



Instituto Brasileiro de Geografia e Estatística
Escola Nacional de Ciências Estatísticas
Bacharelado em Estatística



Classificação de Persuasão em Textos de Memes: Uma Abordagem Baseada em Engenharia de Características

Leonardo Souza Gomes de Lima

Rio de Janeiro

2024

Leonardo Souza Gomes de Lima

Classificação de Persuasão em Textos de Memes: Uma Abordagem Baseada em Engenharia de Características

Monografia apresentada à Escola Nacional de Ciências Estatísticas do Instituto Brasileiro de Geografia e Estatística, como requisito parcial à obtenção do título de Bacharel em Estatística.

Orientador: Prof Eduardo Corrêa Gonçalves

Rio de Janeiro

2024

Ao meu pai, com quem pude compartilhar o início dessa minha nova jornada acadêmica e que nos deixou durante meu período na ENCE. Momentos de vitória despertam fortemente as lembranças, mas levo comigo toda sua presença e companheirismo em minha vida, sabendo do potencial que ele sempre viu em mim. À minha mãe, que celebrou cada uma das minhas conquistas ao longo dos meus nove semestres de ENCE como se fossem as suas próprias, sempre acreditando no meu potencial para ir cada vez mais longe. Agora, também, nesse novo momento como bacharel em Estatística que se inicia. Sempre levarei comigo essas duas referências de vida, tanto no âmbito pessoal quanto profissional.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me proporcionar esta grande vitória em minha vida, concedendo-me força nos momentos difíceis e determinação para avançar nessa jornada que é a graduação em Estatística.

Aos meus pais e à minha irmã, que sempre me apoiaram em todos os momentos da minha vida e acreditaram em mim. O incentivo constante, apoio e exemplo de dedicação foram fundamentais para a minha trajetória.

Aos professores da ENCE, com quem tive o prazer de aprender essa incrível ciência, tão essencial em nossas vidas. Agradeço especialmente ao Professor Eduardo Corrêa, por sua notável orientação e apoio neste trabalho e durante nossa Iniciação Científica. Sua dedicação e conhecimento foram fundamentais para meu desenvolvimento acadêmico.

Aos meus amigos e amigas da ENCE, com quem tive a alegria de compartilhar esses quatro anos de graduação. Sabemos que não foi nada fácil, mas a rede de apoio que nós formamos foi fundamental para superarmos os desafios.

A todos, meu muito obrigado.

Onde há um desejo, há um caminho.

- Albert Einstein

RESUMO

A propaganda visa influenciar a opinião pública para promover uma agenda específica, incentivando opiniões contrárias ou a favor. Para isso, emprega técnicas de persuasão, utilizando uma comunicação direcionada a moldar a opinião de quem recebe a informação, podendo ser disseminada por diversos meios. Paralelamente, nas redes sociais, os memes se mostram como uma poderosa ferramenta comunicativa, capazes de se espalhar rapidamente e atingir um grande número de pessoas. Assim, os memes podem ser um veículo eficaz para a disseminação de propaganda. Na literatura, já existem estudos sobre a detecção computacional de propaganda em textos de artigos. No entanto, apenas recentemente os pesquisadores começaram a investigar a classificação de persuasão em memes, que representa ambiente mais desafiador. Este trabalho aborda a classificação de persuasão em textos de memes, utilizando algoritmos de aprendizado de máquina. A abordagem é baseada na engenharia de características, uma estratégia usada para transformar texto (dado não estruturado) em um formato adequado para os algoritmos de classificação. Serão utilizados modelos de árvore de decisão, floresta aleatória e regressão logística.

Palavras-chaves: Persuasão. Propaganda. Meme. Classificação. Engenharia de Características. Processamento de Linguagem Natural. Aprendizado de Máquina.

ABSTRACT

Propaganda aims to influence public opinion to promote a specific agenda, encouraging both opposing and supporting views. To achieve this objective, it uses persuasion techniques through targeted communication to shape the opinions of its audience, disseminated across various channels. Simultaneously, memes on social media emerge as a powerful communicative tool capable of rapid dissemination and reaching a vast audience. Thus, memes can serve as an effective vehicle for propagandistic spread. In scientific literature, computational studies already exist on detecting propaganda in article texts. However, researchers have only recently begun exploring the classification of persuasion in memes, which presents a more challenging environment. This work addresses the classification of persuasion in meme texts using machine learning algorithms. The approach is based on feature engineering, a strategy used to transform unstructured text into a suitable format for classification algorithms. Decision tree models, random forest, and logistic regression will be employed.

Key-words: Persuasion. Propaganda. Meme. Classification. Feature Engineering. Natural Language Processing. Machine Learning.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Exemplo de meme com persuasão | 13 |
| Figura 2 – Frequência das técnicas de persuasão na base de treino | 22 |
| Figura 3 – Exemplo de meme com a técnica Smears | 23 |
| Figura 4 – Distribuição do total de técnicas de persuasão na base de treino | 24 |
| Figura 5 – Nuvem de palavras da base de treino | 24 |
| Figura 6 – Nuvem de palavras (não-propaganda) | 25 |
| Figura 7 – Nuvem de palavras (propaganda) | 26 |
| Figura 8 – Exemplo de Árvore de Decisão | 34 |
| Figura 9 – Função sigmoide | 39 |
| Figura 10 – QR Code de acesso para o código Python | 59 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Matriz Documento-Termo | 32 |
| Tabela 2 – Base explicativa para Índice de Gini | 36 |
| Tabela 3 – Estrutura da Matriz de Confusão | 41 |
| Tabela 4 – Resultados do Experimento 1 – Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, sem utilizar TF-IDF | 43 |
| Tabela 5 – Resultados do Experimento 2 - Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, utilizando TF-IDF com os 50 bigramas mais frequentes | 44 |
| Tabela 6 – Resultados do Experimento 3 - Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, utilizando TF-IDF com os 50 unigramas mais frequentes | 45 |
| Tabela 7 – Resultados do Experimento 1 – Sem Balanceamento | 46 |
| Tabela 8 – Matriz de Confusão (Experimento 1 Sem Balanceamento – Regressão Logística) | 47 |
| Tabela 9 – Matriz de Confusão (Experimento 1 com Oversampling – Regressão Logística) | 47 |
| Tabela 10 – Estatísticas dos Atributos (textos persuasivos) | 50 |
| Tabela 11 – Estatísticas dos Atributos (textos não persuasivos) | 51 |

LISTA DE QUADROS

| | | |
|---|---------------------------------|----|
| 1 | Técnicas de Persuasão | 19 |
| 2 | Técnicas de Persuasão | 20 |
| 3 | Técnicas de Persuasão | 21 |

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 11 |
| 1.1 | Justificativa | 13 |
| 1.2 | Objetivos | 14 |
| 2 | REVISÃO BIBLIOGRÁFICA | 15 |
| 3 | BASE DE DADOS | 18 |
| 3.1 | Classificação das Técnicas de Persuasão | 18 |
| 3.2 | Análise Exploratória | 22 |
| 4 | METODOLOGIA | 27 |
| 4.1 | Pré-processamento | 28 |
| 4.2 | Engenharia de Características | 29 |
| 4.2.1 | TF-IDF | 31 |
| 4.3 | Balanceamento da Base | 32 |
| 4.4 | Técnicas de Classificação | 33 |
| 4.4.1 | Árvores de Decisão | 33 |
| 4.4.2 | Floresta Aleatória | 37 |
| 4.4.3 | Regressão Logística | 38 |
| 4.5 | Métricas de Avaliação | 40 |
| 5 | RESULTADOS | 42 |
| 5.1 | Resultados Gerais | 42 |
| 5.2 | Comparação com Modelos Gerados sem Balanceamento de Classes | 46 |
| 5.3 | Análise da Importância dos Atributos com Árvore de Decisão | 48 |
| 6 | CONCLUSÃO | 53 |
| | REFERÊNCIAS | 55 |
| | APÊNDICE A – NOTEBOOK GOOGLE COLABORATORY | 59 |

1 INTRODUÇÃO

Nos últimos anos, os memes emergiram como uma forma peculiar e poderosa de comunicação na era digital. Embora popularizados no ambiente online, é difícil precisar com rigor o momento em que conteúdos que circulam na internet passaram a ser reconhecidos como memes (CHAGAS, 2021). Sabe-se que nos anos 90, tornou-se corriqueiro traduzir memes como piadas e outras formas virais que ganhavam rápido alcance em fóruns de discussão *online* e *newgroups*. Na era da internet, esses elementos de mídia compartilhável têm a habilidade de se espalhar rapidamente pelo mundo através da interação social (WANG; WOOD, 2011) e, hoje, se encontram quase que onipresentes nas redes sociais. Desde um ambiente de humor até a manifestação política, sua natureza altamente compartilhável e adaptável os torna ideais para o ambiente dinâmico das redes sociais, se destacando pela sua capacidade de atrair e envolver as pessoas (CHAGAS, 2021).

De acordo com Merriam-Webster (2023), o fenômeno tem origem ainda na década de 70. A palavra "meme" foi cunhada em 1976 pelo renomado biólogo britânico Richard Dawkins em seu livro "*The Selfish Gene*". Dawkins introduziu o termo como uma unidade cultural que se replica e se transmite de pessoa para pessoa por meio da imitação, semelhante ao conceito de gene na biologia. Ele derivou a palavra "meme" do grego "mimema", que significa "algo imitado". Dawkins usou os memes como uma forma de explicar como ideias, comportamentos, rituais e outras formas de cultura são transmitidas e evoluem ao longo do tempo (DAWKINS, 1976). Segundo Dawkins, exemplos de memes são melodias, ideias, frases de efeito e até mesmo moda, argumentando que, assim como genes, os memes competem entre si pela atenção e reprodução, e aqueles que são mais adaptáveis e atraentes têm mais chances de se espalhar e sobreviver.

Desde então, o conceito de meme expandiu-se para além do domínio da biologia, tornando-se uma parte fundamental da cultura digital contemporânea, onde memes são compartilhados e adaptados em uma escala global. Em uma analogia com a biologia, memes podem ser unidades culturais que se propagam e se adaptam por meio do compartilhamento online. Plataformas como Instagram, Facebook, TikTok, Reddit e X (antigo Twitter) servem como espaços de criação, compartilhamento e discussão de memes.

No entanto, por trás de sua aparente trivialidade, os memes não apenas podem refletir os interesses e preocupações de uma geração conectada, mas também podem ser utilizados como campanhas de influência por parte de governos (ZAKEM; MCBRIDE; HAMMERBERG, 2021), o que pode impactar ativamente a maneira como percebemos e interpretamos o mundo ao nosso redor. Desde comentários sociais até críticas políticas, os memes se tornaram uma importante ferramenta para a expressão (MIHAILIDIS, 2020),

podendo assumir a capacidade de moldar opiniões.

Essa mesma capacidade dos memes de influenciar as percepções e opiniões pode ser manipulada para espalhar desinformação e propaganda, através de técnicas de persuasão. Propagandas possuem o objetivo de influenciar a opinião das pessoas com o propósito de expandir uma agenda específica (MARTINO et al., 2020). Com o rápido compartilhamento e disseminação dos memes nas redes sociais, tornou-se cada vez mais fácil para indivíduos e grupos com interesses específicos explorarem essa forma de comunicação para promover agendas políticas, disseminar teorias da conspiração e distorcer fatos. A natureza rápida e concisa dos memes pode simplificar questões complexas e levar a uma compreensão superficial ou distorcida de eventos e questões importantes. Além disso, a viralidade dos memes pode amplificar mensagens enganosas ou prejudiciais, aumentando sua influência e impacto.

Diferentes técnicas de persuasão podem ser aplicadas em memes, aproveitando sua capacidade de transmitir mensagens de forma rápida e eficaz. O uso de falácias lógicas, com o uso de humor e ironia, pode tornar uma mensagem mais cativante e persuasiva, ao mesmo tempo em que tenta suavizar possíveis resistências do público-alvo. Por exemplo, a Figura 1 ilustra um meme, em inglês, que retrata a congressista americana Ilhan Omar junto ao ex-presidente dos Estados Unidos, Donald Trump. Neste meme, a associação é feita entre a expressão de desgosto em relação ao ex-presidente e o terrorismo.

Este trabalho pretende utilizar técnicas de Processamento de Linguagem Natural (PLN) para criar um sistema capaz de classificar um meme como normal ou persuasivo unicamente em função de seu texto. O Processamento de Linguagem Natural é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana (CASELI; NUNES, 2024). Ele funciona como uma interseção entre ciência da computação, inteligência artificial e linguística computacional, oferecendo as ferramentas necessárias para os computadores entenderem a nossa linguagem (LOPEZ; KALITA, 2017). O trabalho empregará uma abordagem baseada em engenharia de características (*feature engineering*), que consiste na transformação dos textos dos memes (dados brutos) em vetores numéricos que são exigidos pelos algoritmos de classificação (NARGESIAN et al., 2017; ZHENG; CASARI, 2018).

Figura 1 – Exemplo de meme com persuasão



Fonte: (SemEval, 2024)

1.1 Justificativa

A utilização de técnicas de persuasão em memes é um problema do mundo moderno, visto que pode levar a propagação de desinformação e influenciar a percepção do público sobre diversos assuntos através de mídias sociais na internet. Como este problema é relativamente novo, ainda há uma carência de estudos sobre o tema. Embora existam pesquisas focadas na identificação de propaganda em textos de artigos (CRUZ; ROCHA; CARDOSO, 2019; MARTINO; BARRÓN-CEDEÑO; NAKOV, 2019), apenas recentemente os pesquisadores começaram a investigar e estudar a classificação e detecção de

propaganda em textos de memes. Trata-se de um problema importante e relevante, dado o vasto alcance e a rápida disseminação que os memes podem atingir.

1.2 Objetivos

O objetivo deste trabalho é criar um sistema de classificação de textos curtos, especificamente aplicado à análise de textos de memes, de forma a classificar se o conteúdo do texto é persuasivo ou não. Ao desenvolver um sistema capaz de identificar de forma eficaz a presença de propaganda em memes, este trabalho poderá fornecer um material para pesquisas futuras sobre o tema. A classificação em textos de memes se encontra em um contexto desafiador, visto que os textos tendem a ser muito curtos e informais, podendo empregar linguagem coloquial e referências culturais. Com o objetivo de tentar a compreensão sobre o tema, este trabalho terá como foco o uso de técnicas de engenharia de características e de algoritmos capazes de gerar modelos interpretáveis (FREITAS, 2014), isto é, modelos que possuem a habilidade de “explicar” as suas classificações para os usuários. Mais especificamente, os algoritmos de classificação utilizados nesse trabalho são Árvore de Decisão CART (BREIMAN et al., 1984), Regressão Logística (JURAFSKY; MARTIN, 2024a) e Floresta Aleatória (*Random Forest*)¹ (BREIMAN, 2001).

¹ Embora o algoritmo de Floresta Aleatória não gere um modelo que é diretamente interpretável, a técnica proposta em (VIDAL; SCHIFFER, 2020) torna possível transformar uma Floresta Aleatória em uma única árvore de decisão com o mesmo poder preditivo.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo tem como objetivo fazer uma revisão em dois temas de grande relevância para este trabalho: a detecção de propaganda em textos e a classificação de textos utilizando engenharia de características. Para isso, foram analisados diferentes estudos recentes de pesquisa, visando uma compreensão aprofundada e atualizada sobre esses tópicos.

A detecção computacional de propaganda, sob a perspectiva do PLN e análise de rede, foi abordada por [Martino et al. \(2020\)](#). No trabalho, eles revisaram o estado da arte sobre o tema, discutindo sobre a necessidade de esforços combinados entre Processamento de Linguagem Natural, análise de rede e aprendizado de máquina. Os pesquisadores identificaram três conjuntos de dados contendo artigos de notícias. Dois desses conjuntos incluíam classificações sobre a presença de propaganda no artigo como um todo, enquanto o terceiro fornecia classificações por frases. Com esses dados rotulados, torna-se possível desenvolver classificadores para a detecção de propaganda em textos. Além disso, o artigo menciona a existência de abordagens baseadas em redes, em vez de PLN, como a identificação de tráfego atípico em redes sociais. No entanto, o presente TCC não trabalhará com este tipo de abordagem.

Ainda na detecção de propaganda em textos de artigos, [Martino, Barrón-Cedeño e Nakov \(2019\)](#) realizaram uma revisão sobre métodos para identificar propaganda em artigos de notícias. Diversas equipes participaram de uma tarefa compartilhada voltada para a identificação de propaganda, na qual a maioria utilizou classificadores treinados com uso de redes neurais ([JURAFSKY; MARTIN, 2024d](#)) e *embeddings* semânticos ([SENO et al., 2024](#)).

A detecção de propaganda de forma interpretável é abordada por [Yu et al. \(2021\)](#), através da identificação de conteúdo persuasivo em artigos de notícias. Segundo os autores, a desinformação disfarçada como notícia cria a ilusão de confiabilidade, tornando as pessoas mais suscetíveis a julgarem a informação como verdadeira se ela estiver alinhada com suas próprias crenças e opiniões. O objetivo do estudo foi classificar a presença ou ausência de propaganda em frases de artigos de notícias. Utilizaram uma base de dados com 451 artigos, totalizando 20.110 frases, classificadas como propaganda ou não-propaganda. Eles propuseram um método capaz de detectar propaganda em cada frase e explicar qual técnica de propaganda foi empregada, baseado em um classificador SVM treinado com atributos interpretáveis relacionados à semântica e à sintaxe. Os autores concluíram que o método proposto teve uma performance relativamente boa em comparação com métodos tradicionais da literatura.

Cruz, Rocha e Cardoso (2019) exploraram a detecção de propaganda em frases de artigos, avaliando soluções com engenharia de características (NARGESIAN et al., 2017; ZHENG; CASARI, 2018) e *Word Embeddings*. Foram avaliados 350 artigos com um total de 16.965 frases, onde 75% estavam classificadas como não-propaganda e 25% estavam classificadas como propaganda. O desequilíbrio da base leva o modelo a favorecer previsões na classe majoritária (não-propaganda) (CRUZ; ROCHA; CARDOSO, 2019). Para contornar esse problema foi feito o balanceamento da base para o treinamento dos modelos através do método *undersampling* da classe majoritária, que será abordado na seção metodológica deste trabalho.

A classificação de textos utilizando engenharia de características foi abordada por Gualberto e authors (2020), onde eles utilizaram esse método para extrair atributos para classificar e-mails como *phishing*² ou não. Primeiramente, eles aplicaram um pré-processamento no texto dos e-mails. Em seguida, foi gerada uma Matriz Documento-Termo (DTM), a partir do texto processado, com o ranking matricial representado pelo TF-IDF, que será abordado na seção metodológica deste trabalho. Foram utilizados diferentes algoritmos, em diferentes cenários, tais como Árvores de Decisão, Floresta Aleatória, KNN, Naive Bayes, XGBoost, entre outros. O método por Floresta Aleatória obteve bons resultados em diferentes cenários, com a medida F1 atingindo o valor de até 100%.

O trabalho realizado por Ferreira, Dalip e Silva (2023) utiliza conceitos de engenharia de características para a extração de atributos numéricos a partir de informações textuais para classificação. Eles realizaram a classificação de gêneros musicais com base em letras de músicas, utilizando algoritmos de classificação, como, por exemplo, a Floresta Aleatória. O algoritmo de Floresta Aleatória, utilizando os atributos extraídos de cada letra musical, obteve o melhor resultado entre os classificadores testados, tanto para a classificação hierárquica, onde é possível utilizar diferentes agrupamentos e formas de agrupamentos, quanto para a classificação tradicional.

Nagy e Kapusta (2023) tiveram como foco a detecção de notícias falsas em textos não estruturados. Para a extração de atributos preditivos, eles utilizaram a técnica TW-IDW (*Term Weight-Inverse Document Weight*), que é uma variação na medida TF-IDF. A técnica TW-IDW utiliza a profundidade das palavras nos documentos. Os modelos de classificação foram criados utilizando Floresta Aleatória e Redes Neurais. Três conjuntos de dados com notícias sobre COVID-19 foram analisados. A Floresta Aleatória alcançou uma acurácia de 90,2% utilizando a medida TF-IDF, enquanto a Rede Neural obteve a melhor acurácia da pesquisa, com 90,4%, ao utilizar a medida TW-IDW.

Albornoz, Vidal e Plaza (2018) utilizaram a engenharia de características para análise de sentimentos em fóruns de saúde em redes sociais. Para isso, eles consideraram diferentes atributos preditivos, incluindo lexicais, semânticos, recursos baseados em sen-

² Ameaça baseada em comunicação fraudulenta, geralmente recebida por e-mail.

timentos e *embeddings*. Foram analisados mais de 3.500 posts *online* em fóruns de saúde, abordando câncer de mama, doença de Crohn e diferentes tipos de alergia. Utilizando algoritmos de aprendizado de máquina, eles conseguiram alcançar uma acurácia de aproximadamente 70% ao utilizar *embeddings* para a extração de atributos.

Este trabalho se distingue dos anteriores ao focar especificamente em textos de memes, enquanto os estudos prévios geralmente analisam artigos ou frases extraídas de artigos. Além disso, esta abordagem não se limita ao uso de *undersampling*. Também foi empregado *oversampling* para balancear os dados, que será abordado na seção metodológica.

3 BASE DE DADOS

O trabalho utilizará a base de dados fornecida na competição SemEval 2024 (SemEval, 2024). Nesta competição, diversas equipes competem entre si com o objetivo de construir modelos para identificar as diferentes técnicas de persuasão que podem estar sendo utilizadas ou não nos textos dos memes. As equipes tiveram seus desempenhos medidos através de métricas avaliativas, tais como o F1-Score (JURAFSKY; MARTIN, 2024c). A competição, iniciada nos últimos meses de 2023 e finalizada em 2024 foi organizada por pesquisadores da Itália, França, Bulgária, Emirados Árabes Unidos e Catar. Até o momento do desenvolvimento deste trabalho, não havia sido publicado nenhum artigo elaborado por membros das equipes participantes da competição.

A base de dados fornecida pela organização da competição foi dividida em três conjuntos distintos: treino, validação e teste. Cada conjunto de dados possui as seguintes variáveis: "id" (identificador único para cada observação), "text" (texto do meme), "labels" (classificações relacionadas aos métodos de persuasão presentes no texto, ou indicando a ausência deles) e "link" (link para abrir a imagem do meme). A coluna "labels" pode estar rotulada com mais de uma classificação de persuasão diferente ao mesmo tempo. Essa estruturação dos dados permite uma análise detalhada e sistemática dos textos, auxiliando a construção e avaliação de modelos de classificação.

A base de treino possui 7.000 observações e será usada nesse trabalho para treinar diferentes modelos de classificação. A base de validação possui 500 observações e foi utilizada pelos competidores em um estágio preliminar da competição. Ela não será utilizada no presente TCC. Já a base de teste possui 1.000 observações e será utilizada para estimar o desempenho preditivo dos modelos, onde serão verificadas as medidas de acurácia, precisão, recall e F1-Score (JURAFSKY; MARTIN, 2024c).

3.1 Classificação das Técnicas de Persuasão

De acordo com os organizadores da competição SemEval 2024, há 20 técnicas de persuasão distintas que podem ser aplicadas em textos. Essas técnicas podem ser aplicadas de forma isolada ou em conjunto com outras de forma simultânea. Os Quadros 1, 2 e 3 apresentam uma síntese das técnicas pontuadas juntamente com uma definição³.

³ <<https://propaganda.math.unipd.it/semEval2024task4/definitions.html>>

Quadro 1 – Técnicas de Persuasão

| Técnica | Descrição | Exemplo |
|---|---|---|
| Presenting Irrelevant Data (Red Herring) | Apresentar material irrelevante ao assunto em discussão, para que a atenção de todos seja desviada dos pontos levantados. | <i>Temos recebido muitas críticas em relação à qualidade do nosso produto. Decidimos, então, fazer uma promoção, por um preço muito menor!</i> |
| Misrepresentation of Someone's Position (Straw Man) | Substituir a proposição de um oponente por outra semelhante, que é então refutada no lugar da proposição original. | <i>Você afirmou que fornecer cuidados de saúde para todos os cidadãos seria custoso, então você não se importa se as pessoas morrerem por não terem assistência médica.</i> |
| Whataboutism | Tentar desacreditar a posição de um oponente, acusando-o de hipocrisia, sem refutar diretamente o seu argumento. | <i>O Catar gasta muito com o Neymar e não combate o terrorismo.</i> |
| Causal Oversimplification | Assumir uma única causa ou motivo quando na verdade existem múltiplas causas para um problema. | <i>Se a França não tivesse declarado guerra com a Alemanha, a Segunda Guerra Mundial nunca teria acontecido.</i> |
| Obfuscation, Intentional vagueness, Confusion | Usar palavras deliberadamente pouco claras para que o público possa ter as suas próprias interpretações. | <i>Nós vamos desenvolver a blockchain com interconectores baseados em inteligência artificial para maximizar o ROI.</i> |
| Appeal to authority | Afirmar que uma afirmação é verdadeira simplesmente porque uma autoridade ou especialista válido no assunto disse que era verdade, sem qualquer outra evidência de apoio oferecida. A referência pode, ainda, não ser uma especialista no assunto e ser tratada como tal. | <i>De acordo com Serena Williams, nossa política estrangeira é a melhor do mundo. Logo, estamos no caminho certo.</i> |
| Black-and-white Fallacy, Dictatorship | Apresentar duas opções alternativas como as únicas possibilidades, quando na verdade existem mais possibilidades. | <i>Não há outra alternativa que não seja a guerra.</i> |

Quadro 2 – Técnicas de Persuasão

| Técnica | Descrição | Exemplo |
|------------------------------|---|--|
| Name calling or labeling | Rotular o objeto da campanha de propaganda como algo que o público-alvo teme, odeia, considera indesejável ou ama, elogia. | <i>Posso não ser perfeito, mas não sou falso!</i> |
| Loaded Language | Usar palavras e frases específicas com fortes implicações emocionais (positivas ou negativas) para influenciar o público. | <i>Como as coisas se tornaram estúpidas e mesquinhas em Washington.</i> |
| Exaggeration or Minimisation | Representar algo de forma excessiva: tornando as coisas maiores, melhores, piores ou fazendo algo parecer menos importante ou menor do que realmente é. | <i>Eu não briguei com ela, foi apenas uma brincadeira.</i> |
| Flag-waving | Jogar com um forte sentimento nacional (ou de qualquer grupo; por exemplo, raça, gênero, preferência política) para justificar ou promover uma ação ou ideia. | <i>Entrar nessa guerra trará um futuro melhor para o nosso país.</i> |
| Doubt | Questionar a credibilidade de alguém ou de alguma coisa. | <i>Eu olho para meu oponente e me pergunto: ele está realmente apto para ser prefeito?</i> |
| Appeal to fear/prejudice | Procurar construir apoio para uma ideia incutindo ansiedade e/ou pânico na população em relação a uma alternativa. | <i>Se meu oponente for eleito, ele vai roubar sua renda e sua casa.</i> |
| Slogans | Introduzir uma frase breve e marcante que pode incluir rotulagem e estereótipos. Os slogans tendem a atuar como apelos emocionais. | <i>O que importa é o aqui e o agora.</i> |

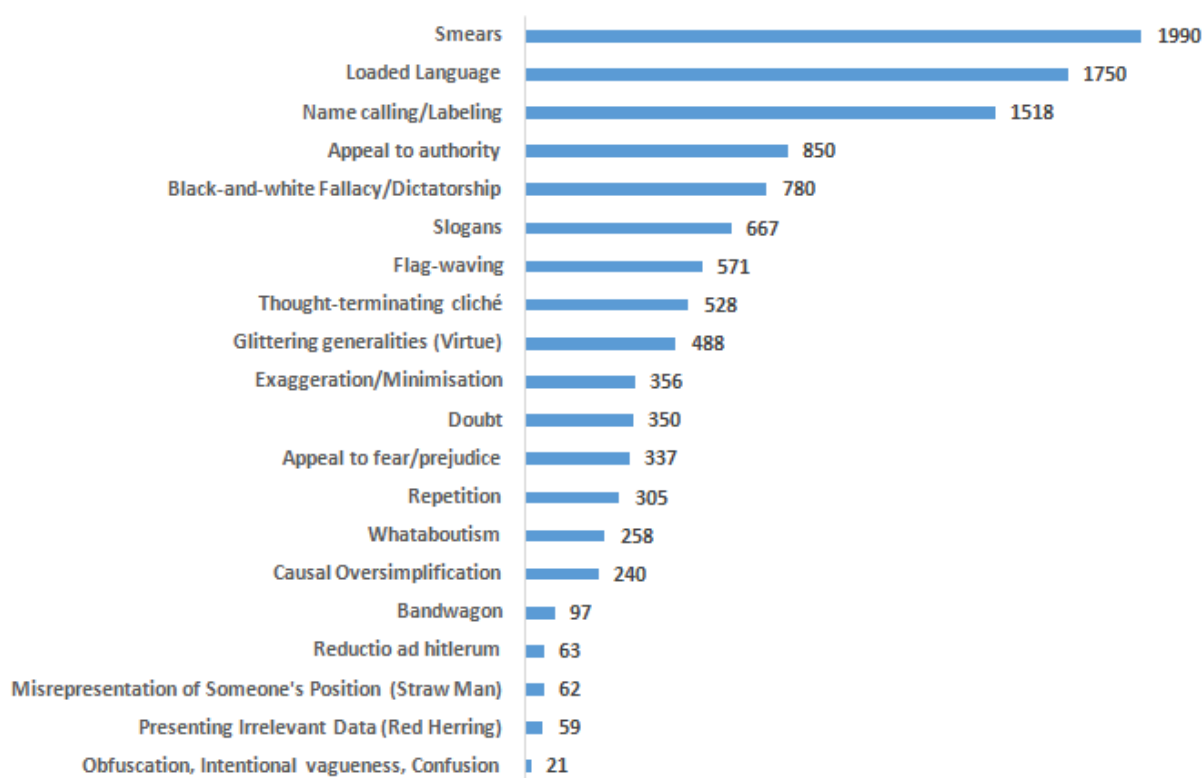
Quadro 3 – Técnicas de Persuasão

| Técnica | Descrição | Exemplo |
|----------------------------|--|---|
| Thought-terminating cliché | Usar palavras ou frases que desencorajam o pensamento crítico e a discussão significativa sobre um determinado tópico. Geralmente são frases curtas e genéricas que oferecem respostas aparentemente simples a questões complexas ou que desviam a atenção de outras linhas de pensamento. | <i>É apenas bom senso.</i> |
| Bandwagon | Tentar persuadir o público-alvo a aderir e tomar a atitude porque “todos os outros estão tomando a mesma atitude”. | <i>Todo mundo vai comprar o novo smartphone que será lançado essa semana!</i> |
| Reductio ad hitlerum | Persuadir um público a desaprovar uma ação ou ideia, sugerindo que a ideia é popular entre grupos odiados com desprezo pelo público-alvo. Pode referir-se a qualquer pessoa ou conceito com conotação negativa. | <i>Sabe quem mais já fez isso? Hitler!</i> |
| Repetition | Repetir a mesma mensagem continuamente para que o público acabe por aceitá-la. | <i>Oppenheimer merece o Oscar. Outros filmes têm potencial, mas não merecem o Oscar como Oppenheimer merece. Os outros filmes podem merecer uma menção honrosa, mas Oppenheimer merece o Oscar.</i> |
| Smears | Tentar prejudicar ou questionar a reputação de alguma instituição ou alguém, propondo propaganda negativa. Pode ser aplicado a indivíduos ou grupos. | <i>Meu oponente tem um histórico de mentiras e tenta encobrir seus negócios desonestos com um sorriso agradável. Não deixe que ele o convença a acreditar em suas palavras.</i> |
| Glittering Generalities | Usar palavras ou símbolos do sistema de valores do público-alvo que produzem uma imagem positiva quando associados a uma pessoa ou assunto. Paz, esperança, felicidade, segurança, liderança sábia, etc. | <i>Eu sei que ele é uma boa pessoa e um homem sábio para liderar o país.</i> |

3.2 Análise Exploratória

Cada observação de cada uma das três bases está rotulada com as técnicas de persuasão (informadas no Quadro ??) que estão sendo utilizadas no texto do meme, caso o texto tenha conteúdo persuasivo. A base de treino possui todas as 20 técnicas em seus textos. A Figura 2 exibe a frequência de cada técnica na base de treino. No total, as técnicas foram empregadas 11.290 vezes.

Figura 2 – Frequência das técnicas de persuasão na base de treino



Fonte: Elaboração própria.

A técnica Smears destaca-se como a principal estratégia de persuasão adotada nos memes, respondendo por 17,63% de todas as técnicas utilizadas. Em seguida, a técnica Loaded Language ocupa a posição de segunda mais frequente, representando 15,50% do total. Por sua vez, a técnica Name calling/Labeling figura como a terceira mais prevalente, com uma representatividade de 13,45%. Quanto à técnica de persuasão com a menor incidência na base de dados, observa-se que é a Obfuscation, Intentional vagueness, Confusion, registrando apenas 0,19% do total. Na Figura 3 a seguir, apresenta-se um exemplo de meme da base com a técnica Smears, o método mais frequente utilizado na base de treino. No exemplo tem-se o texto: “como eu durmo sabendo que nunca segui nenhuma diretriz governamental”. A técnica Smears tem o objetivo de tentar prejudicar ou questionar a reputação de alguma instituição ou alguém, aqui representado como o governo.

Figura 3 – Exemplo de meme com a técnica Smears



Fonte: Facebook.

Cada texto pode ser rotulado com diferentes técnicas de persuasão simultaneamente. Dentro do conjunto de observações da base de treinamento, 3.274 foram identificadas com duas ou mais técnicas, enquanto 2.462 foram rotuladas com apenas uma técnica. Além disso, há 1.263 textos na base que não apresentam nenhuma técnica de persuasão em seu conteúdo. A Figura 4 a seguir ilustra a distribuição do número de textos em relação ao total de técnicas aplicadas.

Uma análise da distribuição das palavras presentes nos textos dos memes também

4 METODOLOGIA

O processo de construção do sistema para detecção de persuasão proposto nesse trabalho é composto por 5 etapas:

1. Pré-processamento.
2. Engenharia de Características.
3. Balanceamento da base de treino.
4. Treinamento do modelo.
5. Teste e avaliação do modelo.

A classificação está no cerne tanto da inteligência humana quanto da inteligência artificial (JURAFSKY; MARTIN, 2024c). O objetivo da classificação é, a partir de dados de entrada, gerar um rótulo de saída para esses dados. Este trabalho, por exemplo, tem o objetivo de classificar textos de memes com o rótulo “persuasivo” ou “não persuasivo” a partir de atributos extraídos desses textos. Um sistema de classificação é composto pelos seguintes componentes (JURAFSKY; MARTIN, 2024c):

1. Um conjunto $X = \{X_1, X_2, \dots, X_d\}$ dos atributos de entrada, onde d é o total de atributos que serão usados para a classificação.
2. Um conjunto $L = \{l_1, \dots, l_q\}$ de q rótulos, onde $q \geq 2$. No caso deste trabalho, temos que $q = 2$, ou seja, trata-se de um problema de classificação binária.
3. Uma base de treinamento $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ de N objetos (observações).
4. Um vetor $\mathbf{x} = [x_1, x_2, \dots, x_n]$, onde cada x_i representa a i -ésima observação da base de treinamento D e armazena valores para os d atributos preditivos de X .
5. Um vetor de classificação $\mathbf{Y} = [y_1, y_2, \dots, y_n]$, onde cada $y_i \in L$ corresponde à classificação associada à x_i .
6. Aprendizado de uma função f (classificador ou modelo de classificação) que, dado um objeto não rotulado $t = (x, ?)$, seja capaz de prever de forma efetiva o seu rótulo de classe y .

Para um classificador probabilístico, tal como a regressão logística, de acordo com Jurafsky e Martin (2024a), a função f de classificação vai calcular \hat{y} , a classe estimada, através da probabilidade condicional $p(y|x)$. Também há uma função objetivo para aprendizado, geralmente envolvendo a minimização de erros no treino e um algoritmo para otimização da função objetivo. Além da regressão logística, esse trabalho utilizará outras duas técnicas de classificação: Árvore de Decisão e Floresta Aleatória.

4.1 Pré-processamento

Em um primeiro momento, foi realizado o pré-processamento dos textos das bases de treino e teste. O passo inicial foi a remoção das *stopwords* utilizando a biblioteca “spaCy”⁴ do Python. Para a língua inglesa, a biblioteca “spaCy” considera uma lista de 326 palavras como *stopwords*. A remoção dessas palavras auxilia na análise exploratória da base e na criação de atributos, pois são termos que podem aparecer com alta frequência, mas não fornecem informações relevantes sobre o conteúdo do texto.

Em seguida, foi realizada a lematização das palavras. A lematização é uma forma de normalização das palavras, as convertendo para uma forma padrão (FINATTO et al., 2024). As frases “eu corro” e “eles correm”, por exemplo, passariam a ser “eu correr” e “eles correr”, respectivamente. Da mesma forma, isto foi feito para os textos dos memes, em inglês, das bases de treino e teste. A lematização é útil para converter formas variantes das palavras (plural, feminino, gerúndio, etc.) para uma representação única e concisa.

Por fim, o processo de tokenização foi utilizado para transformar cada texto em um vetor de palavras. A tokenização envolve a separação do texto em unidades linguísticas mínimas (*tokens*), separando as palavras por meio de delimitadores (FINATTO et al., 2024). Neste trabalho, os espaços e os sinais de pontuação foram utilizados como delimitadores.

De forma ilustrativa, apresenta-se a seguir um exemplo do processo de remoção de *stopwords*, lematização e tokenização. Considere a seguinte frase: “O gato pulou alto!”. O seu pré-processamento se daria da seguinte forma:

1. Remoção das stopwords, resultando na frase: “gato pulou alto!”.
2. Lematização, resultando na frase: “gato pular alto!”.
3. Tokenização, resultando no vetor: [“gato”, “pular”, “alto”, “!”].

⁴ <<https://spacy.io/api/doc>>

4.2 Engenharia de Características

Para obter atributos numéricos a partir dos dados textuais, foi empregada a engenharia de características. A engenharia de características é o ato de extração de atributos dos dados brutos e a transformação desses atributos em formatos adequados, como vetores numéricos, necessários para os algoritmos de aprendizado de máquina (NARGESIAN et al., 2017; ZHENG; CASARI, 2018).

De forma a obter-se um vetor numérico representativo do texto do meme, foram considerados os seguintes atributos utilizados por Cruz, Rocha e Cardoso (2019), que foram adaptados para o contexto do problema e extraídos para o texto de cada meme da base de treino:

1. Número total de frases: representa o total de frases do texto do meme.
2. Média de caracteres por frase: representa a média de caracteres por frase do texto do meme, onde cada frase é delimitada por ponto final, ponto de exclamação ou ponto de interrogação.
3. Variância de caracteres por frase: representa a variância de caracteres por frase do texto do meme, utilizando a mesma delimitação de frase do atributo (2).
4. Total de caracteres: representa o total de caracteres presente em todo o texto do meme.
5. Média de caracteres por palavra: representa a média de caracteres por palavra do texto do meme.
6. Variância de caracteres por palavra: representa a variância de caracteres por palavra do texto do meme.
7. Frequência de pontuação: representa a frequência absoluta de todos os símbolos de pontuação presentes no texto do meme.
8. Frequência de letras maiúsculas: representa a frequência absoluta de todas as letras maiúsculas presentes no texto do meme.
9. Proporção de token sobre palavras lematizadas: representa o total de palavras lematizadas dividido pelo total de tokens do texto do meme (“andou” e “andei” são dois tokens diferentes, mas ao serem lematizados, ambos se tornam “andar”).
10. TF-IDF.

De forma ilustrativa, considere o seguinte texto, em português, para passar pelo processo de extração de atributos numéricos através da engenharia de características:

“As Olimpíadas chegaram! Os atletas acabam de chegar em Paris para a maior competição esportiva do mundo. O Time Brasil está preparado? De acordo com os atletas, eles estão!”

Considerando as variáveis apresentadas previamente (desconsiderando o TF-IDF), temos os seguintes atributos numéricos extraídos:

1. Número total de frases: 4.
2. Média de caracteres por frase: 42,25.
3. Variância de caracteres por frase: 499,69.
4. Total de caracteres: 172.
5. Média de caracteres por palavra: 4,79.
6. Variância de caracteres por palavra: 7,06.
7. Frequência de pontuação: 5.
8. Frequência de letras maiúsculas: 8.
9. Proporção de *tokens* sobre palavras lematizadas: 0,74.

Analisando o atributo (9), de mais complexo entendimento que os outros, podemos detalhar seu processo de cálculo. Os *tokens* gerados para o texto formam o seguinte vetor de palavras e pontuações:

["As", "Olimpíadas", "chegaram", "!", "Os", "atletas", "acabam", "de", "chegar", "em", "Paris", "para", "a", "maior", "competição", "esportiva", "do", "mundo", ".", "O", "Time", "Brasil", "está", "preparado", "?", "De", "acordo", "com", "os", "atletas", ",", "eles", "estão", "!"]

Este vetor tem o total de 34 elementos. O próximo passo consiste em gerar um vetor com as palavras lematizadas, que resulta no seguinte vetor de palavras e pontuações:

['!', ',', ':', '?', 'Brasil', 'Olimpíadas', 'Paris', 'Time', 'acabar', 'acordo', 'atleta', 'chegar', 'com', 'competição', 'de', 'de o', 'ele', 'em', 'esportivo', 'estar', 'grande', 'mundo', 'o', 'para', 'preparado']

No vetor lematizado há um total de 25 elementos. Neste caso, não há repetições. Desta forma, o atributo é calculado por $25/34$, resultado no valor de 0,74.

4.2.1 TF-IDF

Uma das formas a ser utilizada para a extração de atributos a partir do conteúdo dos textos será a Matriz Documento-Termo (*Document-Term Matrix* - DTM) com pesos TF-IDF. Para explicar o TF-IDF, será apresentado um exemplo de como essa medida é usada no contexto de busca de documentos na Internet. Para ordenar documentos em resposta à uma consulta realizada, há duas premissas que podem ser utilizadas:

1. Documentos que contêm mais vezes os termos da consulta terão mais chances de serem relacionados a ela e, então, serem relevantes.
2. Os termos mais raros na coleção de documentos são úteis para a diferenciação de conteúdo nos documentos (MOREIRA, 2023).

Por exemplo, se uma pessoa decide pesquisar sobre artigos de notícias do cenário político nos EUA em um corpus de documentos (um site de notícias, por exemplo), alguns termos serão mais relevantes que outros. Palavras como “Biden” ou “Casa Branca” serão termos úteis para a pesquisa, enquanto termos como “basquete” ou “previsão do tempo” já não terão tanta utilidade.

O TF-IDF e a DTM são usados não apenas para recuperação da informação (busca de documentos), mas também em problemas de classificação. Na DTM, cada linha representa um documento em uma coleção de documentos e cada coluna representa uma palavra ou grupo de palavras, onde cada célula da matriz representa o peso do termo no documento (JURAFSKY; MARTIN, 2024e). Para atribuir pesos aos termos em relação aos documentos, é muito comum utilizar a medida TF-IDF, onde TF significa “*term-frequency*” (frequência do termo) e IDF significa “*inverse document-frequency*” (frequência inversa do termo) (MOREIRA, 2023). A medida TF-IDF é, então, o produto de TF e IDF, conforme apresentado na Equação 4.1.

$$\text{TF-IDF}_{(t,d)} = \text{TF}_{(t,d)} \times \text{IDF}_t \quad (4.1)$$

Onde:

- **TF-IDF_(t,d)**: Valor TF-IDF do termo t no documento d .
- **TF_(t,d)**: Valor TF do termo t no documento d .
- **IDF_t**: Valor IDF do termo t no corpus de documentos.

O cálculo dos pesos TF-IDF foi realizado utilizando a biblioteca Scikit-learn, no Python. A biblioteca calcula o termo TF_(t,d) como sendo a frequência absoluta em que o

termo t aparece no documento d . Já o termo IDF_t é calculado de acordo com a Equação 4.2.

$$IDF_t = \ln \left(\frac{N}{df_t} \right) + 1 \quad (4.2)$$

Onde:

- n : valor total de documentos no corpus;
- df_t : valor total de documentos no corpus que contêm o termo t .

O componente IDF tem o objetivo de dar um peso maior aos termos mais raros, pois eles são úteis para discriminar o conteúdo dos documentos (MOREIRA, 2023). Portanto, quanto mais raro o termo for no corpus, maior será o valor IDF. A Tabela 1 exibe um exemplo de como funciona uma Matriz Documento-Termo com pesos TF-IDF.

Tabela 1 – Matriz Documento-Termo

| | Termo 1 | Termo 2 | Termo 3 | ... | Termo n |
|-------------|---------|---------|---------|-----|---------|
| Documento 1 | 0,31 | 2,90 | 7,14 | ... | 1,21 |
| Documento 2 | 5,32 | 0,90 | 2,21 | ... | 3,72 |
| Documento 3 | 3,98 | 9,14 | 0,23 | ... | 7,35 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Documento j | 12,01 | 6,49 | 4,15 | ... | 8,92 |

Neste trabalho, cada documento corresponde a um texto de meme, e o corpus é composto pelo conjunto de todos os textos de memes presentes na base de dados. Nos experimentos, os termos serão representados de duas formas distintas:

1. Os 50 unigramas mais frequentes.
2. Os 50 bigramas mais frequentes.

Os unigramas e bigramas são dois exemplos de n-gramas. O n-grama é uma sequência de n palavras (JURAFSKY; MARTIN, 2024b). Portanto, o unigrama é uma sequência de uma palavra e o bigrama é uma sequência de duas palavras.

4.3 Balanceamento da Base

Conforme abordado na seção de revisão bibliográfica deste trabalho, bases desbalanceadas podem levar o modelo a favorecer previsões na classe majoritária (CRUZ;

ROCHA; CARDOSO, 2019). Como a base de treino contém 78% de suas observações classificadas como propaganda, tem-se um caso em que os modelos podem favorecer a previsão dos textos da base de teste dessa forma. Portanto, com o objetivo de retirar essa tendência de classificação, serão aplicadas duas formas distintas de balanceamento: *undersampling* e *oversampling*.

Ao se utilizar dados do mundo real, classes desbalanceadas podem ser esperadas. Se o grau de desbalanceamento dos dados para a classe majoritária for algo extremo então o classificador pode produzir uma alta acurácia na previsão, dado que o modelo prevê a maioria das instâncias como pertencentes à classe majoritária (LEEVEY et al., 2018).

O método *undersampling* consiste na redução da quantidade de observações da classe majoritária. A forma mais popular para a aplicação de *undersampling*, que também será utilizada neste trabalho, é o *undersampling* aleatório. O método consiste em remover observações da classe majoritária, aleatoriamente, até que a base de dados alcance o balanceamento entre as classes (Imbalanced learn, Python)⁵.

Os dois métodos de balanceamento serão aplicados na base de treino, de forma a balancear as classes para tentar obter um melhor desempenho dos modelos de classificação. Resultados empíricos de trabalhos relevantes apontam que o método de *oversampling* aleatório alcança melhores resultados do que o método de *undersampling* aleatório (LEEVEY et al., 2018).

4.4 Técnicas de Classificação

4.4.1 Árvores de Decisão

As árvores de decisão são um método aprendizado não paramétrico usado para classificação e regressão. De acordo com a biblioteca Scikit-learn⁶, o objetivo é criar um modelo que preveja o valor de uma variável alvo, aprendendo regras de decisão simples inferidas a partir dos dados. Uma vantagem deste algoritmo é a facilidade na sua interpretação.

As árvores de decisão são construídas através da análise de uma base de treino, no qual a classificação de cada observação é conhecida, sendo utilizada posteriormente para classificar dados novos e desconhecidos (KINGSFORD; SALZBERG, 2008).

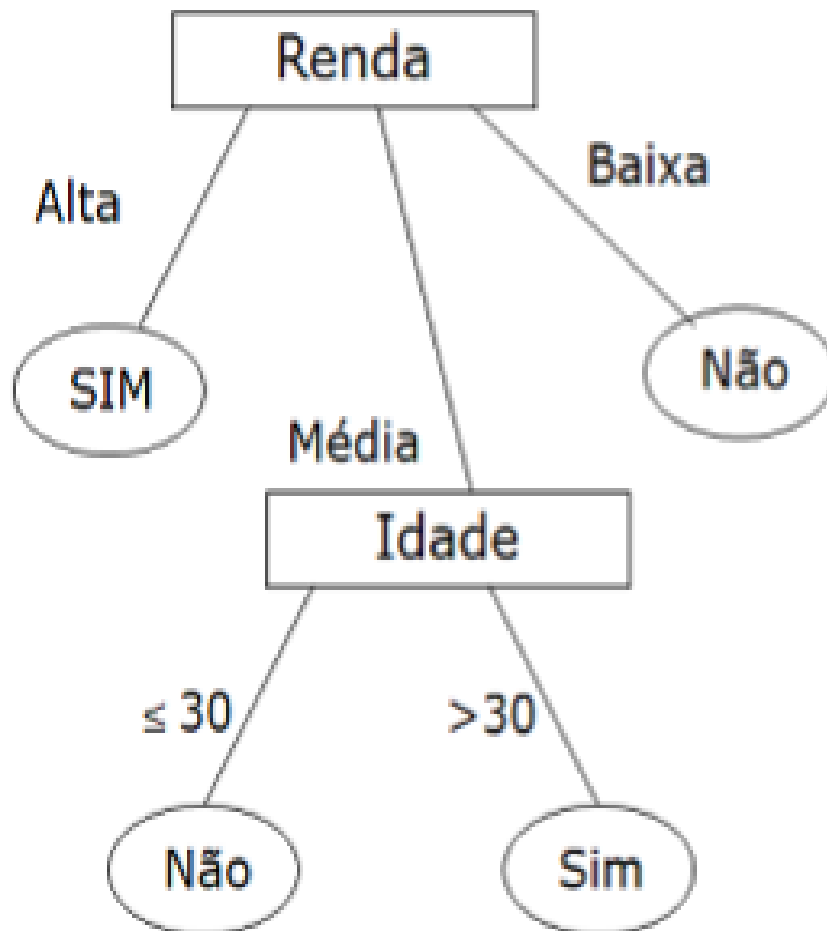
De acordo com Ali et al. (2012), a ideia do algoritmo veio a partir da estrutura de uma árvore real, que é composta por raiz e nós, sendo que os nós seriam as ramificações e os locais onde elas se dividem. A árvore de decisão, então, é construída a partir dos nós e os ramos são representados pelos segmentos que conectam os nós. Ela começa da raiz

⁵ <https://imbalanced-learn.org/stable/under_sampling.html>

⁶ <<https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>>

e se move para baixo. O nó onde as ramificações se encerram é conhecido como folha. Cada nó representa uma certa característica enquanto os ramos representam uma gama de valores. A Figura 8 a seguir apresenta um exemplo de uma Árvore de Decisão que classifica um cliente como possível comprador de carro importado com base em sua renda e idade (SILVA; GONÇALVES, 2021).

Figura 8 – Exemplo de Árvore de Decisão



Fonte: (SILVA; GONÇALVES, 2021)

Uma Árvore de Decisão é formada por um conjunto de regras de classificação, uma vez que existe sempre um único caminho da raiz para cada folha, onde o caminho representa uma expressão lógica da regra utilizada para classificar um objeto (SILVA; GONÇALVES, 2021). Por exemplo, as regras da Figura 8 para classificar o cliente como possível comprador de carro importado são:

- SE (renda = “alta”), ENTÃO (possível comprador = “sim”).
- SE (renda = “média”) e (idade > 30), ENTÃO (possível comprador = “sim”).

O algoritmo CART (BREIMAN et al., 1984) é um dos mais populares para a geração de árvore de decisão. Este algoritmo realiza a decisão de como dividir a árvore

através do Índice de Gini, que será utilizado para montar a árvore através de decisões binárias. Portanto, cada nó gerado terá duas ramificações. O Índice de Gini é calculado de acordo com a Equação 4.3.

$$\text{Gini}(D) = 1 - \sum_{i=1}^q p_i^2 \quad (4.3)$$

Onde:

- q : número de classes;
- p_i : probabilidade de uma tupla t pertencer à classe C_i na base D .

Como o algoritmo CART realiza apenas divisões binárias em cada nó, sempre serão feitas divisões que resultem em duas ramificações (BREIMAN et al., 1984). Suponha que um atributo preditivo “tempo” tenha as seguintes classificações possíveis: sol, chuva e nublado. Neste caso, o algoritmo CART faria partições da seguinte forma: {sol, chuva}, {sol, nublado}, {chuva, nublado}, {sol}, {chuva} e {nublado}. Então, seria calculado o Índice de Gini para cada combinação possível entre as repartições. Dessa forma, é feita a soma ponderada de cada partição de acordo com a Equação 4.4.

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2) \quad (4.4)$$

A Tabela 2 exibe um exemplo de uma base sobre as condições climáticas e a decisão de sair ou não, representada pela classificação binária “Sim” ou “Não”.

Para gerar uma árvore de decisão para essa base utilizando o algoritmo CART, as Equações 4.3 e 4.4 serão empregadas para determinar a melhor divisão da árvore. A primeira decisão do algoritmo é definir o primeiro nó, que será a raiz da árvore. Nesse caso, a decisão será baseada no cálculo do Índice de Gini para cada atributo e suas possíveis partições (BREIMAN et al., 1984). Então, o que apresentar o menor valor para o Índice de Gini será o escolhido pelo algoritmo para ser a raiz da árvore, conforme exemplo de cálculo demonstrado a seguir.

Tempo:

$$\{\text{Nublado}\} \{ \text{Chuva}, \text{Sol} \} : \frac{4}{14} \times \left(1 - \left(\frac{4}{4} \right)^2 - \left(\frac{0}{4} \right)^2 \right) + \frac{10}{14} \times \left(1 - \left(\frac{5}{10} \right)^2 - \left(\frac{5}{10} \right)^2 \right) = 0.3571$$

$$\{\text{Chuva}\} : \{\text{Nublado}, \text{Sol}\} : \frac{5}{14} \times \left(1 - \left(\frac{3}{5} \right)^2 - \left(\frac{2}{5} \right)^2 \right) + \frac{9}{14} \times \left(1 - \left(\frac{6}{9} \right)^2 - \left(\frac{3}{9} \right)^2 \right) = 0.4571$$

Tabela 2 – Base explicativa para Índice de Gini

| Tempo | Temperatura | Humidade | Ventania | Sair? |
|--------------|--------------------|-----------------|-----------------|--------------|
| Nublado | Quente | Alta | FALSO | Sim |
| Chuva | Ameno | Alta | FALSO | Sim |
| Chuva | Frio | Normal | FALSO | Sim |
| Nublado | Frio | Normal | VERDADEIRO | Sim |
| Sol | Frio | Normal | FALSO | Sim |
| Chuva | Ameno | Normal | FALSO | Sim |
| Sol | Ameno | Normal | VERDADEIRO | Sim |
| Nublado | Ameno | Alta | VERDADEIRO | Sim |
| Nublado | Quente | Normal | FALSO | Sim |
| Sol | Quente | Alta | FALSO | Não |
| Sol | Quente | Alta | VERDADEIRO | Não |
| Chuva | Frio | Normal | VERDADEIRO | Não |
| Sol | Ameno | Alta | FALSO | Não |
| Chuva | Ameno | Alta | VERDADEIRO | Não |

Fonte: Elaboração própria, com base em weather.nominal.arff

$$\{\text{Sol}\} : \{\text{Nublado}, \text{Chuva}\} : \frac{5}{14} \times \left(1 - \left(\frac{2}{5}\right)^2 - \left(\frac{3}{5}\right)^2\right) + \frac{9}{14} \times \left(1 - \left(\frac{7}{9}\right)^2 - \left(\frac{2}{9}\right)^2\right) = 0.3937$$

Temperatura:

$$\{\text{Frio}\} : \{\text{Quente}, \text{Ameno}\} : \frac{4}{14} \times \left(1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2\right) + \frac{10}{14} \times \left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) = 0.4500$$

$$\{\text{Quente}\} : \{\text{Frio}, \text{Ameno}\} : \frac{4}{14} \times \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) + \frac{10}{14} \times \left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) = 0.4429$$

$$\{\text{Ameno}\} : \{\text{Frio}, \text{Quente}\} : \frac{6}{14} \times \left(1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2\right) + \frac{8}{14} \times \left(1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2\right) = 0.4583$$

Humidade:

$$\{\text{Alta}\} : \{\text{Normal}\} : \frac{7}{14} \times \left(1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2\right) + \frac{7}{14} \times \left(1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2\right) = 0.3673$$

Ventania:

$$\{\text{FALSO}\} : \{\text{VERDADEIRO}\} : \frac{8}{14} \times \left(1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2\right) + \frac{6}{14} \times \left(1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2\right) = 0.4286$$

Dessa forma, neste exemplo, a raiz da árvore de decisão ocorre no atributo “tempo”, com a partição Nublado e Chuva, Sol. Portanto, a primeira decisão que o algoritmo CART irá tomar será com base na pergunta “o tempo está nublado?”. Caso sim, ele seguirá por uma ramificação da árvore. Caso contrário, seguirá pela outra ramificação. O algoritmo CART seguirá dessa forma para decidir todos os nós da árvore.

O exemplo apresentado aborda atributos categóricos. No caso de atributos numéricos, o algoritmo CART ordenará os valores de forma ascendente e testará todas as possibilidades considerando o valor médio de cada par de valores. Por exemplo, caso um atributo numérico tenha observações com os valores 10 e 20, o algoritmo considerará o valor 15 para avaliar as condições “menor ou igual a 15” e “maior que 15”. Esse processo será repetido para todos os valores na base de dados.

4.4.2 Floresta Aleatória

O método Floresta Aleatória é uma combinação de árvores de decisão em que cada árvore depende de valores de um vetor aleatório amostrado, em que os vetores são independentes e identicamente distribuídos, sendo que o k -ésimo vetor gerado governa a k -ésima árvore de decisão (BREIMAN, 2001).

Segundo descrito por Ali et al. (2012), cada árvore é gerada a partir dos seguintes passos:

1. Gerando uma amostra aleatória com reposição e de tamanho N , se o número de casos na base de treinamento for igual a N , da base de dados. Essa amostra é usada como treinamento para a árvore de decisão.
2. Sendo d o valor total de atributos preditivos, m atributos são selecionados aleatoriamente de d , sendo que $m \ll d$. A melhor divisão dos m atributos selecionados é utilizada para a divisão do nó. Durante o processo de geração de árvores aleatórias, o valor de m é constante.
3. Cada árvore gerada cresce o máximo possível.

O método Floresta Aleatória foi desenvolvido por (BREIMAN, 2001), que o define formalmente como um classificador que consiste em uma coleção de classificadores de árvores de decisão $\{h(x, \Theta_k), k = 1, \dots\}$, onde $\{\Theta_k\}$ são vetores independentes e identicamente distribuídos e cada árvore dá um voto unitário para a classe mais popular para o

input x . Depois que um grande número de árvores é gerado, a classificação se dará pelo voto majoritário dos classificadores das árvores.

4.4.3 Regressão Logística

A Regressão Logística é um classificador probabilístico, sendo uma das ferramentas analíticas mais importantes das ciências sociais e naturais (JURAFSKY; MARTIN, 2024a). No Processamento de Linguagem Natural, ela é o algoritmo de aprendizado de máquina supervisionado básico para classificação e também tem uma relação muito próxima com redes neurais (JURAFSKY; MARTIN, 2024a).

Conforme Jurafsky e Martin (2024a) destacam, o principal propósito da regressão logística binária reside em tornar um classificador capaz de tomar uma decisão dicotômica sobre a classe de uma nova observação. Para isso, temos o conjunto $X = \{X_1, X_2, \dots, X_d\}$ de atributos numéricos para cada observação x , sendo que o output y do classificador será igual a 1 quando a observação for classificada como parte da classe e 0 caso contrário. Neste trabalho, foi definido que o valor 1 significa texto com persuasão e 0 significa texto sem persuasão. Queremos saber a probabilidade condicional $p(y = 1|x)$. Caso essa seja maior que 0,5, então teremos a classificação $y = 1$ e, caso contrário, $y = 0$.

Ainda segundo Jurafsky e Martin (2024), a regressão logística resolve essa tarefa aprendendo um vetor de pesos (coeficientes) e um termo de viés (intercepto). Cada peso w_i é um número real associado ao atributo preditivo X_i . Então, para tomar uma decisão, primeiro multiplica-se cada X_i pelo peso w_i , soma-se o resultado e adiciona-se o termo de viés b , resultando em um valor z que expressa a soma ponderada para a classe, conforme a Equação 4.5.

$$z = \sum_{i=1}^n w_i X_i + b \quad (4.5)$$

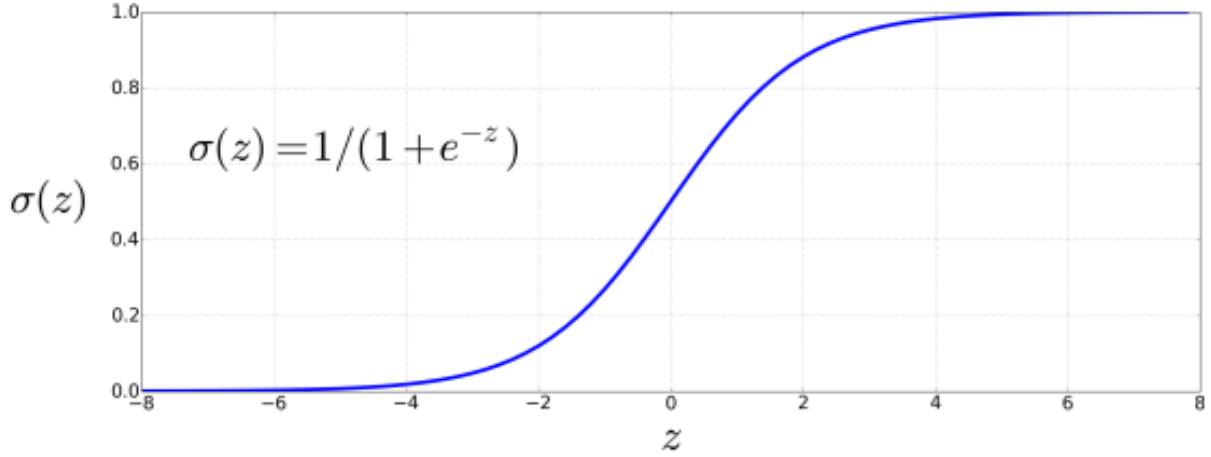
O vetor de pesos \mathbf{w} vai indicar o quão importante cada atributo é para a classificação. O valor z também pode ser representado pelo produto escalar dos vetores \mathbf{w} e \mathbf{X} . O produto escalar de dois vetores \mathbf{a} e \mathbf{b} pode ser escrito como $\mathbf{a} \cdot \mathbf{b}$ e será a soma dos produtos dos elementos correspondentes de cada vetor. Podemos, então, representar z conforme a Equação 4.6.

$$z = \mathbf{w} \cdot \mathbf{X} + b \quad (4.6)$$

Para representar z como uma probabilidade, a função sigmoide (também chamada de função logística), apresentada na Equação 7, é introduzida e apresentada graficamente na Figura 9, de forma a termos valores entre 0 e 1. Como cada um dos pesos possuem valores reais, o valor de z na Equação 4.7 pode variar de $-\infty$ até $+\infty$.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.7)$$

Figura 9 – Função sigmoide



Fonte: (JURAFSKY; MARTIN, 2024a)

A função sigmoide possui muitos valores próximos de 0 e 1 em seu domínio, a tornando adequada para aplicação em métodos de classificação (STOJILJKOVIĆ, 2020). Segundo Jurafsky e Martin (2024a), se aplicarmos a função sigmoide para a soma ponderada z , teremos um número entre 0 e 1. Para torná-la uma probabilidade, precisamos que $P(y = 1|x) + P(y = 0|x) = 1$. Portanto, temos:

$$P(y = 1|x) = \sigma(\mathbf{w} \cdot \mathbf{X} + b) = \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{X} + b))}$$

$$P(y = 0|x) = 1 - \sigma(\mathbf{w} \cdot \mathbf{X} + b) = \frac{\exp(-(\mathbf{w} \cdot \mathbf{X} + b))}{1 + \exp(-(\mathbf{w} \cdot \mathbf{X} + b))}$$

Por fim, a entrada para a função sigmoide, isso é, o valor z , é frequentemente chamada de logit. Isso ocorre porque a função logit é o inverso da função sigmoide. A função logit é o log da razão de chances $\frac{p}{1-p}$, conforme mostrado na Equação 4.8:

$$\text{logit}(p) = \sigma^{-1}(p) = \ln \left(\frac{p}{1-p} \right) \quad (4.8)$$

O processo de treinamento da regressão logística envolve uma função objetiva para aprendizado, que visa minimizar os erros e otimizar o desempenho do modelo. A função objetiva utilizada é a perda de entropia cruzada. A função $L(\hat{y}, y)$ mede o quão próximo \hat{y} , a classe estimada, está de y , a classe verdadeira (JURAFSKY; MARTIN, 2024a).

Como a classe estimada possui apenas dois valores possíveis, 0 ou 1, trata-se de uma distribuição de Bernoulli. Ao aplicarmos a função logarítmica na função de densidade

de probabilidade de Bernoulli, obtemos a função log-verossimilhança (DOBSON; BARNETT, 2018). Essa transformação é útil matematicamente e quaisquer valores que maximizem uma probabilidade, também irão maximizar o log da probabilidade (JURAFSKY; MARTIN, 2024a). Essa transformação é expressa na Equação 4.9.

$$\log(p(y|x)) = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \quad (4.9)$$

Para transformá-la em uma função de perda, aplica-se o sinal negativo, pois é algo a ser minimizado (JURAFSKY; MARTIN, 2024a). Dessa forma, a função de perda de entropia cruzada é definida pela fórmula expressa na Equação 4.10.

$$L_{CE}(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (4.10)$$

A otimização da função de perda ocorre através do gradiente descendente. Seu objetivo é encontrar o conjunto de pesos que irá minimizar a função de perda (JURAFSKY; MARTIN, 2024a). Na Equação 4.11, tem-se a representação explícita de que a função de perda L é parametrizada pelos pesos, geralmente definidos por θ . No caso da regressão logística, $\theta = (\mathbf{w}, \mathbf{b})$ (JURAFSKY; MARTIN, 2024a).

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{m} \sum_{i=1}^m L_{CE}(f(x^{(i)}; \theta), y^{(i)}) \quad (4.11)$$

O gradiente descendente é um método que encontra o mínimo de uma função ao descobrir, no espaço de parâmetros θ , a direção em que a inclinação da função está aumentando de forma mais acentuada. Então, ele se move na direção oposta (JURAFSKY; MARTIN, 2024a).

4.5 Métricas de Avaliação

Uma vez que o modelo é treinado, o próximo passo consiste em obter as suas predições em uma base de teste. A base de teste é composta por objetos rotulados, mas que não foram utilizados na construção do modelo. A partir das predições que o modelo faz na base de teste, tona-se possível construir uma matriz de confusão. A avaliação dos classificadores se dará pela acurácia, precisão, recall e F1-Score, métricas de avaliação calculadas a partir da matriz de confusão (JURAFSKY; MARTIN, 2024c). A estrutura da matriz de confusão utilizada neste trabalho é apresentada na Tabela 3, onde cada elemento da matriz corresponde à frequência absoluta de classificações. A matriz de confusão possui as seguintes classificações:

1. **VN**: Verdadeiro Negativo (classificou que não era persuasivo e acertou).

2. **FP**: Falso Positivo (classificou que era persuasivo e errou).
3. **FN**: Falso Negativo (classificou que não era persuasivo e errou).
4. **VP**: Verdadeiro Positivo (classificou que era persuasivo e acertou).

Tabela 3 – Estrutura da Matriz de Confusão

| | Não persuasivo | Persuasivo |
|----------------|----------------|------------|
| Não persuasivo | VN | FP |
| Persuasivo | FN | VP |

Fonte: Elaboração própria.

A acurácia mede a proporção de predições corretas feita pelo modelo, ou seja, a quantidade de exemplos que foram corretamente classificados em relação ao total de exemplos, conforme exibido na Equação 4.12.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.12)$$

A precisão indica a proporção de exemplos positivos classificados corretamente pelo modelo em relação a todos os exemplos que o modelo classificou como positivos, conforme a Equação 4.13.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (4.13)$$

O recall mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo em relação a todos os exemplos que são realmente positivos, conforme evidenciado na Equação 4.14.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (4.14)$$

Por fim, a medida F1-Score é média harmônica da precisão e do recall, fornecendo uma métrica que balanceia precisão e recall, conforme exibido na Equação ??.

$$\text{F1 Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4.15)$$

5 RESULTADOS

Neste capítulo são apresentados os resultados obtidos a partir dos experimentos realizados conforme a metodologia previamente descrita. Os experimentos foram realizados com o objetivo de criar e avaliar diferentes modelos de classificação para detectar a presença de conteúdo persuasivo no texto do meme. Para isso, a base de treino foi utilizada para treinar os classificadores e a base de teste foi utilizada para estimar o desempenho preditivo dos modelos.

Para resolver o problema do desbalanceamento da base, utilizou-se os métodos de balanceamento de base descritos na metodologia. Foram utilizados os atributos preditivos apresentados na seção metodológica, que foram obtidos através da engenharia de características. A partir desses atributos, diferentes combinações foram realizadas nos experimentos. A seguir, apresentam-se os atributos utilizados em cada um dos experimentos.

1. Atributos utilizados por Cruz et al. (2019), desconsiderando a medida TF-IDF.
2. Atributos utilizados por Cruz et al. (2019) com a medida TF-IDF para os 50 bigramas mais frequentes.
3. Atributos utilizados por Cruz et al. (2019) com a medida TF-IDF para os 50 unigramas mais frequentes.

5.1 Resultados Gerais

No experimento 1 podemos obter o efeito isolado dos atributos gerados pela engenharia de características, sem considerar o TF-IDF. Os experimentos 2 e 3 nos fornecem o efeito conjunto dos atributos e a medida TF-IDF, com bigramas e unigramas, respectivamente. A Tabela 4 a seguir exibe os resultados obtidos no experimento 1, onde não foi incluída a medida TF-IDF.

No experimento 1, ao se utilizar *undersampling*, a Floresta Aleatória se destaca com uma acurácia de 0,6940 e um F1-Score de 0,7916, superando tanto a Árvore de Decisão quanto a Regressão Logística. A alta precisão, de 0,9366, indica que a Regressão Logística tem um baixo índice de falsos positivos.

No cenário de *oversampling*, todos os classificadores apresentam uma melhoria na acurácia, recall e F1-Score. Novamente, a Floresta Aleatória lidera com uma acurácia de 0,8020 e um F1-Score de 0,8833. A Árvore de Decisão também apresenta bons resultados, com uma acurácia de 0,7730, precisão de 0,8791 e um F1-Score de 0,8638. A Regressão

Tabela 4 – Resultados do Experimento 1 – Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, sem utilizar TF-IDF

| | Classificador | Acurácia | Precisão | Recall | F1-Score |
|---------------|---------------------|---------------|---------------|---------------|---------------|
| Undersampling | Árvore de Decisão | 0,6510 | 0,9091 | 0,6517 | 0,7591 |
| | Floresta Aleatória | 0,6940 | 0,9311 | 0,6884 | 0,7916 |
| | Regressão Logística | 0,6520 | 0,9366 | 0,6303 | 0,7535 |
| Oversampling | Árvore de Decisão | 0,7730 | 0,8749 | 0,8531 | 0,8638 |
| | Floresta Aleatória | 0,8020 | 0,8791 | 0,8874 | 0,8833 |
| | Regressão Logística | 0,6660 | 0,9337 | 0,6505 | 0,7668 |

Fonte: Elaboração própria.

Logística, embora tenha melhorado em relação ao cenário com *undersampling*, ficou atrás com uma acurácia de 0,6660 e um F1-Score de 0,7668.

Em uma análise global do experimento 1, a Floresta Aleatória apresenta os maiores valores de acurácia, recall e F1-Score no cenário de *oversampling*. A Árvore de Decisão, que é um método mais simples, alcançou resultados próximos dos resultados obtidos pela Floresta Aleatória com o *oversampling*. Esse é um ponto relevante, dado que a Árvore de Decisão gera um modelo interpretável e de fácil entendimento. Já a Regressão Logística vence na precisão, no cenário de *undersampling*. O experimento ainda foi realizado em um cenário com balanceamento SMOTE (CHAWLA et al., 2002), técnica de balanceamento bastante tradicional que gera observações sintéticas da classe minoritária. Porém, esta técnica obteve resultados inferiores e foi desconsiderada deste trabalho.

A Tabela 5 a seguir apresenta os resultados do experimento 2, incluindo a medida TF-IDF com os 50 bigramas mais frequentes. Esse experimento utiliza os mesmos atributos gerados por Cruz, Rocha e Cardoso (2019).

Neste experimento, no cenário com *undersampling*, a Floresta Aleatória obtém a maior acurácia, recall e F1-Score. Contudo, com exceção da Regressão Logística, os desempenhos dos modelos foram inferiores ao cenário com *oversampling*. A Regressão Logística vence na precisão, com um valor de 0,9427, apenas um pouco maior do que o obtido no experimento 1.

Quando o *oversampling* é aplicado, a Árvore de Decisão e a Floresta Aleatória

Tabela 5 – Resultados do Experimento 2 - Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, utilizando TF-IDF com os 50 bigramas mais frequentes

| | Classificador | Acurácia | Precisão | Recall | F1-Score |
|---------------|---------------------|---------------|---------------|---------------|---------------|
| Undersampling | Árvore de Decisão | 0,6370 | 0,8962 | 0,6446 | 0,7498 |
| | Floresta Aleatória | 0,6930 | 0,9269 | 0,6908 | 0,7916 |
| | Regressão Logística | 0,6660 | 0,9427 | 0,6434 | 0,7648 |
| Oversampling | Árvore de Decisão | 0,7870 | 0,8852 | 0,8590 | 0,8719 |
| | Floresta Aleatória | 0,7980 | 0,8759 | 0,8863 | 0,8810 |
| | Regressão Logística | 0,6630 | 0,9424 | 0,6398 | 0,7622 |

Fonte: Elaboração própria.

apresentam fortes aumentos de acurácia e F1-Score. Neste cenário, a Árvore de Decisão alcança um valor de 0,7870 na acurácia e 0,8719 no F1-Score. Já a Floresta Aleatória, 0,7970 e 0,8810, respectivamente, o que a coloca como o classificador com os maiores valores nessas duas métricas. A Regressão Logística obtém valores levemente menores na acurácia e F1-Score, com 0,6630 e 0,7622, respectivamente. Na precisão, ela consegue uma leve melhora, atingindo o valor de 0,9424.

Analisando todo o experimento 2 conjuntamente, destacam-se os resultados obtidos pela Floresta Aleatória no cenário com *oversampling*. Ela obteve uma acurácia de 0,7980, um recall de 0,8863 e um F1-Score de 0,8810, sendo esses os maiores valores dessas métricas no experimento. A Regressão Logística se destaca com a maior precisão de todas, com um valor de 0,9427, no cenário com *undersampling*.

A Tabela 6 a seguir apresenta os resultados do experimento 3, incluindo a medida TF-IDF com os 50 unigramas mais frequentes

No experimento 3, ao utilizar do *undersampling*, a acurácia, recall e F1-Score de todos os modelos são inferiores em relação ao cenário de *oversampling*. A precisão dos modelos, contudo, foi superior. A Regressão Logística, novamente, apresenta destaque com sua alta precisão de 0,9472.

Com a aplicação do balanceamento por *oversampling*, todos os classificadores apresentam melhoria na acurácia, recall e F1-Score. Novamente, a Floresta Aleatória demons-

Tabela 6 – Resultados do Experimento 3 - Avaliação de Modelos com Atributos Obtidos por Engenharia de Características, utilizando TF-IDF com os 50 unigramas mais frequentes

| | Classificador | Acurácia | Precisão | Recall | F1-Score |
|---------------|---------------------|---------------|---------------|---------------|---------------|
| Undersampling | Árvore de Decisão | 0,6320 | 0,9007 | 0,6339 | 0,7441 |
| | Floresta Aleatória | 0,6980 | 0,9288 | 0,6955 | 0,7954 |
| | Regressão Logística | 0,6640 | 0,9472 | 0,6374 | 0,7620 |
| Oversampling | Árvore de Decisão | 0,7810 | 0,8834 | 0,8531 | 0,8680 |
| | Floresta Aleatória | 0,8130 | 0,8878 | 0,8910 | 0,8894 |
| | Regressão Logística | 0,6720 | 0,9433 | 0,6505 | 0,7700 |

Fonte: Elaboração própria.

tra um desempenho superior em relação aos outros classificadores, atingindo a maior acurácia, com 0,8130, e o maior F1-Score, com 0,8894. Já a Árvore de Decisão, dessa vez, ultrapassa a Regressão Logística e atinge o segundo maior F1-Score entre os classificadores, com um total de 0,8680 contra 0,7700 da Regressão Logística. Porém, novamente, a Regressão Logística se destaca com sua alta precisão, obtendo um valor de 0,9433, levemente inferior ao que obteve no cenário com *undersampling*.

Analisando o experimento 3 como um todo, o maior destaque vai para a Floresta Aleatória com balanceamento por *oversampling*. Ela apresenta os maiores valores de acurácia, recall e F1-Score para todo o experimento 3, se destacando com o valor de 0,8894 na medida F1-Score.

Observando todos os 3 experimentos, podemos ver resultados muito próximos entre eles, com a Floresta Aleatória apresentando os melhores resultados no cenário com *oversampling* do experimento 3. Ela alcançou uma acurácia de 0,8130, um recall de 0,8910 e um F1-Score de 0,8894. Foram os maiores valores para essas métricas entre todos os classificadores em todos os experimentos realizados. Porém, ao levarmos em consideração a proximidade dos resultados, é interessante considerar o experimento 1 como o melhor entre eles, pois trata-se de um cenário em que os atributos gerados pela engenharia de características são de maior simplicidade e melhor entendimento. Além disso, embora a Floresta Aleatória apresente o melhor desempenho, também é válido notar que os resultados com Árvore de Decisão foram bastante próximos. A Árvore de Decisão é um modelo

interpretável e é de fácil entendimento o processo feito por ela para classificar uma nova observação. Então, por uma questão de maior poder interpretativo, seus resultados neste trabalho possuem um destaque em relação aos outros. Já a Regressão Logística obteve a maior precisão de todos os experimentos no cenário com *undersampling*, também no experimento 3, alcançando um valor de 0,9472.

5.2 Comparação com Modelos Gerados sem Balanceamento de Classes

Os resultados apresentados na seção anterior, onde foram aplicadas técnicas de balanceamento de classes, evidenciam uma tentativa de melhoria dos resultados originais, sem balanceamento. Conforme apresentado nos capítulos de revisão bibliográfica e metodologia, uma base desbalanceada pode levar o modelo a favorecer previsões na classe majoritária (CRUZ; ROCHA; CARDOSO, 2019), sendo esse o caso da base de treino utilizada neste trabalho.

O experimento 1, com os atributos gerados pela engenharia de características e sem considerar a medida TF-IDF, alcança resultados próximos aos outros experimentos. Porém, é o experimento de mais fácil entendimento, pois são atributos facilmente interpretáveis que foram extraídos do texto. Portanto, esse experimento será utilizado como base de comparação ao cenário sem balanceamento. A Tabela 7 a seguir exibe os resultados do experimento 1 sem nenhum balanceamento de base.

Tabela 7 – Resultados do Experimento 1 – Sem Balanceamento

| Classificador | Acurácia | Precisão | Recall | F1-Score |
|---------------------|----------|----------|--------|----------|
| Árvore de Decisão | 0,7770 | 0,8755 | 0,8578 | 0,8666 |
| Floresta Aleatória | 0,8170 | 0,8644 | 0,9289 | 0,8955 |
| Regressão Logística | 0,8430 | 0,8523 | 0,9846 | 0,9137 |

Fonte: Elaboração própria.

Ao analisar os resultados e comparar com os cenários onde *undersampling* e *oversampling* foram aplicados, pode se ter a conclusão de que o cenário sem balanceamento algum é melhor e mais promissor, dado que os valores de acurácia, recall e F1-Score são superiores. Contudo, pode se tratar de uma interpretação equivocada. Neste cenário, é importante analisar não só os valores das métricas, mas o contexto em que esse experimento ocorreu.

Ao realizar o experimento em uma base que não passou pelo processo de *under-sampling* ou *oversampling*, há um forte desbalanceamento. A base de treino possui um total de 78% das observações com textos de memes classificados como persuasivos, contra 22% de textos não persuasivos. Conforme apontado por Cruz, Rocha e Cardoso (2019), isso tende a favorecer previsões na classe majoritária, nesse caso, criando um viés para classificar novos textos de memes como persuasivos.

Assim como a base de treino, a base de teste também possui a maior parte das observações com textos persuasivos, representando 84% das observações. Essa alta proporção, aliada com um viés dos modelos para classificar textos como persuasivos, tendem a levar a um cenário em que os modelos terão um alto índice de acertos. Porém, dado o contexto, isso não representa um bom resultado.

De forma a ilustrar esta análise, a Tabela 8 a seguir apresenta a matriz de confusão gerada para a Regressão Logística, modelo com o maior F1-Score no cenário sem balanceamento.

Tabela 8 – Matriz de Confusão (Experimento 1 Sem Balanceamento – Regressão Logística)

| | Não persuasivo | Persuasivo |
|----------------|----------------|------------|
| Não persuasivo | 12 (VN) | 144 (FP) |
| Persuasivo | 13 (FN) | 831 (VP) |

Fonte: Elaboração própria.

Na matriz de confusão, as linhas representam a classe verdadeira e as colunas representam a classe predita. Como se pode observar, quase todas as observações da base de teste foram preditas como texto persuasivo. Com 84% da base sendo de fato texto persuasivo, a quantidade de acertos é bastante alta. Porém, isso não indica um bom modelo, mas apenas evidencia o viés gerado pela base não balanceada. A Tabela 9 a seguir evidencia a forte diferença na matriz de confusão para a Regressão Logística no cenário com balanceamento por *oversampling*. Foi escolhido o cenário com *oversampling* para a comparação por este ter apresentado os melhores resultados.

Tabela 9 – Matriz de Confusão (Experimento 1 com Oversampling – Regressão Logística)

| | Não persuasivo | Persuasivo |
|----------------|----------------|------------|
| Não persuasivo | 117 (VN) | 39 (FP) |
| Persuasivo | 295 (FN) | 549 (VP) |

Fonte: Elaboração própria.

Portanto, embora apresente métricas elevadas, conclui-se que não se trata de uma boa classificação e que o balanceamento da base é uma importante estratégia para se ter resultados mais promissores e confiáveis.

5.3 Análise da Importância dos Atributos com Árvore de Decisão

Os modelos treinados com Árvore de Decisão obtiveram um desempenho preditivo muito próximo da Floresta Aleatória. Diferentemente da Floresta Aleatória, a Árvore de Decisão gera um modelo diretamente interpretável, em uma estrutura gráfica, de forma é apresentado de forma clara o que levou o modelo a fazer uma determinada classificação. Esta seção tem por objetivo analisar a importância dos atributos da Árvore de Decisão do experimento 1, no cenário com *oversampling*. Este experimento foi escolhido pela maior simplicidade nos atributos utilizados e por seu bom desempenho. O cenário com *oversampling* foi escolhido por apresentar os melhores resultados do experimento.

A Árvore de Decisão gerada possui diferentes regras, onde cada regra leva a classificação do texto do meme como persuasivo ou não persuasivo. Uma elevada quantidade de regras foi gerada, onde 1.263 regras possuem 100% de confiança, ou sejam, classificam perfeitamente um texto como persuasivo ou não persuasivo. Dado o elevado número de regras definidas, serão apresentadas as duas regras com maior cobertura para os textos persuasivos e as duas regras com maior cobertura para os textos não persuasivos, todas com 100% de confiança.

Como a análise está sendo feita para o cenário com *oversampling*, é importante lembrar que estamos lidando com uma base de treino que contém mais observações do que teria sem a aplicação dessa técnica de balanceamento. Nesse contexto, a base de treino possui um total de 11.472 textos de memes, devido à replicação de textos da classe minoritária.

Primeiramente, apresentam-se as regras com maior cobertura para a classificação de textos como persuasivos. A primeira regra cobre em 525 textos persuasivos, enquanto a segunda, cobre 187 textos persuasivos, todos provenientes da base de treino.

- SE (total de caracteres $> 125,5$) e ($4,005 < \text{média de caracteres por palavra} \leq 5,544$) e ($\text{média de caracteres por frase} < 27,829$) e ($1,349 < \text{variância de caracteres por palavra} \leq 7,933$) e ($3,5 < \text{frequência de pontuação} \leq 9,5$) e ($\text{frequência de letras maiúsculas} > 6,5$) e ($\text{proporção de tokens sobre palavras lematizadas} > 0,489$), ENTÃO (Texto = “Persuasivo”).
- SE (total de caracteres $> 325,5$) e ($\text{média de caracteres por palavra} > 4,005$) e ($\text{média de caracteres por frase} > 27,829$) e ($\text{variância de caracteres por frase}$

$\leq 51.948,834$) e (frequência de pontuação $> 9,5$) e (proporção de tokens sobre palavras lematizadas $> 0,489$), ENTÃO: (Texto = “Persuasivo”).

Por fim, apresentam-se as regras com maior cobertura para a classificação de textos como não persuasivos. A primeira regra foi baseada em 78 textos não persuasivos, enquanto a segunda, cobre 70 textos não persuasivos, todos provenientes da base de treino.

- SE (total de caracteres $\leq 32,5$) e ($4,9 < \text{média de caracteres por palavra} \leq 5,708$) e (variância de caracteres por palavra $\leq 10,32$) e (frequência de pontuação $\leq 4,5$) e ($1,5 < \text{frequência de letras maiúsculas} \leq 8,5$) e (frequência de pontuação $> 0,5$) e (variância de caracteres por frase $\leq 44,5$), ENTÃO (Texto = “Não persuasivo”).
- SE (total de caracteres $\leq 32,5$) e (frequência de letras maiúsculas $\leq 0,5$) e (frequência de pontuação $\leq 4,5$) e (variância de caracteres por frase $\leq 44,5$), ENTÃO (Texto = “Não persuasivo”).

Essas regras cobrem um total de 860 textos de memes, o que representa um total de aproximadamente 7,5% da base de treino balanceada por *oversampling*. Podemos ver pelas regras a diferença entre textos persuasivos e não persuasivos, por exemplo, no atributo “total de caracteres”. As regras exibidas nos mostram que um elevado valor de caracteres tende a levar o texto a ser persuasivo.

Para uma análise comparativa, as tabelas a seguir apresentam as principais estatísticas dos atributos para os textos persuasivos, na Tabela 10, e para os textos não persuasivos, na Tabela 11.

Alguns atributos possuem diferenças mais significativas entre os textos persuasivos e não persuasivos. Conforme as regras apresentadas da Árvore de Decisão sugeriram, textos persuasivos possuem, em média, 126% mais caracteres do que textos não persuasivos. Essa diferença também pode ser vista na quantidade máxima de caracteres, onde os textos persuasivos representam um aumento de 195% em relação aos textos não persuasivos. O desvio padrão desse atributo em textos não persuasivos, menor que a metade para textos persuasivos, indica uma dispersão menor dessa variável nesse contexto, fortalecendo a regra de que quanto menor for o total de caracteres no texto, mais propenso ele é a ser não persuasivo.

O atributo “frequência de letras maiúsculas” também apresenta diferenças mais notáveis entre as duas classificações. Textos persuasivos possuem, em média, quase o dobro da quantidade de letras maiúsculas de textos não persuasivos. O valor máximo desse atributo, por exemplo, aumenta em 220% nos textos persuasivos. Essas estatísticas

Tabela 10 – Estatísticas dos Atributos (textos persuasivos)

| Atributo | Média | Desvio padrão | Mínimo | Mediana | Máximo |
|---|--------------|--------------------------|---------------|----------------|---------------|
| Total de frases | 1,72 | 1,33 | 1,00 | 1,00 | 18,00 |
| Média de caracteres por frase | 80,80 | 57,94 | 4,00 | 67,50 | 838,00 |
| Variância de caracteres por frase | 665,04 | 2715,55 | 0,00 | 0,00 | 105625,00 |
| Total de caracteres | 124,90 | 108,38 | 4,00 | 94,00 | 1511,00 |
| Média de caracteres por palavra | 4,73 | 0,90 | 2,67 | 4,63 | 26,50 |
| Variância de caracteres por palavra | 5,27 | 2,91 | 0,00 | 4,91 | 33,25 |
| Frequência de pontuação | 4,15 | 4,75 | 0,00 | 3,00 | 59,00 |
| Frequência de letras maiúsculas | 42,00 | 49,59 | 0,00 | 25,00 | 769,00 |
| Proporção de tokens por palavras lematizadas | 0,88 | 0,12 | 0,09 | 0,91 | 1,00 |

Fonte: Elaboração própria.

Tabela 11 – Estatísticas dos Atributos (textos não persuasivos)

| Atributo | Média | Desvio padrão | Mínimo | Mediana | Máximo |
|---|--------------|--------------------------|---------------|----------------|---------------|
| Total de frases | 1,28 | 0,72 | 1,00 | 1,00 | 7,00 |
| Média de caracteres por frase | 44,94 | 30,95 | 2,00 | 39,00 | 271,00 |
| Variância de caracteres por frase | 126,90 | 1759,70 | 0,00 | 0,00 | 61009,00 |
| Total de caracteres | 55,10 | 43,67 | 2,00 | 46,00 | 513,00 |
| Média de caracteres por palavra | 4,74 | 1,28 | 2,00 | 4,50 | 15,00 |
| Variância de caracteres por palavra | 4,45 | 5,29 | 0,00 | 3,45 | 69,00 |
| Frequência de pontuação | 1,95 | 2,87 | 0,00 | 1,00 | 35,00 |
| Frequência de letras maiúsculas | 21,31 | 24,44 | 0,00 | 12,00 | 240,00 |
| Proporção de tokens por palavras lematizadas | 0,93 | 0,11 | 0,42 | 1,00 | 1,00 |

Fonte: Elaboração própria.

também reforçam as regras da Árvore de Decisão, que indicam que um número maior de letras maiúsculas no texto os aproxime da classificação como persuasivo.

6 CONCLUSÃO

Este trabalho abordou a classificação de persuasão em textos de memes, utilizando uma abordagem baseada em engenharia de características. Para isso, utilizou-se dos modelos de Árvore de Decisão, Floresta Aleatória e Regressão Logística. A exploração do problema de classificação de persuasão em textos de memes é algo novo, levando este trabalho a realizar este estudo em um contexto mais desafiador do que a classificação de persuasão em outros tipos de texto, como artigos de notícias, por exemplo.

A engenharia de características foi empregada para extrair atributos numéricos dos textos presentes nos memes, visando sua aplicação em algoritmos de classificação. Os atributos utilizados foram baseados na metodologia descrita por [Cruz, Rocha e Cardoso \(2019\)](#), que realizaram a classificação de persuasão em textos de artigos. No contexto dos textos de memes, estamos lidando com conteúdos mais curtos e informais.

Os experimentos realizados apontaram que os resultados alcançados pela Árvore de Decisão e Floresta Aleatória foram, em geral, superiores do que a Regressão Logística. A Regressão Logística obteve um desempenho melhor apenas na precisão, enquanto a Árvore de Decisão e Floresta aleatória tiveram melhores acurácia, recall e F1-Score. A Árvore de Decisão e a Floresta Aleatória tiveram desempenhos aproximados entre si. Neste caso, a vantagem da Árvore de Decisão se dá pelo fato de se tratar de um modelo facilmente interpretável, com regras claras para a classificação.

Três diferentes experimentos foram realizados, com diferentes atributos e balanceamentos. O experimento 3, no cenário com *oversampling*, apresentou três métricas de avaliação com os maiores valores entre todos os experimentos. Esses resultados foram alcançados com a Floresta Aleatória, que obteve uma acurácia de 0,8130, um recall de 0,8910 e um F1-Score de 0,8894. Contudo, por terem desempenhos próximos entre si, o experimento 1 apresenta destaque em relação aos demais, por sua maior simplicidade e clareza nos atributos utilizados. Neste caso, utilizaram-se os atributos propostos por Cruz et al. (2019), mas sem a inclusão da medida TF-IDF. Neste experimento, o cenário com balanceamento por *oversampling* se saiu melhor.

No experimento 1, com *oversampling*, a Floresta Aleatória aparece com o melhor desempenho, alcançando uma acurácia de 0,8020 e um F1-Score de 0,8833. A Árvore de Decisão alcança valores próximos, com uma acurácia de 0,7730 e um F1-Score de 0,8638. A Regressão Logística se destaca na precisão, com um valor de 0,9337. Embora a Floresta Aleatória tenha um desempenho levemente melhor do que a Árvore de Decisão, destaca-se a vantagem da Árvore de Decisão por sua interpretabilidade.

O balanceamento de classes se mostrou uma importante estratégia para gerar um

equilíbrio nos dados, dividindo a base de treino com 50% de observações como textos persuasivos e 50% de observações como textos não persuasivos. Isso gerou um maior equilíbrio entre falsos positivos e falsos negativos, uma vez que diminuiu a tendência do modelo em classificar novas observações como persuasivas. Antes do balanceamento, a maioria das observações da base de teste era classificada como persuasiva. Como a maior parte de base de teste é, de fato, persuasiva, isso gerava um alto valor de falsos positivos, bem como de verdadeiros positivos.

Por fim, apresenta-se a seguir ideias para trabalhos futuros, tais como:

1. Criação de um sistema em duas etapas. A primeira etapa, realizada neste trabalho, classifica o texto como persuasivo ou não. A segunda etapa, então, consiste em classificar qual ou quais técnicas de persuasão foram utilizadas nos memes classificados como persuasivos na primeira etapa.
2. Gerar novos atributos preditivos a partir do texto, como por exemplo, elementos semânticos. Desta forma, pode-se incluir atributos como total de adjetivos, total de verbos, entre outros.
3. Na geração de novos atributos pode-se, ainda, criar listas com palavras positivas ou negativas, gerando uma contagem para cada caso e utilizando esses valores como atributos. Essas listas podem ser criadas a partir de adjetivos ou de palavras específicas que façam parte do contexto do problema.
4. Avaliação de embeddings semânticos como atributos, método que já apresentou bons resultados em outros trabalhos recentes. Contudo, neste caso, há a desvantagem de não ser um método interpretável, diferentemente da Árvore de Decisão, por exemplo.
5. Criação de um léxico com unigramas e bigramas frequentemente presentes em textos persuasivos, podendo utilizar a frequência absoluta de cada caso como atributo preditivo.
6. Balancear a base de treino com diferentes proporções entre as duas classes, além da divisão realizada de 50% para persuasão e 50% para não persuasão.

REFERÊNCIAS

ALBORNOZ, J. Carrillo-de; VIDAL, J. R.; PLAZA, L. Feature engineering for sentiment analysis in e-health forums. *PloS one*, Public Library of Science San Francisco, CA USA, v. 13, n. 11, p. e0207996, 2018. Citado na página 16.

ALI, J. et al. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, International Journal of Computer Science Issues (IJCSI), v. 9, n. 5, p. 272, 2012. Citado 2 vezes nas páginas 33 e 37.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <<http://dx.doi.org/10.1023/A%3A1010933404324>>. Citado 2 vezes nas páginas 14 e 37.

BREIMAN, L. et al. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN 9780412048418. Disponível em: <<https://books.google.com.br/books?id=JwQx-WOmSyQC>>. Citado 3 vezes nas páginas 14, 34 e 35.

CASELI, H. M.; NUNES, M. G. V. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. 2. ed. BPLN, 2024. ISBN 978-65-00-95750-1. Disponível em: <<https://brasileiraspln.com/livro-pln/2a-edicao/>>. Citado na página 12.

CHAGAS, V. Da memética aos memes de internet: uma revisão da literatura. *BIB - Revista Brasileira de Informação Bibliográfica em Ciências Sociais*, n. 95, mar. 2021. Disponível em: <<https://bibanpocs.emnuvens.com.br/revista/article/view/119>>. Citado na página 11.

CHAWLA, N. V. et al. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, v. 16, p. 321–357, 2002. Citado na página 43.

CRUZ, A. F.; ROCHA, G.; CARDOSO, H. L. On sentence representations for propaganda detection: From handcrafted features to word embeddings. In: FELDMAN, A. et al. (Ed.). *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 107–112. Disponível em: <<https://aclanthology.org/D19-5015>>. Citado 8 vezes nas páginas 13, 16, 29, 33, 43, 46, 47 e 53.

DAWKINS, R. *The Selfish Gene*. [S.l.]: Oxford University Press, 1976. 224 p. Citado na página 11.

DOBSON, A. J.; BARNETT, A. G. *An introduction to generalized linear models*. [S.l.]: Chapman and Hall/CRC, 2018. Citado na página 40.

FERREIRA, B. P. J. M.; DALIP, D. H.; SILVA, I. S. Análise e classificação de gêneros musicais com base em letras de músicas. In: *Proceedings of the 38th Brazilian Symposium on Databases*. Belo Horizonte, Minas Gerais, Brasil: Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), 2023. Disponível em: <<https://github.com/>>

[bpoliana/music-genre-analysis/blob/master/18k_features_1_17_normalized.csv](#)>. Citado na página 16.

FINATTO, M. J. B. et al. Sequência de caracteres e palavras: Morfologia e morfossintaxe. In: CASELI, H. M.; NUNES, M. G. V. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. 2. ed. BPLN, 2024. book chapter 4. ISBN 978-65-00-95750-1. Disponível em: <<https://brasileiraspln.com/livro-pln/2a-edicao/parte-palavras/cap-caracteres-palavras/cap-caracteres-palavras.html>>. Citado na página 28.

FREITAS, A. A. Comprehensible classification models: A position paper. *ACM SIGKDD Explorations Newsletter*, v. 15, n. 1, p. 1–10, 2014. Citado na página 14.

GUALBERTO, E. S.; AUTHORS other. The answer is in the text: Multi-stage methods for phishing detection based on feature engineering. *IEEE Access*, v. 8, p. 223529–223547, 2020. Citado na página 16.

JURAFSKY, D.; MARTIN, J. H. Logistic regression. In: *Speech and Language Processing*. 3. ed. Palo Alto: Stanford, 2024. cap. 5, p. 1–25. Feb. 3, 2024 draft. Citado 5 vezes nas páginas 14, 28, 38, 39 e 40.

JURAFSKY, D.; MARTIN, J. H. N-gram language models. In: *Speech and Language Processing*. 3. ed. Palo Alto: Stanford, 2024. cap. 3, p. 1–2. Feb. 3, 2024 draft. Citado na página 32.

JURAFSKY, D.; MARTIN, J. H. Naive bayes, text classification, and sentiment. In: *Speech and Language Processing*. 3. ed. Palo Alto: Stanford, 2024. cap. 4. Feb. 3, 2024 draft. Citado 3 vezes nas páginas 18, 27 e 40.

JURAFSKY, D.; MARTIN, J. H. Neural networks and neural language models. In: *Speech and Language Processing*. 3. ed. Palo Alto: Stanford, 2024. cap. 7. Feb. 3, 2024 draft. Citado na página 15.

JURAFSKY, D.; MARTIN, J. H. Vector semantics and embeddings. In: *Speech and Language Processing*. 3. ed. Palo Alto: Stanford, 2024. cap. 6, p. 11–14. Feb. 3, 2024 draft. Citado na página 31.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? *Nature biotechnology*, Nature Publishing Group US New York, v. 26, n. 9, p. 1011–1013, 2008. Citado na página 33.

LEEVY, J. L. et al. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, Springer, v. 5, n. 1, p. 1–30, 2018. Citado na página 33.

LOPEZ, M. M.; KALITA, J. *Deep Learning applied to NLP*. 2017. Citado na página 12.

MARTINO, G. D. S.; BARRÓN-CEDENO, A.; NAKOV, P. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In: FELDMAN, A. et al. (Ed.). *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 162–170. Disponível em: <<https://aclanthology.org/D19-5024>>. Citado 2 vezes nas páginas 13 e 15.

MARTINO, G. D. S. et al. *A Survey on Computational Propaganda Detection*. 2020. Citado 2 vezes nas páginas 12 e 15.

Merriam-Webster. *The History of ‘Meme’*. 2023. Accessed: 2024-03-11. Disponível em: <<https://www.merriam-webster.com/wordplay/meme-word-origins-history>>. Citado na página 11.

MIHAILIDIS, P. The civic potential of memes and hashtags in the lives of young people. *Discourse: Studies in the Cultural Politics of Education*, v. 41, n. 5, p. 762–781, 2020. Citado na página 11.

MOREIRA, V. P. Recuperação da informação. In: CASELI, H.; NUNES, M. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. BPLN, 2023. cap. 16. Disponível em: <<https://brasileiraspln.com/livro-pln/1a-edicao/parte8/cap16/cap16.html>>. Citado 3 vezes nas páginas 24, 31 e 32.

NAGY, K. S.; KAPUSTA, J. Twidw—a novel method for feature extraction from unstructured texts. *Applied Sciences*, MDPI, v. 13, n. 11, p. 6438, 2023. Citado na página 16.

NARGESIAN, F. et al. Learning feature engineering for classification. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. [s.n.], 2017. p. 2529–2535. Disponível em: <<https://doi.org/10.24963/ijcai.2017/352>>. Citado 3 vezes nas páginas 12, 16 e 29.

SemEval. *SemEval2024 shared task on “Multilingual Detection of Persuasion Techniques in Memes”*. 2024. Accessed: 2024-03-11. Disponível em: <<https://propaganda.math.unipd.it/semEval2024task4/>>. Citado 2 vezes nas páginas 13 e 18.

SENO, E. R. M. et al. Semântica distribucional. In: CASELI, H. M.; NUNES, M. G. V. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. 2. ed. BPLN, 2024. book chapter 10. ISBN 978-65-00-95750-1. Disponível em: <<https://brasileiraspln.com/livro-pln/2a-edicao/parte-significado/cap-semantica-distribucional/cap-semantica-distribucional.html>>. Citado na página 15.

SILVA, V. R. da; GONÇALVES, E. C. Análise comparativa de implementações de algoritmos de árvores de decisão para aplicações no serviço público. In: SBC. *Anais da XXI Escola Regional de Computação Bahia, Alagoas e Sergipe*. [S.l.], 2021. p. 10–19. Citado na página 34.

STOJILJKOVIĆ, M. *Logistic regression in python*. Vancouver: RealPython, 2020. <<https://realpython.com/logistic-regression-python/>>. Acesso em: 23 abr. 2024. Citado na página 39.

VIDAL, T.; SCHIFFER, M. Born-again tree ensembles. In: *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*. [S.l.: s.n.], 2020. p. 9743–9753. Citado na página 14.

WANG, L.; WOOD, B. C. An epidemiological approach to model the viral propagation of memes. *Applied Mathematical Modelling*, v. 35, n. 11, p. 5442–5447, 2011. Citado na página 11.

YU, S. et al. Interpretable propaganda detection in news articles. *CoRR*, abs/2108.12802, 2021. Disponível em: <<https://arxiv.org/abs/2108.12802>>. Citado na página 15.

ZAKEM, V.; MCBRIDE, M. K.; HAMMERBERG, K. *Exploring the Utility of Memes for US Government Influence Campaigns*. 2021. Citado na página 11.

ZHENG, A.; CASARI, A. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly, 2018. ISBN 9781491953242. Disponível em: <<https://books.google.com.br/books?id=Ho0UvgAACAAJ>>. Citado 3 vezes nas páginas 12, 16 e 29.

APÊNDICE A – NOTEBOOK GOOGLE COLABORATORY

Este apêndice apresenta o código Python utilizado para o desenvolvimento deste trabalho. O código pode ser acessado pelo notebook do Google Colaboratory através do QR code ou pelo link <https://tinyurl.com/23trfpg5>.

Figura 10 – QR Code de acesso para o código Python

