UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEONARDO DA SILVA MORENO

APRENDIZAGEM DE MÁQUINA APLICADA PARA IDENTIFICAÇÃO DE PRECONCEITOS EM TEXTOS DE REDES SOCIAIS

PATO BRANCO

2023

LEONARDO DA SILVA MORENO

APRENDIZAGEM DE MÁQUINA APLICADA PARA IDENTIFICAÇÃO DE PRECONCEITOS EM TEXTOS DE REDES SOCIAIS

MACHINE LEARNING APPLIED TO IDENTIFY SOCIAL PREJUDICES IN SOCIAL NETWORK TEXTS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná.

Orientador: Drª. Rúbia Eliza de Oliveira Schultz Ascari

PATO BRANCO

2023

LEONARDO DA SILVA MORENO

APRENDIZAGEM DE MÁQUINA APLICADA PARA IDENTIFICAÇÃO DE PRECONCEITOS EM TEXTOS DE REDES SOCIAIS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 01/junho/2023

Rúbia Eliza de Oliveira Schultz Ascari
Doutora
Universidade Tecnológica Federal do Paraná

Eliane Maria de Bortoli Fávero
Doutora
Universidade Tecnológica Federal do Paraná

Mariza Miola Dosciatti
Doutora
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

RESUMO

As redes sociais são poderosas ferramentas capazes de compartilhar, em segundos, diversas informações com milhões de usuários. Esse poderio pode ser perigoso quando a informação disseminada é um discurso de ódio, prejudicando indivíduos e grupos direta e indiretamente. Sistemas que se alimentam de textos e informações de redes sociais, estão sujeitos a coletar esse tipo de dado, dado esse que pode prejudicar possíveis análises feitas nesses sistemas que se baseiam em textos. Dependendo do objetivo, os algoritmos podem tomar decisões errôneas, afetando diretamente um ou mais indivíduos, resolução essa que poderia ser diferente se textos repletos de intolerância não estivessem no conjunto de dados analisado, prejudicando o julgamento do programa. Utilizando métodos de classificação e de aprendizado de máquina supervisionado, este trabalho buscou identificar textos presentes em redes sociais na língua portuguesa que contenham algum tipo de discriminação ou preconceito, contribuindo tanto para o usuário final da rede social, quanto para desenvolvedores que buscam identificar e retirar este tipo de conteúdo de sua base de dados.

Palavras-chave: aprendizado de máquina; preconceito; discurso de ódio; rede social; língua portuguesa.

ABSTRACT

Social networks are powerful tools capable of sharing, in seconds, diverse information with millions of users. This power can be dangerous when the disseminated information is hate speech, harming individuals and groups directly and indirectly. Systems that feed on texts and information from social networks are subject to collecting this type of data, data that can harm possible analyzes made in these systems that are based on texts. Depending on the objective, the algorithms can make erroneous decisions, directly affecting one or more individuals, a resolution that could be different if texts full of intolerance were not in the analyzed dataset, impairing the program's judgment. Using methods of classification and supervised machine learning, this work sought to identify texts present in social networks in Portuguese that contain some type of discrimination or prejudice, contributing both to the end user of the social network and to developers who seek to identify and remove this type of content from your database.

Keywords: machine learning; preconception; hate speech; social network; portuguese language.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTAGEM DE CÓDIGOS FONTE

LISTA DE ABREVIATURAS E SIGLAS

AJL	*Algorithmic Justice League*
API	*Application Programming Interface*
COMPAS	*Correctional Offender Management Profiling For Alternative Sanctions*
CSV	*Comma-separated values*
FN	Falso Negativo
FP	Falso positivo
IA	Inteligência Artificial
LR	Regressão Logística
MNB	*Multinomial Naive-Bayes*
NB	*Naive-Bayes*
NLTK	*Natural Language Toolkit*
PLN	Processamento de Linguagem Natural
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul
re	*Regular Expression*
SMO	*Sequential Minimal Optimization*
SVM	*Support Vector Machines*
UTFPR	Universidade Tecnológica Federal do Paraná
VN	Verdadeiro Negativo
VP	Verdadeiro positivo

SUMÁRIO

1 INTRODUÇÃO

A Inteligência Artificial (IA), mais especificamente as tecnologias de redes neurais, vêm se desenvolvendo com maestria e velocidade nos últimos anos, sendo elas cada vez mais presentes no meio científico e social. Este trabalho visa contribuir com este avanço, adentrando na questão ética desses algoritmos e na identificação de preconceitos em textos.

Algoritmos de IA se alimentam de dados agrupados em um ou diversos conjuntos de dados que representam a fonte do conhecimento utilizado para treinar modelos inteligentes. Contudo, nem sempre esses dados são filtrados, gerando informações não confiáveis e que podem reproduzir algum tipo de preconceito, tornando o software enviesado, não confiável, e, algumas vezes, deixando de cumprir seu papel social.

Buscar a neutralidade de um algoritmo de IA, ou seja, evitar o viés algorítmico, é um objetivo que muitos tentam atingir. Mittelstadt, Allo e seus parceiros do Oxford Internet Institute e Alan Turing Institute fizeram suas contribuições no artigo “The ethics of algorithms: mapping the debate” (MITTELSTADT *et al.*, 2016), principalmente por meio de um mapa evidenciando a ética dos algoritmos, que não é uma solução para o problema, mas sim, uma sustentação para auxiliar discussões sobre o assunto. Cumprindo seu propósito, ROSSETTI e ANGELUCI (2021) fizeram, em seu artigo, seu próprio quadro (Quadro 1) apresentando sete questões éticas que são derivadas das seis preocupações citadas no trabalho de MITTELSTADT *et al.* (2016).

Quadro 1 – Questões éticas derivadas de preocupações citadas por MITTELSTADT *et al.* (2016)

Preocupações Éticas Trazidas por Algoritmos	**Questões Éticas Tratadas por Rossetti (2021)**
Evidências inconclusivas	Falibilidade
Evidências inextricáveis	Opacidade
Evidências mal direcionadas	Viés
Resultados injustos	Discriminação
Efeitos transformativos	Autonomia
Efeitos transformativos	Privacidade de Informações
Rastreabilidade	Responsabilidade

Fonte: Adaptado de ROSSETTI e ANGELUCI (2021), p. 7.

Algoritmos podem ser preconceituosos dependendo dos dados que são fornecidos a eles, podendo impactar negativamente na vida de diversas pessoas, assim como no caso citado por ROSSETTI e ANGELUCI (2021), o software estadunidense COMPAS (*Correctional Offender Management Profiling For Alternative Sanctions*), semelhante ao Victor[1] no Brasil, é usado

[1] Software utilizado pelo Supremo Tribunal de Justiça com o intuito de resolver ou mitigar os desafios pertinentes a uma maior eficiência e celeridade processuais. https://portal.stf.jus.br/noticias/verNoticiaDetalhe.asp?idConteudo=471331

para auxiliar tribunais em seus julgamentos por meio de processamento de dados textuais, mas acaba demonstrando comportamentos preconceituosos em seus resultados.

Algoritmos de análise de texto fazem uso de técnicas de Aprendizado de Máquina, estatísticas e linguísticas para processar grandes volumes de texto não estruturado ou texto que não tem um formato predefinido, para derivar percepções e padrões (TIBCO, 2022). Os resultados da análise de texto são úteis em vários contextos, e frequentemente são usados com técnicas de visualização de dados para facilitar e apoiar processos de tomada de decisão.

Considerando que algoritmos de IA podem expressar viés não intencional, e tomando como base os dados apresentados no Quadro 1, este trabalho tem como foco a quarta preocupação, ou seja, a discriminação causada por resultados injustos gerados por algoritmos de IA, focando-se em algoritmos que utilizam texto como dados de entrada. Assim, o objeto de estudo deste trabalho são algoritmos de análise de textos que possam ser úteis na identificação de preconceitos, evitando assim uma possível inserção desses dados em outros algoritmos de IA, melhorando a qualidade dos dados utilizados como entrada para treinamento de sistemas inteligentes, e consequentemente, a tomada de decisão. Uma das possíveis formas de alcançar o resultado almejado é realizando uma análise de sentimentos nos textos em questão, identificando possíveis discursos de ódio e eliminando-os do conjunto de dados.

1.1 Objetivos

1.1.1 Objetivo geral

Identificar preconceitos em textos de redes sociais na língua portuguesa a fim de melhorar a integridade e a qualidade da fonte de dados que um algoritmo de IA possa vir a consumir.

1.1.2 Objetivos específicos

- Criar um conjunto de dados contendo textos retirados de redes sociais incluindo termos considerados neutros, positivos e negativos, vinculados a algum preconceito social da língua portuguesa, ou considerados ofensivos.
- Treinar um modelo por meio de algoritmos de Aprendizagem de Máquina para identificar preconceitos sociais em textos da língua portuguesa.
- Contribuir para minimizar a discriminação causada por resultados injustos gerados por algoritmos de IA, evitando assim a disseminação de preconceitos que afligem diversos grupos de pessoas que sofrem por sua raça, etnia, religião ou orientação sexual.

1.2 Justificativa

O desenvolvimento de soluções e tecnologias para reduzir os vieses gerados por algoritmos de aprendizado vem se mostrando cada vez mais frequentes nos dias atuais. Redes sociais, grupos ativistas, programadores independentes e organizações sem fins lucrativos, como a *Algorithmic Justice League* (AJL)[2], contribuem para esse objetivo.

A "Política contra propagação de ódio" da rede social Twitter cita alguns exemplos de conteúdo que são proibidos como "Espero que todos os [nacionalidade] peguem COVID e morram." ou "Estou a fim de socar uns [ofensa racial]. Quem vem comigo?". Esse conteúdo, na maioria das vezes, só é removido da rede social após uma denúncia de outro usuário, porém, o cenário ideal seria que esses textos fossem identificados e removidos automaticamente.

Em 2020 o Twitter recebeu diversas denúncias de viés racista em seu algoritmo de cortes de imagens (TECMUNDO, 2020), uma ferramenta que tem como objetivo melhorar a visualização das imagens publicadas no site. Os usuários começaram a perceber que em muitas ocasiões, o algoritmo possuía uma preferência com indivíduos de etnia branca, cortando da imagem pessoas com outra etnia. Essa onda de denúncias fez com que a empresa fizesse mais testes e melhorias em seu algoritmo, além de prometer abrir o código fonte da ferramenta. No fim isso não foi o suficiente para mitigar o viés e a ferramenta deixou de ser utilizada pela rede social. Casos como esse mostram a responsabilidade que empresas têm perante seus algoritmos e o quão prejudicial eles podem ser para um indivíduo ou grupo de pessoas. Melhorias constantes e novas soluções para combater vieses de algoritmos devem ser um anseio de todos os profissionais da área e empreendedores.

Identificar preconceitos em textos de uma maneira eficaz traz benefícios para a sociedade e técnicas com esse objetivo podem ser aplicadas em diversas áreas, principalmente em redes sociais para automatizar a identificação e tomar medidas de forma mais eficiente para combater discursos de ódio e sua disseminação.

1.3 Estrutura do trabalho

Este trabalho está organizado em capítulos, como descrito a seguir. No Capítulo 2 está o referencial teórico que abrange temas e conceitos relacionados ao desenvolvimento do trabalho, como identificação de preconceitos e mineração de dados em textos e métodos de classificação. No Capítulo 3 são apresentados os trabalhos relacionados com a área de estudo em questão. No Capítulo 4 constam os materiais utilizados para desenvolvimento do trabalho e o método empregado. O Capítulo 5 contém a descrição dos resultados obtidos com a realização deste trabalho. Por fim, as referências utilizadas na composição do texto.

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

2 REFERENCIAL TEÓRICO

2.1 Identificação de Preconceitos

O primeiro passo para atingir os objetivos deste trabalho é entender o que é um preconceito e como ele é disseminado nas redes sociais em forma de discurso de ódio. Em 2010 surgiu uma organização chamada *Dangerous Speech Project* que segundo descrição em seu *site*[1], tem o objetivo de estudar os tipos de discurso que inspiram violência entre grupos de pessoas e buscar maneiras de mitigar estes acontecimentos sem ferir a liberdade de expressão. A organização atua em quatro áreas (DANGEROUS, 2022):

- Rastreamento e estudo de falas perigosas em muitos países.
- Pesquisa por respostas eficazes a falas perigosas e outras formas de expressão nocivas.
- Aconselhamento às mídias sociais e outras empresas de tecnologia sobre suas políticas e incentivo ao envolvimento em pesquisas transparentes.
- Ensino de ideias de fala perigosa para uma variedade de pessoas que as usam para estudar e combater a fala perigosa.

Por meio desses quatro âmbitos, o projeto contribui frequentemente com diversos trabalhos e pesquisas acadêmicas, tanto para identificação quanto para o combate a esse tipo de fala.

O Twitter[2] é uma plataforma de mídia social para comunicação online mediada por computador, que molda uma estrutura social emergente (KARAMI *et al.*, 2020). Segundo dados de julho de 2022, elencados por Ahlgren (2023), essa plataforma de comunicação possui mais de 1 bilhão de contas, com cerca de 500 postagens por dia, sendo que mais de 20 milhões de usuários estão no Brasil. Os usuários do Twitter podem postar comentários conhecidos como *"tweets"*, com quantidade de caracteres restrita. A menos que os *tweets* sejam privados, eles estão disponíveis publicamente e os usuários do Twitter podem mostrar sua reação e envolvimento com um *tweet* compartilhando-o em seu perfil (*retweet*), clicando no botão Curtir, marcando o nome de usuário de alguém ou respondendo ao autor do *tweet* (ARIGO *et al.*, 2018).

Considerando o grande volume de dados gerado nessa plataforma, o Twitter tornou-se uma fonte de dados global e o número de pesquisas envolvendo seu uso cresceu rapidamente na última década (KARAMI *et al.*, 2020). Segundo Bonin, Kirchof e Ripoll (2018), são numerosos os casos de racismo, xenofobia e preconceito no Twitter, e no Brasil, notadamente, tais casos estão relacionados aos homossexuais, transgêneros, mulheres, índios e nordestinos. Nesse

[1] https://dangerousspeech.org/what-we-do/
[2] https://twitter.com/

contexto, considerando que na base de dados do Twitter há uma quantidade bastante grande de textos na língua portuguesa, os quais podem conter frases vinculadas a preconceito, esta foi a rede social escolhida para servir como fonte de dados para este trabalho.

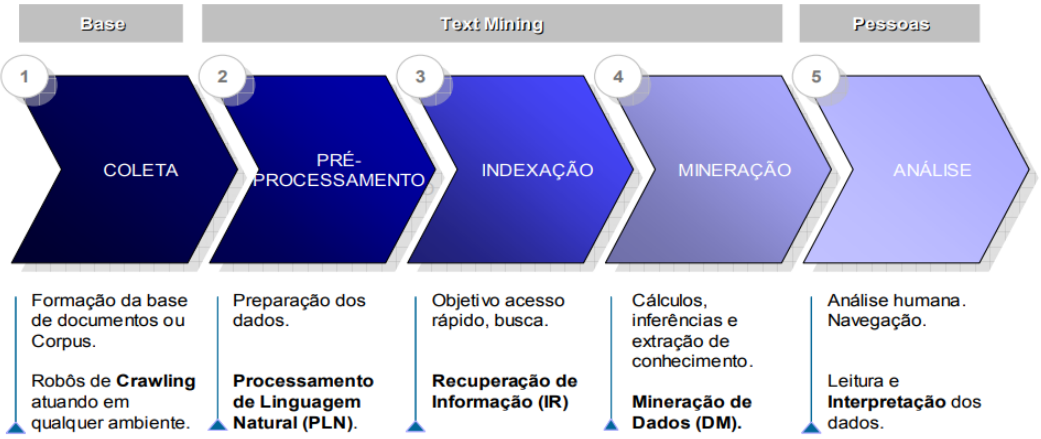
Há diversas maneiras de identificar um discurso de ódio na redes sociais, uma das principais é fazendo uma análise de sentimento nos textos recolhidos dessas plataformas, assim como SOUZA e VIEIRA (2012) fizeram em seu trabalho. Os autores analisaram dados do Twitter na língua portuguesa e enfatizaram as dificuldades em aplicar este método em dados retirados do microblog Twitter, empregando heurística para se adaptar ao conjunto de dados utilizado e aplicando léxicos de análise de sentimentos para a língua portuguesa. A análise de sentimentos é um campo de pesquisa bastante explorado atualmente e corresponde a uma das subcategorias da mineração de texto.

2.2 Mineração de Textos

"A mineração de texto refere-se geralmente ao processo de extração de informações e conhecimentos interessantes a partir de um texto não estruturado"(HOTHO; NÜRNBERGER; PAASS, 2005), para posteriormente, com o auxílio de tecnologias, fazer uso dessas informações para obter dados classificáveis e de fácil compreensão. Considerada uma abordagem eficaz e eficiente, a mineração de texto tem sido aplicada em uma ampla gama de interesses de pesquisa, tendo como principal objetivo organizar e compreender documentos, revelando padrões semânticos ocultos em um corpus (base de dados textual, conhecida também na literatura como Corpora) (KARAMI *et al.*, 2020).

O processo de mineração compreende várias etapas, que pode variar de acordo com o contexto em questão. Segundo ARANHA (2007), esse processo pode ser dividido em coleta, pré-processamento, indexação, mineração e análise, como apresentado na Figura 1. Este trabalho se baseará nos passos do modelo proposto por ARANHA (2007), com exceção das etapas de indexação e mineração, a fim de simplificar o processo. Pela razão da base ter sido criada, não se baseando em outras, não houve a necessidade de inferências e cálculos.

Figura 1 – Diagrama de camadas abstratas de sistema computacional para mineração de textos



Fonte: ARANHA (2007).

2.2.1 Coleta

A coleta de dados é a etapa inicial, e objetiva formar uma base de dados textual, ou seja, um corpus, com dados relacionados ao tipo de conhecimento que se deseja obter. Pode-se utilizar de várias fontes, como livros, e-mails, fóruns de internet, entre outros.

A fonte de dados que busca-se para este trabalho são textos retirados de redes sociais. Esta fonte é volumosa, porém, em muitos casos, se não souber o que e onde buscar, pode-se obter um conjunto de dados pouco efetivo. Diferente de outros tipos de fonte onde há a possibilidade de realizar mineração de textos, a maioria das redes sociais disponibilizam para uso uma *Application Programming Interface* (API), que refere-se a uma forma de comunicação entre a aplicação e o usuário. Por meio desta API, um usuário, geralmente autenticado, pode coletar informações, dados, e textos, que é o que se busca para este trabalho.

Desta forma, é possível criar conjuntos de dados com milhares de textos, como no trabalho de SOUZA e VIEIRA (2012), que para montar seu conjunto de dados, coletaram 1.700 *tweets* por meio da API do Twitter, utilizando filtros de idioma e o uso de *hashtags* para restringir sua busca.

Além das APIs, *web crawlers*[3] também podem servir como forma de mineração de textos, varrendo a *internet* e buscando as informações ideais para extrair. ARANHA (2007) empregou esse tipo de algoritmo na fase de coleta em seu modelo de mineração de textos, conforme apresentado na Figura 1.

Outra opção para coleta de dados é via *Web Scraping*, que segundo VIEIRA, SILVA e CORDEIRO (2019), refere-se à extração de informações em sites por meio da estrutura sintática de códigos HTML para detectar, selecionar e coletar os dados úteis.

[3] Web crawlers são algoritmos criados com o intuito de coletar dados na *internet* https://www.crawly.com.br/blog/o-que-e-crawler-robos-para-coleta-de-dados

2.2.2 Pré-processamento

Ao realizar a coleta de um texto ou documento escrito em linguagem natural, obtém-se dados em formato não estruturado, sendo necessário a realização de formatação, para estruturá-los de maneira padronizada, sem perder o sentido original. Essa etapa da mineração de textos é chamada de pré-processamento e é onde ocorre a preparação dos dados que foram coletados na etapa anterior, utilizando Processamento de Linguagem Natural (PLN).

PLN é um conjunto de técnicas teórico-computacionais que visam representar dados textuais e processar a linguagem natural humana para diversas tarefas (FILHO, 2014). Técnicas presentes neste conjunto, como remoção de *stopwords*, consiste em remover palavras sem significado, palavras vazias, que costumam ser as palavras mais comuns do idioma estudado. Podendo ser realizado durante ou depois da etapa de coleta, esse tratamento tem o intuito de melhorar o desempenho, a qualidade da busca e a análise feita nos textos coletados. Outras técnicas de pré-processamento comumente utilizadas são a lematização e o *stemming*, ambas reduzem as palavras à sua base, contribuindo para redução da dimensão do léxico. A lematização executa essa função de uma maneira mais sofisticada do que o *stemming*, ou seja, em vez de simplesmente cortar sufixos ou prefixos, a lematização usa análise morfológica para retornar à forma base da palavra, que pode ser chamada de "lemma". O lemma é uma palavra presente no dicionário e representa a forma canônica ou o lema da palavra. Por exemplo, a lematização da palavra "correndo" é "correr". A lematização é mais precisa do que o stemming, pois considera a classe gramatical e o contexto da palavra.

Outras técnicas como a tokenização também são comuns, ela consiste em segmentar um texto em unidades menores para que possa ser facilmente processado e analisado. O resultado final da frase "Este é um trabalho acadêmico." após ser tokenizada em palavras individuais fica: ["Este", "é", "um", "trabalho", "acadêmico", "."].

2.3 Léxico

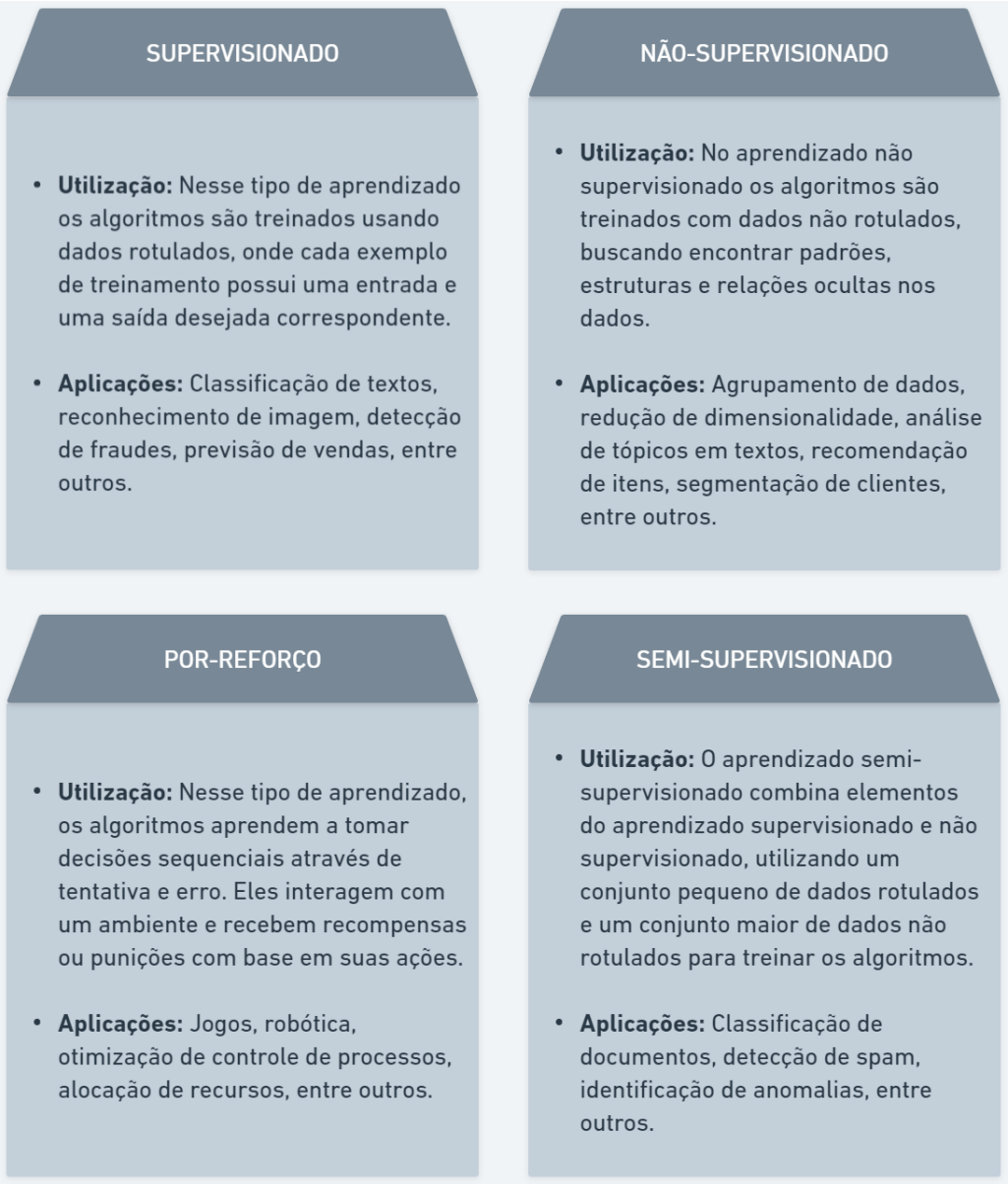
Um léxico é um conjunto de palavras de um idioma específico e pode ser utilizado para analisar morfologicamente e classificar palavras. Com esse dicionário é possível realizar diversos estudos, inclusive uma análise de sentimentos de textos curtos, como no trabalho de SOUZA e VIEIRA (2012) que utilizam o léxico *OpLexicon*, criado pela Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Segundo os autores, o léxico utilizado é constituído por cerca de 15.000 palavras polarizadas classificadas por sua categoria morfológica e notadas com polaridades positiva, negativa e neutra, e pode ser enriquecido com outros léxicos para ter resultados melhores.

Esta abordagem é comumente usada para analisar textos e palavras para classificá-los em duas ou três polaridades (1, 0, -1), assim como é feito em SOUZA e VIEIRA (2012).

2.4 Métodos de Classificação

A classificação de textos é a tarefa de associar textos em linguagem natural a rótulos pré-definidos. Uma abordagem empregada com bastante eficiência na realização dessa tarefa baseia-se no emprego de algoritmos de Aprendizado de Máquina (*Machine Learning*).

Figura 2 – Técnicas de Aprendizado de Máquina



Fonte: Autoria própria (2023).

Usado para ensinar às máquinas sobre como usar os dados com mais eficiência (KHAN *et al.*, 2010), o Aprendizado de Máquina é o campo de pesquisa dedicado ao estudo formal de sistemas de aprendizado. O processo de Aprendizado de Máquina passa pela fase de aprendizado, na qual o sistema analisa os dados e gera regras (encontrando algumas semelhanças entre os dados), e pela fase de validação, na qual as regras geradas são verificadas, com-

putando alguma função de avaliação de desempenho em um novo conjunto de dados (JAIN; KACPRZYK, 2002).

Algoritmos de Aprendizado de Máquina podem empregar diferentes abordagens de aprendizagem (Figura 2), como o aprendizado supervisionado, aprendizado semi-supervisionado, aprendizado não supervisionado e aprendizado por reforço. Segundo Park *et al.* (2022) o aprendizado supervisionado é usado principalmente para sistemas que prevêem valores ou rótulos, o aprendizado não supervisionado é usado para extrair padrões e o aprendizado por reforço é usado para estabelecer sistemas que podem interagir. O aprendizado semi-supervisionado combina técnicas do aprendizado não supervisionado e supervisionado, sendo trabalhado com conjunto de dados no qual apenas alguns pontos de dados são rotulados. A abordagem a ser empregada deve ser escolhida de acordo com a finalidade para a qual o Aprendizado de Máquina é aplicado.

Os algoritmos de aprendizado supervisionado utilizam a detecção de padrões para estabelecer previsões e podem ser subdivididos em algoritmos de classificação e algoritmos de regressão (REBALA; RAVI; CHURIWALA, 2019). Algoritmos de classificação têm como objetivo classificar algo (como por exemplo um texto) em um conjunto distinto de classes, de acordo com as características observadas pelo supervisor. Os algoritmos de regressão são semelhantes aos de classificação, porém neste caso ao invés de classificar algo, o objetivo é fazer uma regressão para tentar prever valores de uma variável contínua, funcionam com a compreensão de relação da máquina.

Neste trabalho será utilizado Aprendizado de Máquina Supervisionado, empregando o algoritmo de classificação *Support Vector Machine* (SVM).

2.4.1 *Support Vector Machine* (SVM)

Support Vector Machine é um modelo de Aprendizado de Máquina Supervisionado criado por CORTES e VAPNIK (1995) e pode ser utilizado tanto para classificar dados quanto para realizar inferências, estimando valores, tomando como base dados já existentes. As principais características desses modelos são (LORENA; CARVAHO, 2003):

- **Boa capacidade de generalização**: classificadores gerados por uma SVM costumam alcançar bons resultados de generalização. Essa capacidade é medida por sua eficiência na classificação de dados que não pertencem ao conjunto utilizado no treinamento.
- **Robustez em grandes dimensões**: SVMs são robustas diante de objetos de grandes dimensões, como imagens. Em outros classificadores gerados por outros métodos inteligentes sobre esses tipos de dados, comumente há a ocorrência de *overfitting*, cenário onde o algoritmo “decora” as regras e ao receber novos dados acaba obtendo resultados insatisfatórios.

- **Convexidade da função objetivo**: a aplicação das SVMs implica na otimização de uma função quadrática que possui apenas um mínimo global, sendo uma vantagem sobre, por exemplo, as Redes Neurais Artificiais, onde há a presença de mínimos locais na função objetivo a ser minimizada.
- **Teoria bem definida**: SVMs possuem uma base teórica bem estabelecida dentro da Matemática e Estatística.

No Aprendizado de Máquina Supervisionado, há uma técnica comum que é utilizada, o *holdout* (KOHAVI *et al.*, 1995). Esse processo serve para avaliar o desempenho de um modelo em dados não vistos, ajustar hiperparâmetros e realizar seleção de modelo. O *holdout* envolve a divisão do conjunto de dados em dois subconjuntos mutuamente exclusivos: um conjunto de treinamento e um conjunto de teste, como por exemplo, dividir a base para 80% dos dados serem utilizados para treinamento e 20% para testes.

3 TRABALHOS RELACIONADOS

3.1 Mineração de textos do Twitter utilizando técnicas de classificação

No campo da mineração de textos, o trabalho de LEITE (2015) busca, além de minerar textos da rede social Twitter, classificá-los em diferentes categorias. Para atingir seu objetivo, o autor, assim como na proposta deste trabalho, faz uso da API disponibilizada pela rede social para coletar as informações desejadas, e após esta etapa é executada uma limpeza nos dados obtidos, removendo acentos, pontuações, *stopwords*, reduzindo palavras ao seu radical (*stemming*) e empregando técnicas como *tokenização*, procedimento usado em análises morfológicas. Após definir as categorias manualmente, o autor aplica o algoritmo de classificação *Naive Bayes* do Apache Mahout para gerar o modelo de classificação de *tweets* (LEITE, 2015), que segundo ele, foi escolhido por possuir um código aberto, escalável e que apresenta resultados satisfatórios em outros trabalhos relacionados. LEITE (2015) alcançou uma acurácia de 97.619% em seus resultados, conseguindo classificar mais de 1600 *tweets* corretamente em categorias como esporte, economia, religião, política e outros.

3.2 Classificação Automática de Discursos de Ódio em Textos do Twitter

O estudo de NASCIMENTO (2019) compara técnicas utilizadas em modelos de aprendizado supervisionado de máquinas para classificar *tweets* que contenham discursos de ódio. Diferente do proposto neste trabalho, o autor divide os *tweets* em três categorias: Discurso normal; Discurso ofensivo; Discurso de ódio. NASCIMENTO (2019) exemplifica como os *tweets* devem ser classificados (Quadro 2).

Quadro 2 – Exemplo de classificação de sentenças encontradas em *tweets*

Classe	Sentença
Discurso de Ódio	@Irineu é um gay safado e fascista
Ofensivo	@Irineu é muito feio
Regular	@Irineu é meu amigo

Fonte: NASCIMENTO (2019).

O autor faz a comparação utilizando três classificadores, *Suport Vector Machine*, *Multinomial Naive-Bayes* (MNB), e Regressão Logística (LR). Ele conquista resultados melhores utilizando *tweets* escritos na língua inglesa, alcançando acurácias próximas a 90%, principalmente com o classificador SVM. *Tweets* escritos em português brasileiro possuem acurácias mais variadas, começando em aproximadamente 40% utilizando MNB balanceado sem *stemming* e chegando por volta dos 77% utilizando MNB desbalanceado sem stemming.

O autor também comenta sobre as limitações encontradas, sendo a principal o não tratamento de ironia e sarcasmo. O motivo do menor desempenho na classificações de textos em português, segundo NASCIMENTO (2019), foi o baixo número de *tweets* de discurso de ódio, afetando diretamente a base de treino do classificador.

3.3 Utilizando Análise de Sentimentos e SVM na Classificação de *Tweets Depressivos*

Utilizando os modelos de classificação SVM e *Naive Bayes*, CORTES e MELO (2021) buscaram classificar *tweets* depressivos encontrados na rede social Twitter e os quatros resultados possíveis elencados pelos autores são (CORTES; MELO, 2021):

- **Verdadeiro positivo (VP):** Ocorre quando a classe que se está buscando foi prevista corretamente.
- **Falso positivo (FP):** Ocorre quando a classe que se está buscando prever foi prevista como verdadeira incorretamente.
- **Verdadeiro Negativo (VN):** Ocorre quando a classe que se está buscando foi prevista como negativa de forma correta.
- **Falso Negativo (FN):** Ocorre quando a classe que se está buscando foi prevista como negativa incorretamente.

O método de otimização Grade de Busca (*Grid-searching*) foi utilizado no modelo SVM. A aplicação deste método tem como objetivo "construir um modelo para cada combinação possível de parâmetros que lhe foi dado e será feito uma validação cruzada para cada modelo" (CORTES; MELO, 2021).

Os resultados obtidos pelos autores indicam acurácias superiores a 90% com o modelo SVM, alcançando seu objetivo e obtendo valores baixos de falsos positivos ou falsos negativos.

3.4 CRIPS-DM 1.0 *Step-by-step data mining guide*

Em seu artigo, CHAPMAN *et al.* (2000) descrevem o modelo de processo para mineração de dados, o CRISP-DM, comumente usado por estudantes e profissionais da área de tecnologia auxiliando-os, principalmente, em seus processos de mineração de textos.

4 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais utilizados para desenvolvimento da proposta deste trabalho, e o método empregado.

4.1 Materiais

O desenvolvimento da proposta deste trabalho e testes foram realizados fazendo uso das seguintes tecnologias:

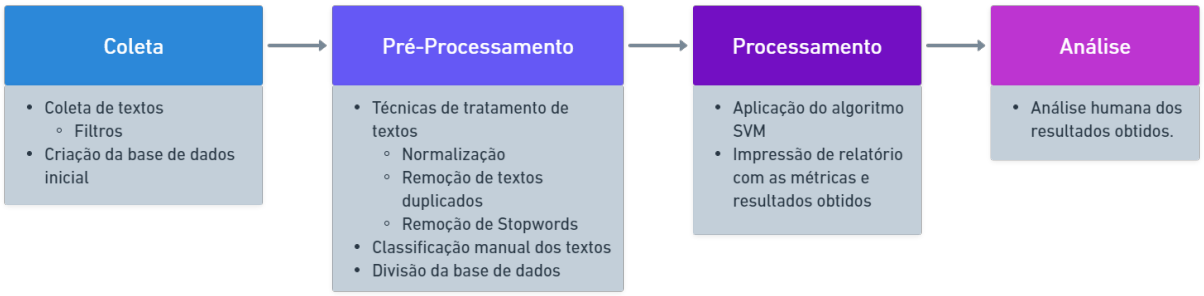
- **Ambiente de Desenvolvimento Integrado:** Visual Studio Code e Jupyter
- **Linguagem de Programação:** Python 3.11
- **Bibliotecas:** requests, os, json, csv, pandas, re, emoji, datetime, unidecode, nltk, sklearn, numpy
- **API:** Twitter v2

4.2 Método

Para desenvolvimento da proposta deste trabalho foi empregado um algoritmo de classificação com base em Aprendizado de Máquina Supervisionado, com o objetivo de identificar preconceitos em frases retiradas da base de dados do Twitter, comparando diferentes métodos de particionamento, buscando por maior desempenho e acurácia. Sendo uma das principais preocupações e motivação para condução deste trabalho, a incidência de possíveis resultados injustos gerados por algoritmos treinados a partir de conjuntos de dados não filtrados, gerando viés não intencional.

As etapas executadas para desenvolvimento deste trabalho são apresentadas de forma resumida na Figura 3.

Figura 3 – Etapas executadas neste trabalho para classificação de textos e identificação de preconceitos



Fonte: Autoria própria (2023).

4.2.1 Coleta

A coleta de dados foi realizada com o objetivo de obter um conjunto representativo de textos publicados em redes sociais na língua portuguesa, para análise e aplicação de técnicas de Aprendizado de Máquina Supervisionado. Nesta etapa, foi realizada a mineração dos textos, coletando *tweets*[1] publicados na rede social Twitter por meio da API disponibilizada pela empresa.

Para acessar a API o usuário necessita pagar por uma licença mensal que fornece ao desenvolvedor o direito de coletar 10.000 *tweets* por mês. Com a conta licenciada, basta criar um aplicativo no portal do desenvolvedor da API para obter o *Bearer Token*, uma chave de acesso única que autentica o usuário para ter acesso às informações da rede social, como *tweets* e informações de perfil.

Fazendo uso do plano inicial contratado, a API da empresa proporcionou acesso apenas aos *tweets* recentes publicados na plataforma, permitindo definir filtros e critérios para a coleta. Durante esse processo, foram estabelecidos os seguintes parâmetros:

- *Tweets* publicados na plataforma há pelo menos oito horas antes da coleta e no máximo há sete dias, a fim de garantir que os dados coletados fossem recentes e refletissem a dinâmica atual da rede social.
- Utilização das palavras-chave definidas no Quadro 3, relacionadas ao tema de interesse deste trabalho, palavras comumente utilizadas em frases preconceituosas ou ofensivas, para filtrar e coletar apenas os *tweets* potencialmente relevantes.

[1] Nome atribuído às publicações feitas na rede social Twitter. São textos de até 280 caracteres

Quadro 3 – Lista de Palavras utlizadas para filtragem dos *tweets*

Palavras utlizadas no filtro de pesquisa dos *tweets*
aleijada, aleijado, alejada, assassinar, atacar, atropelada, atropelado, atropelar, bicha, bixa, bosta, burra, burro, cadela, deficiente, desgraçada, desgraçado, enforcada, enforcado, enforcar, escrota, escroto, espancar, fdp, fedida, fedido, feia, feio, gay, gorda, gordo, herege, idiota, judeu, lesbica, lgbt, louca, louco, macaca, macaco, macumbeiro, maluca, maluco, matar, merda, miserável, mongoloide, mongol, negro, nordestina, nordestino, obeso, otária, otário, pelego, piranha, pobre, porra, preto, prostituta, prostituto, puta, retardada, retardado, sapatona, sapatão, suja, sujo, tiro, traveco, travesti, trouxa, troxa, vaca, vadia, vagabunda, vagabundo, verme, viado, arrombada, arrombado, bichona, fudida, fudido, morra, obesa, sapatao, amarelo, arretado, bixona, crente, jegue, mulherzinha, negra, nordestinada, otaria, otario, paraiba, parda, pardo, queimar, índio

Fonte: Autoria própria (2023).

4.2.2 Pré-Processamento

Na etapa de pré-processamento, foram aplicadas técnicas para tratar os textos coletados na etapa anterior e prepará-los para o processo de Aprendizado de Máquina Supervisionado. Essas técnicas visam melhorar a qualidade e eficácia dos dados textuais, removendo ruídos, padronizando formatos e tratando elementos indesejados que possam interferir na análise e no desempenho dos algoritmos de Aprendizado de Máquina.

Uma das técnicas aplicadas foi a remoção de pontuação, acentuação e caracteres especiais. Isso inclui a eliminação de sinais de pontuação, emojis, *hashtags*, menções a usuários, *links* e imagens, que geralmente não contribuem significativamente para a análise do conteúdo textual. Além disso, é comum converter todas as letras para minúsculas, a fim de evitar a duplicação de palavras devido a diferenças de capitalização.

A criação de colunas auxiliares para o andamento do trabalho (label e label2) também foi feita nesta etapa. Essas colunas são úteis para a classificação da base. Na coluna 'label' consta valores 0 e 1 e que foram preenchidos manualmente, onde 1 são para textos preconceituosos e 0 para não preconceituosos. Já na coluna 'label2' consta os valores 0, 1 e 2, onde 0 são para textos neutros, 1 para textos preconceituosos e 2 para ofensivos.

Além disso, a remoção de *tweets* duplicados, *tweets* em outros idiomas que não seja o português e *stopwords* foi aplicada. Como foi dito anteriormente, *stopwords* são palavras muito comuns, como artigos, preposições e pronomes, que geralmente não possuem um significado discriminativo para a análise do texto. Essas palavras são removidas para evitar que sejam tratadas como características relevantes durante o processo de aprendizado.

A utilização de bibliotecas disponíveis no Python desempenhou um papel fundamental na aplicação das técnicas de pré-processamento mencionadas acima. As seguintes bibliotecas foram utilizadas neste trabalho:

- Unicode: utilizada para lidar com a codificação e manipulação correta dos caracteres textuais;
- NLTK (*Natural Language Toolkit*): ofereceu uma variedade de recursos e funcionalidades que foram aplicados no pré-processamento dos textos;
- emoji: foi utilizada especificamente para tratar e remover emojis dos textos coletados;
- re (*Regular Expression*): foi aplicada para a remoção de caracteres especiais e pontuações indesejadas;
- pandas: foi utilizada para a manipulação e organização dos dados durante o pré-processamento.

Essas bibliotecas ofereceram um conjunto de ferramentas eficientes para a aplicação das técnicas de pré-processamento nos textos coletados. Elas contribuíram para a automação e agilidade do processo, garantindo a qualidade dos dados preparados para a etapa seguinte.

De forma geral, a etapa de pré-processamento desempenha um papel crucial no preparo dos dados para o processo de aprendizado. Ela busca tornar os textos mais estruturados, limpos e representativos, a fim de melhorar a eficácia dos algoritmos de aprendizado de máquina e obter resultados mais precisos e relevantes na análise dos textos coletados.

As *stopwords* removidas estão contidas no Quadro 4. Referem-se às palavras disponíveis na biblioteca nltk.corpus do Python.

Quadro 4 – Lista de *stopwords* removidas dos textos que compõem a base de dados

Lista de *stopwords*
a, ao, aos, aquela, aquelas, aquele, aqueles, aquilo, as, até, com, como, da, das, de, dela, delas, dele, deles, depois, do, dos, e, ela, elas, ele, eles, em, entre, era, eram, essa, essas, esse, esses, esta, estamos, estar, estas, estava, estavam, este, esteja, estejam, estejamos, estes, esteve, estive, estivemos, estiver, estivera, estiveram, estiverem, estivermos, estivesse, estivessem, estivéramos, estivéssemos, estou, está, estávamos, estão, eu, foi, fomos, for, fora, foram, forem, formos, fosse, fossem, fui, fôramos, fôssemos, haja, hajam, hajamos, havemos, haver, hei, houve, houvemos, houver, houvera, houveram, houverei, houverem, houveremos, houveria, houveriam, houvermos, houverá, houverão, houveríamos, houvesse, houvessem, houvéramos, houvéssemos, há, hão, isso, isto, já, lhe, lhes, mais, mas, me, mesmo, meu, meus, minha, minhas, muito, na, nas, nem, no, nos, nossa, nossas, nosso, nossos, num, numa, não, nós, o, os, ou, para, pela, pelas, pelo, pelos, por, qual, quando, que, quem, se, seja, sejam, sejamos, sem, ser, serei, seremos, seria, seriam, será, serão, seríamos, seu, seus, somos, sou, sua, suas, são, só, também, te, tem, temos, tenha, tenham, tenhamos, tenho, terei, teremos, teria, teriam, terá, terão, teríamos, teu, teus, teve, tinha, tinham, tive, tivemos, tiver, tivera, tiveram, tiverem, tivermos, tivesse, tivessem, tivéramos, tivéssemos, tu, tua, tuas, tém, tínhamos, um, uma, você, vocês, vos, à, às, é, éramos

Fonte: Biblioteca nltk.corpus do Python.

Após esse tratamento, a classificação manual dos *tweets* e da distribuição das bases de treino e teste também foram executadas e definidas. Seis cenários foram criados para fim de comparação da acurácia e precisão do algoritmo criado, conforme apresentado no Quadro 5.

Quadro 5 – Particionamento da base de dados em dados para treinamento e teste, considerando diferentes cenários

	Variáveis	**Treino**	**Teste**	**Total**
Cenário 1	Com preconceito/sem preconceito	(60%)	(40%)	(100%)
Cenário 2	Com preconceito/sem preconceito	(80%)	(20%)	(100%)
Cenário 3	Com preconceito/sem preconceito	(90%)	(10%)	(100%)
Cenário 4	Com preconceito/ofensivo/neutro	(60%)	(40%)	(100%)
Cenário 5	Com preconceito/ofensivo/neutro	(80%)	(20%)	(100%)
Cenário 6	Com preconceito/ofensivo/neutro	(90%)	(10%)	(100%)

Fonte: Autoria própria (2023).

No Cenário 1, o algoritmo de classificação foi treinado para identificar apenas frases com ou sem preconceito. Sendo que do total de *tweets* coletados, foram utilizados 60% para treinamento, e 40% para teste. No Cenário 2, foram utilizados os mesmos rótulos do Cenário 1,

porém foi feita a separação dos dados utilizando 80% dos dados para treinamento, e 20% para teste. Já no Cenário 3, ainda com os mesmos rótulos, a separação foi feita em 90% para treino e 10% para teste.

No Cenário 4 o algoritmo de classificação foi treinado para identificar frases com preconceito, sem preconceito, e frases consideradas ofensivas. Neste caso, do total de *tweets* coletados, foram utilizados 60% para treinamento, e 40% para teste. No Cenário 5, foram utilizados os mesmos rótulos do Cenário 4, porém foi feita a separação dos dados utilizando 80% dos dados para treinamento, e 20% para teste. Por fim, no Cenário 6, utilizando os três rótulos, a separação foi feita em 90% para treino e 10% para teste.

Esses cenários foram pensados e modificados durante o desenvolvimento do trabalho a fim de se adequar às necessidades observadas. O principal motivo de criar cenários com três classes (cenários 4, 5 e 6) foi a grande quantidade de *tweets* coletados que possuíam somente conteúdo ofensivo, sem ser de cunho preconceituoso, abrindo caminho para uma divisão de classes mais equilibrada.

4.2.3 Processamento e Análise

O modelo SVM apresentou bom desempenho na maioria das referências citadas neste trabalho, e por essa razão, esse foi o modelo escolhido para realizar a classificação dos textos.

Nesta etapa a base totalmente rotulada foi submetida ao algoritmo de classificação SVM utilizando a biblioteca sklearn do Python. As funções disponibilizadas nessa biblioteca permitem efetuar todo o processo necessário para treinar, classificar, dividir a base e avaliar modelos de Aprendizado de Máquina Supervisionado.

O algoritmo foi desenvolvido para se adequar aos seis cenários definidos e apresentados anteriormente (Quadro 5). As funções do sklearn permitiram utilizar parâmetros para definir as porcentagens das partições de treino e teste da base, vetorizar os textos utilizando a técnica TF-IDF, procedimento comumente utilizado para verificar a importância relativa de um termo específico, e apresentar relatórios de avaliação do modelo, com informações úteis para a análise, como precisão, acurácia, *recall* e f1-score, descritos em Das (2016).

O valor de precisão que o algoritmo retorna mede a exatidão das previsões positivas do modelo. Uma pontuação de precisão alta indica que o modelo tem uma taxa baixa de falsos positivos.

A acurácia mede a capacidade geral do modelo de fazer previsões corretas em todas as classes.

O *recall* mede a capacidade do modelo de encontrar todos os casos positivos. Uma pontuação alta deste índice indica que o modelo tem uma taxa baixa de falsos negativos.

E por fim, o f1-score é a métrica que combina a precisão e o *recall* em um único valor, fornecendo uma medida de desempenho equilibrada. Este valor é útil quando o objetivo é obter

uma avaliação geral do modelo que leve em consideração tanto os falsos positivos quanto os falsos negativos.

Todas as informações desse relatório de avaliação são úteis para identificar a eficácia do modelo.

5 RESULTADOS

5.1 Coleta de Textos

Uma aplicação em Python utilizando a ferramenta Notebook Jupyter foi desenvolvida com o objetivo de realizar a coleta de dados do *Twitter*. Para isso, foi utilizada a API disponibilizada pela plataforma, juntamente com as funções necessárias para efetuar as requisições, aplicando os parâmetros previamente definidos.

O acesso à API do *Twitter* foi configurado na aplicação, permitindo a autenticação e obtenção de permissões para coletar os tweets desejados. Foram utilizadas as bibliotecas e recursos adequados para realizar as chamadas de API, utilizando as palavras-chave e critérios estabelecidos durante a etapa de planejamento da coleta.

Ao executar a aplicação, foram efetuadas as requisições à API do *Twitter*, passando os parâmetros definidos. O período de coleta, como informado anteriormente, foi limitado a *tweets* publicados há pelo menos oito horas antes do momento da coleta e no máximo há sete dias.

Durante o processo de coleta, realizado no período de 11 de abril a 15 de maio de 2023, foram obtidos 20.000 *tweets*. Essa quantidade de dados representa um conjunto significativo para a análise subsequente. Após a realização da etapa de pré-processamento, essa base ficou menor.

A estrutura principal do código desenvolvido para realização desta etapa da mineração de texto pode ser observada em Listagem 1, onde são apresentados os métodos que fazem a requisição na API do *Twitter*, utilizando os parâmetros já definidos, e as funções que armazenam os resultados em um arquivo CSV (*Comma-separated values*), que é um formato de arquivo de texto estruturado como tabela e separado por vírgulas.

Listagem 1 – Coleta de textos via *requests*

```
# Parametros
query_title = "palavra-chave"
bearer_token = ''
url = "https://api.twitter.com/2/tweets/search/recent"
date_today = datetime.now().strftime("%Y-%m-%d-%H-%M-%S")
now = datetime.now()
date_today_end = now - timedelta(minutes=480)
date_today_end = date_today_end.strftime("%Y-%m-%dT%H:%M:%SZ")

params = {
  "query": {query_title},
  "max_results": 100,
  "end_time": {date_today_end},
  "tweet.fields": "lang,public_metrics,created_at,source,
                   context_annotations,referenced_tweets"
}

# Requisicaoo na API Twitter v2
response = requests.request("GET", url, params=params, auth=bearer_oauth)
print(response.status_code)
if response.status_code != 200:
  raise Exception(
    "Request returned an error: {} {}".format(
      response.status_code, response.text
    )
  )

json_response = response.json()

# Criando um DataFrame do pandas
df = pd.json_normalize(json_response)
df = pd.DataFrame(df)

# Escrevendo no arquivo CSV
diretorio_atual = os.getcwd()
df.to_csv(f'{diretorio_atual}\csv\{query_title}_{date_today}.csv',
    sep=';', encoding='utf-8', index=False)
```

Fonte: Autoria própria (2023).

Para cada palavra-chave, uma requisição teve que ser executada, coletando no máximo 100 *tweets* por vez, limitação da empresa para o plano contratado, ou seja, para a coleta dos 20.000 *tweets*, este código foi executado 200 vezes.

O total de palavras-chave únicas utilizadas na coleta foi de 102 palavras. Para manter um valor próximo da quantidade de textos de cada plavavra-chave, o código foi executado mais de uma vez para algumas delas.

Cada requisição gerou um arquivo CSV único e seu nome foi padronizado. A primeira parte do nome é a própria palavra-chave e a segunda parte é a data e hora em que a requisição

foi feita. O resultado final ficou no formato de: *palavra_ano-mês-dia-hora-minuto-segundo.csv*. Dessa forma bastou concatenar todos esses arquivos gerados para formar um único arquivo que futuramente se tornou a base de dados.

Exemplos de textos ofensivos e preconceituosos coletados nesta fase são apresentados no Quadro 6.

Quadro 6 – Textos coletados

Palavra-chave	**Texto**
"vagabunda"	"essa vagabunda vai ver o que é desrespeito ja ja"
"viado"	"Gente ele é viado mesmoooooo"
"macaca"	"Cala boca negrinha, acho que oculos pra macaca nao fizeram ainda"
"macaca"	"me ama menos macaca preta"
"mongoloide"	"Mongoloide. Vc não é artista não doente"
"mongoloide"	"Esse é um acéfalo, mongolóide"
"fudida"	"Que velha porca, essa mão toda fudida sem usar luvas pqp nunca comeria"
"aleijado"	"Sou aleijado?"

Fonte: Autoria própria (2023).

5.2 Pré-processamento

Nesta etapa uma outra aplicação em Python foi desenvolvida com a finalidade de aplicar os filtros e tratamentos citados na seção 4.2.2, referente ao pré-processamento executado neste trabalho.

Exemplos de textos antes e após a aplicação dos tratamentos na base de dados podem ser visualizados no Quadro 7 e Quadro 8.

Quadro 7 – Tratamento de normalização

Setença antes da normalização	**Sentença após normalização**
"@xxx quer ficar aleijada p sempre né"	"quer ficar aleijada p sempre ne"
"@xxx olha outra aleijada 🙂 "	"olha outra aleijada"
"@xxx Caralho, viado. Resumiu fodamente"	"caralho viado resumiu fodamente"

Fonte: Autoria própria (2023).

Quadro 8 – Tratamento de remoção de stopwords

Setença antes da remoção	Sentença após a remoção
"seus dias estao contados vagabunda "	"dias estao contados vagabunda "
"tem razao pegador de traveco"	"razao pegador traveco"
"virou chat essa porra o traveco da avenida brasil"	"virou chat porra traveco avenida brasil"

Fonte: Autoria própria (2023).

É possível observar que na normalização (Quadro 7) todas as menções, que são os caracteres "@" seguidos de "xxx", foram removidas, além também de toda a acentuação, pontuação e *emojis*. Os nomes dos perfis foram substituídos por "xxx" para preservar a identidade do usuários. Já no Quadro 8 é notada a remoção das palavras presentes na lista de *stopwords* (Quadro 4), como artigos, pronomes e preposições.

A quantidade final de *tweets* da base após esse tratamento ficou em 3.043 e todos foram rotulados manualmente, das duas formas já citadas, em suas respectivas colunas, 'label' e 'label2'.

A escolha de criar um segundo modo para classificação dos textos surgiu após análise dos tweets que compõem a base de dados final, e o autor deste trabalho perceber a existência de uma grande quantidade de textos ofensivos que não continham preconceito em seu conteúdo. Dessa forma, foram criados diferentes cenários para testes, considerando em alguns deles esses textos considerados ofensivos, e no estágio de processamento, foi possível efetuar comparações para verificar a acurácia do classificador em identificar cada um deles.

A classificação manual realizada nos *tweets* selecionados, foi executada seguindo os princípios de NASCIMENTO (2019). No Quadro 9 é exemplificado como foram classificadas algumas sentenças, considerando apenas a possibilidade de estarem relacionadas ou não a preconceito.

Quadro 9 – Primeira classificação das sentenças que compõem a base de dados textual.

Classe	Exemplo de Sentença
0 - Sem preconceito	"arrasou viado"
0 - Sem preconceito	"calada vagabunda"
1 - Com preconceito	"e tu uma macaca negrinha"

Fonte: Autoria própria (2023).

No Quadro 10 é exemplificado como foram classificadas algumas sentenças considerando a possibilidade de estarem vinculadas a preconceito, texto ofensivo ou neutro.

Quadro 10 – Segunda clasifficação das sentenças que compõem a base de dados textual.

Classe	Exemplo de Sentença
0 - Neutro	"arrasou viado"
1 - Com preconceito	"e tu uma macaca negrinha"
2 - Ofensivo	"calada vagabunda"

Fonte: Autoria própria (2023).

Vale ressaltar que, por não ter conhecimento do contexto, este não foi levado em consideração na análise. Particularidades da língua como gírias, erros ortográficos e demais mudanças fora do comum também foram deixadas de lado. Situações como alguém dizer "odeio viado", porém se referindo ao animal cervo como "viado", podem acontecer. Um pessoa falando para o próprio amigo que ele é um "otário", pode ter um tom de brincadeira entre amigos. Todos os textos que compõe a base de dados criada, foram classificados atribuindo o sentido exato, literal da frase, sem levar em consideração situações como essas citadas.

5.3 Processamento

A base de dados, já formada e rotulada adequadamente, foi submetida ao algoritmo de classificação SVM, como apresentado na autorefcodigo:svm1, utilizando as funções da biblioteca sklearn do Python.

Listagem 2 – Algoritmo utilizando SVM da biblioteca sklearn

```
1
2  # Dividir os dados em recursos (X) e rótulos (y)
3  X = df['text']
4  y = df['label']
5
6  # Pré-processamento dos textos usando TF-IDF
7  vectorizer = TfidfVectorizer()
8  X_tfidf = vectorizer.fit_transform(X)
9
10 # Dividir os dados em conjunto de treinamento e conjunto de teste
11 X_train,X_test,y_train y_test = train_test_split(X_tfidf,y,test_size=0.2)
12
13 # Treinamento do modelo SVM
14 svm = SVC()
15 svm.fit(X_train, y_train)
16
17 # Classificação dos textos de teste
18 y_pred = svm.predict(X_test)
19
20 # Avaliação do modelo e armazenamento dos resultados para cada classe
21 report = classification_report(y_test, y_pred, output_dict=True)
```

Fonte: Autoria própria (2023).

Os experimentos foram aplicados nos seis cenários citados no Quadro 5 e em cada um deles uma bateria de testes foi executada para obter uma média das métricas.

O algoritmo foi aplicado 100 vezes em cada cenário sem definir um "random state", um parâmetro da função de treino do sklearn que controla a aleatoriedade na divisão dos dados em conjuntos de treinamento e testes, ou seja, sem defini-lo, a cada execução do código a divisão será feita de uma forma diferente.

Os resultados obtidos com a classificação realizada nos cenários 1, 2 e 3 são apresentados na Tabela 1, Tabela 2 e Tabela 3, respectivamente, informando o menor, o maior e a média de todas as métricas obtidas.

Tabela 1 – Resultados Cenário 1 - 60% treinamento e 40 % teste

Métrica	Mínima	Máxima	Média
Acurácia	0.7989	0.8498	0.8220
Classe 0 - Sem preconceito			
Precisão	0.7872	0.8423	0.8156
Recall	0.9692	0.9922	0.9827
f1-score	0.8756	0.9076	0.8913
Support	874	930	904.91
Classe 1 - Com preconceito			
Precisão	0.8014	0.9369	0.8784
Recall	0.3103	0.4510	0.3581
f1-score	0.4584	0.6013	0.5079
Support	288	344	313.09

Fonte: Autoria própria (2023).

Tabela 2 – Resultados Cenário 2 - 80% treinamento e 20 % teste

Métrica	Mínima	Máxima	Média
Acurácia	0.7997	0.8736	0.8336
Classe 0 - Sem preconceito			
Precisão	0.7893	0.8706	0.8261
Recall	0.9618	0.9956	0.9827
f1-score	0.8744	0.9244	0.8975
Support	430	478	452.18
Classe 1 - Com preconceito			
Precisão	0.7632	0.9672	0.8910
Recall	0.3235	0.4966	0.4043
f1-score	0.4741	0.6218	0.5552
Support	131	179	156.82

Fonte: Autoria própria (2023).

Tabela 3 – Resultados Cenário 3 - 90% treinamento e 10 % teste

Métrica	Mínima	Máxima	Média
Acurácia	0.7869	0.8754	0.8377
Classe 0 - Sem preconceito			
Precisão	0.7695	0.8782	0.8351
Recall	0.9585	1.0000	0.9810
f1-score	0.8571	0.9310	0.9020
Support	202	243	226.64
Classe 1 - Com preconceito			
Precisão	0.7750	1.0000	0.8897
Recall	0.3088	0.5714	0.4402

(continua)

Tabela 3 – Resultados Cenário 3 - 90% treinamento e 10 % teste
(continuação)

Métrica	Mínima	Máxima	Média
f1-score	0.4490	0.7218	0.5870
Support	32	100	78.36

Fonte: Autoria própria (2023).

Nesses três primeiros cenários as métricas para a classe 0 (sem preconceito) atingiram resultados satisfatórios, visto que não foram aplicados métodos de otimização. Mesmo no cenário com a menor porcentagem na base de treino (60%) o algoritmo conseguiu encontrar cerca de 98% dos casos positivos, tendo uma baixa incidência de falsos negativos, apontada pela métrica "recall". Por outro lado, a classe 1 (com preconceito) não obteve números de "recall" tão bons quanto a classe 0 alcançou, demonstrando uma grande quantidade de falsos negativos presentes na análise. Um melhora desse índice é vista conforme o aumento da base de treino acontece, mostrando que a quantidade de textos na base de treino pode ser de grande significância para encontrar corretamente *tweets* preconceituosos.

Tanto a classe 0 quanto a classe 1 obtiveram altos índices de precisão, demonstrando baixos níveis de falsos positivos, principalmente nos casos de textos preconceituosos. A precisão neste trabalho, como já dito anteriormente, tem um peso significante, e esses resultados mostraram que o algoritmo chegou a ter até 100% de acerto na classe 1 (Tabela 3), ou seja, em algum momento, todas as vezes que ele classificou um texto como preconceituoso, ele acertou.

Na Tabela 4, Tabela 5 e Tabela 6 são apresentados os resultados obtidos com a classificação realizada nos textos correspondentes aos cenários 4, 5 e 6.

Tabela 4 – Resultados Cenário 4 - 60 % treinamento e 40% teste

Métrica	Mínima	Máxima	Média
Acurácia	0.6544	0.7356	0.6918
Classe 0 - Neutro			
Precisão	0.5776	0.6835	0.6240
Recall	0.8652	0.9400	0.9100
f1-score	0.7081	0.7838	0.7400
Support	518	591	553.30
Classe 1 - Com preconceito			
Precisão	0.7709	0.9091	0.8533
Recall	0.3858	0.5314	0.4580
f1-score	0.5328	0.6475	0.5952
Support	275	354	313.91
Classe 2 - Ofensivo			
Precisão	0.7355	0.8867	0.8082
Recall	0.4643	0.6548	0.5584
f1-score	0.5961	0.7130	0.6595

(continua)

Tabela 4 – Resultados Cenário 4 - 60 % treinamento e 40% teste
(continuação)

Métrica	Mínima	Máxima	Média
Support	321	380	350.79

Fonte: Autoria própria (2023).

Tabela 5 – Resultados Cenário 5 - 80 % treinamento e 20% teste

Métrica	Mínima	Máxima	Média
Acurácia	0.6585	0.7553	0.7056
Classe 0 - Neutro			
Precisão	0.5817	0.6990	0.6374
Recall	0.8550	0.9565	0.8996
f1-score	0.7012	0.7909	0.7458
Support	243	300	274.21
Classe 1 - Com preconceito			
Precisão	0.7525	0.9625	0.8529
Recall	0.3926	0.5850	0.4948
f1-score	0.5289	0.7049	0.6252
Support	139	181	159.77
Classe 2 - Ofensivo			
Precisão	0.7222	0.9115	0.8054
Recall	0.4684	0.7012	0.5956
f1-score	0.5914	0.7573	0.6837
Support	150	207	175.02

Fonte: Autoria própria (2023).

Tabela 6 – Resultados Cenário 6 - 90 % treinamento e 10% teste

Métrica	Mínima	Máxima	Média
Acurácia	0.6590	0.7803	0.7129
Classe 0 - Neutro			
Precisão	0.5515	0.7486	0.6456
Recall	0.8382	0.9580	0.8987
f1-score	0.6837	0.8111	0.7508
Support	119	158	137.56
Classe 1 - Com preconceito			
Precisão	0.7308	1.000	0.8487
Recall	0.4000	0.6571	0.5051
f1-score	0.5532	0.7931	0.6316
Support	61	95	77.99
Classe 2 - Ofensivo			
Precisão	0.6849	0.9231	0.8124

(continua)

Tabela 6 – Resultados Cenário 6 - 90 % treinamento e 10% teste (continuação)

Métrica	Mínima	Máxima	Média
Recall	0.4659	0.7556	0.6091
f1-score	0.5985	0.7895	0.6948
Support	66	107	89.45

Fonte: Autoria própria (2023).

Nesses três últimos cenários a classificação de três rótulos foi utilizada. A classe 0 obteve resultadores piores. Por outro lado, houve um sutil avanço nos resultados da classe 1, onde a precisão se manteve na média anterior mas os números de "recall" subiram, indicando uma melhora para encontrar textos preconceituosos e uma menor incidência de falsos negativos.

Os textos rotulados como "com preconceito", foram rotulados como 1 em todos os cenários, já os textos rotulados como 0 (sem preconceito) na classificação binária (0 e 1), na segunda forma de classificação ou se mantiveram como 0 (desta vez, neutro), ou foram rotulados como 2 (ofensivo). Apesar de um pequeno progresso nas métricas de "recall" da classe 1, esta classificação com três classes teve um desempenho pior no conexto geral do que a classificação binária.

6 CONCLUSÃO

Neste trabalho acadêmico foram empregados métodos de Aprendizado de Máquina Supervisionado para identificar preconceitos em textos de redes sociais. O objetivo principal foi desenvolver um modelo capaz de analisar o conteúdo textual de uma base de dados e detectar automaticamente a presença de discursos preconceituosos, contribuindo para a promoção de um ambiente *online* mais inclusivo e respeitoso.

Ao longo do estudo, foi realizado um processo abrangente de pré-processamento de texto, que incluiu etapas como normalização e remoção de *stop words*. Em seguida, foi empregado o algoritmo de classificação SVM, que baseia-se em Aprendizado de Máquina Supervisionado, para treinar e avaliar os modelos.

Os resultados obtidos demonstraram que a abordagem de Aprendizado de Máquina Supervisionado pode ser eficaz na identificação de preconceitos em textos de redes sociais. Os modelos desenvolvidos apresentaram desempenho satisfatório, com pontuações de precisão e "recall" que indicam a capacidade de detectar de forma precisa e abrangente os discursos preconceituosos. Desta forma, este trabalho pode contribuir para evitar o viés de algoritmos de IA que fazem uso de bases de dados textuais para treinamento de modelos inteligentes.

Apesar do algoritmo nem sempre ter encontrado todos os casos com preconceito, a principal preocupação que buscava-se minimizar com este trabalho foi atingida. Há baixa incidência de falsos positivos. O fato de ser um trabalho utilizando Aprendizado de Máquina Supervisionado torna o fator humano bastante impactante, pois a base de dados corresponde a um fator decisivo para bons resultados. Este trabalho mostrou que com uma base pequena rotulada corretamente e poucas, porém importantes, técnicas e métodos, bons resultados podem ser obtidos.

É importante ressaltar que a detecção de preconceitos em textos de redes sociais é uma tarefa desafiadora devido à natureza complexa da linguagem utilizada nessas plataformas. Além disso, a compreensão de sarcasmo, ironia e contextos sutis pode representar um desafio adicional, além de impactar negativamente em modelos que não tratam essas situações, como o caso do modelo empregado neste trabalho. Portanto, é fundamental continuar aprimorando os modelos e explorar abordagens mais avançadas, como o uso de redes neurais e técnicas de processamento de linguagem natural mais sofisticadas. A identificação de categorias de preconceito também podem ser incluídas posteriormente, distinguindo preconceito contra mulheres, preconceito racial, social, religioso e outras classes.

Por fim, os resultados deste trabalho demonstram a viabilidade e a importância de utilizar técnicas de Aprendizado de Máquina para identificar preconceitos em textos de redes sociais. A continuidade dessa pesquisa e o desenvolvimento de soluções práticas podem ter um impacto significativo na promoção de uma cultura *online* mais inclusiva e na redução do preconceito e da discriminação nas plataformas digitais, promovendo a igualdade e o respeito aos direitos humanos.

O conjunto de dados gerado e analisado neste trabalho está disponível para acesso livre e gratuito no link: https://www.kaggle.com/datasets/leonardosilvamoreno/portuguese-twitter-dataset-prejudice-analysis.

REFERÊNCIAS

AHLGREN, M. **40+ Twitter Statistics Facts For 2023**. 2023. Disponível em: https://www.websitehostingrating.com/twitter-statistics/. Acesso em: 25 may 2023.

ARANHA, C. N. **Uma Abordagem de PréProcessamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional**. 2007. Disponível em: https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=10081@1. Acesso em: 03 nov. 2022.

ARIGO, D. *et al.* Using social media for health research: Methodological and ethical considerations for recruitment and intervention delivery. **Digital health**, SAGE Publications Sage UK: London, England, v. 4, p. 2055207618771757, 2018.

BONIN, I. T.; KIRCHOF, E. R.; RIPOLL, D. Disputas pela representação do corpo indígena no twitter. **Revista Brasileira de Estudos da Presença**, SciELO Brasil, v. 8, p. 219–247, 2018.

CHAPMAN, P. *et al.* **CRIPS-DM 1.0 Step-by-step data mining guide**. 2000. Disponível em: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf. Acesso em: 09 nov. 2022.

CORTES, C.; VAPNIK, V. **Support-Vector Networks**. 1995. Disponível em: https://link.springer.com/content/pdf/10.1007/BF00994018.pdf. Acesso em: 08 nov. 2022.

CORTES, O. A. C.; MELO, W. E. d. O. **Utilizando Análise de Sentimentos e SVM na Classificação de Tweets Depressivos**. 2021. Disponível em: https://periodicos.univali.br/index.php/acotb/article/view/17388. Acesso em: 09 nov. 2022.

DANGEROUS, S. P. **WHAT WE DO**. 2022. Disponível em: https://dangerousspeech.org/what-we-do/. Acesso em: 11 nov. 2022.

DAS, S. Data science using oracle data miner and oracle r enterprise. **New York: Apress Media**, Springer, 2016.

FILHO, J. A. C. **MINERAÇÃO DE TEXTOS: ANÁLISE DE SENTIMENTO UTILIZANDO TWEETS REFERENTES À COPA DO MUNDO 2014**. 2014. Disponível em: https://www.repositoriobib.ufc.br/000017/0000179f.pdf. Acesso em: 03 nov. 2022.

HOTHO, A.; NÜRNBERGER, A.; PAASS, G. **A Brief Survey of Text Mining**. 2005. Disponível em: https://www.researchgate.net/publication/215514577_A_Brief_Survey_of_Text_Mining. Acesso em: 03 nov. 2022.

JAIN, L. C.; KACPRZYK, J. **New learning paradigms in soft computing**. [*S.l.*]: Springer Science & Business Media, 2002. v. 84.

KARAMI, A. *et al.* Twitter and research: A systematic literature review through text mining. **IEEE access**, IEEE, v. 8, p. 67698–67717, 2020.

KHAN, A. *et al.* A review of machine learning algorithms for text-documents classification. **Journal of advances in information technology**, Academy Publisher, PO Box 40 Oulu 90571 Finland, v. 1, n. 1, p. 4–20, 2010.

KOHAVI, R. *et al.* A study of cross-validation and bootstrap for accuracy estimation and model selection. *In*: MONTREAL, CANADA. **Ijcai**. [*S.l.*], 1995. v. 14, n. 2, p. 1137–1145.

LEITE, J. L. A. **MINERAÇÃO DE TEXTOS DO TWITTER UTILIZANDO TÉCNICAS DE CLASSIFICAÇÃO**. 2015. Disponível em: https://repositorio.ufc.br/handle/riufc/25016. Acesso em: 09 nov. 2022.

LORENA, A. C.; CARVAHO, A. C. P. L. F. **Introdução às Máquinas de Vetores Suporte (Support Vector Machines)**. 2003. Disponível em: https://repositorio.usp.br/item/001305413. Acesso em: 09 nov. 2022.

MITTELSTADT, B. D. *et al.* **The ethics of algorithms: mapping the debate**. 2016. Disponível em: https://journals.sagepub.com/doi/full/10.1177/2053951716679679. Acesso em: 29 ago. 2022.

NASCIMENTO, R. M. F. **Classificação Automática de Discursos de Ódio em Textos do Twitter**. 2019. Disponível em: https://repository.ufrpe.br/bitstream/123456789/2439/1/tcc_robsonmuriloferreiradonascimento.pdf. Acesso em: 09 nov. 2022.

PARK, H. *et al.* A framework for energy optimization of distillation process using machine learning-based predictive model. **Energy Science & Engineering**, Wiley Online Library, v. 10, n. 6, p. 1913–1924, 2022.

REBALA, G.; RAVI, A.; CHURIWALA, S. **An introduction to machine learning**. [*S.l.*]: Springer, 2019.

ROSSETTI, R.; ANGELUCI, A. **Ética Algorítmica: questões e desafios éticos do avanço tecnológico da sociedade da informação**. 2021. Disponível em: https://www.scielo.br/j/gal/a/R9F45HyqFZMpQp9BGTfZnyr/?lang=pt&format=pdf. Acesso em: 29 ago. 2022.

SOUZA, M.; VIEIRA, R. **Sentiment Analysis on Twitter Data for Portuguese Language**. 2012. Disponível em: https://www.inf.pucrs.br/linatural/wordpress/wp-content/uploads/2017/08/PROPOR_2012.pdf. Acesso em: 02 nov. 2022.

TECMUNDO. **Twitter se desculpa após denúncias de viés racista de algoritmo**. 2020. Disponível em: https://www.tecmundo.com.br/redes-sociais/198362-twitter-desculpa-denuncias-vies-racista-algoritmo.htm. Acesso em: 11 nov. 2022.

VIEIRA, L. M.; SILVA, N. R. d.; CORDEIRO, D. F. Análise descritiva das fake news da saúde através de mineração de textos no portal da saúde. *In*: **Congresso de Ciências da Comunicação na Região Centro-Oeste**. [*S.l.*: *s.n.*], 2019. p. 1–4.