 18, os resultados do SVM em relação ao MLP sempre foram inferiores, porém não são desprezíveis, pois o seu melhor resultado em uma das configurações de cenários alcançou o valor de 93,76% para a métrica F1-score.

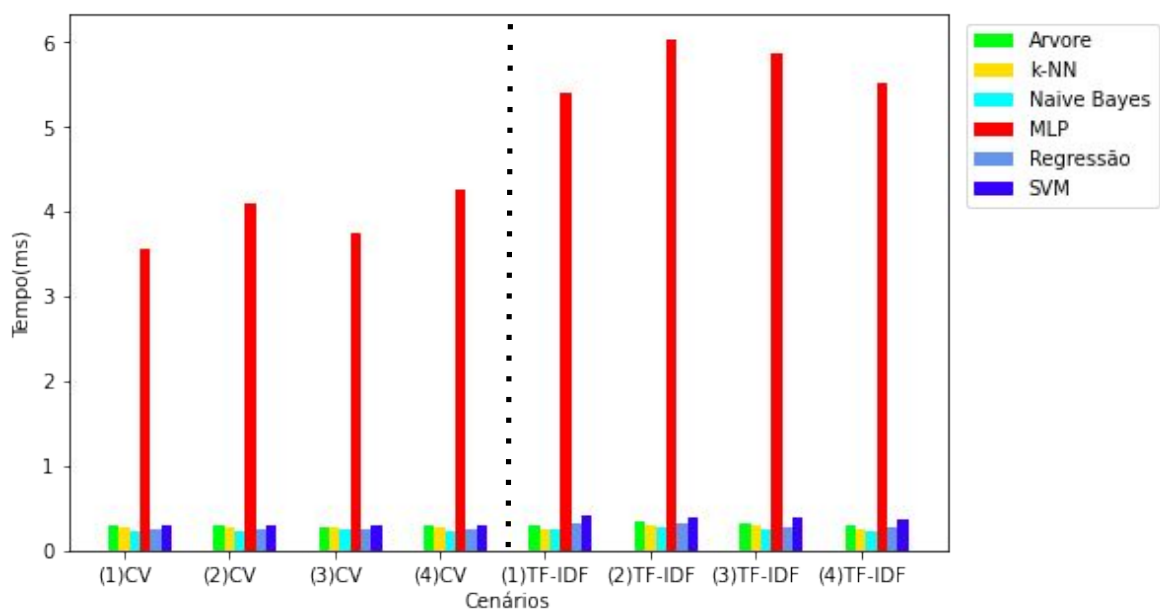
Figura 18 – comparação de F1-score entre os classificadores



Fonte: autor.

Em todos os cenários em que foi utilizado o classificador SVM, o seu tempo de execução não ultrapassou o valor de 1 milissegundo. Entretanto, com o uso do algoritmo MLP, os tempos de execuções desse algoritmo chegaram a alcançar valores de até 6 milissegundos conforme Figura 19.

Figura 19 – comparação de tempo(ms) entre os classificadores



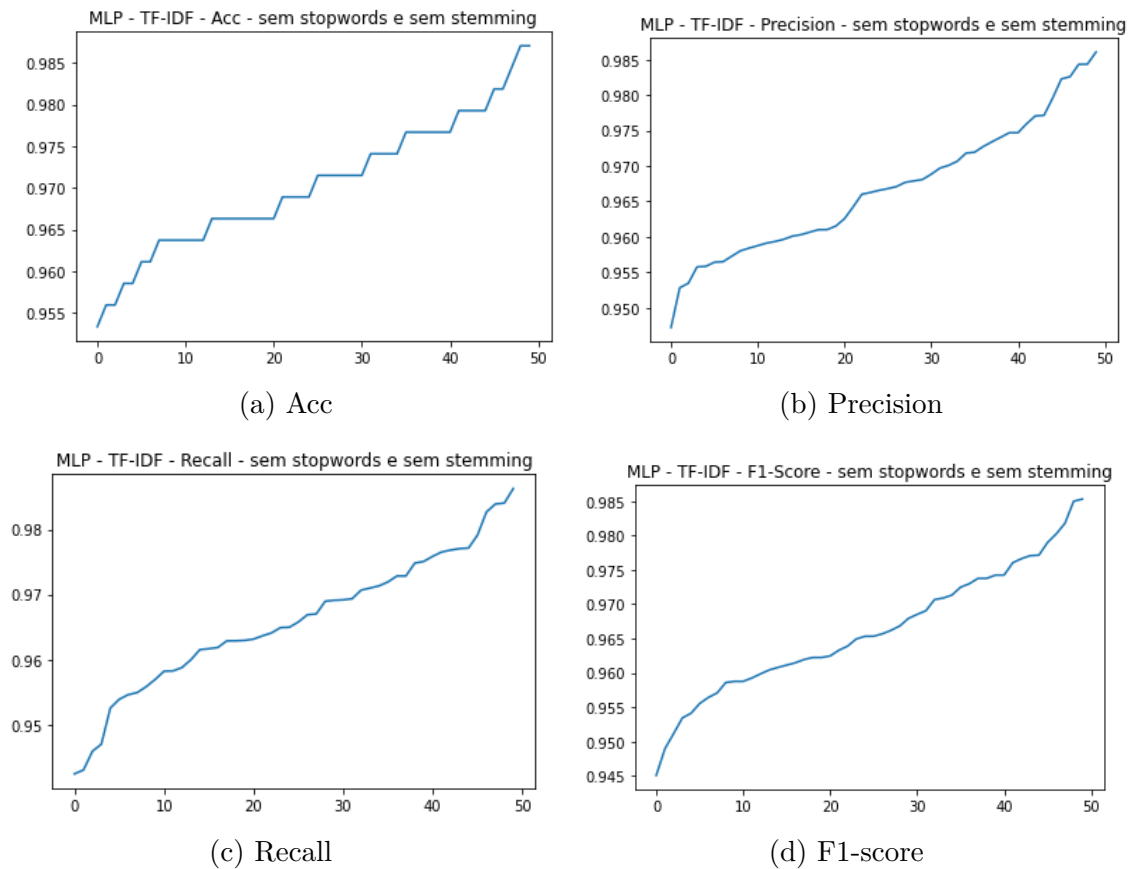
Fonte: autor.

Os valores de tempo de execução do MLP são elevados em comparação ao tempo de execuções dos outros algoritmos. O SVM possui implementação simples, melhor custo computacional e o seu tempo de execução é inferior ao do algoritmo MLP. Logo, este trabalho reforça a sugestão de escolha do algoritmo SVM para o problema proposto.

Se o propósito for realizar classificações assíncronas, o algoritmo MLP pode, então, ser utilizado como a melhor recomendação, mas, para o problema deste estudo, ele não é uma boa sugestão, pois se trata de um processo síncrono, simultâneo e em tempo real.

Na Figura 20, é possível compreender os valores alcançados das métricas de avaliação do classificador MLP para as 50 iterações realizadas. Nas figuras a seguir, apresentam-se os resultados ordenados de forma crescente.

Figura 20 – métricas do MLP

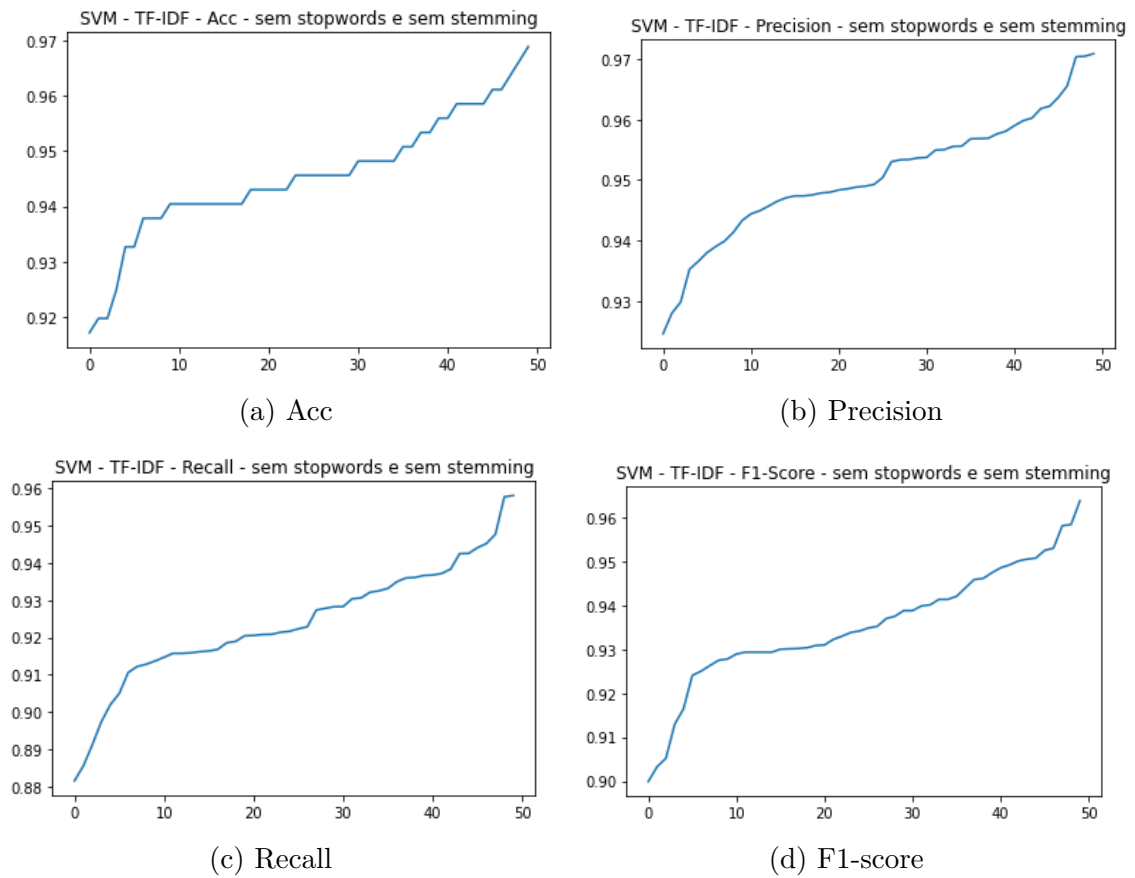


Fonte: autor.

Os valores alcançados nas métricas para o MLP variaram entre 94% e 98%. Para o SVM, as métricas variaram entre 88% e 97% consoante Figura 21. É possível concluir, então, que, se o problema necessitar de uma solução com melhor precisão e somente disso, o MLP será de fato o classificador recomendado, considerando-se que a aplicação terá uma predição mais lenta. Mas, se for possível tolerar um erro percentual um pouco maior e se a base de dados não for muito grande, o SVM será a melhor opção, por ser mais rápido no

treinamento e possuir precisão considerável.

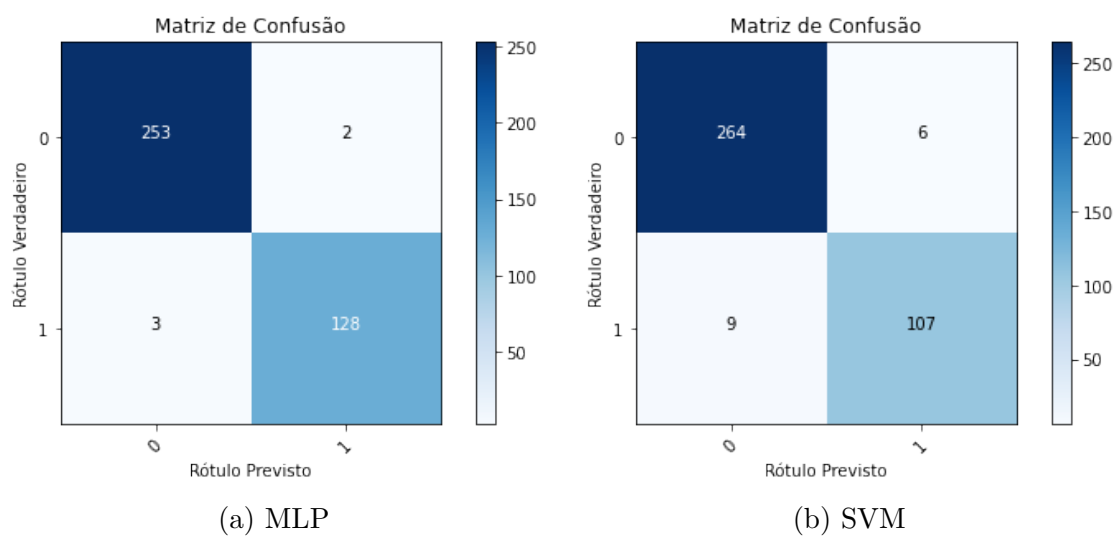
Figura 21 – métricas do SVM



Fonte: autor.

Outra maneira em que é possível compreender o desempenho de classificadores é por meio da visualização das matrizes de confusão. Foram separadas as duas matrizes de confusão das melhores iterações dos classificadores e apresentadas na Figura 22.

Figura 22 – matrizes de confusão da melhor iteração



Fonte: autor.

## 6 Considerações finais

Este trabalho apresentou um modelo de moderação inteligente capaz de classificar as mensagens enviadas em um fórum de um AVA como adequadas ou inadequadas. As mensagens inadequadas são consideradas aquelas contendo comunicação inapropriada à disciplina, frases contendo palavras utilizadas em gírias, ameaças, obscenidade, xingamentos e alguns termos com significados implícitos, comuns do vocabulário utilizado por pessoas privadas de liberdade.

Foram apresentados trabalhos que serviram para construir um caminho para a pesquisa, descobrir quais técnicas precisariam ser estudadas, quais deveriam ser as configurações ou parâmetros essenciais para os classificadores e, baseando-se nesses trabalhos já publicados na literatura científica, saber antecipadamente, quais técnicas não seriam eficientes para solucionar o problema proposto.

Na construção do *dataset*, que contém todas as mensagens que são classificadas, foram utilizados dados de duas fontes, de um curso à distância do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e frases, palavras e gírias que estão descritas em glossários sobre os termos usados por presos.

Em um primeiro momento desta pesquisa, foram utilizados, apenas, os classificadores Naive Bayes e SVM, em que o SVM apresentava melhores resultados, principalmente quando se considerava o tempo de execução do algoritmo.

Alguns autores alcançaram ótimos resultados com a Rede Neural Artificial MLP, valores de até 97% para *F1-score*. Esta pesquisa não foi aplicada no mesmo propósito que outras, mas buscou-se utilizar, também, outros algoritmos, para formar uma visão comparativa, pois há um pouco de semelhança entre as pesquisas.

Após analisar os desempenhos dos seis classificadores, foi possível concluir que a Rede Neural Artificial MLP também obteve os melhores valores para *F1-score*, alcançando uma média de 96,58%. Em seguida, o classificador SVM conseguiu alcançar uma média de 93,76% para *F1-score*.

O modelo precisa ser capaz de classificar as mensagens em um breve período de tempo, para que não interfira no fluxo do uso do fórum em um AVA. O classificador SVM alcançou um tempo médio de 0,411ms, enquanto a MLP alcançou um valor bem elevado no tempo de execução, 5,4144ms. Além disso, observou-se, também, que o algoritmo de extração de características TF-IDF contribui tornando o processo de classificação mais rápido e eficiente em relação ao CV.

Embora tenham-se utilizados outros quatro classificadores, Árvores de Decisão, K-

NN, MLP e Regressão Logística, a Rede Neural Artificial MLP e, mesmo que a MLP supere os resultados do SVM alcançados no primeiro momento da pesquisa, há considerações importantes a se fazer, quanto à escolha do melhor classificador para o problema proposto. O SVM continua como escolhido dentre os classificadores, por ter alcançado ótimos resultados para *F1-score* e, principalmente, por possuir o menor tempo de execução.

Com a moderação dos diálogos realizados pelos alunos privados de liberdade, eleva-se o nível de segurança empregado na educação à distância voltada para esse público, o que incentiva o seu emprego e evita o uso incorreto das ferramentas educacionais.

Como trabalhos futuros, pretende-se incorporar o modelo inteligente de moderação em um ambiente virtual de aprendizagem, como no Moodle, em um curso à distância oferecido para uma comunidade prisional, e utilizar o classificador SVM sem as técnicas de remoção de *stopwords* e *stemming*, utilizando, apenas, o TF-IDF, pois nessa configuração o classificador obteve o seu melhor desempenho.

A função principal do modelo será de classificar, automaticamente, cada mensagem enviada ao fórum e evitar o envio das mensagens inadequadas. Além disso, como diferencial, exibir alertas visíveis somente aos tutores, com informações do usuário e da mensagem que foi impedida de ser enviada, permitindo optar por enviar a mensagem ou não. Pretende-se, também, estender os testes e registrar os novos resultados em publicações.

Além dos objetivos técnicos das pesquisas futuras, espera-se alcançar contribuições sociais para essa comunidade em questão, que seja possível viabilizar a sua formação e desenvolvimento de suas competências, que se possa contribuir com a ressocialização desses alunos e, se possível, mesmo que de maneira indireta, contribuir, positivamente, para a redução do número de reincidências em crimes.

# Referências

- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. *Machine learning - Springer*, v. 6, n. 1, p. 37–66, 1991. Citado na página 30.
- AKYUZ, H. I.; KURT, M. Effect of teacher's coaching in online discussion forums on students' perceived self-efficacy for the educational software development. *Procedia - Social and Behavioral Sciences, World Conference on Learning, Teaching and Administration Papers*, v. 9, p. 633–637, 2010. Citado na página 15.
- ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, Taylor Francis Group*, v. 46, n. 3, p. 175–185, 1992. Citado 2 vezes nas páginas 28 e 29.
- AMADEUS. *Amadeus documentation*. 2021. Disponível em: <<https://softwarepublico.gov.br/social/amadeus>>. Acesso em: 16 jan. 2021. Citado na página 16.
- AMARAL, F. *Aprenda Mineração de Dados: Teoria e prática*. São Paulo: Alta Books, 2016. Citado na página 24.
- AMBRÓSIO, A. P. L.; MORAIS, E. A. M. *Mineração de Textos*. Goiás: Instituto de Informática da Universidade Federal de Goiás, 2007. 29 p. Citado 2 vezes nas páginas 20 e 21.
- ARANHA, C. N. *Uma Abordagem de Pré-Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência*. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 2007. 144 p. Citado na página 20.
- ARANHA, C. N.; PASSOS, E. A tecnologia de mineração de textos. *Revista Eletrônica de Sistemas de Informação*, v. 5, n. 2, p. 1–8, 2006. Citado na página 20.
- ARANHA, C. N.; VELLASCO, M. *Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional*. Rio de Janeiro: Pontifícia: Universidade Católica do Rio de Janeiro, 2006. Citado 2 vezes nas páginas 20 e 21.
- AZEVEDO, B. F. T.; BEHAR, P. A.; REATEGUI, E. B. Análise das mensagens de fóruns de discussão através de um software para mineração de textos. *Simpósio Brasileiro de Informática na Educação - SBIE*, p. 20–29, 2011. Citado 2 vezes nas páginas 21 e 46.
- BAEZA, Y. R.; RIBEIRO NETO, B. *Modern Information Retrieval*. United Kingdom: Addison-Wesley, 1999. Citado na página 15.
- BAKER, J. Preprocessing text data: Learn how to apply the bag-of-words and tf-idf model to unstructured text. *Cambridge Spark*, 2019. Citado na página 23.
- BEN-HUR, A. Support vector clustering. *Journal of machine learning research*, v. 2, n. 1, p. 125–137, 2001. Citado na página 37.
- BRASIL. *Lei 7.210/84 – Lei de Execução Penal*. Brasília: Presidência da República: Casa Civil, 1984. Citado na página 14.

BRASIL. *Infopen - Levantamento Nacional de Informações Penitenciárias - Jun/2016*. Brasília: Ministério da Justiça: DEPEN - Departamento Penitenciário Nacional, 2016. Citado na página 14.

BROWNLEE, J. A gentle introduction to k-fold cross-validation. *Machine Learning Mastery*, 2018. Citado na página 38.

BURIOL, T. M.; ARGENTA, M. A. Acelerando o desenvolvimento e o processamento de análises numéricas computacionais utilizando python e cuda. *Métodos Numéricos e Computacionais em Engenharia - CMNE CILAMCE*, 2009. Citado na página 22.

CANDIDO, A. L. M.; FURTADO JÚNIOR, C. G.; FREITAS, F. A. S. Moderação inteligente de mensagens em ambientes virtuais de aprendizagem para alunos privados de liberdade. *Simpósio Brasileiro de Informática na Educação*, p. 1838–1847, 2019. Citado na página 45.

CATAE, F. S. *Classificação automática de texto por meio de similaridade de palavras: um algoritmo mais eficiente*. São Paulo: Universidade de São Paulo, 2012. (Dissertação). Citado 2 vezes nas páginas 21 e 28.

CHOWDHURY, G. G. Natural language processing. annual review of information science and technology. *Wiley Online Library*, v. 37, n. 1, p. 51–89, 2003. Citado na página 21.

CORTES C.; VAPNIK, V. Support-vector network. *Machine learning - Springer*, v. 20, n. 3, p. 273–297, 1995. Citado 2 vezes nas páginas 36 e 37.

DAVID, P. B. et al. Análise da afetividade em fóruns virtuais: Construção de uma ontologia de domínio. *Nuevas Ideas en Informatica Educativa (TISE 2014)*, p. 821–826, 2014. Citado na página 15.

DOKEOS. *Dokeos documentation*. 2021. Disponível em: <<https://www.dokeos.com>>. Acesso em: 03 jan. 2021. Citado na página 16.

DUMAIS, S. T. Inductive learning algorithms and representations for text categorization. *Proceedings of the seventh international conference on Information and knowledge management - ACM*, p. 148–155, 1998. Citado na página 25.

FELDMAN, R.; SANGER, J. *The text mining handbook: advanced approaches in analyzing unstructured data*. United Kingdom: Cambridge University Press, 2007. Citado 2 vezes nas páginas 20 e 21.

FERREIRA, M. A. D. et al. Mineração de texto aplicada à identificação de colaboração em fóruns educacionais. *Simpósio Brasileiro de Informática na Educação - SBIE*, p. 1437–1446, 2017. Citado 2 vezes nas páginas 46 e 47.

FERREIRA, M. M. *Educação a distância para sistemas prisionais: um estudo sobre viabilidades técnicas de infraestrutura necessária para implementação da educação em rede nas escolas do sistema prisional*. Belo Horizonte: Universidade Federal de Minas Gerais: Faculdade de Educação, 2016. (Dissertação). Citado 2 vezes nas páginas 14 e 15.

FURTADO JUNIOR, C. G. *Contribuições à modelagem e à simulação de sistemas distribuídos, usando redes de petri coloridas: alocação de webcaches e particionamento de modelos em processos independentes*. Fortaleza: Universidade Federal do Ceará, 2016. (Tese). Citado na página 48.

GABRIELLI, K. S. *O papel da mediação pedagógica em fóruns educacionais de cursos on-line de língua estrangeira*. São Paulo: UNESP, 2016. 212 p. Citado na página 45.

GAMA, J. Concept drift in decision-tree learning for data streams. *Proceedings of the Fourth European Symposium on Intelligent Technologies and their implementation on Smart Adaptive Systems*, Germany: Aachen, Verlag Mainz, p. 218–225, 2004. Citado na página 25.

GAO, G.; GUAN, S. Text categorization based on improved rocchio algorithm. *International Conference On Systems And Informatics (icsai2012)*, p. 2247–2250, 2012. Citado na página 43.

GOMES, H.; CASTRO NETO, M. de; HENRIQUES, R. Text mining: Sentiment analysis on news classification. *8th Iberian Conference on Information Systems and Technologies (CISTI)*, p. 1–6, 2013. Citado na página 31.

HAO, P.; YING, D.; LONGYUAN, T. Application for web text categorization based on support vector machine. *International Forum On Computer Science-technology And Applications*, p. 42–45, 2009. Citado na página 42.

HE, W.; ZHA, S.; LI, L. Social media competitive analysis and text mining: A casae study in the pizza industry. *International Journal of Information Management*, p. 464–472, 2013. Citado na página 21.

HERTZ, J.; PALMER, R. G.; KROGH, A. S. *Introdução à teoria da computação neural*. Nova York: Perseus Books, 1990. Citado na página 32.

HOSMER, D. W.; LEMESHOW, S. *Applied Logistic Regression*. Nova York: John Wiley and Sons, 1985. Citado na página 35.

HUANG, J.; LU, J.; LING, C. X. Comparing naive bayes, decision trees, and svm with auc and accuracy. *Data Mining - ICDM 2003. Third IEEE International Conference on*, p. 553–556, 2003. Citado na página 25.

HYAFIL, L.; RIVEST, R. L. Constructing optimal binary decision trees is npcomplete. *Information Processing Letters*, v. 5, n. 1, p. 15–17, 1976. Citado na página 28.

ISLAM, S.; ELAHI, F.; AHMED, S. I. A support vector machine mixed with tf-idf algorithm to categorize bengali document. *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, p. 16–18, 2017. Citado 2 vezes nas páginas 21 e 43.

JAIN, P. Virtual learning environment. *International Journal in IT Engineering* 3.5, p. 75–84, 2015. Citado na página 15.

JO, T. String vector based knn for text categorization. *International Conference on Advanced Communications Technology (ICACT)*, p. 458–463, 2017. Citado na página 44.

KOTSIANTIS, S. B.; ZAHARAKIS, I. D.; PINTELAS, P. E. Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, v. 160, p. 3, 2007. Citado na página 25.

KRUSE, R. et al. Multilayer perceptrons. *Computational Intelligence*, p. 47–81, 2013. Citado na página 46.

- LANDIS, J.; KOCH, G. The measurement of observer agreement for categorical data. *Biometrics*, n. 33, p. 159–174, 1977. Citado na página 45.
- LIN, F.; HSIEH, L.; CHUANG, F. Discovering genres of online discussion threads via text mining. *Computers Education*, p. 481–495, 2009. Citado na página 45.
- LOBATO, M. C. A. A presença docente em fórum educacional virtual. *Linha d'Água*, v. 2, n. 25, p. 269–284, 2012. Citado na página 15.
- LOPES, M. C. S. *Mineração de Dados Textuais*. Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2004. Citado na página 20.
- LUI, A. K.; LI, S. C.; CHOY, S. O. An evaluation of automatic text categorization in online discussion analysis. *IEEE International Conference on Advanced Learning Technologies*, 2007. Citado 2 vezes nas páginas 45 e 47.
- MA, Y.; GUO, G. *Support Vector Machines Applications*. New York: Springer International Publishing, 2014. Citado na página 25.
- MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. *An introduction to information retrieval*. United Kingdom: Cambridge University Press Cambridge, 2008. v. 1. Citado na página 25.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. Citado na página 24.
- MOODLE. *Moodle documentation*. 2021. Disponível em: <<https://www.moodle.org>>. Acesso em: 15 jan. 2021. Citado na página 16.
- MOONEY, R. J.; BENNETT, P. N.; ROY, L. Book recommending using text categorization with extracted information. *Papers of the AAAI-98/ICML-98 Workshop on Learning for Text Categorization and Papers of the AAAI-98 Workshop on Recommender Systems*, p. 70–74, 1998. Citado na página 24.
- MORAIS, E. A. M.; AMBRÓSIO, A. P. L. *Mineração de textos*. Goiás: Relatório Técnico–Instituto de Informática da Universidade Federal de Goiás, 2007. Citado na página 21.
- MOREIRA, S. Rede neural perceptron multicamadas. *Medium*, 2018. Disponível em: <<https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9>>. Acesso em: 28 nov. 2019. Citado na página 34.
- MUBEEN, S.; QASEEM, M. S.; GOVARDHAN, A. Usage of distinctive classifiers for text categorization using distributional features. *Annual IEEE India Conference*, p. 133–137, 2011. Citado na página 42.
- NLTK. Natural language toolkit. *NLTK.ORG*, 2018. Disponível em: <<https://www.nltk.org>>. Citado na página 52.
- OLIVEIRA JUNIOR, R. L. d.; ESMIN, A. A. A. Monitoramento automático de mensagens de fóruns de discussão usando técnica de classificação de texto semisupervisionado. *Simpósio Brasileiro de Informática na Educação - SBIE*, 2011. Citado na página 46.

- QU, Z. et al. Improved bayes method based on tf-idf feature and grade factor feature for chinese information classification. *International Conference on Big Data and Smart Computing*, p. 677–680, 2018. Citado 2 vezes nas páginas 21 e 44.
- QUINLAN, J. R. C4.5: Programs for machine learning. *Morgan kaufmann*, v. 1, p. 147–159, 1993. Citado na página 28.
- RAVI, S.; KIM, J. Profiling student interactions in threaded discussions with speech act classifiers. *Artificial Intelligence in Education*, 2007. Citado 2 vezes nas páginas 45 e 47.
- RENNIE, J. D. M. et al. Tackling the poor assumptions of naive bayes text classifiers. *Proceedings of the Twentieth International Conference on Machine Learning*, p. 616–623, 2003. Citado 2 vezes nas páginas 31 e 32.
- REZENDE, S. O.; MARCACINI, R. M.; MOURA, M. F. O uso da mineração de textos para extração e organização não supervisionada de conhecimento. *Revista de Sistemas de Informação da FSMA*, v. 1, n. 7, p. 7–21, 2011. Citado na página 16.
- RIGO, S. J. et al. O papel do processamento de linguagem natural e da representação de conhecimento na extração de informações em mensagens textuais na educação a distância. *Simpósio Brasileiro de Informática na Educação-SBIE*, v. 24, n. 1, p. 647, 2013. Citado na página 15.
- ROLIM, V. B.; FERREIRA, R.; COSTA, E. Identificação automática de dúvidas em fóruns educacionais. *Simpósio Brasileiro de Informática na Educação - SBIE*, p. 936–945, 2016. Citado na página 46.
- SCHEUER, O.; MCLAREN, B. M. Helping teachers handle the flood of data in online student discussions. *Intelligent Tutoring Systems*, p. 323–332, 2008. Citado 2 vezes nas páginas 45 e 47.
- SEBASTIANI, F. Machine learning in automated text categorization. *ACM - computing surveys (CSUR)*, v. 34, n. 1, p. 1–47, 2002. Citado na página 25.
- SHUNG, K. P. Accuracy, precision, recall or f1? *Towards Data Science*, 2018. Citado na página 39.
- SILVA, E. R. C. C. *Técnicas de data e text mining para anotação de um arquivo digital*. Aveiro: Universidade de Aveiro, 2010. 89 p. Citado na página 20.
- SMOLA, A.; VISHAWNATHAN, S. V. N. *Introduction to Machine Learning*. Cambridge: Cambridge University Press, 2008. Citado na página 24.
- SOARES, F. d. A. *Categorização Automática de Textos Baseada em Mineração de Textos*. Rio de Janeiro: Universidade Católica do Rio de Janeiro: Departamento de Engenharia Elétrica, 2004. 158 p. (Tese). Citado na página 16.
- TAN, A. Text mining: The state of the art and the challenges. *Pacific-Asia Workshop on Knowledge Discovery from Advanced Databases - PAKDD'99. Beijing*, p. 65–70, 1999. Citado na página 15.
- TELEDUC. *TelEduc documentation*. 2021. Disponível em: <<https://www.teleduc.org.br>>. Acesso em: 12 jan. 2021. Citado na página 16.

- VALERO, A. T.; GÓMEZ, M. M.; PINEDA, L. V. Using machine learning for extracting information from natural disaster news reports. *Computación y Sistemas*, v. 13, n. 1, p. 33–44, 2009. Citado na página 42.
- WANG, Z.; QIAN, X. Text categorization based on lda and svm. *2008 International Conference on Computer Science and Software Engineering*, p. 674–677, 2008. Citado na página 42.
- XUE, D.; LI, F. Research of text categorization model based on random forests. *International Conference On Computational Intelligence Communication Technology*, p. 173–176, 2015. Citado na página 43.
- YANG, Y.; ZHANG, J.; KISIEL, B. A scalability analysis of classifiers in text categorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - ACM*, p. 96–103, 2003. Citado na página 25.
- ZHANG, X.; HUANG, H.; ZHANG, K. Knn text categorization algorithm based on semantic centre. *2009 International Conference on Information Technology and Computer Science*, p. 249–252, 2009. Citado na página 42.
- ZHOU, J. G.; WANG, K.; WU, J. A method of chinese text categorization based on proximal support vector machine. *International Conference On Machine Learning And Cybernetics*, p. 1615–1619, 2005. Citado na página 42.