Universidade Federal de São Carlos– UFSCar
Centro de Ciências Exatas e de Tecnologia– CCET
Departamento de Computação– DC
Programa de Pós-Graduação em Ciência da Computação– PPGCC

Thiago Nacrur Maricondi

Análise Comparativa de Métodos de Detecção Automática de Mensagens Ofensivas em Textos Curtos e Ruidosos

São Carlos
2025

Thiago Nacrur Maricondi

Análise Comparativa de Métodos de Detecção Automática de Mensagens Ofensivas em Textos Curtos e Ruidosos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Metodologias e Técnicas de Computação

Orientador: Prof. Dr. Auri Marcelo Rizzo Vincenzi
Coorientador: Prof. Dr. Ricardo Rodrigues Ciferri

São Carlos
2025

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Thiago Nacrur Maricondi, realizada em 06/05/2025.

Comissão Julgadora:

Prof. Dr. Ricardo Rodrigues Ciferri (UFSCar)

Prof. Dr. Tiago Agostinho de Almeida (UFSCar)

Prof. Dr. Renato de Freitas Bulcão Neto (UFG)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Ciência da Computação.

Dedico este trabalho ao meu filho,
que me inspira a sempre persistir e perseverar perante os desafios.

Agradecimentos

A Deus, em primeiro lugar, por me conceder força e esperança mesmo nos momentos mais difíceis.

Aos meus pais, Fernando e Patrícia, por todo o esforço e sacrifício para que seus filhos tivessem a melhor educação possível.

Ao meu filho Heitor, que, com seu inocente sorriso, me motiva todos os dias a seguir em frente e nunca desistir.

Ao meu orientador, Prof. Dr. Auri Marcelo Rizzo Vincenzi, por sua orientação atenciosa, paciência e ensinamentos ao longo desta jornada.

Ao Prof. Dr. Ricardo Rodrigues Ciferri, que me coorientou com dedicação, sempre solícito, compreensivo e disposto a contribuir com sua experiência.

Por fim, a todos os colegas, amigos e familiares que, com palavras gentis e incentivo constante, me deram forças para continuar e concluir este trabalho.

"O ponto cego da Inteligência Artificial é que a consciência não emerge do pensamento; é a fonte dele."

(George Gilder)

Resumo

Com o crescente uso das redes sociais e a facilidade de acesso a conteúdos digitais, especialmente entre crianças e jovens, houve um aumento significativo nos casos de *cyberbullying* e assédio virtual nos últimos anos. Em resposta, diversas ferramentas de moderação de conteúdo foram desenvolvidas, como filtros de comentários, sistemas de denúncia e perfis de usuários dedicados à moderação. No entanto, devido à enorme quantidade de informações geradas continuamente nas redes sociais, a moderação manual tornou-se impraticável, o que destaca a importância da moderação automática na redução da incidência de crimes digitais. Este trabalho aborda a identificação automática de comportamentos agressivos em mensagens ofensivas, presentes em textos curtos e ruidosos, por meio de algoritmos de aprendizado de máquina e aprendizado profundo. Utilizou-se um conjunto de dados público extraído da plataforma X, contendo 20.001 sentenças rotuladas como agressivas 39,1% ou não agressivas 60,9%. Modelos de aprendizado supervisionado foram treinados com validação cruzada estratificada, utilizando técnicas de pré-processamento textual e diferentes algoritmos, incluindo BERT, FastText e métodos ensemble, com o objetivo de avaliar a eficácia dessas abordagens na detecção automática de agressividade textual. Os resultados obtidos demonstraram que os modelos BERT e FastText apresentaram excelente desempenho em revocação, alcançando 96,5% e 95,8%, respectivamente, superando significativamente o modelo baseline na detecção de mensagens ofensivas.

Palavras-chave: Aprendizado de Máquina. Cyberbullying. Classificação Binária.

Abstract

With the increasing use of social media and the ease of access to digital content—especially among children and adolescents—there has been a significant rise in cases of cyberbullying and online harassment in recent years. In response, several content moderation tools have been developed, such as comment filters, reporting systems, and user profiles dedicated to moderation. However, due to the vast amount of information constantly generated on social media platforms, manual moderation has become impractical, highlighting the importance of automated moderation in reducing the incidence of digital crimes. This work addresses the automatic identification of aggressive behavior in offensive messages found in short and noisy texts using machine learning and deep learning algorithms. A public dataset extracted from platform X was used, containing 20,001 sentences labeled as aggressive 39.1% or non-aggressive 60.9%. Supervised learning models were trained using stratified cross-validation, employing text preprocessing techniques and various algorithms, including BERT, FastText, and ensemble methods, with the goal of assessing the effectiveness of these approaches in the automatic detection of textual aggressiveness. The results showed that the BERT and FastText models achieved excellent recall scores, reaching 96.5% and 95.8%, respectively, significantly outperforming the baseline model in detecting offensive messages.

Keywords: Machine Learning. Cyberbullying. Binary Classification.

Lista de ilustrações

Lista de tabelas

Lista de siglas

AM Aprendizado de Máquina

BoW *Bag-of-Words*

BERT *Bidirectional Encoder Representations from Transformers*

GPT *Generative Pre-trained Transformer*

IBGE Instituto Brasileiro de Geografia e Estatística

LSTMs *Long Short-Term Memory networks*

NB Naive Bayes

PLN Processamento de Linguagem Natural

RN Redes Neurais

RNAs Redes Neurais Artificiais

RNNs *Recurrent Neural Networks*

RL Regressão Logística

TF-IDF *Term frequency Inverse Document Frequency*

Sumário

Capítulo 1

Introdução

1.1 Considerações Iniciais

Neste capítulo introdutório, são discutidas a contextualização e a motivação desta pesquisa de Mestrado, que se concentra na classificação de mensagens impróprias em redes sociais, com ênfase em comportamentos agressivos. Além disso, são apresentadas as hipóteses e os objetivos deste estudo, concluindo com o detalhamento da estrutura da dissertação.

1.2 Contextualização

Os avanços nas tecnologias de comunicação tornaram possível que qualquer pessoa, com acesso à Internet, possa se comunicar de diversas maneiras, independentemente de sua localização geográfica. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE),[1] em 2021, aproximadamente 85 a cada 100 pessoas com 10 anos ou mais de idade acessaram a Internet no Brasil. Esse dado evidencia o crescimento contínuo do uso da Internet como parte integrante do cotidiano da população brasileira. Dentre esses usuários, crianças e adolescentes representam uma parcela significativa e crescente, impulsionada pelas diversas facilidades proporcionadas pelo ambiente digital, como o aumento da interação social, o acesso a conteúdos educacionais gratuitos e recursos voltados ao lazer e entretenimento. Especificamente entre crianças com idade entre 10 e 13 anos, o percentual de utilização da Internet foi de aproximadamente 82%, demonstrando uma forte presença desse público no espaço virtual.

[1] <https://www.ibge.gov.br/pt/inicio.html>

Entretanto, apesar de todos os benefícios, a comunicação online também apresenta riscos consideráveis, especialmente para crianças e adolescentes. O ambiente virtual pode ser comparado a uma terra sem leis, onde as regras são muitas vezes confusas e a supervisão de adultos nem sempre está presente (WACHS et al., 2021). Com o crescimento da popularização da Internet, problemas como o cyberbullying e o assédio sexual se tornam cada vez mais comuns entre os jovens. Segundo a ONG Plan International Brasil,[2] uma pesquisa com 14 mil meninas revelou que 39% delas relataram ter sofrido assédio sexual no Facebook, enquanto 23% afirmaram que os abusos ocorreram no Instagram.

As redes sociais são amplamente utilizadas em todo o mundo para conectar, comunicar e colaborar de maneiras inovadoras. Essa popularização permitiu que os usuários dessas plataformas gerassem impactos significativos, tanto em outros indivíduos quanto em escala global. Um exemplo notável é a plataforma X (antigo Twitter), uma rede social na qual os usuários podem publicar postagens curtas, conhecidas como tweets, e acompanhar as atividades de outras pessoas ao segui-las. No X, os tweets são limitados a um número restrito de caracteres, o que incentiva a criação de mensagens concisas para expressar opiniões, iniciar discussões ou informar sobre eventos recentes. Essa característica torna a plataforma dinâmica e influente no cenário das redes sociais.

Contudo, apesar do impacto positivo das redes sociais na sociedade, elas também deram origem a comportamentos antissociais, como o *trolling* (BISHOP, 2013). Trolls são responsáveis por desentendimentos online, debates sem propósito e pela disseminação de notícias falsas. Esses indivíduos são conhecidos por criar confusão e discórdia, aproveitando-se do ambiente anônimo da Internet (BUCKELS; TRAPNELL; PAULHUS, 2014). Segundo Bishop (2013), o trolling é definido como a atividade de postar mensagens em redes de comunicação com a intenção de provocar, ofender ou ameaçar. De acordo com Hardaker (2010), a verdadeira intenção de um troll é causar perturbação, desencadear ou intensificar conflitos com o objetivo de obter entretenimento pessoal.

Outro comportamento negativo amplamente presente nas redes sociais é o cyberbullying, caracterizado pelo assédio, humilhação e intimidação de indivíduos em plataformas digitais. Segundo Alqahtani e Ilyas (2023a), essa situação é agravada pelo fato de que muitos usuários dessas plataformas são jovens, especialmente adolescentes.

De acordo com resultados estatísticos apresentados por Hemmatian e Sohrabi (2019), 25% dos usuários da Internet são vítimas de cyberbullying, e 1 em cada 3 adolescentes sofre ameaças no ambiente online. Além disso, cerca de 145 milhões de usuários ativos publicam tweets diariamente no X, abordando uma ampla gama de tópicos, e 36% desses usuários relataram serem vítimas de bullying em 2020 (LUO, 2021). Embora seja desafiador combater o *cyberbullying* nas mídias sociais devido a diversos fatores, soluções inteligentes podem ajudar a mitigar esse problema. Tendo em vista as dificuldades no

[2] <https://abrir.link/wXvDv>

reconhecimento automático de conteúdos de *cyberbullying*, modelos de aprendizado de máquina e aprendizado profundo mostram-se promissores no enfrentamento dessas agressões online.

Embora muitas vezes considerado uma forma menos grave de comportamento antissocial, o trolling está profundamente ligado ao cyberbullying. Ambos compartilham o objetivo de causar danos psicológicos e desconforto no ambiente digital. No entanto, enquanto o cyberbullying tende a ser mais direto e persistente, o trolling frequentemente se disfarça de provocação ou "brincadeira". A suposta anonimidade e impunidade que o ambiente online proporciona cria um terreno fértil para ambos os comportamentos, permitindo que agressores explorem as vulnerabilidades de suas vítimas e perpetuem um ciclo de violência online.

1.3 Motivação

Em virtude da grande popularidade de Redes Sociais como Facebook, Instagram e X, que conectam bilhões de usuários ao redor do mundo, as organizações enfrentam um grande desafio de banco de dados, buscando extrair e analisar informações potencialmente importantes geradas pelos usuários, tendo assim, uma demanda por estratégias analíticas avançadas, como: análise de sentimentos, mineração de dados, aprendizado de máquina, processamento de linguagem natural, etc (YANG et al., 2022).

Parte dos esforços estão envolvidos na moderação de conteúdo, sendo que cada plataforma adota diversas medidas a fim de reduzir os danos causados pelo assédio, *bullying*, palavras ofensivas e insultos raciais. No Instagram,[3] por exemplo, existem algumas abordagens para lidar com o assédio, como ocultar comentários nos quais haja presença de palavras, frases ou emojis ofensivos, sendo apenas visível para o autor que realizou o comentário; também é possível criar um filtro personalizado de comentários, buscando assim ocultar comentários que contenham palavras adicionadas pelo usuário. Tais ferramentas de moderação visam reduzir o conteúdo tóxico gerado por usuários, contudo, os métodos atuais mostram-se pouco eficazes na redução de hostilidades por diversos motivos. Lidar com o assédio e outros comportamentos hostis online é um desafio em parte porque as pessoas, as políticas e as leis discordam sobre o que constitui um comportamento inaceitável (LIU et al., 2018). Além disso, outro problema enfrentado é a quantidade, velocidade e variedade de conteúdo gerado; dificultado, o controle manual por partes dos usuários ou mesmo das plataformas sociais, tornando assim imperativo o uso de ferramentas de moderação automatizadas (GILLESPIE, 2020).

Em contrapartida, os agressores utilizam diversos artifícios para burlar a moderação, tais como gírias, abreviações e emojis, que são amplamente utilizados, especialmente por jovens, e tendem a ser difíceis de serem identificados pelos mecanismos atuais. Segundo

[3] <https://help.instagram.com/700284123459336>

Vasantharajan e Thayasivam (2021), nos últimos anos, com o aumento do número de usuários que falam diferentes idiomas nas redes sociais, as postagens e comentários são publicados em um formato misto, ou seja, com frases que contêm vocabulário e sintaxe de várias línguas. Devido a falta de conhecimento e habilidade de se expressar, muitos internautas transmitem seus pensamentos para a comunidade em geral utilizando fragmentos de outros idiomas juntamente com sua língua nativa. Isso leva a sérios desafios para identificação de fala ofensiva, justamente pela falta de metodologias para tratamento de texto misto.

Nesse cenário, o uso de Aprendizado de Máquina (AM) pode auxiliar na tarefa de classificação automática de comentários agressivos em redes sociais, reduzindo o tempo e os erros associados à análise manual, especialmente em contextos com grande volume de dados. Para isso, modelos de AM podem ser treinados com um conjunto de dados rotulado em duas classes, comentários agressivos e comentários não agressivos, permitindo que o algoritmo aprenda a identificar padrões linguísticos associados à agressividade. Uma vez treinado, o modelo é capaz de prever automaticamente o rótulo de novos comentários, mesmo que não estejam previamente classificados (TOLBA; OUADFEL; MESHOUL, 2021).

1.4 Hipóteses e Objetivos

A presente pesquisa busca explorar uma ampla gama de algoritmos e técnicas para a detecção automática de comentários agressivos em textos curtos e ruidosos, com ênfase em modelos pré-treinados. Serão ainda comparados modelos tradicionais de aprendizado de máquina, modelos baseados em Transformers e ainda técnicas de ensemble que combinam as previsões de diversos modelos. O objetivo é desenvolver uma solução eficaz para a detecção de textos ofensivos em redes sociais, com ênfase na identificação de comentários agressivos que possam comprometer a qualidade das interações online. Com isso, espera-se contribuir para que as mídias sociais se tornem um ambiente mais seguro e respeitoso para seus usuários.

Mais especificamente, esta pesquisa de Mestrado tem como objetivo principal a detecção automática de mensagens ofensivas em textos curtos e ruidosos, por meio da aplicação de modelos de linguagem pré-treinados baseados na arquitetura transformer, como *Bidirectional Encoder Representations from Transformers* (BERT), RoBERTa e DistilBERT. Além disso, algoritmos de aprendizado de máquina, como FastText, e métodos ensemble, como o Bootstrap Aggregating (Bagging), também serão avaliados. Para lidar com o desafio do desbalanceamento de classes, presente no *dataset* desenvolvido por DataTurks (2018), e evitar o *overfitting*, os modelos serão treinados utilizando a abordagem de validação cruzada k-fold, garantindo maior robustez e generalização nos resultados obtidos.

Com base nesse objetivo, este trabalho investiga a seguinte hipótese:

A Inteligência artificial e seus algoritmos potencializam a detecção automática de mensagens ofensivas em textos curtos e ruidosos, combatendo cyberbullying e outras ameaças de forma eficiente.

Em função desta hipótese são analisadas as seguintes questões de pesquisa:

QP1: Algoritmos de classificação binária são eficazes na detecção de mensagens ofensivas em textos curtos e ruidosos?

QP2: Modelos transformers superam algoritmos tradicionais na identificação de tweets ofensivos?

QP3: Técnicas ensemble, como Bagging, aumentam a precisão e robustez em datasets desbalanceados?

QP4: A validação cruzada k-fold contribui para mitigar overfitting e melhorar a capacidade de generalização dos modelos?

QP5: Modelos leves como o DistilBERT podem ser competitivos, equilibrando desempenho e eficiência computacional?

A partir da investigação das questões de pesquisa, espera-se obter as seguintes contribuições:

- ❏ Desenvolver e implementar soluções baseadas em aprendizado de máquina (AM), aprendizado profundo e processamento de linguagem natural (PLN) que explorem diferentes algoritmos para identificar com alta precisão tweets ofensivos. O objetivo é encontrar modelos que apresentem maior acurácia na detecção de comentários agressivos e inapropriados, contribuindo assim para a melhoria das práticas de moderação em redes sociais.

- ❏ Comparação sistemática entre modelos transformers e algoritmos de aprendizado de máquina tradicionais, como o FastText, visando a identificar qual modelo oferece melhor desempenho na detecção de conteúdo ofensivo.

- ❏ Avaliação do impacto de métodos ensemble, como o Bagging, na melhoria da precisão e robustez dos modelos, especialmente em datasets desbalanceados, como o CyberTroll, proporcionando soluções mais confiáveis e eficazes para lidar com esse tipo de problema.

- ❏ Verificação da eficácia da validação cruzada k-fold na redução do *overfitting* e na melhoria da capacidade de generalização dos modelos, aumentando a confiabilidade e a consistência dos resultados em diferentes cenários de dados.

- ❏ Investigar se modelos mais leves como o DistilBERT, podem alcançar um desempenho competitivo em relação a modelos mais robustos, oferecendo um equilíbrio vantajoso entre eficiência computacional e precisão, especialmente em cenários com recursos computacionais limitados.

1.5 Estrutura da Dissertação

Este capítulo abordou o contexto em que este trabalho se insere, a motivação para a definição do tema, assim como os objetivos e hipóteses estabelecidos. Os quatro capítulos restantes estão organizados da seguinte forma:

O Capítulo 2 apresenta a fundamentação teórica que será utilizada no desenvolvimento deste trabalho, abordando tópicos relacionados ao processamento de linguagem natural, técnicas de pré-processamento, métodos ensembles e, por fim, modelos baseados em transformers. O Capítulo 3 analisa os trabalhos relacionados à classificação binária de sentenças, explorando tanto aprendizado de máquina quanto aprendizado profundo. O Capítulo 4 detalha a metodologia adotada, incluindo o tipo de pesquisa, os métodos empregados, os conjuntos de dados utilizados e as métricas escolhidas, além dos resultados obtidos na detecção de comentários agressivos em redes sociais. Por fim, o Capítulo 5 apresenta a conclusão, discutindo as principais contribuições do estudo, as implicações dos resultados e sugestões para futuras pesquisas.

Capítulo 2

Fundamentação Teórica

2.1 Considerações Iniciais

Neste capítulo, são apresentados os conceitos e fundamentos teóricos essenciais para a compreensão da metodologia de classificação de comentários agressivos. Serão abordados os princípios de Aprendizado de Máquina, com ênfase nos algoritmos de classificação; Aprendizado Profundo, destacando os modelos pré-treinados; Processamento de Linguagem Natural, incluindo as principais técnicas de pré-processamento de textos; métodos Ensemble, com foco nas abordagens de *bagging*, *boosting* e *stacking*; e, por fim, será explorada a arquitetura transformers, que desempenha um papel central neste trabalho.

2.2 Processamento de Linguagem Natural

O termo Processamento de Linguagem Natural (PLN) é normalmente utilizado para descrever a função dos componentes de software ou hardware em um sistema computacional que analisa e sintetiza a linguagem falada ou escrita (JACKSON; MOULINIER, 2002). O propósito é dar aos computadores a capacidade de lidar com textos escritos por humanos, isso abrange identificar o seu contexto, retirar informações e até formular textos em resposta. A denominação "natural" significa distinguir a fala e a escrita humanas de línguas mais formais, tais como notações matemáticas ou lógicas, ou linguagens de programação, tais como Java, Cobol e C++.

Segundo Khurana et al. (2022), na literatura existente, a maior parte do trabalho em PLN é realizado por cientistas da computação, porém profissionais de outras áreas, incluindo linguística, psicologia e filosofia, também estão interessados nesta linha de atuação. O campo do PLN está relacionado com diferentes teorias e técnicas que tratam do

problema da linguagem natural de comunicação com os computadores. Algumas das tarefas principais de PLN são a sumarização automática, resolução de correferências, análise do discurso, tradução automática, segmentação morfológica, reconhecimento de entidade nomeada, reconhecimento óptico de caracteres e etiquetador *Part-Of-Speech*.

Aplicações que fazem uso do PLN são construídas usando uma enorme quantidade de dados. Em outras palavras, pode se dizer que uma grande coleção de dados é chamada de *corpus*. *Corpus* é uma coletânea de textos naturais, escolhidos para caracterizar um estado ou variedade de linguagem, e que pode ser usado como base para pesquisa linguística (SINCLAIR, 1991). No PLN, um *corpus* contém dados de texto e fala que podem ser usados para treinar sistemas de inteligência artificial e AM. Se houver um problema ou objetivo específico a ser alcançado, será preciso uma coleção de dados que suporte, ou pelo menos seja uma representação dos resultados desejados.

2.2.1 Pré-Processamento

O pré-processamento de texto é uma etapa fundamental no processo de classificação, que ocorre no início de todo o procedimento. Ele é responsável por transformar o texto de entrada em um formato mais adequado a fim de permitir que os algoritmos de classificação tenham um melhor desempenho. A limpeza de texto e a representação de dados visam aumentar a eficiência do classificador (KADAM; PANISKAKI, 2020).

Durante o processo de pré-processamento diversas técnicas podem ser utilizadas, como remoção de espaços em branco extras; remoção de *tags* HTML; remoção de atributos XML; substituição ou remoção de URLs; remoção ou substituição de pontuação, números e caracteres especiais; converter todo texto em letras minúsculas; remoção de *stopwords*, etc. Segundo KADAM e PANISKAKI (2020), *Stemming* e *Lemmatization* são as principais técnicas usadas para limpar e preparar a entrada antes de passá-la para os modelos a serem treinados.

A seguir descrevem-se as técnicas de pré-processamento comumente empregadas na literatura. Vale ressaltar que a escolha das técnicas que serão aplicadas e sua ordem dependem tanto do conjunto de dados quanto do contexto da tarefa de classificação.

2.2.1.1 Stemming

O *Stemming* é uma técnica de normalização de texto utilizada no processamento de linguagem natural (PLN) para reduzir palavras aos seus radicais. O processo envolve a remoção de sufixos e, em alguns casos, prefixos, com o objetivo de obter uma forma base da palavra. No entanto, diferentemente da lematização, o *stemming* nem sempre resulta em uma palavra com significado claro ou forma gramaticalmente correta. Isso ocorre porque o *stemming* aplica regras básicas de truncamento, sem levar em consideração o contexto ou as regras linguísticas. Por exemplo, palavras como “cars” e “caring” podem ser reduzidas

ao mesmo radical "car", embora tenham significados diferentes. Apesar dessas limitações, o *stemming* é amplamente utilizado devido à sua eficiência em sistemas de recuperação de informação e motores de busca, nos quais a precisão linguística completa pode não ser necessária.

2.2.1.2 Lematização

A lematização, por outro lado, é um processo mais sofisticado de normalização de texto, que envolve a redução de uma palavra à sua forma canônica ou base, conhecida como *lemma*. Ao contrário do *stemming*, a lematização considera o contexto gramatical e morfológico da palavra para garantir que a forma reduzida mantenha seu significado original. Isso é feito por meio da análise de dicionários e regras linguísticas, garantindo que as palavras sejam convertidas para formas válidas e com sentido. Por exemplo, a palavra "better" seria lematizada para "good", uma vez que leva em conta o contexto de comparativos. Embora a lematização seja computacionalmente mais cara do que o *stemming*, ela é preferida em aplicações que requerem alta precisão semântica, como análises linguísticas profundas e sistemas de compreensão de linguagem natural.

Na Tabela 1, são apresentadas as diferenças no uso das técnicas de *Stemming* e *Lemmatization*, considerando as palavras presentes na coluna *"Word"*. Na Coluna *Stemming*, as palavras da Coluna 1 foram reduzidas ao seu radical, retornando, assim, *"chang"*; na Coluna *Lemmatization*, as palavras foram reduzidas ao seus "lemmas", retornando *"change"*.

Tabela 1 – Pré-processamento com *Stemming* e Lematização

Word	***Stemming***	**Lematização**
change	chang	change
changing	chang	change
changes	chang	change
changed	chang	change
changer	chang	change

2.2.1.3 Tokenização

A tokenização é uma etapa fundamental no pré-processamento de dados textuais, que consiste em segmentar o texto em unidades menores, denominadas tokens, que podem ser palavras, frases ou até mesmo caracteres. Essas unidades servem como entrada para modelos de aprendizado de máquina ou processamento de linguagem natural (NLP). A granularidade da tokenização pode variar, dependendo da aplicação e do modelo adotado, podendo enfocar palavras inteiras, subpalavras ou caracteres individuais. O tratamento da pontuação, entretanto, é um aspecto que requer atenção. Conforme apontado por

Manning (1999), a remoção da pontuação interna das frases durante o processo de tokenização pode não ser a melhor abordagem, visto que sinais de pontuação fornecem informações relevantes sobre a estrutura macro do texto e ajudam a indicar relações entre seus elementos.

2.2.1.4 Stopwords

As stopwords são termos que ocorrem com alta frequência em textos, como preposições, artigos e conjunções, mas que, em geral, possuem baixo valor semântico para o entendimento ou classificação de documentos. No contexto do Processamento de Linguagem Natural (PLN), a remoção dessas palavras é uma prática comum na etapa de pré-processamento de dados, com o objetivo de reduzir a dimensionalidade e otimizar o desempenho dos algoritmos. A exclusão das stopwords contribui para uma análise mais eficiente, uma vez que esses termos raramente auxiliam na diferenciação entre classes de texto. Diversas técnicas são utilizadas nesse processo, como o uso de listas predefinidas de stopwords ou a geração dinâmica dessas listas, removendo termos infrequentes que aparecem apenas uma vez no corpus, como apontado por Saif, Fernandez e Alani (2014).

2.2.2 Saco de Palavras

O modelo Sacos de Palavras ou no inglês *Bag-of-Words* (BoW) é uma representação simplificada usada no processamento de linguagem natural, especialmente em problemas de classificação de texto. Nesse modelo, um texto é representado como multiconjunto de suas palavras, desconsiderando a gramática e até mesmo a ordem das palavras, mas mantendo a multiplicidade. O modelo BoW é comumente usado em métodos de classificação de documentos nos quais a frequência de cada palavra é usada como um recurso para treinar um classificador.

Empiricamente, o modelo BoW é usado, principalmente, como ferramenta de geração de características. Depois de transformar o texto em um "saco de palavras", é possível calcular diversas medidas para caracterizar o texto, como por exemplo a frequência do termo, representando, assim, o número de instâncias de uma determinada palavra.

Na Tabela 2 temos a representação de um saco de palavras de 3 frases, exibidas abaixo:

1. O gato sentou;
2. O gato sentou no chapéu;
3. O gato com o chapéu.

Cada frase ou sentença acima pode ser considerada um documento de texto, e o conjunto destes documentos é chamado de corpus. O corpus contém todas as palavras encontradas no conjunto de documentos. Em seguida, é construído um vocabulário de

todas as palavras únicas nos três documentos acima, totalizando 6 palavras: chapéu, com, gato, no, o, sentou. Finalmente, é verificada a quantidade de vezes que uma palavra aparece em cada uma das frases.

Tabela 2 – Exemplo do Modelo Saco de Palavras

Documento	**o**	**gato**	**sentou**	**no**	**chapéu**	**com**
1	1	1	1	0	0	0
2	1	1	1	1	1	0
3	2	1	0	0	1	1

2.2.3 N-grama

Um n-grama é uma sequência contígua de n itens de uma determinada amostra de texto. Seus componentes podem ser letras, sílabas, palavras ou grupos de palavras, dependendo da necessidade da aplicação. O modelo n-grama tenta amenizar o problema da não-captura da ordem das palavras que ocorre no modelo saco de palavras, visando, desse modo, reduzir a perda do contexto. Com prefixos numéricos latinos, um n-grama de tamanho 1 é chamado de unigrama, um de tamanho 2 é chamado de bigrama e um de tamanho 3 é chamado de trigrama. Na lista abaixo, Nasser et al. (2021) apresentam um exemplo de n-gramas de palavra para a sentença 'coronavírus é contagioso':

- ❏ Unigrama: coronavírus, é, contagioso;
- ❏ Bigrama: coronavírus é, é contagioso;
- ❏ Trigrama: coronavírus é contagioso.

2.2.4 TF-IDF

Term frequency Inverse Document Frequency (TF-IDF) é considerado um dos esquemas de ponderação de termo mais usados e relevantes. TF-IDF é uma estatística numérica que informa o quão importante é uma palavra em relação a um documento em um corpus. É utilizado, principalmente, como um fator de ponderação em diversos processos usados para recuperação de informação e também mineração de texto (KUMARI; JAIN; BHATIA, 2016).

O algoritmo TF-IDF é resultado do produto de duas métricas, *Term Frequency* (TF) e *Inverse Document Frequency* (IDF). TF estima quantas vezes um termo ocorre em um determinado documento. Considerando que documentos em um corpus podem possuir tamanhos diferentes, um termo pode se repetir mais frequentemente em documentos mais longos do que em arquivos menores. Desse modo, para normalizar estas contagens, é dividido o número de ocorrências de uma determinada palavra pelo comprimento do

documento. Já o IDF ajuda a determinar a relevância de uma palavra ou termo. Na contagem da frequência de termos (TF), é dada igual importância a todos os termos. Mas certas palavras, como “para”, “que” e “de”, podem aparecer muito frequentemente e não apresentar nenhuma importância. Desse modo, é necessário reduzir os pesos dos termos frequentes e aumentar dos termos incomuns.

2.3 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é o estudo de algoritmos que pegam dados e informações de observações e interações como entrada e generalizam a partir de entradas específicas para exibir traços do pensamento humano. A generalização é um processo pelo qual exemplos específicos são abstraídos para conceitos ou regras de decisão mais abrangentes (VARSHNEY, 2022). O AM refere-se à capacidade de um sistema de adquirir e integrar conhecimento por meio de observações em larga escala, e melhorar e se estender aprendendo novos conhecimentos, em vez de ser programado com esse conhecimento (WOOLF, 2008). Com a constante evolução da tecnologia, o AM tornou-se uma ferramenta cada vez mais poderosa, contribuindo para a melhoria em diversas áreas, incluindo profissionais, pessoais e acadêmicas. Sua aplicação é ampla, abrangendo desde filtros de spam até sistemas de recomendação. Algumas das utilizações mais comuns do aprendizado de máquina incluem:

- ❏ Reconhecimento de Imagem;
- ❏ Reconhecimento de fala;
- ❏ Recomendações de produtos;
- ❏ Carros autônomos;
- ❏ Assistente pessoal virtual;
- ❏ Criação de *chatbots*;
- ❏ Sumarização de textos.

O foco desta sessão é descrever, brevemente, os principais métodos de AM utilizados nesta pesquisa. O enfoque será no aprendizado supervisionado, método utilizado neste trabalho. Contudo, existem basicamente 4 modos de aprendizado, a saber: supervisionado, semi-supervisionado, não supervisionado e reforço.

De acordo com Dukart (2015), o aprendizado supervisionado refere-se ao caso em que se conhece o número de classes que devem ser aprendidas e a atribuição de casos de treinamento a essas classes, a chamada rotulagem. Esse tipo de algoritmo de aprendizado de

máquina visa identificar padrões que podem ser usados para distinguir as diferentes classes no conjunto de dados de treinamento. O classificador obtido pode então ser aplicado a novos dados com rótulos de classe desconhecidos. No Aprendizado Supervisionado, os *datasets* são treinados em um conjunto de treinamento e, em seguida, serão usados para rotular novas observações do conjunto de teste. Quanto ao conjunto de treinamento, as variáveis de entrada são as características que irão influenciar a precisão da variável predita. Contém variáveis quantitativas e qualitativas; a variável de saída é a classe de rótulo que o Aprendizado Supervisionado irá rotular as novas observações. De acordo com os diferentes tipos de variáveis de saída, as tarefas de Aprendizagem Supervisionada podem ser divididas em dois tipos: tarefa de classificação e tarefa de regressão. As variáveis de saída da tarefa de classificação são variáveis categóricas e as da tarefa de regressão são variáveis contínuas. Por exemplo, a classificação de comentários impróprios ou não é uma tarefa de classificação e a previsão do preço das ações é uma tarefa de regressão (WANG et al., 2021).

Os algoritmos de aprendizado não supervisionado são utilizados para agrupar casos com base em atributos semelhantes, tendências ou relações que ocorrem naturalmente nos dados, sem possuir uma classe alvo pré-determinada. Modelos não supervisionados incluem técnicas de agrupamento e mapas auto-organizados. Cada algoritmo usa estratégias distintas para dividir os dados em grupos, alguns são relativamente diretos, dividindo rapidamente os casos em grupos com base em atributos comuns ou semelhanças (MCCUE, 2015).

O aprendizado semi-supervisionado é uma mistura entre aprendizado supervisionado e aprendizado não supervisionado. Normalmente, os métodos de aprendizado supervisionado exigem grandes quantidades de dados rotulados. Enquanto a coleta de dados geralmente é barata, a rotulagem de dados geralmente só pode ser obtida com custos enormes porque os especialistas precisam anotar os dados manualmente. O aprendizado semi-supervisionado combina o aprendizado supervisionado e o uso de dados não rotulados (REINDERS et al., 2019).

O aprendizado por reforço é um tipo de programação dinâmica que treina algoritmos usando um sistema de recompensa e penalidade. O sistema de aprendizagem aprende com um ambiente interativo, selecionando e executando ações e, em virtude disso, recebendo recompensas pelo desempenho correto e penalidades pelo desempenho incorreto. No aprendizado por reforço, o sistema de aprendizagem aprende sozinho a melhor estratégia para maximizar a recompensa em uma situação particular usando programação dinâmica (BOUCHEFRY; SOUZA, 2020).

2.3.1 Algoritmos de classificação

Algoritmos de Classificação são técnicas de Aprendizagem Supervisionada usadas para categorizar novas observações. Na classificação, um programa usa o conjunto de dados

ou observações fornecidas para aprender como categorizar novas observações em várias classes ou grupos. Por exemplo, 0 ou 1, vermelho ou azul, sim ou não, spam ou não spam, etc. Alvos, rótulos ou categorias podem ser usados para descrever classes. Um algoritmo de classificação, em geral, é uma função que pesa as características de entrada para que a saída divida uma classe em valores positivos e a outra em valores negativos. O treinamento do classificador é realizado para identificar os pesos que fornecem a separação mais precisa e melhor das duas classes de dados (NETOFF, 2019).

Nesta subseção são destacados os principais algoritmos de AM para classificação de texto utilizados nesta pesquisa. Como a tarefa principal deste trabalho é a classificação binária de comentários, as métricas de classificação multi-classe estão fora do escopo deste trabalho.

2.3.1.1 FastText

O FastText é uma ferramenta eficiente voltada à obtenção de vetores de palavras e à classificação de textos. Seu principal objetivo é proporcionar bons resultados em tarefas de PLN com menor custo computacional e tempo de treinamento reduzido, em comparação a redes neurais profundas (YAO; ZHAI; GAO, 2020). A arquitetura do FastText se baseia em uma simplificação dos modelos tradicionais de redes neurais, incorporando características adicionais como n-gramas de palavras, que permitem capturar parcialmente a ordem e a estrutura do texto, superando algumas limitações do modelo tradicional de saco de palavras.

Diferentemente do modelo *bag-of-words*, que trata as palavras de forma isolada, o FastText considera, além dos vetores individuais de palavras, os n-gramas de caracteres ou palavras como atributos adicionais. Isso permite ao FastText representar melhor a semântica e a sintaxe do texto. Na camada oculta, o modelo realiza a média dos vetores das palavras e dos n-gramas, gerando uma representação densa e eficiente da sentença. Já na camada de saída, é utilizado o hierarchical softmax, uma técnica que reduz significativamente o tempo de treinamento, especialmente em tarefas com um grande número de classes. Esses aspectos tornam o FastText uma alternativa viável e poderosa para tarefas de classificação de texto em contextos com recursos computacionais limitados.

2.3.1.2 Naive Bayes

Considerado um algoritmo simples e poderoso, Naive Bayes (NB) é um classificador probabilístico baseado no teorema de Bayes, que assume que cada recurso faz uma contribuição independente e igual para a classe alvo. O classificador é fácil de implementar, computacionalmente rápido e funciona bem em grandes conjuntos de dados com alta dimensionalidade. Além disso, o classificador NB é propício para aplicações em tempo real, não sendo sensível a ruídos (MISRA; LI, 2020).

Na Figura 1 é dada a fórmula do teorema de Bayes.

Figura 1 – Teorema de Bayes

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

A abaixo está a descrição do que significa cada item da fórmula acima:

- P(A|B): Probabilidade do evento A acontecer quando ocorre B;
- P(B|A): Probabilidade de B acontecer, dado que A já ocorreu;
- P(A): Probabilidade de A ocorrer;
- P(B): Probabilidade de B acontecer.

2.3.1.3 Regressão Logística

A regressão logística é uma das ferramentas analíticas mais importantes nas ciências sociais e naturais. No processamento de linguagem natural, a regressão logística é o algoritmo *baseline* de aprendizado de máquina supervisionado para classificação e também tem uma estreita relação com as redes neurais. A regressão logística pode ser usada para classificar uma observação em uma das duas classes, como sentimento positivo e sentimento negativo, ou em uma de muitas classes (JURAFSKY; MARTIN, 2020).

Segundo Bartosik e Whittingham (2021) a vantagem mais significativa da regressão logística é que pode ser usado tanto para classificação quanto para estimativa de probabilidade de classe, pois está vinculado à distribuição de dados logísticos. Ela recebe uma combinação linear de recursos e aplica a eles uma função sigmoide não linear. A matemática da regressão logística baseia-se no conceito de razão de chances do evento, que é a probabilidade de um evento ocorrer dividida pela probabilidade de um evento não ocorrer. Assim como na regressão linear, a regressão logística tem pesos associados às dimensões dos dados de entrada. Ao contrário da regressão linear, a relação entre os pesos e a saída do modelo é exponencial, não linear. A função sigmoide pode ser representada matematicamente pela seguinte fórmula:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Descrição dos elementos da fórmula:

- $\sigma(x)$: representa o valor da função sigmoide para um dado valor de entrada x. Seu resultado sempre estará entre 0 e 1.
- x: valor de entrada (qualquer número real, positivo ou negativo).

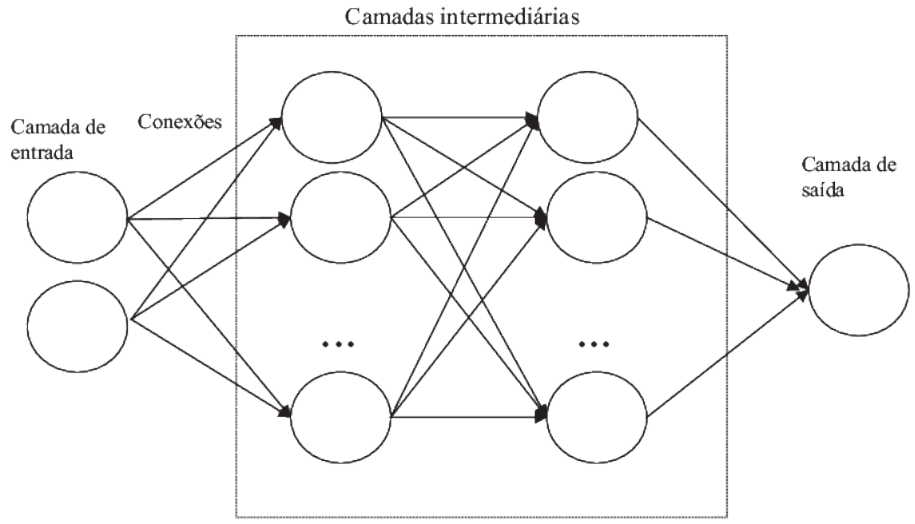
- e: constante de Euler, aproximadamente igual a 2,718.
- e^{-x}: exponencial negativa de x, responsável pela forma em "S"da função.

2.3.1.4 Redes Neurais

As Redes Neurais (RN), também conhecidas como Redes Neurais Artificiais (RNAs), são um modelo de inteligência artificial que ensina computadores a processar dados de forma a imitar o funcionamento do cérebro humano. Inicialmente, foram introduzidos como alternativa na resolução de problemas geográficos durante a década de 1990 e são amplamente usadas graças aos avanços na área da engenharia da computação, tecnologias de inteligência artificial e disponibilidade de dados (ANTONSICH et al., 2020).

Conforme apresentado pela Figura 2, uma RN pode ser descrita como uma coleção de nós conectados, chamados de neurônios artificiais, que são organizados em camadas. A camada de entrada é responsável por receber os dados de iniciais; enquanto as camadas intermediárias, comumente conhecidas como camadas ocultas, são responsáveis por extrair características importantes dos dados de entrada e transmiti-las para a camada de saída. A camada de saída tem o papel de produzir a saída final, correspondente à previsão da classe (GORA et al., 2020).

Figura 2 – Exemplo de arquitetura de uma rede neural de múltiplas camadas



Fonte: Caldeira, Souza e Machado (2009)

2.3.2 Treinamento e Teste

A divisão de treino e teste é uma técnica para avaliar o desempenho de um algoritmo de aprendizado de máquina. Ela pode ser usada para problemas de classificação ou regressão e pode ser usada para qualquer algoritmo de aprendizado supervisionado. O

procedimento envolve pegar um conjunto de dados e dividi-lo em dois subconjuntos. O primeiro subconjunto é usado para ajustar o modelo e é referido como o conjunto de dados de treinamento. O segundo subconjunto, os elementos de entrada do conjunto de dados são fornecidos ao modelo, então as previsões são feitas e comparadas com os valores esperados. Esse segundo conjunto de dados é chamado de conjunto de dados de teste. Contudo, é um desafio decidir a proporção dos conjuntos de dados de treinamento e teste. É altamente dependente do tamanho e da natureza do conjunto de dados. A presença de ruído e valores ausentes precisam ser considerados para dividir um conjunto de dados. A relação incorreta leva a problemas de *overfitting* e *underfitting*. Se os dados de treinamento contiverem todos os casos muito próximos dos dados de teste, ele enfrentará o problema de *overfitting*. Caso o conjunto de dados de treinamento seja muito pequeno, ele não conseguirá representar com precisão a variação e a complexidade da distribuição dos dados. Desse modo, o modelo apresentará *underfitting*, ou seja, um modelo que não se ajusta adequadamente aos dados de treinamento (PRADHAN et al., 2020).

2.3.3 Métricas de Avaliação

Perante tantas opções de classificadores, incluindo classificadores *ensemble*, a escolha de um classificador mais adequado para cada caso prático pode ser um desafio. Em virtude disto, as métricas de avaliação desempenham um papel crítico na obtenção de um classificador ideal. Desse modo, uma seleção de métricas de avaliação adequadas é a chave para discriminar e obter um classificador que melhor irá se adequar para o problema de pesquisa em questão. Nesta subseção serão apresentadas as principais métricas de avaliação utilizadas nesta pesquisa de Mestrado.

2.3.3.1 Matriz de Confusão

A matriz de confusão é uma medida muito usada na solução de problemas de classificação e pode ser usada para analisar o potencial de um classificador. Pode ser empregada na classificação de duas classes ou classificação multi-classe. Na Tabela 3 é apresentado um exemplo de matriz de confusão para classificação binária.

A matriz de confusão permite verificar o desempenho de um modelo preditivo, comparando os valores preditos com os valores reais dados por um classificador. Na matriz de confusão são apresentadas duas linhas e duas colunas nas quais têm-se o número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos. Os valores apresentados como falsos positivos são na verdade valores negativos classificados erroneamente. Da mesma forma, os valores falsos negativos são, de fato, de sentenças positivas classificadas na classe oposta. Os valores tidos como verdadeiros positivos e verdadeiros negativos são de sentenças classificadas corretamente.

Tabela 3 – Matriz de confusão binária

		Valor Predito	
		Positivo	Negativo
Valor Real	Positivo	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativo	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: autoria própria

2.3.3.2 Acurácia

A acurácia é tida como uma das métricas mais simples e importantes, comumente utilizada na avaliação de problemas de classificação de aprendizado de máquina. Ela é determinada por meio do número de previsões corretas sobre o tamanho da saída. Abaixo segue a fórmula para se calcular a acurácia:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN}$$

2.3.3.3 Precisão

A precisão mede a taxa de verdadeiros positivos em relação a todos os positivos apresentados pelo classificador. Não considera os valores de falso negativo e verdadeiro negativo.

$$Precisão = \frac{VP}{VP + FP}$$

2.3.3.4 Revocação

A revocação, também chamada de sensibilidade ou *recall*, é uma métrica que indica, dentre o total das amostras positivas existentes, quantas o modelo conseguiu classificar corretamente.

$$Revocação = \frac{VP}{VP + FN}$$

2.3.3.5 Medida-F

A medida-F ou F-*score* é a média harmônica da combinação entre precisão e revocação. É conhecida também como medida F_1, pois suas medidas são ponderadas uniformemente. Na fórmula abaixo considera-se $P = Precisão$ e $R = Revocação$.

$$F_1 = 2 * \frac{P * R}{P + R}$$

2.3.3.6 Curva ROC e AUC

A curva ROC ou *Receiver Operating Characteristic* é uma ferramenta muito usada para classificação binária. Ela expõe o quão bom o modelo criado pode discernir entre duas coisas, sejam positivas ou negativas, cães ou gatos, 0 ou 1, etc. A curva ROC possui dois parâmetros:

Taxa de verdadeiro positivo (TVP), que é sinônimo para revocação e, portanto, definido da seguinte forma:

$$TVP = \frac{VP}{VP + FN}$$

Taxa de Falso Positivo (TFP) que é a razão de instâncias negativas que foram erroneamente classificadas como positivas sobre o total de instâncias negativas, é definida da seguinte forma:

$$TFP = \frac{FP}{FP + VN}$$

A área sob a Curva (AUC) mede toda a área bidimensional abaixo de toda a curva ROC, seu valor varia de 0 até 1, sendo 0,5 o limiar entre classes. Os valores acima desse limiar são classificados em uma classe, enquanto que valores abaixo de 0,5 na outra. Um modelo cujas previsões estão totalmente erradas tem um AUC de 0, já um modelo com todas as previsões corretas tem um AUC de 1.

2.4 Aprendizado Profundo

O aprendizado profundo e a área da inteligência artificial têm alcançado significativa relevância nos últimos anos, apresentando um crescimento contínuo em sua popularidade. Fundamentado na estrutura do cérebro humano, o aprendizado profundo é uma vertente do campo de aprendizado de máquina que emprega redes neurais, uma série de camadas interconectadas, para processar informações de forma progressiva. Cada camada absorve informações distintas durante o processo, culminando em resultados relevantes. O aprendizado profundo emergiu como uma força impactante na indústria de aprendizado de máquina, especialmente no contexto do uso de Big Data (JANECZKO; SRIVASTAVA, 2022).

Os métodos de Deep Learning são primordialmente empregados para a representação e aprendizado de padrões, utilizando RNAs. A aplicação do aprendizado profundo pode ocorrer de forma supervisionada, não supervisionada ou semi-supervisionada. Essas

abordagens estão ganhando popularidade devido à sua notável precisão na detecção de anomalias em setores como infraestrutura crítica, saúde, defesa e diversos outros domínios (SIKDER; BATARSEH, 2023).

Segundo Perconti e Plebe (2020), um dos aspectos mais interessantes relacionados ao aprendizado profundo é que a tecnologia contém pequenas melhorias provenientes das RNAs, campo que estava estagnado no início deste século. É dito que uma das diferenças mais distintas entre a primeira geração de redes neurais artificiais e o atual empreendimento de aprendizagem profunda está relacionada ao seu foco. A principal motivação para o desenvolvimento das primeiras redes neurais foi o estudo da cognição. Por outro lado, a aprendizagem profunda tem se destacado por sua aplicabilidade em problemas complexos de reconhecimento de padrões, processamento de linguagem natural, visão computacional e muitas outras áreas. Essa mudança de enfoque impulsionou o desenvolvimento de arquiteturas de redes neurais mais profundas e eficazes, permitindo avanços notáveis na resolução de tarefas desafiadoras em diversos campos.

2.4.1 Modelos Pré-Treinados

Modelos pré-treinados são modelos de aprendizado profundo que passam por uma fase inicial de treinamento em grandes volumes de dados antes de serem aplicados a tarefas específicas. Durante essa fase de pré-treinamento, os modelos aprendem representações gerais dos dados, capturando padrões e estruturas relevantes. Posteriormente, esses modelos podem ser ajustados (*fine-tuned*) em tarefas mais específicas, como classificação de texto ou reconhecimento de imagens, utilizando um volume menor de dados. Essa abordagem permite que os modelos pré-treinados, como BERT[1] e *Generative Pre-trained Transformer* (GPT),[2] alcancem resultados superiores em uma ampla variedade de tarefas de processamento de linguagem natural (NLP) e outras áreas, devido à sua capacidade de reutilizar o conhecimento adquirido durante o pré-treinamento.

2.4.2 Transformers

Os transformers são uma arquitetura de rede neural que revolucionou o campo do processamento de linguagem natural (NLP). Introduzida por Vaswani (2017), essa arquitetura utiliza um mecanismo de atenção chamado *self-attention* para capturar dependências entre palavras em um texto, independentemente de sua distância no contexto da frase. Ao contrário de modelos tradicionais como *Recurrent Neural Networks* (RNNs)[3] ou *Long Short-Term Memory networks* (LSTMs),[4] os transformers permitem um alto grau de paralelismo durante o treinamento, resultando em maior eficiência e melhor desempenho em

[1] <https://huggingface.co/docs/transformers/model_doc/bert>
[2] <https://www.ibm.com/think/topics/gpt>
[3] <https://aws.amazon.com/pt/what-is/recurrent-neural-network/>
[4] <https://www.sciencedirect.com/science/article/pii/S0167278919305974>

tarefas que envolvem grandes volumes de dados. O paralelismo refere-se à capacidade de processar simultaneamente várias partes de um modelo ou de dados, ao invés de processá-los de forma sequencial. No contexto dos transformers, esse paralelismo é facilitado pelo mecanismo de *self-attention*, que permite ao modelo analisar todas as palavras de uma frase em paralelo, em vez de uma por vez. Essa arquitetura serve de base para modelos como BERT e GPT, amplamente utilizados em tarefas como tradução, classificação de texto e geração de linguagem.

2.4.3 Transferência de Aprendizagem

O aprendizado profundo é uma técnica que depende da disponibilidade de um grande conjunto de dados para alcançar o nível desejado de precisão. No entanto, muitas vezes não é possível coletar conjuntos de dados rotulados e curados em alguns campos de estudo. Em virtude disso, surge a transferência de aprendizado.

A transferência de aprendizado é uma abordagem valiosa quando a disponibilidade de dados de treinamento é limitada, conforme apontado por Weiss, Khoshgoftaar e Wang (2016). Isso pode ocorrer devido à escassez de dados, ao custo elevado de coletar e rotular dados ou à inacessibilidade dos mesmos. Com os repositórios de big data se tornando mais prevalentes, o uso de conjuntos de dados existentes, relacionados, mas não idênticos, a um domínio de interesse, torna as soluções de aprendizado por transferência uma abordagem atraente.

A abordagem de transferência de aprendizado elimina a necessidade de um grande conjunto de dados de treinamento rotulado e torna o processo de treinamento mais eficiente em termos de utilização de recursos. Em vez de começar do zero a cada vez, o conhecimento anterior pode ser aproveitado de um modelo existente, e um novo modelo pode ser construído iterativamente com otimização de hiperparâmetros e ajuste fino do modelo existente (PATRA et al., 2023).

2.4.4 BERT

O BERT (Bidirectional Encoder Representations from Transformers) é um modelo baseado em transformadores que opera em duas etapas principais: pré-treinamento e ajuste fino (fine-tuning). O BERT foi projetado para pré-treinar representações bidirecionais profundas a partir de textos não rotulados, condicionando conjuntamente os contextos à esquerda e à direita em todas as camadas do modelo. Após o pré-treinamento, o BERT passa pela fase de ajuste fino, no qual seus parâmetros previamente aprendidos são utilizados como base, e o modelo é então treinado adicionalmente em dados rotulados específicos para a tarefa a ser realizada. Essa abordagem torna o BERT altamente eficaz em diversas tarefas de processamento de linguagem natural (PLN), como classificação de texto, reconhecimento de entidades e respostas a perguntas (DEVLIN et al., 2019).

2.5 Métodos de Reamostragem

Os métodos de reamostragem alteram a distribuição de várias classes em um conjunto de dados com o intuito de equilibrar seus rótulos. Eles ajustam os dados de treinamento diretamente desbalanceados via subamostragem ou sobreamostragem e podem ser combinados com diversos classificadores devido à sua independência dos processos de aprendizagem do classificador. Desequilíbrio e sobreposição de classes são fatores que influenciam diretamente o desempenho dos modelos de classificação. O desequilíbrio entre classes pode ocasionar em uma capacidade menor de representação da classe minoritária, devido à escassez de suas amostras. Já a sobreposição de classes implica na existência de regiões no espaço amostral onde há probabilidades anteriores aproximadamente iguais e podem resultar em um limite de classificação ambíguo (SHI et al., 2022).

A fim de aprender com um conjunto de dados enviesados, é necessário resolver os desequilíbrios presentes nos dados de treinamento. Para isso, os chamados métodos de reamostragem são a solução para remediar os desequilíbrios presentes nos dados de treinamento. A reamostragem harmoniza o número de instâncias nos dados de treinamento ajustando o número de sentenças das classes majoritária e minoritária (SASADA et al., 2020).

Além disso, a partir de um *dataset* é possível extrair, repetidamente, amostras utilizando métodos de reamostragem, a fim de calcular métricas e estatísticas em cada uma dessas amostras, objetivando obter mais informações sobre o desempenho do modelo. Os métodos de reamostragem, no caso de uma análise estatística, podem fornecer dados interessantes e visões diferentes a respeito do comportamento de algum parâmetro.

2.5.1 Validação Cruzada k-fold

Na validação cruzada é possível definir um número fixo de *folds* ou partições dos dados. Se os dados forem divididos em três partições aproximadamente iguais, cada uma, por sua vez, será usada para teste e o restante para treinamento. Ou seja, dois terços dos dados para treinamento, e um terço para teste. O procedimento é repetido três vezes para que, no final, cada instância tenha sido usada exatamente uma vez para teste. Este método é conhecido como validação cruzada tripla e, caso a estratificação também seja utilizada, é chamado de validação cruzada tripla estratificada.

Segundo Witten, Frank e Hall (2011), a maneira padrão de prever a taxa de erro de uma técnica de aprendizado, dada uma única amostra fixa de dados, é a validação cruzada estratificada de dez *folds*. Os dados são fracionados, aleatoriamente, em 10 partições nas quais as classes são representadas aproximadamente nas mesmas proporções presentes no conjunto de dados. Então, um subconjunto é utilizado para teste e os demais nove para estimação dos parâmetros, realizando cálculos da acurácia do modelo. Este processo é realizado dez vezes alternando de forma circular o subconjunto de teste.

2.6 Métodos Ensemble

Um *ensemble* é um método de aprendizado no qual uma coleção de um número finito de classificadores é treinada para a mesma tarefa de classificação e, assim, podendo obter um melhor desempenho. Nos últimos anos, os métodos *ensemble* foram empregados para aumentar a precisão na classificação além do nível alcançado por classificadores individuais. Normalmente, o aprendizado de conjunto envolve classificadores paramétricos estatísticos ou redes neurais treinadas nos mesmos dados e um método que combina suas saídas em uma única (YANG, 2011). Segundo Rahmani et al. (2021), o objetivo destes métodos é criar classificadores com viés relativamente constante. Além disso, o *ensemble* combina as saídas dos classificadores por meio do esquema de média ou outros métodos para reduzir a variação e melhorar a precisão. A precisão preditiva de um ensemble tende a melhorar mais se os modelos individuais forem, além de precisos, também diversos, ou seja, se cometerem erros diferentes. Sem diversidade, a previsão combinada cometeria erros semelhantes às previsões individuais. Portanto, os algoritmos de criação de modelos ensemble consistem em diferentes técnicas para criar modelos individuais diversos, e não apenas precisos (MINKU, 2016).

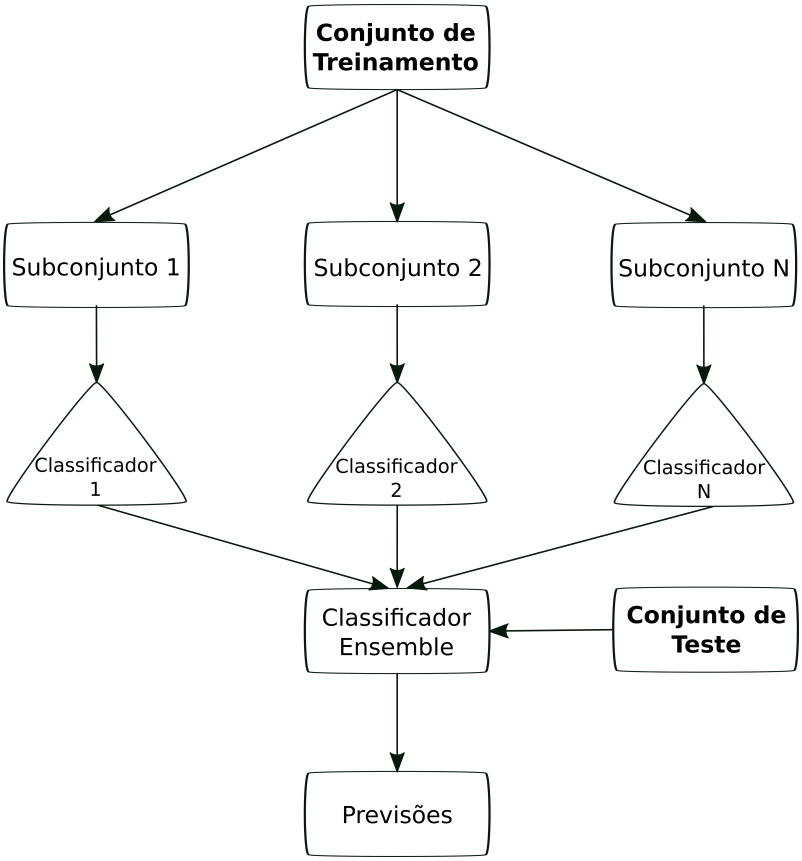
2.6.1 Bagging

O método *Bootstrap Aggregation*, ou Bagging, proposto por Breiman (1996), é um meta-algoritmo para construção de classificadores agregados. Bagging treina um conjunto de classificadores-base de forma independente por diferentes amostragens do conjunto de treinamento. A amostragem é realizada com reposição e tem o mesmo tamanho do conjunto de treinamento original. A classificação final é realizada por meio de um sistema de votação majoritária, no caso de um problema de classificação ou média dos valores no caso de regressão.

De acordo com Song et al. (2011), *Bagging* oferece várias vantagens em relação a outros algoritmos: ele pode evitar *overfitting*, problema no qual o classificador se ajusta demais aos dados de treinamento. Além disso, torna a classificação mais estável, pois qualquer classificador único, não importa quão forte, não pode ter um desempenho muito bom quando a distribuição do conjunto de teste é bastante diferente da do conjunto de treinamento. *Bagging* utiliza o resultado da votação dos classificadores-base, cada um com uma distribuição única do mesmo modelo, sendo assim mais estável em estatísticas.

A Figura 3 ilustra de forma geral o funcionamento do ensemble *Bagging*. Primeiramente, a partir de um conjunto de dados, são gerados novos subconjuntos por amostragem com reposição. Esses subconjuntos são utilizados para treinar os classificadores base ou estimadores. Em seguida, utilizando um conjunto de teste, os novos dados são preditos por meio do voto majoritário dos classificadores base.

Figura 3 – Infográfico de funcionamento do bagging



Fonte: De autoria própria

2.6.2 Boosting

O método *Boosting* combina diversos modelos buscando explicitamente os que melhor se complementam. Assim como *Bagging*, *Boosting* usa votação ou média, dependendo se é um problema de classificação ou regressão, para combinar a saída de modelos individuais. Enquanto no *Bagging* os modelos individuais são construídos separadamente e de maneira paralela, no *Boosting* cada novo modelo é influenciado pelo desempenho dos construídos anteriormente, criando um processo iterativo, em que em cada iteração é criado um modelo com base em amostras de treinamento e, além disso, instâncias preditas de forma incorreta têm seus pesos aumentados para a próxima iteração. O *Boosting* encoraja novos modelos a se tornarem especialistas em instâncias tratadas incorretamente por modelos anteriores, atribuindo maior peso a essas instâncias (WITTEN; FRANK; HALL, 2011).

De acordo com Simske (2019), *Boosting* é uma forma de aprendizado em conjunto, no qual o peso das amostras muda ao longo do tempo, a fim de permitir que o sistema otimize sua decisão considerando os resultados das amostras em proporção ao seu impacto na precisão geral do sistema. No *boosting*, inicialmente, as amostras são igualmente ponderadas. Após cada iteração do algoritmo, as amostras atribuídas corretamente recebem peso menor em relação às amostras atribuídas incorretamente.

2.6.3 Stacking

Stacking, também conhecido como *stacked generalization*, é uma técnica de *Ensemble Learning* que combina as previsões feitas por múltiplos classificadores de base, gerados, preferencialmente, usando diferentes algoritmos de aprendizado. Em virtude desta variedade de modelos, os resultados obtidos a partir de uma tarefa de classificação ou regressão possuem desempenhos superiores aos obtidos por quaisquer um dos modelos únicos presentes no conjunto (CUI et al., 2021).

A arquitetura de um modelo stacking, geralmente, é representada por meio de dois níveis, chamados de modelos de nível 0 e modelo de nível 1. Os modelos de nível 0 compreendem os modelos base, responsáveis por fornecer predições para o metamodelo; No nível 1, o metamodelo aprende como combinar melhor as previsões realizadas pelos modelos de base usando técnicas como voto majoritário para produzir uma previsão final (YAO et al., 2022).

2.7 Ferramentas de Ciência de Dados

Ferramentas de Ciência de Dados têm um papel essencial na exploração, análise e interpretação de conjuntos de dados complexos. Elas englobam uma ampla variedade de softwares e tecnologias que auxiliam em diferentes estágios do ciclo de vida da ciência de dados, desde a coleta e preparação dos dados até a modelagem, visualização e comunicação dos resultados. Essas ferramentas também oferecem interfaces amigáveis para a manipulação de dados. Muitas delas incorporam algoritmos de aprendizado de máquina e técnicas de mineração de dados, permitindo identificar padrões ocultos e relações complexas nos conjuntos de dados.

2.7.1 Orange Data Mining

Orange[5] é um conjunto de ferramentas para visualização de dados, aprendizado de máquina e mineração de dados de código aberto[6]. O software foi projetado para ser intuitivo e de fácil uso, principalmente, para iniciantes. Contudo, oferece recursos avançados para usuários mais experientes como, por exemplo, um módulo para a linguagem de programação Python.

Os componentes visuais do Orange são chamados de *widgets*. Suas aplicações e usos são diversos, indo desde simples visualizadores de dados e ferramentas de pré-processamento até modelagem preditiva e aplicação de algoritmos de aprendizado de máquina.

[5] <https://orangedatamining.com/>
[6] <https://github.com/biolab/orange3>

A Figura 4 ilustra um exemplo de pré-processamento com Orange, no qual são exibidos três componentes: *Corpus*, responsável por carregar o conjunto de dados; *Preprocess Test*, componente utilizado para pré-processar o conjunto de dados; *word cloud*, utilizado para visualizar as palavras presentes no *dataset* e suas frequências.

Figura 4 – Exemplo de pré-processamento com Orange

Fonte: De autoria própria

2.7.2 Jupyter Lab

No âmbito da análise de dados avançada, o Jupyter Lab[7] assume um papel crucial. Com sua abordagem interativa e flexível, essa ferramenta ajuda os pesquisadores a explorar informações, criar modelos analíticos e compartilhar resultados de forma eficaz. Usando linguagens como Python e R, junto com gráficos e textos explicativos, o Jupyter Lab fornece um ambiente colaborativo que incentiva a experimentação e facilita a comunicação clara de descobertas. Isso faz do Jupyter Lab uma ferramenta vital para impulsionar a pesquisa em análise de dados, oferecendo uma plataforma versátil para conduzir estudos e desenvolver soluções inovadoras.

2.8 Considerações Finais

Neste capítulo foi apresentada a fundamentação teórica necessária para o desenvolvimento desta pesquisa, destacando-se, dentre os demais tópicos: as técnicas de PLN, responsáveis pela preparação e pré-processamento de textos; Aprendizado de Máquina Supervisionado, no qual é construído um modelo que trabalha em cima de dados e respostas conhecidos e pré-estabelecidos; Medidas de desempenho dos classificadores, métricas que identificam a qualidade do classificador; Validação Cruzada, uma importante etapa em que ocorre a validação do modelo, verificando a confiabilidade dos resultados; classificadores *ensemble*, métodos que combinam os resultados de múltiplos modelos a fim de produzir um melhor modelo preditivo. Por fim, foram apresentadas duas ferramentas de ciência de dados: o *software Orange*, que é um kit de ferramentas para visualização de dados, aprendizado de máquina e mineração de dados, e o Jupyter Lab, utilizado nesta pesquisa para o desenvolvimento de modelos Ensemble e Deep Learning.

[7] <https://jupyter.org/>

Capítulo 3

Trabalhos Relacionados

3.1 Considerações Iniciais

O crescimento e proliferação de casos de assédio e *bullying* cibernético, motivam o surgimento de trabalhos que objetivam a identificação automática destas ocorrências nas mídias sociais. Neste capítulo, são apresentados os trabalhos científicos relacionados à classificação, identificação e moderação de comentários envolvendo assédio ou discurso de ódio em redes sociais. Na Tabela 4 (p. 55) são expostas, para fins de comparação, as principais características dos trabalhos identificados como relacionados à esta pesquisa.

3.2 Detecção de cyber-trolls

Capistrano, Suarez e Jr (2019) desenvolveram um método chamado SALSA para detectar comportamentos ciber agressivos, como trolling, em plataformas de mídia social. Com base em pesquisas da área da psicologia, os autores definiram os principais atributos do comportamento de trolls em mídias sociais: comportamentos agressivos, sutis ou explícitos; pouca preocupação com a estrutura gramatical de suas postagens; e sentimentos predominantemente negativos. O SALSA utiliza quatro categorias principais de características extraídas dos dados textuais do *dataset* Cyber-Troll: análise de sentimentos, análise de agressão, análise lexical e análise sintática.

Cada uma dessas categorias contribui com uma funcionalidade específica: a análise de sentimentos identifica e classifica o sentimento do texto em categorias negativas, neutras ou positivas; a análise de agressão mede a presença e intensidade da linguagem agressiva;

a análise lexical examina a frequência e o uso das palavras; e a análise sintática avalia a estrutura gramatical das sentenças. Ao integrar essas análises, o SALSA cria um conjunto de características robusto que aprimora a detecção de comportamentos ciber agressivos.

3.3 Identificação de Assédio Sexual

O trabalho realizado por Karlekar e Bansal (2018) utilizou classificação binária e classificação multiclasse com o objetivo de classificar diversas formas de assédio sexual presentes em histórias de abuso, compartilhadas no meio digital por intermédio do fórum SafeCity[1], uma plataforma de *crowdsourcing* de histórias pessoais de assédio sexual. Para a classificação binária, existem 3 classes com duas possíveis saídas: 1. *commenting* e *non-commenting*, 2. *ogling* e *non-ogling*, 3. *groping* e *non-groping*.

O conjunto de dados utilizado por Karlekar e Bansal (2018) inclui 9.892 casos de assédio, contendo uma descrição do incidente, o local e as formas marcadas de assédio. Os dados foram extraídos da SafeCity, contendo descrições de textos enviados por usuários do fórum, juntamente com 13 formas de assédio sexual. Contudo, apenas três categorias principais foram utilizadas, já as outras, por terem menos informações, foram descartadas.

Os modelos de rótulo único foram avaliados de acordo com sua precisão. Todos os modelos foram configurados com tamanho de vocabulário de 10.000 e usaram AdamOptimizer[2] com uma taxa de aprendizado de $1e^{-4}$. Os principais algoritmos avaliados foram: Linear SVM[3], Gaussian NB[4], Regressão Logística[5], SVM, CNNLogística[6], RNN[7] e CNN-RNN[8]. Desses, o modelo CNN-RNN obteve a melhor performance em comparação com várias linhas de base não neurais e neurais, com resultados de 81,6% para a classe *Commenting*, 84,1% para *Ogling* e 86,5% para *Groping*.

3.4 Identificação de Comentários Ofensivos

O estudo realizado por Pelle (2019) propõe uma abordagem para detectar comentários ofensivos na Web, denominada Hate2Vec[9]. Composta por um *ensemble* de classificadores no qual um meta-classificador decide se um comentário é ou não ofensivo com base na saída de três classificadores base: (i) um classificador baseado em léxico que utiliza a

[1] <https://www.safecity.in/>
[2] <https://www.sciencedirect.com/science/article/abs/pii/S0020025522006260>
[3] <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html>
[4] <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html>
[5] <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>
[6] <https://en.wikipedia.org/wiki/Convolutional_neural_network>
[7] <https://en.wikipedia.org/wiki/Recurrent_neural_network>
[8] <https://arxiv.org/abs/1604.04573>
[9] <https://github.com/rogersdepelle/hate2vec>

proximidade semântica das representações vetoriais de palavras; (ii) um classificador de regressão logística baseado em representações vetoriais de comentários; e (iii) um classificador *bag-of-words* baseado nos uni-gramas do texto.

A composição dos *datasets* utilizados na pesquisa de Pelle (2019) teve como base a diversidade de fontes de dados. Para isso, foram selecionados dados em Português e Inglês, compostos por postagens em redes sociais e comentários extraídos de sites. A pesquisa de Pelle (2019) contribuiu ainda com a criação de dois *datasets* em Português, OFFCOMBR-2 e OFFCOMBR-3. O primeiro possui 1250 instâncias, sendo que a classe atribuída a cada comentário foi a escolhida por pelo menos dois anotadores. O segundo dataset é um conjunto de dados mais restrito, chamado OFFCOMBR-3. Além disso, mais 2 datasets foram utilizados Tweets-EN[10] e Kaggle[11]. O primeiro, é um conjunto de dados com aproximadamente 16 mil tweets em Inglês anotados pela presença de discurso de ódio; o segundo é *dataset* é composto por 6 mil tweets em Inglês. Todas as instâncias foram anotadas como ofensivas e não ofensivas.

Foram realizados experimentos utilizando conjuntos de dados em Inglês e em Português, o método proposto Hate2Vec produziu bons resultados de classificação com valores de medida F_1 variando de 0,90 a 0,97 e com valores de ROC-AUC[12] variando de 0,88 a 0,98.

3.5 Detecção de Conteúdo Sexual Impróprio

No trabalho de Parnell et al. (2020) foi abordado o problema da identificação e detecção de conteúdo erótico inapropriado em mídias sociais usando técnicas de Processamento de Linguagem Natural (NLP). Seguindo uma abordagem baseada em aprendizado de máquinas, foram avaliados 12 modelos resultantes da combinação de três métodos: Bag of Words, *Term Frequency-Inverse Document Frequency* e Word2vec, juntamente com quatro classificadores: *Support Vector Machine* (SVM), Regressão Logística, k-vizinhos mais próximos e florestas aleatórias. Essas alternativas foram avaliadas em um novo conjunto de dados criado e extraído de dados públicos do Fórum Reddit[13].

A base de dados principal foi criada usando uma versão filtrada do conjunto de dados públicos do Reddit. O conjunto de dados original continha 98.753.936 de arquivos, com comentários postados no Fórum Reddit, que abrangiam diferentes tópicos e temas. Após a filtragem por assunto, foi obtido um conjunto de dados com 111.834 arquivos: 50.921 eróticos e os restantes 60.913 neutros.

Dos 12 modelos avaliados, o que apresentou melhor resultado de desempenho foi obtido pela combinação do TF-IDF e do classificador SVM com kernel linear com precisão de

[10] <https://github.com/zeeraktalat/hatespeech>
[11] <https://www.kaggle.com/c/detecting-insults-in-social-commentary/data>
[12] <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html>
[13] <https://www.reddit.com/>

0,97 e F_1 0,96. O classificador RF obtive os resultados menos satisfatórios, quando a profundidade das árvores é igual a 5, seu desempenho é bastante ruim ao usar BOW e TF-IDF, enquanto o Word2Vec tem desempenho bastante aceitável, atingindo uma precisão máxima de 0,9351.

3.6 Detecção de Ódio Online

O ódio online pode ser descrito como linguagem abusiva, agressão, cyberbullying, ódio, insultos, ataques pessoais, provocação, racismo, sexismo, ameaças ou toxicidade. É identificada como uma grande ameaça nas plataformas online de mídias sociais. O trabalho de Salminen et al. (2020) objetivou o desenvolvimento de um classificador que identifique as diversas formas de ódio online em diferentes plataformas. Para isso, foram avaliados vários algoritmos de classificação (Regressão logística, Naïve Bayes, SVM, XGBoost e Redes Neurais), diferentes recursos (BOW, TF-IDF, Word2Vec e BERT) e suas combinações.

Para construção do conjunto de dados, foram coletados um total de 197.566 comentários extraídos de quatro plataformas: Youtube, Reddit, Wikipedia e Twitter, contendo 80% dos comentários classificados como não odiosos e o restante como odiosos. Para seleção dos *datasets*, foram utilizados 3 critérios: (i) Inglês como idioma oficial, (ii) disponibilidade do conjunto de dados e (iii) avaliação manual do conjunto de dados.

O desempenho do classificador foi medido usando o conjunto de teste, aproximadamente 25% do conjunto total de dados, com duas métricas: (i) medida F_1 e (ii) área sob a curva (ROC-AUC). Os melhores resultados, considerando as duas métricas, na detecção de comentários odiosos em redes sociais, foram com o classificador XGBoost e com o recurso BERT, com pontuações entre 92% e 99%.

3.7 Detecção de Linguagem Ofensiva e Hate Speech

Segundo Vargas et al. (2021), no Brasil, o discurso de ódio é proibido, mas sua regulamentação enfrenta desafios na identificação, quantificação e classificação desse conteúdo na Internet. Para abordar essa questão relevante, os autores do artigo intitulado "HateBR: A Large Expert Annotated Corpus of Brazilian Instagram Comments for Offensive Language and Hate Speech Detection" propuseram o corpus "HateBR". Este corpus contém 7 mil sentenças coletadas de diversas contas de políticos brasileiros na plataforma de mídia social Instagram e possui três camadas diferentes, a saber: classificação ofensiva, classificação de ofensividade e classificação de discurso de ódio.

Os modelos foram criados usando duas representações diferentes e quatro métodos de aprendizado de máquina. As representações implementadas foram n-grams, mais especificamente o modelo de linguagem unigram, e o bag-of-ngrams com pré-processamento tf-idf. Os métodos de aprendizado de máquina aplicados foram Naive Bayes (NB), Sup-

port Vector Machine (SVM) com kernel linear, Multilayer Perceptron (MLP) com apenas uma camada oculta e Regressão Logística (LR). Além disso, os experimentos foram realizados utilizando a linguagem de programação Python e as bibliotecas Scikit-learn[14] e pandas[15]. Por fim, os dados foram divididos em 80% de treinamento, 10% de teste e 10% de validação.

Os resultados foram apresentados e avaliados em duas tarefas diferentes: detecção de linguagem ofensiva e detecção de discurso de ódio. Para a tarefa de detecção de linguagem ofensiva, Vargas et al. (2021) implementaram ambas as representações, unigram e TF-IDF, sobre os comentários do HateBR, com metade dos rótulos contendo sentenças ofensivas. Os resultados foram avaliados usando a medida-F e alcançaram uma pontuação de 85% com o algoritmo SVM e a representação TF-IDF. Quanto aos resultados da detecção de discurso de ódio, os melhores desempenhos foram obtidos com o algoritmo Naive Bayes usando a representação TF-IDF, alcançando uma medida-F de 78%.

3.8 Detecção de Agressão no X

Diversas abordagens têm sido propostas para arquiteturas de Redes Neurais Profundas (DNN) no contexto de processamento de linguagem natural e classificação de texto. Sadiq et al. (2021) descrevem uma arquitetura de DNN e a comparam com modelos de referência, como CNN-LSTM e CNN-BiLSTM. A metodologia inclui três módulos principais: extração de características, seleção e classificação. Foi utilizada uma Rede Neural Profunda simplificada baseada em camadas totalmente conectadas de um Perceptron Multicamadas (MLP). O modelo foi treinado utilizando a abordagem de validação cruzada com 10 divisões (10-fold), prevenindo assim o viés e garantindo a generalização dos resultados.

3.9 Detecção de Agressão em Mídias Sociais

No trabalho de Khan et al. (2022), foram propostas e incorporadas oito características emocionais em uma rede neural profunda (DNN) projetada com apenas três camadas para identificar declarações agressivas. O modelo DNN proposto foi comparado com modelos de última geração utilizando métricas de avaliação chave, como precisão, recall e acurácia. Além disso, os autores compararam os resultados de modelos tradicionais de aprendizado de máquina, como SVM, LR, NB, KNN, GB, DT e LDA, com modelos de aprendizado profundo, incluindo LSTM, BiLSTM, CNN e DNN. A combinação de embedding de palavras e as oito diferentes características emocionais foi alimentada na

[14] <https://scikit-learn.org/stable/>
[15] <https://pandas.pydata.org/>

DNN, visando uma melhoria significativa na detecção de tweets contendo cyberbullying, mantendo um design de DNN simples e computacionalmente eficiente. O modelo DNN proposto demonstrou o melhor desempenho em comparação com os outros modelos.

3.10 Detecção de Tweets Ofensivos

Alqahtani e Ilyas (2023b) propuseram um método automático para detectar cyberbullying em tweets utilizando dois modelos de deep learning para alcançar alta precisão. Os modelos utilizados foram a Long Short-Term Memory (LSTM) e a Convolutional Neural Network (CNN), que classificam tweets como ofensivos ou não-ofensivos. Eles combinaram datasets de discurso de ódio de cinco fontes diferentes, resultando em 162.000 registros rotulados como ofensivos ou não-ofensivos. O dataset foi dividido, com 90% usado para treinamento e os 10% restantes para teste. O sistema de detecção é composto por quatro partes principais: preparação do dataset, pré-processamento, *word embedding* e classificação. A maior precisão alcançada pelos modelos foi de aproximadamente 93%.

3.11 Comparativo Entre Trabalhos

A Tabela 4 exibe um comparativo entre os trabalhos relacionados identificados na literatura, nos quais se buscam a classificação binária de comentários contendo discurso de ódio, bullying cibernético, assédio sexual, racismo e outros. Além de destacar os objetivos pretendidos, os algoritmos empregados, os recursos utilizados e resultados alcançados. Em relação às informações contidas na coluna resultados, foram utilizadas as seguintes métricas e nomenclaturas: medida F (F_1), Revocação (R), Acurácia (Ac), Precisão (P) e ROC-AUC (AUC).

Na Tabela 4, é possível verificar os modelos que obtiveram melhor desempenho por meio das colunas “Algoritmos” e “Técnicas PLN”. Os algoritmos destacados em negrito na tabela foram avaliados na coluna “Resultados”, enquanto os demais foram utilizados em suas respectivas pesquisas, mas não tiveram suas métricas exibidas na Tabela 4. Todas as técnicas de PLN apresentadas foram utilizadas pelos seus respectivos algoritmos, exceto na 4ª e 6ª pesquisa, nas quais apenas a técnica de TF-IDF, destacada em negrito, foi utilizada para obter as métricas alcançadas.

3.12 Discussão

Usuários de mídias sociais, conhecidos como “trolls”, têm como objetivo criar discórdia, caos e disseminar desinformação, aproveitando-se do anonimato que a internet frequentemente oferece para cometer suas ações sem serem identificados. Nesse contexto,

Tabela 4 – Trabalhos relacionados à classificação de comentários em Redes Sociais

Autor∗	**Objetivo**	**Algoritmos**	**Resultados**	**Técnicas PLN**
[1]	Detecção de ciber-trolls	**MLP**	93,1% (R)	Análise de sentimento
[2]	Identificação de várias formas de assédio sexual	LSTM-RNN, **CNN-RNN**, CNN	86,5% (Ac)	Word embedding
[3]	Detecção de comentários ofensivos	**Hate2Vec**	93% (F_1) 93% (AUC)	Word2Vec, Doc2Vec
[4]	Detecção de conteúdo erótico inapropriado	**SVM**, LR, KNN, RF	97% (Ac) 96% (F_1)	BoW, **TF-IDF**, Word2Vec
[5]	Detecção de ódio online	LR, **XGBoost**, SVM, NB, NN	92% (F_1)	BoW, BERT, TF-IDF, Word2Vec
[6]	Detecção de Linguagem Ofensiva e Hate Speech	LR, MLP, **SVM**, NB	85% (F_1)	Unigram, **TF-IDF**

∗Autor: [1] (CAPISTRANO; SUAREZ; JR, 2019) [2] (KARLEKAR; BANSAL, 2018), [3] (PELLE, 2019), [4] (PARNELL et al., 2020), [5] (SALMINEN et al., 2020), [6] (VARGAS et al., 2021)

Capistrano, Suarez e Jr (2019) propuseram um método baseado em características textuais variadas para identificar se um tweet possui intenção provocativa. O método utiliza uma rede neural perceptron multicamadas e busca a melhor combinação de recursos com o intuito de otimizar o desempenho nas métricas de precisão, revocação, acurácia e F1. Os resultados obtidos foram satisfatórios, com destaque para a revocação, que atingiu um valor máximo de 93,12%.

Karlekar e Bansal (2018), apresentaram uma solução baseada na classificação binária e multiclasse, com a finalidade de identificar em histórias reais de assédio, três formas de violência apresentadas: *commenting*, *ogling* e *groping*. Segundo o autores, com as análises e dados obtidos, diversos caminhos para trabalhos futuros podem se abrir, principalmente para: (i) construção de uma ferramenta para analisar as circunstâncias mais comuns em torno de cada forma distinta de assédio, além de fornecer conselhos de segurança mais detalhados e precisos; (ii) um mapa de áreas inseguras para ajudar as pessoas a evitar espaços perigosos; (iii) um registro não oficial de criminosos sexuais.

Por outro lado, o estudo realizado por Pelle (2019) buscou a identificação de discursos de ódio em mídias sociais. Expôs que a detecção de discurso de ódio depende fortemente do contexto, que comentários considerados ofensivos em uma comunidade podem não ser em outras. Ademais, esta subjetividade pode ser um problema para a construção de *datasets*, que é agravada pela necessidade de se classificar muitas sentenças a fim de se conseguir uma amostra relevante. Os resultados apresentados na pesquisa foram expressivos: o

método composto por um conjunto de classificadores, ou *ensemble* de classificadores, se mostrou eficaz na tarefa de classificação de discurso de ódio, com resultados variando de 0,90 a 0,97 na medida F_1.

O estudo realizado por Parnell et al. (2020) focou na identificação e detecção de conteúdos eróticos inapropriados no contexto das mídias sociais. Com o objetivo de avaliar a possibilidade de desenvolver filtros para menores de idade ou qualquer usuário que tenha interesse em bloquear este tipo de conteúdo. Para isso, foram avaliadas as performances de diferentes classificadores implementando diferentes métodos e métricas. Os melhores resultados foram obtidos pelos classificadores SVM e LR, ambos com acurácia acima de 96% usando TF-IDF como técnica principal.

Ódio online pode ser entendido também como discurso de ódio e abrange diversas atitudes e ações criminosas online, como: agressão, *cyberbullying*, racismo, sexismo, etc. Nessa perspectiva, Salminen et al. (2020) propuseram um método de classificação binária para detecção de ódio online em mídias sociais. Os autores fizeram uso de diversos classificadores e recursos a fim de alcançar este objetivo. Os resultados apresentados foram expressivos, com valores 92,4% de medida F_1 e 99,5% de ROC-AUC, sendo que ambas as pontuações foram alcançadas utilizando o classificador XGBoost e a técnica BERT.

O estudo conduzido por Vargas et al. (2021) abordou o problema das mensagens ofensivas e do discurso de ódio que permeiam as redes sociais, especialmente no contexto político. Para enfrentar essa questão, foi apresentado um corpus anotado em três camadas distintas: (i) uma classificação binária entre comentários ofensivos e não ofensivos; (ii) um classificador para determinar o nível de ofensividade, categorizando os comentários em alta ofensividade, moderada ofensividade e ligeira ofensividade; (iii) além disso, os comentários foram classificados em nove grupos de discurso de ódio. Para atingir esses objetivos, avaliaram o desempenho de vários classificadores que empregaram diferentes métodos e métricas. Os resultados mais promissores foram obtidos com o classificador SVM, alcançando uma medida-F de 85% ao usar a técnica principal TF-IDF.

3.13 Considerações Finais

Neste capítulo, procurou-se evidenciar a importância dos estudos relacionados aos crimes de assédio sexual online, suas diferentes abordagens, os principais problemas enfrentados e, também, possibilidades de trabalhos futuros. As seções deste capítulo foram divididas com o intuito de expor diferentes pontos de vista ao efetivo enfrentamento do assédio sexual nas Redes Sociais, considerando o foco principal: a classificação binária de textos.

O próximo capítulo descreve as investigações desta pesquisa de Mestrado sobre classificação de comentários impróprios nas redes sociais.

Capítulo 4

Detecção Automática de Mensagens Ofensivas

4.1 Considerações Iniciais

O foco desta pesquisa de Mestrado está na detecção automática de mensagens ofensivas em textos curtos e ruidosos, utilizando um conjunto de dados composto por postagens extraídas da plataforma X (anteriormente conhecida como Twitter).

Partindo desta perspectiva este capítulo tem como objetivo detalhar a proposta desta pesquisa de Mestrado. Primeiramente, a Seção 4.2 aborda o contexto da investigação em relação ao objetivo apresentado na Seção 1.4. Em seguida, a Seção 4.3 apresenta o conjunto de dados utilizado, ressaltando suas principais características relevantes para este estudo. A Seção 4.4 descreve a metodologia empregada para a detecção de mensagens ofensivas. Por fim, a Seção 4.5 expõe os resultados obtidos pelos modelos de classificação, os quais são discutidos na Seção 4.6.

4.2 Contextualização

Embora as redes sociais implementem tecnologias automatizadas de moderação, o desafio de remover comentários agressivos em postagens ainda persiste. Comentários ofensivos, com linguagem agressiva ou provocativa, muitas vezes permanecem visíveis, mesmo quando denunciados por outros usuários. Esse cenário evidencia a necessidade

de aprimorar sistemas de detecção automática para identificar e remover rapidamente conteúdos prejudiciais, melhorando a experiência do usuário e fortalecendo a percepção das redes sociais como ambientes mais seguros.

A interação entre usuários nas redes sociais frequentemente facilita a exposição a comentários agressivos ou provocativos, impulsionada por diversos fatores, como a facilidade de criação de perfis falsos, a exposição excessiva de informações pessoais e o uso irrestrito por menores de idade. Nesse contexto, é comum que ao compartilhar uma foto ou vídeo, o autor seja alvo de comentários mal-intencionados, ofensivos ou inadequados. A detecção desses comentários nem sempre ocorre de forma eficiente, tanto por mecanismos automáticos quanto por moderadores, o que pode comprometer a saúde mental e emocional dos destinatários.

Nesse contexto, os algoritmos de classificação desempenham um papel muito importante, sendo um recurso confiável na categorização de sentenças. Com isso em vista, a escolha de um classificador adequado tende a ser influenciada pela quantidade e qualidade das *features* utilizadas no processo de classificação (MORÁN-FERNÁNDEZ; BÓLON-CANEDO; ALONSO-BETANZOS, 2022). No caso da detecção de comentários agressivos, a seleção de *features* é essencial para identificar palavras ou expressões que indicam comportamentos hostis. Essas *features* podem incluir, por exemplo, palavras associadas à violência verbal, ofensas diretas, ameaças ou linguagens desrespeitosas. A identificação dessas características é fundamental para aprimorar a precisão dos modelos na categorização de conteúdos como agressivos ou neutros.

A disponibilidade de *datasets* específicos para a detecção de comportamento agressivo, como o Cyber-Troll, possibilita a criação de modelos mais robustos e alinhados com os desafios reais encontrados nas redes sociais. Ao utilizar dados anotados de forma criteriosa, é possível capturar padrões complexos de agressividade verbal que, de outro modo, poderiam passar despercebidos. Isso contribui para o desenvolvimento de sistemas mais eficazes na identificação e mitigação de interações hostis online.

4.3 Conjuntos de Dados

Nesta pesquisa, foram utilizados dois conjuntos de dados: o Cyber-Troll, de DataTurks (2018), e o conjunto de dados desenvolvido por Davidson et al. (2017), ambos disponíveis publicamente no Kaggle[1,2], uma plataforma dedicada a cientistas de dados e profissionais de aprendizado de máquina.

A escolha desses conjuntos se justifica por sua ampla utilização em estudos relacionados à detecção de linguagem ofensiva e agressiva em redes sociais, o que possibilita a comparação dos resultados obtidos com os de trabalhos anteriores. Ademais, ambos os

[1] <https://www.kaggle.com/datasets/dataturks/dataset-for-detection-of-cybertrolls>
[2] <https://www.kaggle.com/datasets/eldrich/hate-speech-offensive-tweets-by-davidson-et-al>

datasets apresentam características alinhadas com os objetivos da pesquisa, como textos curtos, linguagem informal e presença de mensagens ofensivas anotadas manualmente, o que favorece a avaliação da eficácia dos métodos propostos em contextos realistas e desafiadores.

O conjunto de dados Cyber-Troll está estruturado no formato JSON e contém 10 campos distintos, dos quais apenas dois foram utilizados nesta pesquisa: o campo content, que representa o texto do tweet, e o campo label, que indica a categoria atribuída ao registro. O dataset é composto por tweets em inglês, classificados em duas categorias: ciberagressivos e não ciberagressivos (DataTurks, 2018). Os tweets ciberagressivos contêm mensagens destinadas a ofender e desrespeitar outros usuários, enquanto os tweets não ciberagressivos apresentam mensagens consideradas neutras. Como ilustrado na Figura 5, o conjunto de dados possui um total de 20.001 tweets, sendo 60,9% classificados como não ciberagressivos, o que equivale a aproximadamente 12.179 tweets, e 39,1% como ciberagressivos, totalizando cerca de 7.822 tweets. Para os experimentos, 10% dos dados foram separados previamente para compor o conjunto de teste. Os 90% restantes foram utilizados nas etapas de treinamento e validação dos modelos.

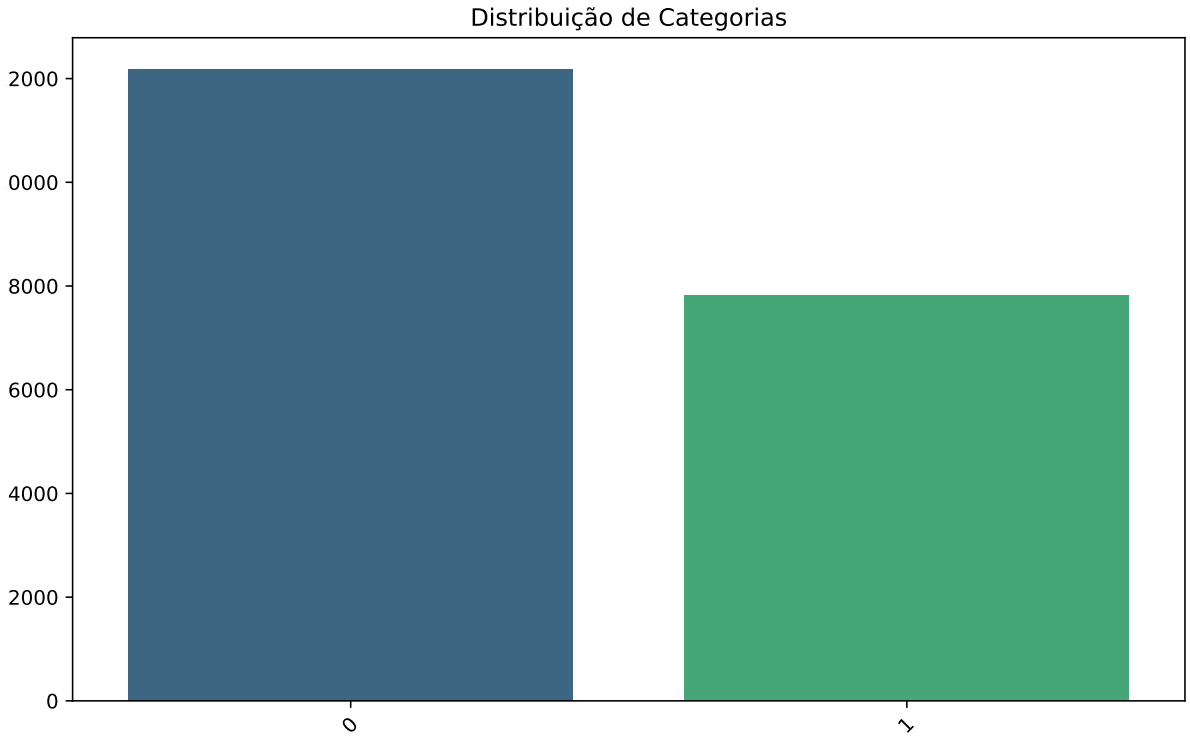
O segundo conjunto de dados utilizado foi o desenvolvido por Davidson et al. (2017), baseado em um vocabulário colaborativo de discurso de ódio obtido no Hatebase.org[3], com o objetivo de coletar tweets que contivessem palavras-chave associadas a discursos de ódio. Uma amostra aleatória de 25.000 tweets foi rotulada manualmente por trabalhadores da plataforma CrowdFlower[4] em três categorias: discurso de ódio, linguagem ofensiva e nenhuma das anteriores. Cada tweet foi revisado por, no mínimo, três trabalhadores, alcançando um índice de concordância intercoder de 92%, sendo atribuído o rótulo da maioria. Esse processo resultou em um subconjunto composto por 24.802 tweets rotulados. Para esta pesquisa, foram considerados apenas os tweets classificados como ofensivos ou neutros. Dessa forma, todos os tweets rotulados como hate speech foram removidos, a fim de alinhar o conjunto de dados à abordagem de classificação binária adotada no Cyber-Troll. O conjunto de dados Davidson foi então empregado para avaliar o desempenho dos modelos treinados, utilizando-os para prever tweets ofensivos e neutros.

4.4 Metodologia

Esta seção apresenta uma análise detalhada do *framework* utilizado para a identificação de tweets agressivos, empregando tanto modelos de aprendizado de máquina quanto de aprendizado profundo. A metodologia começa com a descrição do processo de manipulação dos dados, abrangendo desde o carregamento e a divisão do conjunto de dados,

[3] <https://hatebase.org/>
[4] <https://www.appen.com/>

Figura 5 – Distribuição de classes no conjunto de dados Cyber-Troll.



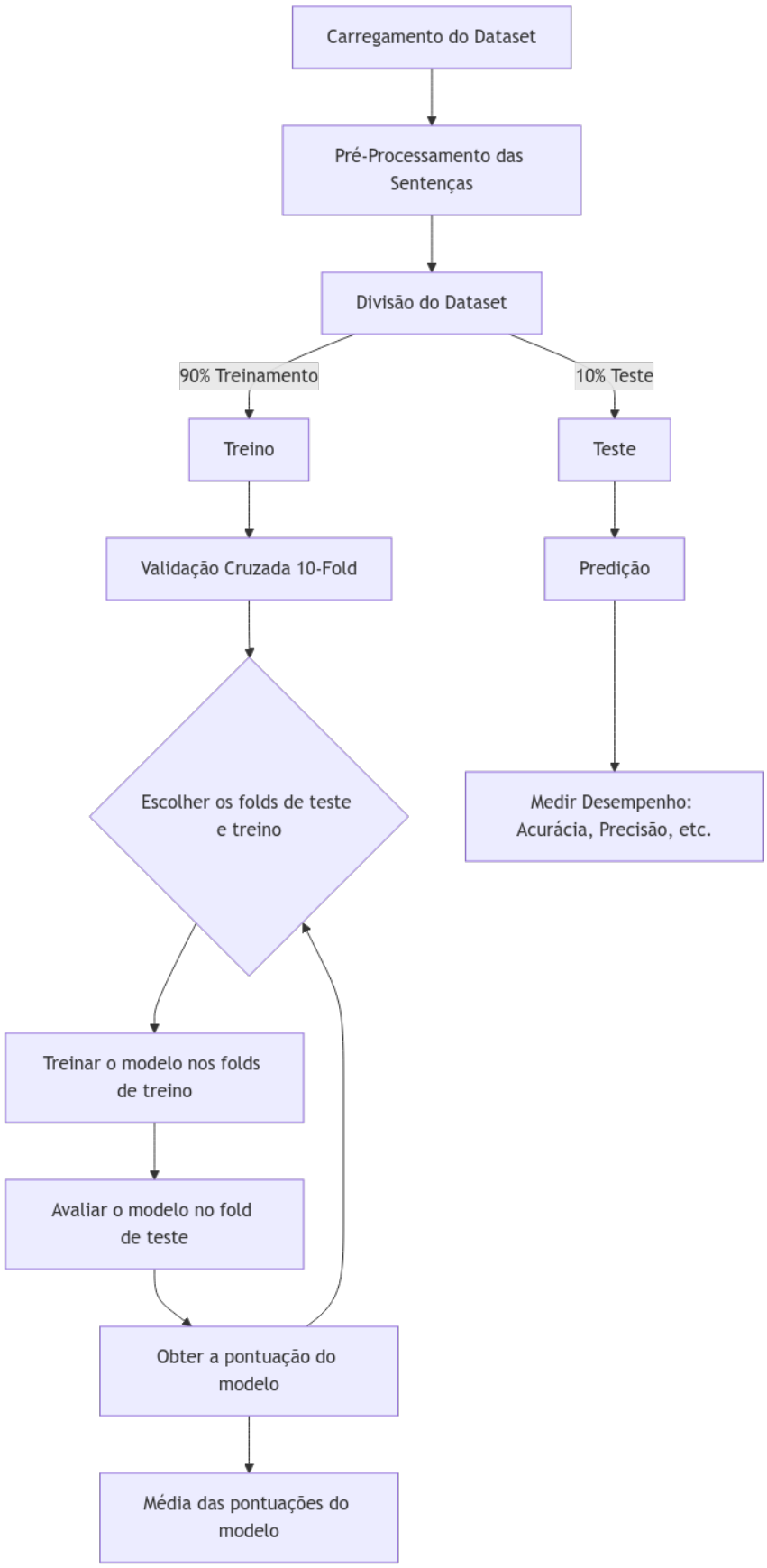
passando pelas etapas de pré-processamento aplicadas, até as fases de validação e predição dos modelos. A Figura 5, apresenta o *framework* utilizado nesta pesquisa para detecção de conteúdo agressivo no X, apontando as principais etapas empregadas.

4.4.1 Carregamento e Embaralhamento dos Datasets

Inicialmente, os dados do conjunto de dados Cyber-Troll estavam armazenados no formato JSON, sendo convertidos para o formato CSV com o objetivo de facilitar a visualização e manipulação das informações. Durante essa etapa, foram removidas colunas que não continham valores em nenhum registro, bem como colunas que armazenavam metadados irrelevantes para os experimentos. Em seguida, os dados foram carregados utilizando a função "read_csv" da biblioteca pandas, sendo convertidos para o formato DataFrame para permitir maior flexibilidade em operações de análise e processamento. Após o carregamento, os dados foram randomizados para minimizar possíveis vieses relacionados à ordem original dos registros. No caso do modelo FastText,[5] foi necessária uma modificação adicional, envolvendo a conversão do dataset para o formato TXT e ajustes estruturais específicos para atender aos requisitos de entrada desse algoritmo. Para garantir a transparência e facilitar a replicação dos experimentos, todos os algoritmos e datasets utilizados nesta pesquisa estão disponibilizados no repositório GitHub[6].

[5] <https://fasttext.cc/>
[6] <https://github.com/researchstudy2025/offensive_tweets_detection>

Figura 6 – Estrutura proposta para detecção de tweets agressivos.



Para avaliar a robustez dos resultados obtidos com o modelo BERT pré-treinado (bert-base-uncased) no conjunto de dados Cyber-Troll, foi utilizado, para fins de comparação, o conjunto de dados Davidson et al. (2017), amplamente empregado em pesquisas sobre discurso ofensivo. Para garantir consistência nas análises, o conjunto de dados Davidson passou por modificações, incluindo a remoção de todas as instâncias rotuladas como discurso de ódio (rótulo 0) e a reclassificação das instâncias originalmente rotuladas como "nenhum" (rótulo 2) para o rótulo 0. Essas alterações alinharam as classes entre os dois datasets, permitindo a aplicação do modelo treinado no conjunto de dados Cyber-Troll para prever sentenças ofensivas no conjunto Davidson, facilitando, assim, uma comparação direta e sistemática dos resultados obtidos.

4.4.2 Pré-processamento das Sentenças

Os modelos passaram por etapas comuns de pré-processamento no conjunto de dados, exceto pelas etapas específicas de cada modelo. Três operações principais foram realizadas para garantir que os dados estivessem adequadamente preparados para obter o melhor desempenho. Primeiramente, os dados foram carregados e embaralhados aleatoriamente, evitando vieses de ordenação (PRUSTY; PATNAIK; DASH, 2022). Em seguida, a tokenização foi realizada utilizando o TweetTokenizer[7], que separa emoticons individualmente, mantém hashtags e outros símbolos especiais. Essa escolha se justifica por ser mais adequada ao tipo de dados em questão, ou seja, comentários nos quais, muitas vezes, letras são substituídas por símbolos para dificultar seu reconhecimento por mecanismos de moderação. Além disso, todos os caracteres foram convertidos para letras minúsculas, assegurando consistência.

O WordNetLemmatizer[8] foi empregado para lematizar os tokens, reduzindo-os às suas formas básicas e, assim, potencializando a generalização dos modelos. Por fim, os tokens processados foram reunidos em cadeias de texto, com remoção de espaços extras para garantir uniformidade.

Além dessas etapas padrão, cada modelo foi submetido a procedimentos de pré-processamento específicos. Para o modelo de bagging, utilizou-se o TfidfVectorizer,[9] que transformou os textos em vetores de frequência ponderados pela frequência inversa dos termos, destacando a relevância de palavras menos comuns. O modelo FastText exigiu a formatação do conjunto de dados para garantir compatibilidade, ajustando sua estrutura às especificações do algoritmo. Já no caso do BERT, os textos foram vetorizados utilizando o tokenizer pré-treinado (bert-base-uncased), que converteu os textos em tensores

[7] <https://www.nltk.org/api/nltk.tokenize.casual.html>
[8] <https://www.nltk.org/api/nltk.stem.WordNetLemmatizer.html?highlight=wordnet>
[9] <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>

numéricos, incorporando informações semânticas contextuais. A vetorização foi padronizada para um comprimento máximo de 64 tokens, aplicando preenchimento (padding) e truncamento para uniformizar o tamanho das sequências.

4.4.3 Divisão do Dataset

Nesta etapa, os dados do conjunto Cyber-Troll foram inicialmente randomizados utilizando a função "sample" da biblioteca pandas. Em seguida, foram divididos por meio da função "train_test_split" da biblioteca Scikit-learn,[10] alocando 90% dos dados para os subconjuntos de treinamento e validação, e reservando 10% para o subconjunto de teste. Para assegurar a reprodutibilidade do experimento, foi definido um valor fixo para a variável "random_state".

4.4.4 Validação Cruzada K-Fold

A técnica de validação cruzada k-fold é uma metodologia robusta para avaliar o desempenho de modelos de aprendizado de máquina, garantindo uma avaliação mais confiável e menos suscetível às variações dos dados. Neste estudo, utilizou-se a validação cruzada estratificada com dez dobras (stratified k-fold), implementada por meio da classe StratifiedKFold da biblioteca Scikit-learn. A escolha da validação estratificada se deve ao fato de ela preservar a proporção das classes em cada subdivisão, assegurando uma representação consistente dos dados minoritários em todas as etapas do treinamento e validação.

Após o pré-processamento, o conjunto de dados foi dividido em 18.001 sentenças para treinamento e validação, enquanto as sentenças restantes foram alocadas no conjunto de teste. Durante a validação cruzada, o conjunto de treinamento foi particionado em dez dobras. A cada iteração, nove dessas dobras foram utilizadas para treinar o modelo, enquanto a décima dobra foi usada para validação, repetindo o processo dez vezes. Esse procedimento foi essencial para calcular métricas como precisão, revocação e F1-score em cada iteração, fornecendo uma avaliação detalhada do desempenho do modelo.

Após o término da validação cruzada, o modelo foi avaliado com o conjunto de teste previamente separado. Esse conjunto foi submetido ao mesmo pré-processamento aplicado aos dados de treinamento, garantindo consistência no tratamento dos dados. A avaliação final com o conjunto de teste permitiu verificar a capacidade de generalização do modelo para dados não vistos, fornecendo uma estimativa robusta e realista de seu desempenho em cenários do mundo real.

[10] <https://scikit-learn.org/stable/>

4.4.5 Predição

Após as etapas de treino e validação, o modelo foi submetido à fase de predição utilizando um conjunto de teste composto por dados que não foram empregados nas etapas anteriores. O objetivo dessa etapa foi avaliar a capacidade do modelo de generalizar para dados não vistos. Além disso, com o intuito de verificar a robustez do modelo em um cenário de domínio distinto, foi utilizado um segundo conjunto de dados na etapa de predição. Nesse segundo experimento, considerou-se exclusivamente o algoritmo BERT por se tratar de um modelo amplamente reconhecido na literatura por sua eficácia na detecção de linguagem ofensiva, servindo, portanto, como um baseline robusto para essa análise complementar. O modelo BERT foi treinado e validado utilizando o conjunto de dados Cyber-Troll, sendo então salvo e posteriormente empregado para realizar predições no conjunto de dados proposto por Davidson et al. (2017). Ressalta-se que o modelo completo foi utilizado como classificador, preservando os pesos obtidos na etapa de treinamento original.

4.5 Resultados Obtidos

Nesta seção, serão apresentados os resultados obtidos a partir dos experimentos descritos nos capítulos anteriores. Esses resultados são essenciais para avaliar a eficácia das metodologias e para responder às questões de pesquisa propostas.

E neste sentido, foram empregadas quatro métricas principais na avaliação do desempenho dos modelos de classificação de sentenças impróprias:

- ❏ acurácia – que mensura a capacidade geral de classificação correta das sentenças;
- ❏ precisão – que avalia a habilidade do modelo em evitar falsos positivos;
- ❏ revocação – que quantifica a proporção de sentenças impróprias corretamente identificadas em relação ao total de sentenças impróprias no conjunto de dados; e
- ❏ Medida-F – a média harmônica entre precisão e revocação.

No contexto desta pesquisa de Mestrado, a métrica de maior relevância na análise da classificação de sentenças agressivas em redes sociais é a revogação (recall). Essa escolha se fundamenta no objetivo principal de identificar e sinalizar o maior número possível de conteúdos potencialmente agressivos, mesmo que isso ocasione um aumento nos falsos positivos. Em sistemas de detecção automática utilizados como APIs de suporte à moderação de conteúdo, as mensagens classificadas como ofensivas não são removidas automaticamente da plataforma, mas sim enviadas para revisão ou marcadas para moderação humana. Nesse cenário, falsos negativos são mais prejudiciais, pois representam casos de mensagens agressivas que passam despercebidas pelo sistema, permanecendo visíveis e

potencialmente causando dano aos usuários. Já os falsos positivos, embora indesejáveis, podem ser revisados e revertidos, causando menor impacto negativo. Portanto, o foco na maximização do recall é justificado pela necessidade de sensibilidade ao problema, priorizando a não omissão de conteúdos danosos, ainda que com algum custo de precisão.

Para treinar os modelos, especialmente o BERT, em um período de tempo razoável, foi necessário utilizar um hardware com especificações adequadas. Para todas as etapas deste estudo, empregamos um notebook com Windows 11 Pro, equipado com um processador Intel(R) Core(TM) i7-11800H de 11ª geração, operando a 2,30 GHz, 32 GB de RAM, uma GPU NVIDIA GeForce RTX 3060 com 8 GB de VRAM e um SSD de 2 TB para armazenamento. Essas especificações atenderam às demandas computacionais exigidas para o treinamento dos modelos, permitindo o manuseio eficiente de grandes conjuntos de dados e reduzindo significativamente o tempo de processamento.

4.5.1 Modelos de Aprendizado de Máquina

Neste trabalho, foram avaliados os desempenhos de diferentes algoritmos de aprendizado de máquina na tarefa de classificação de sentenças ofensivas, incluindo FastText, Redes Neurais Perceptron Multicamadas (MLP), Regressão Logística e Naive Bayes. A Tabela 5 apresenta uma comparação detalhada entre os modelos, destacando as métricas avaliadas, as técnicas de validação empregadas e os resultados obtidos.

Tabela 5 – Comparativo entre os algoritmos convencionais

Algoritmos	**P**	**R**	**F-1**	**Ac**
Redes Neurais	81,9%	95,2%	88,1%	89,9%
Regressão Logística	58,0%	39,7%	47,2%	64,1%
BernoulliNB	51,0%	56,6%	53,6%	60,5%
FastText	94,5%	95,8%	95,2%	94,4%

Precisão (P), Revocação (R), Medida-F (F-1), Acurácia (Ac)

Sem considerar o modelo FastText, o algoritmo que apresentou a melhor performance foi o de Redes Neurais. Apesar do tamanho reduzido do conjunto de dados, os valores alcançados ficaram acima de 81% para todas as métricas analisadas. Nesse modelo, os parâmetros que proporcionaram o melhor desempenho foram o Adam, utilizado como solver para atualizar os pesos da rede em pequenos incrementos a cada iteração, e a função de ativação ReLU, empregada em cada camada da rede neural para introduzir não linearidades nas previsões. Além disso, o modelo foi configurado com 100 neurônios nas camadas ocultas e um número máximo de 20 iterações, correspondente à quantidade máxima de vezes que o modelo percorreu todo o conjunto de treinamento para atualizar seus parâmetros. A validação cruzada estratificada com 10 folds apresentou melhor desempenho em relação às demais opções testadas.

A Regressão Logística (RL) foi o modelo que apresentou o pior desempenho entre os avaliados neste cenário, alcançando seus melhores resultados com 64,1% de precisão e 58% de revocação. Para garantir uma avaliação uniforme, foi empregada a técnica de validação cruzada estratificada com 10 folds. Além disso, a escolha do tipo de regularização influenciou diretamente no desempenho do modelo, uma vez que esse mecanismo contribui para a redução do *overfitting*, diminuindo a variância e favorecendo a capacidade de generalização do modelo. Na RL, existem dois tipos principais de regularização: L1 (Lasso) e L2 (Ridge). A principal diferença entre elas está na forma como penalizam os coeficientes das variáveis. A técnica de regularização Lasso, adotada nesta pesquisa, tem como característica a redução dos coeficientes de atributos menos relevantes a zero, promovendo, assim, a exclusão automática de variáveis. Esse comportamento torna o Lasso especialmente útil em cenários com grande número de variáveis, contribuindo para a seleção de atributos

O classificador Bernoulli Naive Bayes (BernoulliNB) foi escolhido devido à sua capacidade de lidar com dados discretos, sendo especialmente adequado para representações binárias de texto, onde a presença ou ausência de uma palavra é um fator determinante. No entanto, seu desempenho, juntamente com o da Regressão Logística (RL), foi inferior ao dos demais algoritmos analisados, apresentando 51,0% de precisão, 56,6% de revocação e 60,5% de acurácia. Esses resultados foram obtidos por meio de validação cruzada estratificada com 10 *folds*, evidenciando a limitação do modelo na classificação de sentenças ofensivas, principalmente no que diz respeito à precisão.

Por fim, nesta categoria de algoritmos, foi implementado o modelo FastText, uma biblioteca de código aberto amplamente utilizada para aprendizado de representações textuais e construção de classificadores de texto. O modelo foi treinado com 20 épocas, taxa de aprendizado de 0,4, n-gramas de tamanho 4 e uma janela de contexto de 7 palavras. Os resultados obtidos superaram os demais algoritmos convencionais em todas as métricas, com destaque para a revocação, que alcançou 95,8%.

A técnica de TF-IDF (Term Frequency-Inverse Document Frequency) foi empregada como representação vetorial dos textos exclusivamente no modelo Bagging. Os hiperparâmetros definidos incluíram o uso de term frequency com escala sublinear e a suavização da frequência dos documentos (smooth IDF), o que contribuiu para uma melhor distribuição dos pesos das palavras nesse contexto. Além disso, a tokenização foi realizada utilizando o TweetTokenizer, e o pré-processamento dos textos incluiu lemmatização para normalizar as palavras. Essas configurações proporcionaram uma abordagem padronizada e consistente na análise dos dados, alinhando estes modelos com os principais algoritmos testados neste estudo.

4.5.1.1 Modelos Ensemble

Os modelos *Ensemble* foram treinados e testados por meio do JupyterLab, uma versão moderna do Jupyter Notebook. Este ambiente de desenvolvimento oferece recursos que simplificam a busca pelos melhores hiperparâmetros, possibilitando assim encontrar modelos de classificação com desempenhos superiores. Para o desenvolvimento dos modelos, foi utilizado a biblioteca Scikit-learn em Python, que é especificamente projetada para AM e de código aberto. Ela é construída sobre pacotes como NumPy, SciPy e Matplotlib, e oferece ferramentas simples e eficientes para análise preditiva de dados.

O método Bagging foi implementado utilizando 10 estimadores Extra-TreesClassifier e avaliado por meio de validação cruzada estratificada com 10 folds. Essa abordagem garante que a proporção de cada classe seja aproximadamente a mesma em cada fold, proporcionando uma avaliação mais robusta e representativa do modelo. Em cada iteração, as métricas de desempenho foram calculadas no conjunto de validação. A média dessas métricas resultou em uma acurácia de 92,95% e um recall de 90,51%. Além disso, os resultados obtidos no conjunto de teste foram ligeiramente superiores, apresentando uma acurácia de 93,20%, recall de 91,32%, precisão de 91,5% e medida-F de 91,5%. Esses resultados indicam um desempenho consistente do modelo tanto na validação cruzada quanto na avaliação final.

O modelo Bagging, configurado conforme descrito neste capítulo, apresentou um desempenho sólido em todas as métricas avaliadas, destacando-se especialmente em acurácia 93,2% e precisão 91,7%. Esses resultados superaram os obtidos pelos demais modelos ensemble testados nos experimentos, tanto em termos de métricas de desempenho quanto em eficiência computacional.

4.5.2 Modelos de Aprendizado Profundo

Para avaliar a eficácia de um modelo baseado em redes neurais profundas (DNN), foi implementada uma arquitetura sequencial composta por múltiplas camadas densas. O modelo utiliza um vetor de características gerado a partir da representação TF-IDF dos textos, permitindo capturar a importância relativa dos termos presentes nos tweets. A arquitetura definida inclui três camadas ocultas densas com 256, 128 e 64 neurônios, respectivamente, todas ativadas pela função ReLU. A camada de saída, responsável pela decisão binária, adota a ativação sigmoide para mapear as probabilidades de cada instância pertencer à classe positiva. O treinamento do modelo foi realizado com o otimizador Adam e a função de perda “binary_crossentropy”, garantindo a adaptação eficiente dos pesos da rede. Para garantir maior confiabilidade nos resultados, foi adotada a estratégia de validação cruzada estratificada em 10 *folds*, além de uma avaliação final em um conjunto de testes previamente separado.

Os resultados obtidos evidenciam a capacidade da DNN em identificar tweets agressivos com um desempenho competitivo. O modelo alcançou bons resultados, com uma acurácia média de 90,4%, refletindo um alto percentual de previsões corretas. Além disso, a revocação foi de 96,4%, indicando uma excelente capacidade de identificar verdadeiros positivos. A média harmônica entre precisão e revocação também alcançou resultados satisfatórios, 89% de medida-F. Por outro lado, a precisão de 82,7% sugere que o modelo tem maior tendência a classificar falsos positivos, o que pode impactar sua especificidade. De modo geral, a DNN demonstrou ser uma abordagem viável para a detecção de sentenças agressivas, destacando-se pelo tempo de treinamento reduzido em comparação com outros modelos avaliados.

4.5.3 Modelos Transformers

O modelo BERT utilizado nesta pesquisa é a variante BERT-Base uncased, um modelo transformer pré-treinado da família BERT projetado para lidar com tarefas de classificação de sequências. Esta versão do BERT foi selecionada por ser amplamente validada na literatura e apresentar um equilíbrio eficaz entre desempenho e custo computacional. Com 12 camadas, 768 unidades ocultas e 12 cabeças de atenção, totalizando cerca de 110 milhões de parâmetros, ela oferece uma arquitetura robusta para tarefas de NLP. Além disso, seu pré-treinamento em um grande corpus da Wikipédia e do BookCorpus confere versatilidade para tarefas de *downstream*, inclusive em domínios mais restritos, como a detecção de agressividade em redes sociais. Essa escolha também favorece a reprodutibilidade dos experimentos, por se tratar de uma configuração padrão amplamente adotada em trabalhos acadêmicos. O modelo foi ajustado para classificação binária de tweets, distinguindo entre conteúdo agressivo e neutro. O processo de fine-tuning envolveu o treinamento com dados rotulados de tweets, com o modelo otimizado por meio do otimizador AdamW, utilizando uma taxa de aprendizado de $5e-5$ e um epsilon de $1e-8$.

A fim de determinar a configuração mais adequada para o modelo BERT, diversas abordagens e parâmetros foram testados, priorizando o desempenho alcançado sem levar em consideração o tempo de treinamento. Como resultado, o modelo que obteve as melhores métricas de acurácia e revocação apresentou um alto custo computacional, demandando aproximadamente 7 horas e 48 minutos para a execução de 20 épocas em cada um dos dez folds da validação cruzada. Além disso, a escolha dos hiperparâmetros teve um papel crucial na eficiência do treinamento, especialmente o "Batch Size", que define o número de amostras processadas antes da atualização do modelo, e o "Max Length", que determina o tamanho máximo das sequências de entrada consideradas.

Para avaliar a robustez dos resultados obtidos com o modelo BERT pré-treinado utilizando o dataset DataTurks (2018), foi realizado uma comparação com um segundo dataset amplamente utilizado em pesquisas sobre discurso ofensivo, o dataset criado por Davidson et al. (2017). Para garantir a consistência da análise, preparamos o dataset de Davidson

removendo todas as instâncias classificadas como discurso de ódio, rótulo 0, e reclassificamos os casos rotulados como "neither", rótulo 2, para o rótulo 0. Esse ajuste tornou as classes comparáveis às do dataset Cyber-Troll, permitindo aplicar o modelo treinado no Cyber-Troll para prever sentenças ofensivas no dataset de Davidson e facilitando a comparação direta dos resultados.

Enquanto o modelo BERT alcançou altos índices de revocação e precisão no dataset Cyber-Troll, seu desempenho diminuiu quando testado no dataset de Davidson. Especificamente, a revocação caiu para 77,64% e a acurácia para 75,62%. No entanto, o F1 score permaneceu relativamente estável em 83,96%, principalmente devido à taxa de precisão de 91,39%.

Esses resultados representam uma ameaça significativa à validade dos achados desta pesquisa: a variabilidade no desempenho do modelo entre diferentes datasets dentro do mesmo domínio. As diferenças observadas entre os datasets Cyber-Troll e Davidson sugerem que, embora o modelo BERT seja poderoso, ele pode não se generalizar igualmente bem em datasets que apresentam variações nos critérios de rotulagem, distribuição de dados ou características de conteúdo, apesar de serem semelhantes em sua natureza.

Outro modelo transformer avaliado foi o DistilBERT, uma alternativa ideal para cenários com recursos computacionais limitados, oferecendo um equilíbrio entre desempenho e eficiência. Durante o experimento realizado, o DistilBERT completou o treinamento em um tempo total de 15 minutos e 27 segundos, significativamente menor em comparação ao tempo necessário para o treinamento do modelo BERT. No entanto, os resultados obtidos pelo DistilBERT foram inferiores aos do BERT, conforme mostrado na Tabela 6, que detalha a comparação dos resultados entre os modelos transformers.

Tabela 6 – Comparativo entre os modelos transformers

Modelos	**P**	**R**	**F-1**	**Ac**	**Validação Cruzada**
BERT	89,6%	96,5%	93,0%	94,0%	10-fold
DistilBERT	64,8%	48,0%	55,1%	61,0%	10-fold

Precisão (P), Revocação (R), Medida-F (F-1), Acurácia (Ac)

4.6 Discussão dos Resultados

Neste capítulo, foram abordadas diversos algoritmos e métodos para a tarefa de classificação de sentenças agressivas: classificadores convencionais como Naive Bayes, Redes Neurais e Regressão Logística; Redes Neurais Profundas; métodos ensemble, como Bagging; e modelos transformers. Além disso, foram conduzidos testes envolvendo diferentes datasets a fim de validar os resultados obtidos. Adicionalmente, todas as análises e

experimentos foram orquestrados com o auxílio do JupyterLab, um ambiente de desenvolvimento integrado para análise de dados interativa, onde foram avaliados e testados todos os modelos utilizados neste trabalho.

Os resultados obtidos com o modelo BERT foram particularmente expressivos, especialmente em termos de revocação, na qual alcançou o maior valor entre todos os modelos avaliados, superando tanto o FastText quanto a abordagem baseada em Bagging, conforme apresentado na Tabela 7. No entanto, o BERT apresentou um desempenho inferior em precisão, sugerindo dificuldades na correta classificação de tweets neutros, o que resultou em um maior número de falsos positivos.

Tabela 7 – Comparação das Métricas de Desempenho dos Modelos Mais Eficientes

Algoritmos	**P**	**R**	**F-1**	**Ac**	**Validação Cruzada**
BERT	89,6%	96,5%	93,0%	94,0%	10-fold
Bagging	91,7%	91,3%	91,5%	93,2%	10-fold
FastText	94.5%	95.8%	95.2%	94.4%	10-fold
DNN	82,7%	96,4%	89,0%	90,4%	10-fold

Precisão (P), Revocação (R), Medida-F (F-1), Acurácia (Ac)

Apesar de seus resultados impressionantes, o modelo BERT impõe um alto custo computacional durante o treinamento, sendo o tempo de treinamento amplamente dependente da configuração de seus parâmetros, como o número de épocas, o comprimento máximo da sequência e o tamanho do lote. Esses fatores influenciam diretamente o tempo de processamento e o consumo de recursos computacionais. Reduzir o tempo de treinamento sem comprometer o desempenho do modelo é um desafio, pois ajustes inadequados nesses parâmetros podem resultar em perda de precisão e revocação, impactando negativamente a qualidade da previsão. Isso é demonstrado na Tabela 8, que mostra que, embora dois modelos tenham o mesmo número de épocas, a variação no parâmetro Max Length resultou em uma duplicação do tempo de processamento de um modelo para o outro, embora os resultados tenham sido quase idênticos.

Tabela 8 – Comparação entre modelos BERT com diferentes configurações

Épocas	**Max Length**	**Acc**	**F1**	**Training Time**
3	64	93.75%	92.68%	1h11m
10	64	93.95%	92.91%	3h59m
20	64	94.05%	93.01%	7h48m
20	128	93.25%	92.14%	13h19m

O modelo FastText demonstrou um desempenho robusto, superando o modelo de bagging em todas as métricas avaliadas. Esse resultado indica que, apesar de sua simplicidade em comparação com o BERT, o FastText pode fornecer resultados competitivos, especialmente em contextos nos quais a velocidade de treinamento e a eficiência computacional são fatores críticos.

Em relação ao modelo FastText, a criação de um conjunto de previsões separado foi essencial. O modelo foi avaliado usando validação cruzada de 10 partes, sendo o conjunto de dados dividido em 10 partes, com 9 partes usadas para treinamento e 1 para teste, alternando a parte de teste ao final de cada ciclo de treinamento. No entanto, ao avaliar o desempenho do modelo em cada parte usando o método "test" do FastText, observou-se que as métricas de acurácia e precisão permaneceram idênticas, independentemente do "fold" avaliado. Esse padrão também foi observado nos estudos de Bhattacharjee (2018) e Qiu et al. (2020). Devido a esse comportamento inesperado, foi utilizado um método que retorna a pontuação de precisão e revocação para cada rótulo. Para garantir a confiabilidade do modelo, o conjunto de testes separado foi vital para validar os resultados obtidos durante as fases de treinamento e validação, onde resultados semelhantes foram alcançados em ambas as fases.

4.7 Ameaças à Validade

Esta seção discute possíveis ameaças à validade da pesquisa, com foco em fatores que podem comprometer a confiabilidade, a capacidade de generalização e a robustez dos resultados. O estudo investiga a classificação de *tweets* agressivos por meio de algoritmos de aprendizado de máquina, com ênfase em modelos do tipo *transformer*, como o BERT. Foram utilizados dois conjuntos de dados distintos: o Cyber-Troll, empregado para treinamento e predição, e o conjunto de dados desenvolvido por Davidson et al. (2017), utilizado para avaliar a capacidade de generalização e realizar a comparação de desempenho entre os modelos.

4.7.1 Validade de Conclusão

A Validade de Conclusão (*Conclusion Validity*) é fundamental para garantir que as inferências extraídas dos resultados sejam estatisticamente robustas e confiáveis. Neste estudo, foram identificadas algumas limitações, como diferenças estruturais entre os conjuntos de dados utilizados, o que pode reduzir a capacidade de generalização dos modelos e impactar negativamente a confiabilidade dos resultados obtidos. Além disso, a reprodutibilidade dos experimentos é outro fator essencial, uma vez que é importante que outros pesquisadores consigam replicar os estudos e obter resultados consistentes. Essas ameaças foram mitigadas por meio da implementação de uma metodologia detalhada e da reestruturação do conjunto de dados de Davidson et al. (2017), conforme descrito neste capítulo.

4.7.2 Validade Interna

O desequilíbrio significativo entre as classes no conjunto de dados Cyber-Troll representa uma ameaça à validade interna (*Internal Validity*), uma vez que pode enviesar o modelo em favor da classe majoritária. Para mitigar parcialmente esse problema, foram empregadas técnicas de validação cruzada com 10 *folds*, contribuindo para uma avaliação mais robusta do desempenho do modelo. Além disso, considerando a possibilidade de que a configuração dos hiperparâmetros dos modelos possa influenciar os resultados, levando potencialmente a *underfitting* ou *overfitting*, um segundo conjunto de dados foi utilizado. A análise comparativa com esse segundo conjunto permite avaliar a capacidade de generalização do modelo e validar os resultados obtidos.

4.7.3 Validade de Construção

No contexto da classificação de tweets agressivos, é essencial que os rótulos atribuídos pelos anotadores estejam alinhados de forma consistente com as definições teóricas de agressividade. Variações na interpretação do que constitui um comportamento agressivo ou ofensivo podem introduzir vieses nos dados rotulados, comprometendo a capacidade dos modelos de capturar com precisão esses conceitos. Ademais, a escolha das métricas de avaliação é crítica para a validade de construção (*Construct Validity*), pois essas métricas devem refletir adequadamente os objetivos da pesquisa. Para mitigar essas ameaças, foram utilizados conjuntos de dados amplamente reconhecidos para a detecção de tweets agressivos e adotadas métricas de avaliação amplamente aceitas na literatura. Além disso, entre as métricas empregadas, identificamos e priorizamos aquelas mais adequadas com base nos requisitos específicos deste problema de pesquisa.

4.7.4 Validade Externa

No contexto da detecção de comentários ofensivos por meio de algoritmos de aprendizado de máquina, a validade externa (*External Validity*) refere-se à capacidade de generalizar os resultados obtidos para diferentes contextos e conjuntos de dados. Nesse sentido, ao utilizar o modelo treinado e testado no conjunto de dados DataTurks (2018) para prever sentenças do conjunto de dados de Davidson et al. (2017), foram observados bons resultados nas métricas de precisão e medida-F. No entanto, os resultados de revocação e acurácia foram menos satisfatórios. Essa discrepância pode ser atribuída a diversos fatores. Primeiramente, diferenças nas características dos dados, como o comprimento médio dos tweets, o uso de gírias ou emojis e o estilo linguístico, podem dificultar a capacidade de generalização do modelo. Em segundo lugar, as origens dos tweets, incluindo seus contextos temporais, culturais ou temáticos, podem resultar em padrões divergentes que não são adequadamente capturados pelo modelo treinado. Além disso, as definições subjetivas dos anotadores sobre o que constitui um tweet ofensivo podem variar entre os

conjuntos de dados, levando a inconsistências no processo de rotulação e impactando o desempenho do modelo. Por fim, como os *datasets* utilizados pertencem a um mesmo idioma, os modelos desenvolvidos ficam limitados à predição de sentenças nesse idioma específico, dificultando sua generalização para outras línguas.

Para lidar com essas ameaças, pesquisas futuras devem se concentrar na diversificação dos conjuntos de dados utilizados, incluindo a criação de dados em outros idiomas, como o português, assegurando que os modelos sejam treinados e avaliados com dados que apresentem características variadas. Além disso, a aplicação de técnicas de regularização ou a combinação de múltiplos conjuntos de dados durante o treinamento pode aumentar a robustez e a capacidade de generalização dos modelos. Essas estratégias podem contribuir para melhorar o desempenho dos modelos em diferentes contextos e ajudar a mitigar o risco de overfitting a conjuntos de dados específicos. Embora essas abordagens apresentem potencial para melhorias, pesquisas e experimentações adicionais serão necessárias para avaliar sua eficácia na prática.

4.8 Considerações Finais

Neste capítulo, foi proposta uma metodologia para a classificação de sentenças agressivas, com a definição de um baseline estabelecido pelo método Bagging, que, embora não seja uma técnica recente, alcançou um desempenho superior a 90% em todas as métricas avaliadas. Esses resultados são competitivos em comparação com os trabalhos relacionados ao tema de detecção de conteúdo agressivo e inadequado. De modo geral, o modelo FastText apresentou as melhores métricas de desempenho, além de um tempo de treinamento mais rápido.

No entanto, para minimizar os falsos negativos — casos em que sentenças verdadeiramente agressivas são classificadas erroneamente como apropriadas —, o modelo BERT destacou-se por alcançar a melhor taxa de revocação entre todos os modelos e métodos empregados neste estudo. Por outro lado, o modelo DNN, embora não tenha apresentado resultados competitivos em relação aos demais modelos, obteve uma alta taxa de revocação, sendo o segundo modelo a superar 96% nesta métrica. Contudo, devido à vasta gama de configurações disponíveis para uma rede neural profunda, é necessário o uso de técnicas como Grid Search ou Random Search para ajuste de hiperparâmetros, o que pode resultar em um aumento considerável no tempo de treinamento.

Desse modo, considerando a hipótese apresentada nesta dissertação "A Inteligência artificial e seus algoritmos potencializam a detecção automática de mensagens ofensivas em textos curtos e ruidosos, combatendo cyberbullying e outras ameaças de forma eficiente" entende-se que os resultados obtidos corroboram tal hipótese. Entretanto, estudos futuros poderão ser realizados com o objetivo de buscar resultados ainda mais consistentes, especialmente em relação à métrica de revocação.

Capítulo 5

Conclusão

Este estudo explorou o uso do modelo transformer BERT para a classificação de tweets ofensivos. Foi implementado um *framework* para manipulação e preparação de dados, com o objetivo de melhorar a capacidade do modelo em prever sentenças ofensivas e evitar o *overfitting*. Um dos passos mais relevantes foi a validação cruzada k-fold estratificada, que não apenas fornece uma estimativa de desempenho mais robusta, mas também ajuda a lidar com conjuntos de dados desbalanceados. Apesar de a tokenização e o pré-treinamento serem intrínsecos ao BERT, foram realizadas tokenização e lematização adicionais, o que impactou positivamente o desempenho do modelo. Os experimentos demonstraram que o BERT superou o método baseline em todas as métricas. No entanto, em comparação ao modelo FastText, o BERT se destacou apenas em revocação. Embora as métricas de ambos os modelos sejam semelhantes, o BERT apresentou um desempenho inferior em precisão, indicando desafios em lidar com classes neutras e uma maior taxa de falsos positivos. Além disso, o tempo de treinamento do BERT foi significativamente maior.

5.1 Principais Contribuições

Esta pesquisa de mestrado apresentou soluções baseadas em aprendizado de máquina, aprendizado profundo e processamento de linguagem natural, com contribuições para a identificação de mensagens ofensivas em textos curtos e ruidosos. Os modelos BERT e FastText se destacaram, alcançando as melhores métricas entre os métodos explorados. No entanto, o modelo BERT não demonstrou desempenho superior ao FastText em todas as métricas, exceto na revocação, considerada a mais importante para a tarefa alvo desta pesquisa. Métodos como o Bagging também foram avaliados e apresentaram resultados consistentes em todas as métricas, mesmo diante do desbalanceamento de classes no da-

taset utilizado. A aplicação de validação cruzada k-fold mostrou-se eficaz na mitigação do overfitting e na melhoria da capacidade de generalização dos modelos, proporcionando maior consistência nos resultados. Por fim, foi realizada uma comparação entre diferentes modelos transformers, evidenciando ainda mais o potencial do BERT na detecção de conteúdos ofensivos.

5.2 Trabalhos Futuros

Dado os desafios do modelo BERT em termos de precisão, uma estratégia potencial para a moderação de conteúdo seria adotar uma abordagem híbrida, combinando o BERT com um modelo mais focado em precisão, como o FastText. Essa abordagem permitiria ao sistema aproveitar a alta revocação do BERT para capturar uma ampla gama de conteúdos potencialmente ofensivos, enquanto o FastText atuaria como um filtro secundário para aumentar a precisão e reduzir a probabilidade de falsos positivos, evitando que conteúdos não ofensivos sejam incorretamente sinalizados. Adicionalmente, poderia ser incorporada uma camada de revisão manual para as classificações de baixa confiança, mitigando ainda mais o risco de erros. Nos casos em que o BERT identificasse o conteúdo como ofensivo, mas com baixa confiança, moderadores poderiam revisar esses casos antes que qualquer ação fosse tomada, minimizando falhas e garantindo uma abordagem eficaz para a moderação de conteúdos ofensivos.

Outra abordagem seria a aplicação da técnica de *early stopping*, que monitora a perda de validação durante o treinamento e interrompe o processo quando não houver melhora após um número predefinido de épocas consecutivas. Essa estratégia visa evitar o *overfitting* e garantir que o modelo mantenha uma boa capacidade de generalização. Além disso, considerando o tempo elevado de treinamento do modelo BERT, salvar a versão com o melhor desempenho em cada fold pode ser uma prática vantajosa, evitando a perda de informações críticas e facilitando a recuperação das melhores iterações do treinamento.

Adicionalmente, aprimorar o modelo com técnicas de análise de sentimentos poderia fornecer um contexto adicional, ajudando o BERT a diferenciar melhor entre conteúdos verdadeiramente ofensivos e expressões neutras ou não ofensivas. Ao integrar a análise de sentimentos no pipeline de classificação, o modelo pode se tornar mais criterioso, melhorando sua precisão sem sacrificar a revocação.

Acreditamos também que a criação de um novo dataset em português, contendo sentenças ofensivas extraídas de mídias sociais, pode contribuir significativamente para o avanço das pesquisas na área de detecção automática de linguagem agressiva. Além de suprir a escassez de dados rotulados nesse idioma, tal iniciativa pode ampliar a aplicabilidade dos modelos desenvolvidos, permitindo sua adaptação a diferentes contextos

linguísticos e culturais. Essa abordagem também pode ajudar a mitigar os impactos causados por trolls e outros usuários mal-intencionados, fortalecendo as ferramentas de moderação automática em ambientes digitais.

Para trabalhos futuros, acreditamos que a incorporação de técnicas de extração de características, como a análise de sentimento, pode aumentar a eficácia dos modelos transformers pré-treinados durante o fine-tuning. Além disso, a exploração de modelos leves, como o ALBERT, ou de modelos mais robustos e refinados, como o RoBERTa, pode gerar resultados promissores. Esses ganhos podem ser potencializados com a aplicação de técnicas como early stopping e data augmentation, que ajudam a evitar o overfitting e a melhorar a generalização do modelo.

Referências

ALQAHTANI, A. F.; ILYAS, M. Using deep learning lstm and cnn with word embedding for the detection of offensive text on twitter. 2023.

_____. Using deep learning lstm and cnn with word embedding for the detection of offensive text on twitter. **Journal of Social Media and Information Technologies**, 2023.

ANTONSICH, M. et al. **International Encyclopedia of Human Geography**. 2. ed. [S.l.]: Elsevier, 2020. ISBN 9780081022962.

BARTOSIK, A.; WHITTINGHAM, H. Chapter 7 - evaluating safety and toxicity. In: ASHENDEN, S. K. (Ed.). **The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry**. [S.l.]: Academic Press, 2021. p. 119–137. ISBN 978-0-12-820045-2.

BHATTACHARJEE, J. **fastText Quick Start Guide: Get started with Facebook's library for text representation and classification**. [S.l.]: Packt Publishing Ltd, 2018. 33 p.

BISHOP, J. The effect of de-individuation of the internet troller on criminal procedure implementation: An interview with a hater. **International journal of cyber criminology**, v. 7, n. 1, 2013.

BOUCHEFRY, K. E.; SOUZA, R. S. de. Learning in big data: Introduction to machine learning. In: **Knowledge discovery in big data from astronomy and earth observation**. [S.l.]: Elsevier, 2020. p. 225–249.

BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, p. 123–140, 1996.

BUCKELS, E. E.; TRAPNELL, P. D.; PAULHUS, D. L. Trolls just want to have fun. **Personality and individual Differences**, Elsevier, v. 67, p. 97–102, 2014.

CALDEIRA, A. M.; SOUZA, R. C.; MACHADO, M. A. S. Identificação automática das ordens dos modelos garch utilizando redes neurais. **Engevista**, 2009.

CAPISTRANO, J. L. C.; SUAREZ, J. J. P.; JR, P. C. N. Salsa: detection of cybertrolls using sentiment, aggression, lexical and syntactic analysis of tweets. In: **Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics**. [S.l.: s.n.], 2019. p. 1–6.

CUI, S. et al. A stacking-based ensemble learning method for earthquake casualty prediction. **Applied Soft Computing**, v. 101, p. 107038, 2021. ISSN 1568-4946.

DataTurks. **Tweets Dataset for Detection**. 2018. <https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls>. Accessed on 18 July 2024.

DAVIDSON, T. et al. Automated hate speech detection and the problem of offensive language. In: **Proceedings of the 11th International AAAI Conference on Web and Social Media**. [S.l.: s.n.], 2017. (ICWSM '17), p. 512–515.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: **North American Chapter of the Association for Computational Linguistics**. [s.n.], 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:52967399>.

DUKART, J. Basic concepts of image classification algorithms applied to study neurodegenerative diseases. In: TOGA, A. W. (Ed.). **Brain Mapping**. Waltham: Academic Press, 2015. p. 641–646.

GILLESPIE, T. Content moderation, ai, and the question of scale: Big data & society. **Stigma: Notes on the Management of Spoiled Identity**, 2020.

GORA, P. et al. On a road to optimal fleet routing algorithms: a gentle introduction to the state-of-the-art. In: **Smart Delivery Systems**. [S.l.]: Elsevier, 2020. p. 37–92.

HARDAKER, C. Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions. Walter de Gruyter GmbH & Co. KG, 2010.

HEMMATIAN, F.; SOHRABI, M. K. A survey on classification techniques for opinion mining and sentiment analysis. **Artificial intelligence review**, Springer, v. 52, n. 3, p. 1495–1545, 2019.

JACKSON, P.; MOULINIER, I. Natural language processing for online applications: Text retrieval, extraction and categorization. **John Benjamins Publishing Company**, p. 225, 2002.

JANECZKO, B.; SRIVASTAVA, G. The use of deep learning in image analysis for the study of oncology. In: **Internet of Multimedia Things (IoMT)**. [S.l.]: Elsevier, 2022. p. 133–150.

JURAFSKY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 3. ed. [S.l.: s.n.], 2020. 84 p.

KADAM, S. H.; PANISKAKI, K. **Text analysis for email multi label classification**. Dissertação (Mestrado) — UNIVERSITY OF GOTHENBURG, Gothenburg, Sweden, 2020.

KARLEKAR, S.; BANSAL, M. Safecity: Understanding diverse forms of sexual harassment personal stories. **arXiv preprint arXiv:1809.04739**, 2018.

KHAN, U. et al. Aggression detection in social media from textual data using deep learning models. **Applied Sciences**, MDPI, v. 12, n. 10, p. 5083, 2022.

KHURANA, D. et al. Natural language processing: State of the art, current trends and challenges. **Multimedia Tools and Applications**, Springer, p. 1–32, 2022.

KUMARI, M.; JAIN, A.; BHATIA, A. Synonyms based term weighting scheme: An extension to tf.idf. **Procedia Computer Science**, v. 89, p. 555–561, 2016. ISSN 1877-0509.

LIU, P. et al. Forecasting the presence and intensity of hostility on instagram using linguistic and social features. **Twelfth international aaai conference on web and social media**, v. 59, 2018.

LUO, X. Efficient english text classification using selected machine learning techniques. **Alexandria Engineering Journal**, Elsevier, v. 60, n. 3, p. 3401–3409, 2021.

MANNING, C. D. **Foundations of statistical natural language processing**. [S.l.]: The MIT Press, 1999.

MCCUE, C. Identification, characterization, and modeling. In: MCCUE, C. (Ed.). **Data Mining and Predictive Analysis (Second Edition)**. Second edition. Boston: Butterworth-Heinemann, 2015. p. 385–393. ISBN 978-0-12-800229-2.

MINKU, L. L. The wisdom of the crowds in predictive modeling for software engineering. In: MENZIES, T.; WILLIAMS, L.; ZIMMERMANN, T. (Ed.). **Perspectives on Data Science for Software Engineering**. Boston: Morgan Kaufmann, 2016. p. 199–204. ISBN 978-0-12-804206-9.

MISRA, S.; LI, H. Chapter 9 - noninvasive fracture characterization based on the classification of sonic wave travel times. In: MISRA, S.; LI, H.; HE, J. (Ed.). **Machine Learning for Subsurface Characterization**. [S.l.]: Gulf Professional Publishing, 2020. p. 243–287.

MORÁN-FERNÁNDEZ, L.; BÓLON-CANEDO, V.; ALONSO-BETANZOS, A. How important is data quality? best classifiers vs best features. **Neurocomputing**, Elsevier, v. 470, p. 365–375, 2022.

NASSER, N. et al. n-gram based language processing using twitter dataset to identify covid-19 patients. **Sustainable Cities and Society**, v. 72, p. 103048, 2021. ISSN 2210-6707.

NETOFF, T. I. The ability to predict seizure onset. In: IAIZZO, P. A. (Ed.). **Engineering in Medicine**. [S.l.]: Academic Press, 2019. p. 365–378.

PARNELL, A. C. et al. Machine learning techniques for the detection of inappropriate erotic content in text. **International Journal of Computational Intelligence Systems**, Atlantis Press, v. 13, n. 1, p. 591, 2020.

PATRA, G. et al. Deep learning methods for scientific and industrial research. **Handbook of Statistics**, 2023.

PELLE, R. P. d. **Identificação de comentários ofensivos na web**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, 2019.

PERCONTI, P.; PLEBE, A. Deep learning and cognitive science. **Cognition**, Elsevier, v. 203, p. 104365, 2020.

PRADHAN, N. et al. Diabetes prediction using artificial neural network. In: **Deep Learning Techniques for Biomedical and Health Informatics**. [S.l.]: Academic Press, 2020. p. 327–339.

PRUSTY, S.; PATNAIK, S.; DASH, S. K. Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer. **Frontiers in Nanotechnology**, Frontiers Media SA, v. 4, p. 972421, 2022.

QIU, M. et al. Convolutional-neural-network-based multilabel text classification for automatic discrimination of legal documents. **Sensors & Materials**, v. 32, 2020.

RAHMANI, A. et al. Machine learning (ml) in medicine: Review, applications, and challenges. **Mathematics**, v. 9, 11 2021.

REINDERS, C. et al. Learning convolutional neural networks for object detection with very little training data. In: YANG, M. Y.; ROSENHAHN, B.; MURINO, V. (Ed.). **Multimodal Scene Understanding**. [S.l.]: Academic Press, 2019. p. 65–100. ISBN 978-0-12-817358-9.

SADIQ, S. et al. Aggression detection through deep neural model on twitter. **Future Generation Computer Systems**, Elsevier, v. 114, p. 120–129, 2021.

SAIF, H.; FERNANDEZ, M.; ALANI, H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. **Proceedings of the 9th International Language Resources and Evaluation Conference (LREC'14)**, p. 810–817, 01 2014.

SALMINEN, J. et al. Developing an online hate classifier for multiple social media platforms. **Hum.-Centric Comput. Inf. Sci.**, Springer-Verlag, Berlin, Heidelberg, v. 10, n. 1, jan 2020. ISSN 2192-1962.

SASADA, T. et al. A resampling method for imbalanced datasets considering noise and overlap. **Procedia Computer Science**, v. 176, p. 420–429, 2020. ISSN 1877-0509. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

SHI, H. et al. Resampling algorithms based on sample concatenation for imbalance learning. **Knowledge-Based Systems**, v. 245, p. 108592, 2022. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705122002659>.

SIKDER, M. N. K.; BATARSEH, F. A. Outlier detection using ai: a survey. **AI Assurance**, Elsevier, p. 231–291, 2023.

SIMSKE, S. **Meta-Analytics: Consensus Approaches and System Patterns for Data Analysis**. Elsevier Science, 2019. ISBN 9780128146231. Disponível em: <https://books.google.com.br/books?id=M2s3vwEACAAJ>.

SINCLAIR, J. **Corpus Concordance Collocation**. [S.l.]: OUP, 1991. 171 p.

SONG, L. et al. Bagging-based system combination for domain adaption. In: **MTSUMMIT**. [S.l.: s.n.], 2011.

TOLBA, M.; OUADFEL, S.; MESHOUL, S. Hybrid ensemble approaches to online harassment detection in highly imbalanced data. **Expert Systems with Applications**, v. 175, p. 114751, 2021. ISSN 0957-4174.

VARGAS, F. A. et al. Hatebr: A large expert annotated corpus of brazilian instagram comments for offensive language and hate speech detection. **arXiv e-prints**, p. arXiv–2103, 2021.

VARSHNEY, K. R. **Trustworthy Machine Learning**. Chappaqua, NY, USA: Independently Published, 2022.

VASANTHARAJAN, C.; THAYASIVAM, U. Towards offensive language identification for tamil code-mixed youtube comments and posts. **SN Computer Science**, Springer Science and Business Media LLC, v. 3, n. 1, dez. 2021. ISSN 2661-8907. Disponível em: <http://dx.doi.org/10.1007/s42979-021-00977-y>.

VASWANI, A. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017.

WACHS, S. et al. Online correlates of cyberhate involvement among young people from ten european countries: An application of the routine activity and problem behaviour theory. **Computers in Human Behavior**, v. 123, 2021.

WANG, L. et al. Chapter one - a deep-forest based approach for detecting fraudulent online transaction. In: HURSON, A. R.; WU, S. (Ed.). **AI and Cloud Computing**. [S.l.]: Elsevier, 2021, (Advances in Computers, v. 120). p. 1–38.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big data**, SpringerOpen, v. 3, n. 1, p. 1–40, 2016.

WITTEN, I.; FRANK, E.; HALL, M. **Data Mining: Practical Machine Learning Tools and Techniques**. Elsevier Science, 2011. (The Morgan Kaufmann Series in Data Management Systems). ISBN 9780080890364. Disponível em: <https://books.google.com.br/books?id=bDtLM8CODsQC>.

WOOLF, B. **Building Intelligent Interactive Tutors, Student-Centered Strategies for Revolutionizing E-Learning**. [S.l.]: Elsevier & Morgan Kaufmann, 2008.

YANG, J. et al. Social media data analytics for business decision making system to competitive analysis. **Information Processing & Management**, v. 59, 2022.

YANG, L. Classifiers selection for ensemble learning based on accuracy and diversity. **Procedia Engineering**, v. 15, p. 4266–4270, 2011. ISSN 1877-7058. CEIS 2011.

YAO, J. et al. Applications of stacking/blending ensemble learning approaches for evaluating flash flood susceptibility. **International Journal of Applied Earth Observation and Geoinformation**, v. 112, p. 102932, 2022. ISSN 1569-8432.

YAO, T.; ZHAI, Z.; GAO, B. Text classification model based on fasttext. In: **2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)**. [S.l.: s.n.], 2020. p. 154–157.