João Pedro Cigliato Augusto

Identificação do discurso de ódio de cunho homofóbico a partir de métodos de Aprendizado de Máquinas

Niterói - RJ, Brasil

14 de Dezembro de 2023

Universidade Federal Fluminense

João Pedro Cigliato Augusto

Identificação do discurso de ódio de cunho homofóbico a partir de métodos de Aprendizado de Máquinas

Trabalho de Conclusão de Curso

Monografia apresentada para obtenção do grau de Bacharel em Estatística pela Universidade Federal Fluminense.

Orientador(a): Jessica Kubrusly

Niterói - RJ, Brasil

14 de Dezembro de 2023

Universidade Federal Fluminense

João Pedro Cigliato Augusto

Identificação do discurso de ódio de cunho homofóbico a partir de métodos de Aprendizado de Máquinas

Monografia de Projeto Final de Graduação sob o título *"Identificação do discurso de ódio de cunho homofóbico a partir de métodos de Aprendizado de Máquinas"*, defendida por João Pedro Cigliato Augusto e aprovada em 14 de Dezembro de 2023, na cidade de Niterói, no Estado do Rio de Janeiro, pela banca examinadora constituída pelos professores:

Profa. Dra. Jessica Kubrusly
Departamento de Estatística – UFF

Prof. Dr. Douglas Rodrigues Pinto
Departamento de Estatística – UFF

Profa. Dra. Karina Yuriko Yaginuma
Departamento de Estatística – UFF

Niterói, 14 de Dezembro de 2023

Ficha catalográfica automática - SDC/BIME
Gerada com informações fornecidas pelo autor

A923i Augusto, João Pedro Cigliato
Identificação do discurso de ódio de cunho homofóbico a partir de métodos de Aprendizados de Máquinas / João Pedro Cigliato Augusto. - 2023.
44 f.: il.

Orientador: Jessica Quintanilha Kubrusly.
Trabalho de Conclusão de Curso (graduação)-Universidade Federal Fluminense, Instituto de Matemática e Estatística, Niterói, 2023.

1. Aprendizado de máquina. 2. Mineração de texto. 3. Discurso de ódio. 4. Homofobia. 5. Produção intelectual. I. Kubrusly, Jessica Quintanilha, orientadora. II. Universidade Federal Fluminense. Instituto de Matemática e Estatística. III. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Resumo

O objetivo deste estudo é definir, por técnicas de Aprendizado de Máquinas, um classificador de discurso de ódio de cunho homofóbico para postagens na rede social *Twitter*. A base de dados utilizada é composta por *tweets* relacionados ao tema e foi utilizado técnicas de mineração de texto e e de pré-processamento para preparar esses dados para realizar classificações. Serão utilizadas técnicas de balanceamento da base dados como *undersampling*, *oversampling* e *SMOTEENN*. Dois tipos de vetorização serão analisadas: Matriz Termo Documento e TF-IDF. Para a classificação serão realizados modelos de Floresta Aleatória e *Extreme Gradient Boosting* (*XGBoost*). Ao todo, foram realizados 24 modelos no trabalho, e os modelos que obtiveram um melhor resultado foram os modelos que foram treinados com um balanceamento pela técnica de *undersampling*. Dentre esses modelos, os modelos de Floresta Aleatória realizados com vetorização de Matriz Termo Documento tiveram melhores resultados de sensibilidade, com uma média de 84,14%. Já os modelos de *XGBoost* apresentaram uma média na sensibilidade de 75,96%.

Palavras-chave: Aprendizado de máquina. Mineração de texto. Discurso de ódio. Homofobia.

Dedicatória

A minha família, que em nenhum momento deixou de acreditar em mim.

Agradecimentos

Primeiramente, gostaria de agradecer a Deus e à minha família por terem me dado a força que eu precisava para seguir em frente, sempre.

Gostaria, também, de agradecer à Universidade Federal Fluminense por ter me proporcionado uma estrutura excelente para que pudesse concluir o meu curso.

Agradeço a todos os professores que ajudaram, mesmo que apenas um pouco, para a conclusão da minha graduação. Agradecimento especial à minha orientadora deste trabalho, Jessica Kubrusly, que desde o começo me ajudou e ensinou muito.

Agradeço a todos os amigos que fiz durante todo esse processo de graduação. Também agradeço aos amigos externos à faculdade. Agradeço, especialmente, à Mila Grigorovski, que, nos últimos meses, me motivou dia após dia para que pudesse finalizar este trabalho.

Por fim, agradeço a ajuda que tive, também, dos meus companheiros de trabalho no FGV IBRE, e que sempre acreditaram em mim.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Com o passar dos anos, cresceu de forma acelerada o uso de redes sociais. Porém, no meio de tantos pontos positivos que isso poderia trazer, evidentemente que teria também os seus negativos. Más condutas vistas no dia a dia fora da internet foram, naturalmente, ganhando um espaço online.

Cada vez mais são vistos os discursos de ódio contra minorias ganhando mais força nas redes sociais e isto fez com que redes sociais como *Facebook*, *Twitter* e *Youtube* têm declarado o fenômeno do ódio online como prioridade em suas políticas de conteúdo (SILVA et al., 2019). Contudo, esta regulação tem causado questionamento de diversos segmentos da sociedade (CAPPI et al., 2017). Essas redes interligam bilhões de pessoas no mundo todo. Com isso, há de se pensar que é quase impossível controlar as atitudes de todos os usuários. Logo, fica impraticável para que se tenha um olhar humano analisando aquilo que todas essas pessoas publicam e conversam, o que torna os usuários suscetíveis a ataques de cunho odioso (COUTINHO; MALHEIROS, 2020).

Desta forma, sabe-se que existem diversas camadas de discurso de ódio presentes no dia a dia. Por conseguinte, as declarações de ódio e a discriminação contra grupos LGBTQIA+ são as mais comuns aparições nas redes sociais. Com isto, o presente trabalho terá seu foco nos tipos de discurso de ódio de cunho homofóbico que são encontrados hoje em dia na rede social *Twitter*, que, em 2013, como mostrado por Cappi et al. (2017), foi responsável por aumentar a disseminação dos discursos de ódio em 30% em comparação a anos anteriores.

O objetivo deste estudo é definir, por técnicas de Aprendizado de Máquinas, um classificador de discurso de ódio de cunho homofóbico para postagens na rede social *Twitter*. Se tem como finalidade, também, comparar resultados de técnicas diferentes para possíveis trabalhos futuros, onde se queira estudar discursos de ódio envolvendo outros temas.

O Capítulo 2 explicita para o leitor os métodos usados no trabalho, assim como o material usado. Neste capítulo serão apresentadas as técnicas de análise de sentimentos,

de classificação, como as Árvores de Decisão, a Floresta Aleatória, o *Gradient Boosting Machine* e o *XGBoost*. Assim também como os métodos para testar a qualidade desses modelos de classificação. Também é apresentada a base de dados que será usada no trabalho. Logo em seguida, no Capítulo 3, serão apresentadas discussões sobre resultados derivados das técnicas apresentadas no Capítulo 2. Por fim, no Capítulo 4, será mostrada uma conclusão sobre os resultados apresentados no Capítulo 3 e como este pode ser útil para o tema proposto e para possíveis trabalhos futuros.

2 Materiais e Métodos

Neste capítulo será apresentada a base de dados em que o estudo se desenvolverá, assim como os métodos utilizados para que se atinja o objetivo do trabalho.

2.1 Materiais

Para realizar as análises conforme o objetivo deste estudo, utilizou uma base composta por postagens da rede social *Twitter*, elaborada por Almeida e Kubrusly (2022) e pode ser encontrado no *GitHub* [1] . Esta base foi coletada no período de janeiro a agosto de 2022. No processo de criação da base, foram usados termos específicos para fazer as pesquisas de *tweets* relacionados a alguns temas de preconceito como, por exemplo, a homofobia. Portanto, neste caso, foram procuradas postagens relacionadas aos grupos LGBTQIA+. Após essa etapa, a base foi, cuidadosamente, rotulada, de forma manual, como positiva, negativa ou neutra para discurso de ódio. Nesta base cada observação é associada à um *tweet*, forma como é chamada a postagem na rede social. Esta base possui 44730 observações com 9 variáveis, dividida em alguns temas de preconceito, e como o foco deste trabalho será em volta do tema homofobia, tem-se uma base de dados, agora, de 8425 observações.

Por fim, foram retiradas variáveis que, visivelmente, não agregariam ao estudo, que foram: id do usuário, id do usuário para quem foi a resposta e a variável que indicava o tema de preconceito das postagens. As variáveis utilizadas para este estudo são explicitadas na Tabela 1.

[1] link:⟨https://github.com/jessicakubrusly/banco-discurso-odio⟩ (Consultado em Julho de 2023)

Variável	Descrição	Tipo
TWEET	Postagem publicada (texto livre)	Texto
RESPOSTA	Indicador se é uma resposta à outro usuário	Qualitativa
DISPOSITIVO	Tipo do dispositivo usado para a postagem	Qualitativa
DIA_SEMANA	Dia da semana da postagem	Qualitativa
FAIXA_HORA	Faixa de hora da postagem	Qualitativa
TIPO	Indicador se é ofensivo, não ofensivo ou discurso de ódio	Qualitativa

Tabela 1: Tabela de descrição das variáveis da base de dados

As variáveis DIA_SEMANA e FAIXA_HORA foram criadas baseadas nas variáveis DATA e HORA, respectivamente, para que se pudesse dividi-las em níveis. A variável "TIPO" inicialmente compreende três níveis distintos. As postagens rotuladas como discurso de ódio são postagens que, claramente, vê-se uma atitude odiosa por trás e, provavelmente, é uma fala criminosa. Os casos ofensivos são tratados como falas que possuem uma maldade por trás, porém não são tão pesadas quanto as de ódio. Por fim, as postagens consideradas não ofensivas são *tweets* relacionados ao tema mas que não ferem ninguém. Como este estudo tem como objetivo analisar a qualidade de modelos ao classificar que uma postagem é discurso de ódio, optou-se por consolidar os casos ofensivos e não ofensivos, reconfigurando a variável em dois níveis discretos: *tweets* categorizados como discurso de ódio e *tweets* não classificados como tal.

2.2 Análise de Sentimentos

A análise de sentimentos é uma área de estudo que busca, a partir de textos livres, identificar o sentimento expresso em um texto e classificá-lo como positivo, negativo ou neutro. Diversas técnicas são utilizadas nesta análise. As técnicas podem ser divididas em diferentes níveis (MEDHAT; HASSAN; KORASHY, 2014). Primeiramente, pode ser baseado em documento, que classifica, de forma geral, o texto inteiro, assumindo que este é apenas sobre um assunto. Baseado na sentença, que classifica cada frase encontrada. E por último, baseado no aspecto, determinando a polaridade do sentimento expressado.

Na Figura 1, pode-se ver quais os caminhos que podem ser tomados em todo o processo.

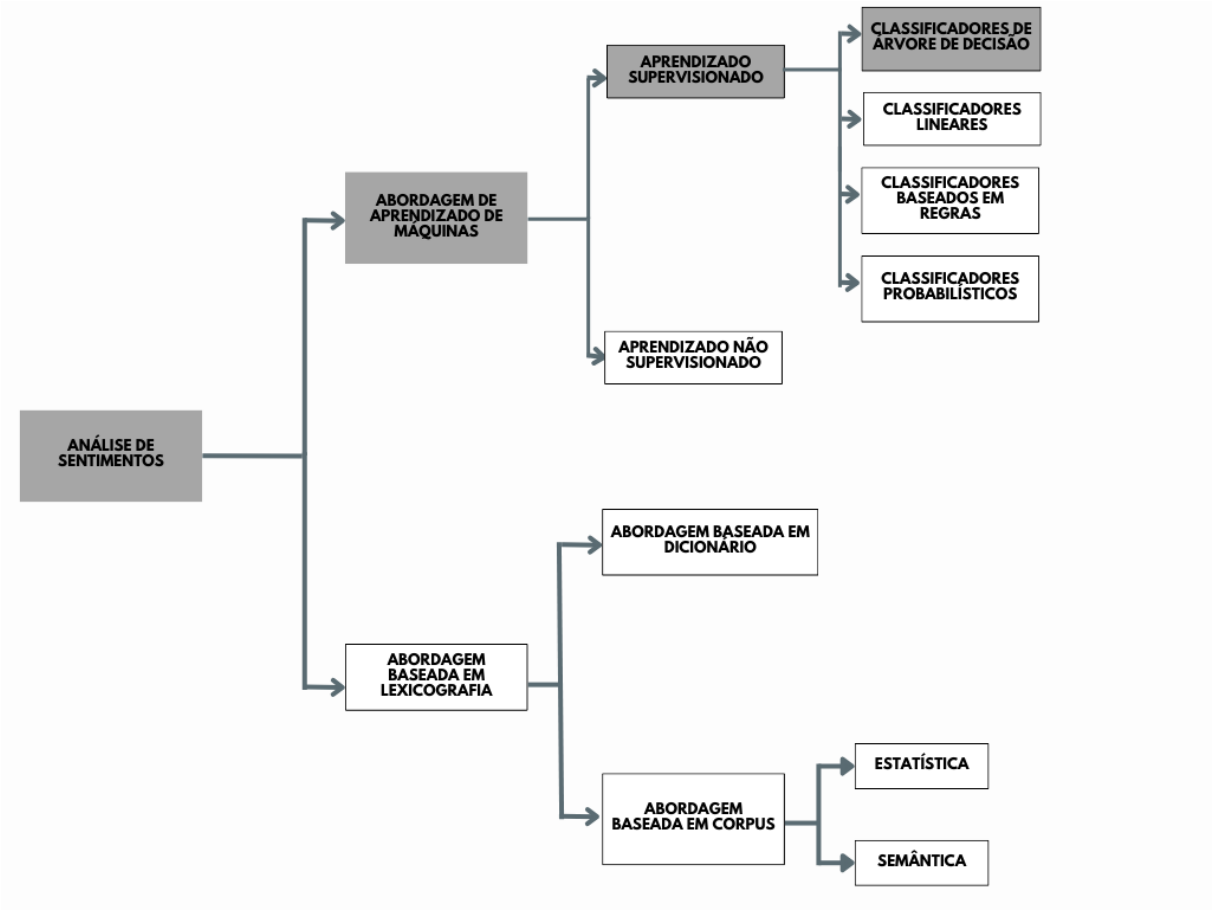

Figura 1: Fluxograma dos caminhos da Análise de Sentimentos. Fonte: Medhat, Hassan e Korashy (2014). Traduzida pelo autor do trabalho.

Durante todo o processo, várias práticas podem ser utilizadas, seja na vetorização dos documentos que serão analisados ou nos métodos de classificação, que comumente são utilizadas práticas de Aprendizado de Máquinas *(Machine Learning)*, mas também pode ser utilizado a Abordagem Baseada em Lexicografia *(Lexicon-based Approach)*.

A Abordagem Baseada em Lexicografia é uma técnica que baseia-se em um conjunto de palavras-chave e as polaridades associadas à ela (MOREO et al., 2012). Durante o processo, palavras do documento à ser estudado são comparadas com os chamados "léxicos", que contém palavras de um glossário pré-estabelecido, para definir a pontuação do sentimento gerado por aquela palavra. Tais glossários são dicionários de palavras criados para indicar a intensidade dos termos que estão sendo estudados (TABOADA et al., 2011). Este trabalho não se profundará neste método, visto que não foi o escolhido para a análise.

O caminho que será seguido no presente estudo, é o dado pela cor cinza na Figura 1.

A prática de Aprendizado de Máquinas na Análise de Sentimentos se baseia em métodos de classificação que utilizam informações linguísticas e sintáticas (MEDHAT; HASSAN; KORASHY, 2014). O caminho seguinte é utilizar algoritmos de aprendizado supervisionado, que são algoritmos que são treinados com bases já rotuladas, e, no caso deste trabalho, são algoritmos que utilizam Árvores de Decisão. Estes métodos e algoritmos serão apresentados nas seções seguintes.

2.3 Vetorização dos Documentos

Para que seja possível realizar os métodos de Aprendizado de Máquinas, o primeiro passo é vetorizar os documentos que serão utilizados no estudo. Portanto, transforma-se esses documentos, geralmente dividido por palavras, em vetores para que se possa usar técnicas computacionais (HVITFELDT; SILGE, 2021). Esse método é utilizado para que se consiga transformar a base de texto em uma linguagem que o computador entende, ou seja, uma linguagem numérica. Depois, é realizada uma mineração dos textos para detectar padrões com técnicas estatísticas (KUBRUSLY; NEVES; MARQUES, 2022).

Neste trabalho serão abordadas duas vetorizações de documentos: a partir da matriz termo documento e o TF-IDF. Porém, é preciso, primeiro, realizar um pré-processamento da base de texto para que se possa criar os dois vetores, a partir destas duas metodologias.

2.3.1 Pré-Processamento de Texto

O primeiro passo é a Tokenização, técnica que consiste em dividir um texto em *tokens*, que são unidades deste texto. Estas unidades podem ser palavras, caracteres, frases e outras características de um texto (HVITFELDT; SILGE, 2021). As palavras são mais comumente utilizadas para esta técnica (KUBRUSLY; NEVES; MARQUES, 2022). O tipo do texto a ser estudado vai influenciar muito na forma que a Tokenização será feita, principalmente em casos que se utiliza postagens de redes sociais - que é o caso deste trabalho - pois sabe-se que, em redes sociais, os usuários geralmente procuram utilizar o máximo possível de palavras contraídas e pontuações em excesso, o que pode dificultar um pouco o processo da análise e, por isso, estudar a forma do texto antes de qualquer método é crucial para lidar com possíveis desafios.

Após a Tokenização, é realizada a exclusão das chamadas *stop words* dos textos, que são palavras que comumente são utilizadas em qualquer texto e que, normalmente, não agregam semanticamente na classificação da análise de sentimentos (KUBRUSLY;

NEVES; MARQUES, 2022). Com o avanço da tecnologia e das técnicas computacionais para Análise de Sentimentos, é facilmente achado listas de *stop words*, pré-estabelecidas, que são chamadas de *stoplist*, em qualquer linguagem de computação.

Um outro procedimento utilizado nesse pré-processamento é a Lematização. A Lematização é uma prática utilizada para colocar certas palavras dentro de "lemas"para que se possa diminuir estas a uma base em comum, ainda possuindo um significado, para fins de diminuição do volume de *tokens* na base. Isto é, agrupar diferentes formas de uma mesma palavra como uma única palavra, como, por exemplo, as palavras "Análise", "Analisar", "Analisando"e "Analisado" seriam transformadas na palavra "Análise".

Com todas as etapas anteriores realizadas, é feita a seleção dos termos para retirar palavras que, de certa forma, não tenham valor semântico no contexto do estudo, ou seja, que tenham um grau alto ou baixo de frequência nos documentos (KUBRUSLY; NEVES; MARQUES, 2022). Esta seleção é feita a partir da análise da frequência dos termos dentro da base de documentos. Logo, os termos de grau baixo ou alto de frequência são excluídos. Contudo, não existe uma frequência de corte adequada, porém, que haja um número de termos que seja possível o processamento.

2.3.2 Criação da Matriz Termo Documento

A Matriz Termo Documento é uma matriz que representa a frequência dos *tokens* dentro dos documento. Só é possível construí-la com os passos realizados na seção anterior.

A matriz M é composta por i linhas e j colunas, e os termos a_{ij} desta matriz são interpretados como a frequência do j-ésimo termo no i-ésimo documento.

Um exemplo de Matriz Termo Documento pode ser visto na Tabela 2. Esta matriz foi construída baseada nas seguintes frases: "O céu está bonito hoje.","Amei a paisagem.", e "Que bonita a paisagem."

	céu	bonito	hoje	amor	paisagem
Texto 1	1	1	1	0	0
Texto 2	0	0	0	1	1
Texto 3	0	1	0	0	1

Tabela 2: Exemplo de uma Matriz Termo Documento

Analisando a Tabela 2, vê-se que, após retirada das *stop words*, as palavras "céu"e "amor"aparecem apenas 1 vez em todos os documentos. Já as palavras "bonito", "hoje"e "paisagem"aparecem 2 vezes nos três documentos.

2.3.3 TF-IDF

O TF-IDF (*Term Frequence-Inverse Document Frequency*) é utilizado para obter uma maior precisão na avaliação do peso das palavras, evitando depender exclusivamente da frequência do termo dentro do documento em questão. Essa abordagem combina dois métodos: o *term frequency* (TF), que calcula a frequência de ocorrência do termo dentro do documento específico, e o *inverse document frequency* (IDF), que mensura a raridade da palavra ou termo em todo o conjunto de documentos.

Assim como para a Matriz Termo Documento, para criação de uma matriz TF-IDF, é necessário que sejam feitos todos os passos da seção 2.3.3. Com tudo realizado, pode-se chegar às medidas TF e IDF e, assim, multiplicá-las, gerando o TF-IDF. Com o resultado, pode-se ver que termos com maior TF-IDF têm maior relevância para aquele documento em que ele está inserido e com isso ajudar mais na análise (SILGE; ROBINSON, 2017). Esta é uma forma mais sofisticada de representar os documentos de forma vetorial. Fica claro que quanto mais certeza do peso que uma palavra tem para definir a polaridade, melhor para o modelo que será testado posteriormente. Abaixo pode-se ver como é calculado o TF, o IDF e o TF-IDF:

$$TF_{i,j} = \frac{\text{número de vezes que o termo } i \text{ aparece no documento } j}{\text{número de termos no documento } j} \tag{2.1}$$

$$IDF_i = \ln\left(\frac{\text{número total de documentos}}{\text{número de documentos que contém o termo } i}\right) \tag{2.2}$$

$$TF\text{-}IDF_{i,j} = TF_{i,j} \times IDF_i \tag{2.3}$$

Na Tabela 3 é possível ver como ficaram os resultados do TF-IDF dos *tokens*, mostrados anteriormente na Tabela 2:

	céu	bonito	hoje	amor	paisagem
Texto 1	0.22	0.08	0.08	0	0
Texto 2	0	0	0	0.37	0.14
Texto 3	0	0.10	0	0	0.10

Tabela 3: Exemplo de uma matriz TF-IDF

Observando a Tabela 3, pode-se ver que as palavras “amor”e “paisagem”possuem um maior peso nos três documentos.

2.4 Métodos de Classificação

Nesta seção, serão abordados os métodos de classificação a serem empregados neste estudo. Conforme mencionado anteriormente, esses métodos são aplicáveis a conjuntos de dados supervisionados, nos quais existe uma variável resposta qualitativa. De forma geral, esses métodos fornecem probabilidades para cada classe considerada, sendo a classe com a maior probabilidade a selecionada como resultado final da previsão.

2.4.1 Árvores de Decisão

Além de estabelecerem regras para a tomada de decisões, as Árvores de Decisão são estruturadas como um conjunto hierárquico de "nós", construídos para partir recursivamente a base de dados (MAIMON; ROKACH, 2005), onde cada nó representa uma condição ou pergunta baseada em características dos dados, encontradas nas variáveis. O problema central a ser estudado se divide em sub-problemas mais simples e, então, as técnicas são aplicadas nestes sub-problemas (JUNIOR, 2007).

Segundo Maimon e Rokach (2014), no processo da construção da árvore nós formam uma Árvore Raiz, ou seja, é uma Árvore Direcionada com um nó chamado "raiz"que não possui arestas de entrada. Todos os outros nós têm exatamente uma aresta de entrada. Um nó com arestas de saída é chamado de nó "interno"ou nó "de teste".

Para medir o grau de heterogeneidade dos dados, é utilizado o Índice Gini (JUNIOR, 2007), que mede a impureza de um certo conjunto de dados. A fórmula do índice Gini é dada pela equação 2.4.

$$\text{Índice de Gini} = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.4}$$

Onde:
p_i é a proporção da classe i dentro do conjunto de dados
c é o número de classes

Quando este índice é igual a zero, o nó é puro, ou seja, possui exemplos apenas de uma única classe. Por outro lado, quando ele se aproxima do valor um, o nó é impuro. Quando, nas árvores de classificação com partições binárias, se utiliza o critério de Gini e tende-se a isolar num ramo os registros que representam a classe mais frequente (JUNIOR, 2007).

Cada nó criado na árvore representa uma decisão a ser tomada com base nas condições

definidas pelos atributos, enquanto os nós no final representam as previsões finais. Em cada nó da Árvore é avaliada diferentes variáveis e condições a fim de criar um fluxograma capaz de ajudar a generalizar e inferir com precisão sobre novos dados.

Um exemplo da base com o seu domínio partido e a montagem de uma Árvore de Decisão para duas variáveis preditivas pode ser visto na Figura 2:

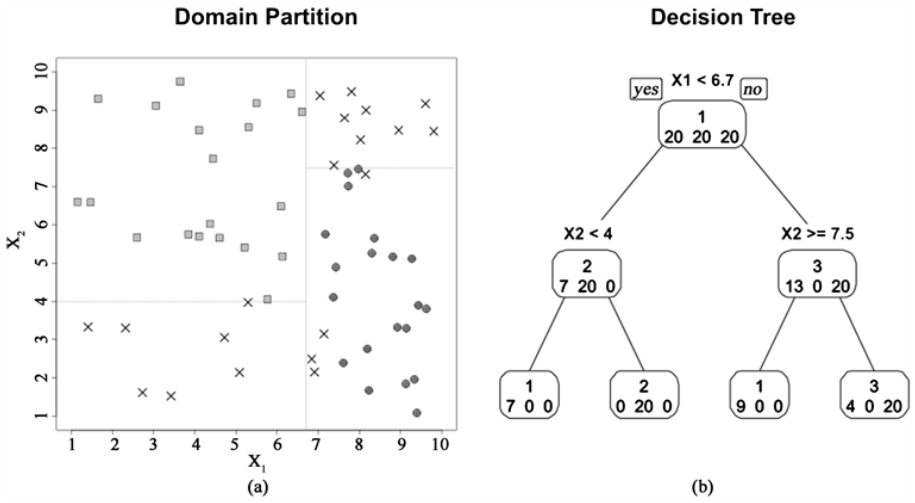

Figura 2: Exemplo de um Domínio Partido da base de dados (a). Exemplo de uma Árvore de Decisão (b) Fonte: Kubrusly, Neves e Marques (2022)

Algoritmo de Árvore de Decisão

1. Estabeleça um critério de parada, podendo ser a quantidade mínima de elementos em um nó ou um índice de Gini mínimo aceitável;

2. Defina X como o conjunto completo de dados de treinamento;

3. Escolha a variável k e o valor s que minimizam a soma dos índices Gini nas regiões $R_1 = \{X|X_k < s\}$ e $R_2 = \{X_k \geq s\}$;

4. Seja X_1 o conjunto de pontos de X que pertencem à região R_1 e X_2 o conjunto de pontos de X que pertencem à região R_2;

5. Verifique se o critério de parada foi atingido pelo conjunto de dados X_1. Se sim, termine o processo. Se não, faça $X = X_1$ e retorne ao passo 3;

6. Verifique se o critério de parada foi atingido pelo conjunto de dados X_2. Se sim, termine o processo. Se não, faça $X = X_2$ e retorne ao passo 3.

2.4.2 Floresta Aleatória

Com a técnica da Árvore de Decisão explicada, é simples entender a Floresta Aleatória. Ela nada mais é que o conjunto de muitas Árvore de Decisão independentes para que seja realizado uma modelagem para classificação ou regressão. Esta prática dá um aumento significativo na acurácia neste conjunto de árvores fazendo com que elas votem e decidam para a classe mais popular, no caso de problemas de classificação (BREIMAN, 2001). Esta prática é realizada para que sejam reduzidos os problemas e erros, como o *overfitting*, que é quando um modelo se ajusta excessivamente para os dados de treinamento e perde a capacidade de gerar previsões corretas para dados em que esse modelo será aplicado.

Maimon e Rokach (2014) explicam que, por ser uma técnica de *bagging*, ou seja, uma junção de vários modelos, a Floresta Aleatória tende a ter uma melhor acurácia que um simples modelo de Árvore de Decisão. Nas Florestas Aleatórias, a estratégia consiste em fazer com que cada árvore retire aleatoriamente amostras do conjunto de dados, permitindo a repetição dessas seleções. Ou seja, cada árvore consegue escolher diferentes amostras, com repetição. Com isso, cada árvore é construída com conjunto de dados parcialmente diferente das outras árvores, fazendo com que o modelo tenha uma riqueza e seja bem diverso.

Na criação da Floresta Aleatória, no momento da criação de um nó da árvore, é selecionado uma amostra aleatória de k preditores dentre K no conjunto total. De acordo com James et al. (2013), normalmente é utilizado $k = \sqrt{K}$.

Algoritmo de Floresta Aleatória

1. Faça $i = 1$;
2. Selecione a partir de métodos de reamostragem, como o *bootstrap*, uma amostra de tamanho n das observações;
3. Construa uma árvore i usando como base de treino os dados selecionados no passo anterior, de forma que em cada partição da árvore seleciona-se uma entre $\sqrt{k}$ variáveis sorteadas de forma aleatória entre todas;
4. Faça $i = i + 1$ e se $i \leq F$, volte para o passo 2.

Seguindo o algoritmo acima, o processo será realizado até que sejam feitas o número escolhido de F árvores. Com isso, escolhe-se para um dado novo a classe que prevaleça entre as F árvores criadas.

2.4.3 Gradient Boosting Machine

Conforme explicado por Ayyadevara e Ayyadevara (2018), Diferente da Floresta Aleatória, que é um algoritmo que combina em paralelo diferentes modelos para fazer previsões, o *Gradient Boosting Machine* utiliza de modelos de forma sequencial, com cada modelo sendo melhorado conforme os erros dos modelos anteriores.

Para exemplificar como funciona esta técnica, pode ser chamado de $M(X)$ um modelo de árvore de decisão utilizando as x variáveis independentes e a variável resposta Y , com determinada taxa de acurácia, como é mostrado na Equação 2.5.

$$Y = M(X) + \epsilon \tag{2.5}$$

Na Equação 2.6, é criado um novo modelo para prever o erro da equação acima, onde $G(X)$ é uma nova árvore de decisão com parte da base. Geralmente, é feita com metade do conjunto de dados a partir de um sorteio aleatório simples, sem reposição, para que se tenha uma generalização do modelo e, então, reduzir o risco de *overfitting*.

Inclui-se também aos próximos passos, uma taxa de aprendizado λ, que é um valor entre 0 e 1, é multiplicada à árvore de correção.

$$\epsilon = \lambda * G(X) + \epsilon' \tag{2.6}$$

De forma similar, na Equação 2.7, é criado um modelo para prever o erro ϵ', onde $H(X)$ é uma nova árvore de decisão, usando as X variáveis independentes e a taxa de aprendizado λ.

$$\epsilon' = \lambda * H(X) + \epsilon'' \tag{2.7}$$

Agora, com esses três modelos criados, junta-se eles na Equação 2.8.

$$Y = M(X) + \lambda * G(X) + \lambda * H(X) + \epsilon'' \tag{2.8}$$

Elaborado por Ayyadevara e Ayyadevara (2018), o exemplo acima leva a um modelo na Equação 2.8 que, muito provavelmente, possui uma acurácia maior que a definida para

o primeiro modelo na Equação 2.5, derivado da árvore de decisão $M(X)$, pois o modelo final possui três árvores de decisão.

Algoritmo de *Gradient Boosting Machine*

1. Inicialize as previsões com uma árvore de decisão simples com um único nó;
2. Calcule o resíduo da previsão atual;
3. Construa outra árvore de decisão que prevê o resíduo com base em todas as variáveis independentes;
4. Atualize a previsão original com a nova previsão multiplicada pela taxa de aprendizado;
5. Repita os passos 2 a 4 por um certo número de iterações (o número de iterações será o número de árvores).

Para se ter uma possível melhora na qualidade e eficiência da previsão, outros algoritmos podem ser utilizados, como, por exemplo, o *XGBOOST*, desenvolvido por Chen e Guestrin (2016), que é uma implementação otimizada do *Gradient Boosting* (CHEN et al., 2015). Da mesma forma que o *Gradient Boosting*, este algoritmo é baseado em um conjunto de árvores de decisão, onde cada árvore é construída de forma sequencial, corrigindo os erros residuais dos modelos anteriores. Este método foi feito para tornar os modelos mais eficientes, de forma que sejam construídos mais rápido e precisamente, comparando com o próprio *Gradient Boosting* e outras implementações dele.

2.5 Medidas de Qualidade do Ajuste

2.5.1 Curva ROC

Para que se possa comparar modelos de classificação, primeiro é preciso que seja feita a escolha dos cortes dos modelos pois o têm-se apenas probabilidades para as categorias. Portanto, define-se um número onde probabilidades igual ou acima dele são classificadas para a categoria de interesse. Por exemplo, se definido que o corte é 0,7 e a probabilidade de uma observação ser classificada como ódio é 0,8, rotula-se esta como ódio. Para definir o corte ideal de um modelo, pode ser usada a curva ROC *(Receiver Operating Characteristic)* (LLOYD, 1998). A curva ROC é a curva formada pelos pontos (x,y), e dado um valor de

corte q, x é o valor de 1-especificidade e y é a sensibilidade. Conhecendo a curva, define-se o valor de corte q* que maximiza a soma da sensibilidade e especificidade.

Juntamente com a análise dos cortes da curva ROC, pode ser analisada a medida AUC *(Area Under the Curve)*, que analisa a área abaixo de sua curva. Esta medida pode ser estudada para comparações dos modelos pois quanto mais perto de 1, melhor o poder do modelo em classificar as classes de forma correta. Ou seja, ela está diretamente ligada às medidas de sensibilidade e especificidade. Logo, será uma medida também utilizada no trabalho para comparação.

2.5.2 Medidas a partir da Matriz de Confusão

Com todos os modelos prontos e treinados é preciso testar seus rendimentos e o quanto se adaptam aos conjuntos de dados. Os resultados são dados pelas medidas de Qualidade do Ajuste.

Para se medir a qualidade de um modelo criado, geralmente é criada uma tabela chamada Matriz de Confusão que indica a disposição dos acertos e erros da previsão. Com esta matriz, que pode ser vista na Tabela 4, pode-se calcular as medidas de Qualidade do Ajuste dos modelos de classificação utilizadas no trabalho, que serão apresentadas a seguir.

Tabela 4: Matriz de Confusão

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

2.5.3 Sensibilidade ou *Recall*

A Sensibilidade ou *Recall* é a proporção dos valores que foram previstos verdadeiramente na classe positiva dentre aqueles que são de fato da classe positiva. Esta medida é representada pela equação 2.9.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{2.9}$$

2.5.4 Acurácia

A Acurácia é a proporção dos valores que foram previstos de forma correta dentre todas as previsões. Pode-se ver como é esta medida na equação 2.10.

$$\text{Acurácia} = \frac{VP + VN}{VP + FN + VN + FP} \tag{2.10}$$

2.6 Balanceamento de Bases de Dados

Uma base de dados, normalmente, não é disposta de uma forma perfeita para se trabalhar. Um problema que comumente é visto nela é o desbalanceamento de dados. Isto é, uma base que haja um rótulo com uma prevalência elevada.

Com uma base de dados desbalanceada, é possível que, ao treinar um modelo de classificação, o rótulo com menor frequência seja subutilizado para criar previsões. Portanto, para evitar que este tipo de problema ocorra, são utilizados métodos para balancear os dados.

Primeiro, pode-se utilizar o método de *undersampling*, que consiste em diminuir o número de observações com o rótulo prevalente até que se atinja um número ideal de observações classificadas como tal, utilizando um sorteio aleatório simples das observações da classe mais frequente, sem reposição. Outra técnica que pode ser utilizada é o *oversampling*, que é aumentar o número de observações com o rótulo menos frequente até que totalize um número adequado de observações rotuladas com esta classe, através de um sorteio aleatório simples para os registros rotulados como tal, com reposição. Por último, pode-se fazer a técnica de *SMOTEENN* (BATISTA; PRATI; MONARD, 2004), que é dada pela junção das duas técnicas anteriores, e, para a seleção, junta-se os dois sorteios aleatórios simples mostrados para os dois métodos.

3 Análise dos Resultados

Com todos os métodos já apresentados, o próximo passo é apresentar, na prática, como estes foram feitos neste trabalho.

Como visto na Seção 2.1, a base de dados é composta por 8425 observações, contendo informações de postagens na rede social *Twitter* que envolvem comentários relacionados aos grupos LGBTQIA+.

Todas as técnicas que serão mostradas em prática neste trabalho foram realizadas usando a linguagem de programação R (R Core Team, 2014).

3.1 Divisão da Base de Treino e Teste

Primeiramente, nossa base de *tweets* relacionados aos grupos LGBTQIA+ foi dividida utilizando o pacote *caret* (Kuhn; Max, 2008) entre treino e teste com 70% e 30% da base, respectivamente. Estas bases ficaram com 5891 observações para treino e 2517 observações para teste.

Com isso, na base de treino, tivemos 5644 observações classificadas como não discurso de ódio e 247 observações como ódio.

3.2 Pré-Processamento

Nesta seção serão apresentados todos os processos utilizados para preparação da base de dados para que sejam feitas as modelagens estatísticas. Todos estes processos foram realizados nas bases de treino e teste.

Primeiramente, foram retirados números e caracteres especiais. Foi retirados também espaços que não fossem normais, ou seja, mais de um espaço entre uma palavra e outra. Depois, foi realizada a tokenização. Normalmente, para esta etapa seria usado o pacote *tm*

(FEINERER; HORNIK; MEYER, 2008), porém as funções deste pacote não consideram os *emojis* como *tokens* e acabam excluindo-os. Portanto, para fins de mantenimento dos *emojis*, foi criada uma função simples para separar as postagens em *tokens*.

Palavras consideradas como *stopwords* foram retiradas com uso do pacote *stopwords* (BENOIT; MUHR; WATANABE, 2021), utilizando palavras da língua portuguesa. Há 203 palavras nesta chamada *stoplist* e, conforme o trabalho foi avançando, outras foram inseridas manualmente. A lista final de *stopwords* pode ser encontrada no Anexo 1.

Seguindo com o pré-processamento dos dados, foi realizada a lematização, que é utilizada para normalizar morfologicamente os *tokens* da base de dados. Esta técnica foi utilizada com auxílio de um dicionário específico para a língua portuguesa. Ao fim desta etapa, a base de treino ficou com 89727 *tokens*.

Em seguida, realizou-se uma seleção de termos na base de treino dos 200 termos mais frequentes. Com os termos mais frequentes, a próxima etapa é, então, criar as matrizes Termo Documento e TF-IDF, como explicado nas Seções 2.3.2 e 2.3.3, respectivamente, para treino e teste. Com esses termos selecionados, identificou-se que a atitude realizada anteriormente na tokenização para manter os *emoticons* teve valor, pois dos 200 termos, 8 são *emojis*.

Considerando a vetorização dos documentos de dimensão 200, quando adicionada a informação sobre as demais variáveis, o resultado é uma base de dados com 205 variáveis no total. Assim, tanto para a Matriz Termo Documento quanto para a matriz TF-IDF, se tem uma base de treino com uma dimensão de 5891 X 205 e uma base de teste com dimensão de 2517 X 205.

Como visto na Seção 3.1, a base de dados de treino é bem desbalanceada, contendo apenas 247 observações dela classificada como discurso de ódio. Como mostrado na Seção 2.6, o desbalanceamento dos dados pode gerar problemas para os modelos de Aprendizado de Máquinas, e, portanto, foram propostas duas formas distintas de balanceamento, que serão comparadas em breve.

3.3 Base de Dados Desbalanceada

Primeiramente, foi utilizada a base normal com 5891 observações e que tem um desbalanço entre o número dos dois rótulos. Com isso, é factível que o processo de treinamento de modelos possa ser afetado. Por exemplo, nesta base, temos apenas 4% classificada

como discurso de ódio e, com isso, sabe-se que mesmo um modelo que classifique todas as observações como o rótulo predominante de não discurso de ódio, teremos medidas de qualidade de ajuste com valores altíssimos, o que, na teoria, é algo bom para um modelo. Porém, como, de antemão, já sabe-se que há este desbalanceamento, isto vira algo para se atentar nos resultados.

Contudo, mesmo assim, serão utilizados modelos com este tipo de base para fins de comparação.

3.4 Bases de Dados Balanceadas

Foi feita então 2 divisões da base. Na primeira, reduziu-se o número de observações na base que estão classificadas como não discurso de ódio, através da técnica de *under-sampling*, para 247 observações através de um sorteio aleatório simples sem reposição.

Logo após, foi criada uma terceira base de treino com 1000 observações classificadas de cada rótulo por meio da técnica de *SMOTEENN*. Para aquelas classificadas como não discurso de ódio foi realizado um sorteio aleatório simples sem reposição, e para aquelas rotuladas como discurso de ódio foi feito um sorteio aleatório simples com reposição.

3.5 Modelagem Estatística

Para este estudo, serão utilizados os algoritmos de classificação de Floresta Aleatória e *XGBOOST* para que se possa compará-los. Além da comparação entre os dois métodos de classificação, serão estudadas diferentes implementações da base que será usada como treinamento para discussão de diferentes resultados. Serão implementados modelos com diferentes variáveis, bases balanceadas e desbalanceadas, e, por fim, duas técnicas de vetorização de documentos apresentadas: Matriz Termo Documento e Matriz TF-IDF.

Para criação desses modelos, serão utilizados os pacotes *randomForest* (LIAW; WIENER, 2002) e *xgboost* (CHEN et al., 2023).

Foram utilizados, ao total, 24 modelos neste estudo. Estes modelos são descritos na Tabela 5, onde FA se refere à Floresta Aleatória, XGB à *XGBoost* e MTD à Matriz Termo Documento.

Tabela 5: Descrição dos Modelos

Modelo	**Variáveis**	**Algoritmo**	**Balanceamento da base de treino**	**Vetorização**
Modelo 1	Somente *Tokens*	FA	Base Completa	MTD
Modelo 2	Todas as variáveis	FA	Base Completa	MTD
Modelo 3	Somente *Tokens*	FA	247 observações/rótulo	MTD
Modelo 4	Todas as variáveis	FA	247 observações/rótulo	MTD
Modelo 5	Somente *Tokens*	FA	1000 observações/rótulo	MTD
Modelo 6	Todas as variáveis	FA	1000 observações/rótulo	MTD
Modelo 7	Somente *Tokens*	XGB	Base Completa	MTD
Modelo 8	Todas as variáveis	XGB	Base Completa	MTD
Modelo 9	Somente *Tokens*	XGB	247 observações/rótulo	MTD
Modelo 10	Todas as variáveis	XGB	247 observações/rótulo	MTD
Modelo 11	Somente *Tokens*	XGB	1000 observações/rótulo	MTD
Modelo 12	Todas as variáveis	XGB	1000 observações/rótulo	MTD
Modelo 13	Somente *Tokens*	FA	Base Completa	TF-IDF
Modelo 14	Todas as variáveis	FA	Base Completa	TF-IDF
Modelo 15	Somente *Tokens*	FA	247 observações/rótulo	TF-IDF
Modelo 16	Todas as variáveis	FA	247 observações/rótulo	TF-IDF
Modelo 17	Somente *Tokens*	FA	1000 observações/rótulo	TF-IDF
Modelo 18	Todas as variáveis	FA	1000 observações/rótulo	TF-IDF
Modelo 19	Somente *Tokens*	XGB	Base Completa	TF-IDF
Modelo 20	Todas as variáveis	XGB	Base Completa	TF-IDF
Modelo 21	Somente *Tokens*	XGB	247 observações/rótulo	TF-IDF
Modelo 22	Todas as variáveis	XGB	247 observações/rótulo	TF-IDF
Modelo 23	Somente *Tokens*	XGB	1000 observações/rótulo	TF-IDF
Modelo 24	Todas as variáveis	XGB	1000 observações/rótulo	TF-IDF

Vê-se então que temos 12 modelos para cada tipo de algoritmo de Aprendizado de Máquinas. Para todos modelos de Floresta Aleatória, foram treinados modelos com 500 árvores de decisão, que é o padrão do *R*. Foram utilizadas $\sqrt{k}$ covariáveis em cada partição da floresta, onde k é o número total de variáveis, que é o padrão do *R*. Já os modelos de *XGBoost* foram instruídos com 100 iterações de árvores, com uma taxa de aprendizado de 0,1.

3.6 Discussão de Resultados

Com todos os modelos apresentados, chega-se a hora de apresentar na prática os resultados para que se possa compará-los e que seja possível tomar decisões quanto à qual tipo de modelo se adapta melhor ao estudo.

Como visto na Seção 2.5 escolheu-se, através da curva ROC, utilizando o pacote *pROC* (ROBIN et al., 2011), pontos ideais de corte para os modelos a serem trabalhados. Com estes estabelecidos, assumiu-se uma das medidas de Qualidade do Ajuste para que se possa fazer as devidas comparações iniciais. Visto que é de importância do trabalho ver a qualidade dos modelos em classificar corretamente uma postagem como ódio, escolheu-se a medida da sensibilidade.

Fazendo as análises, notou-se que, ao comparar a sensibilidade de todos os modelos, de forma global, não houve uma notória diferença ao comparar diferentes tipos de algoritmo de Aprendizado de Máquinas. Também não viu-se, de forma geral, evidência que mudar o tipo de vetorização traga uma melhora para os resultados, assim como a inserção do restante das variáveis do banco de dados.

Para uma melhor análise, os modelos foram ordenados para que se tenha, de forma agrupada, os casos onde foi feito o uso da mesmo método de classificação e vetorização, utilizando as mesmas variáveis, para que se possa analisar o quanto a sensibilidade varia ao balancear a base de treino. Para melhor entendimento, pode-se pegar como exemplo os modelos 1, 3 e 5. Estes foram treinados com a técnica de Floresta Aleatória e com apenas os *tokens* como variáveis.

Nas Figuras 3 e 4, temos os valores da sensibilidade dos 24 modelos nas bases de treino e teste, respectivamente, mostrados por um gráfico de barras. Nestes gráficos, as barras em rosa, roxo e verde claro formam um agrupamento de modelos - que apenas se diferenciam no balanceamento das bases em que foram treinados - com a base completa, com balanceamento por *undersampling* e por *SMOTEENN*, respectivamente.

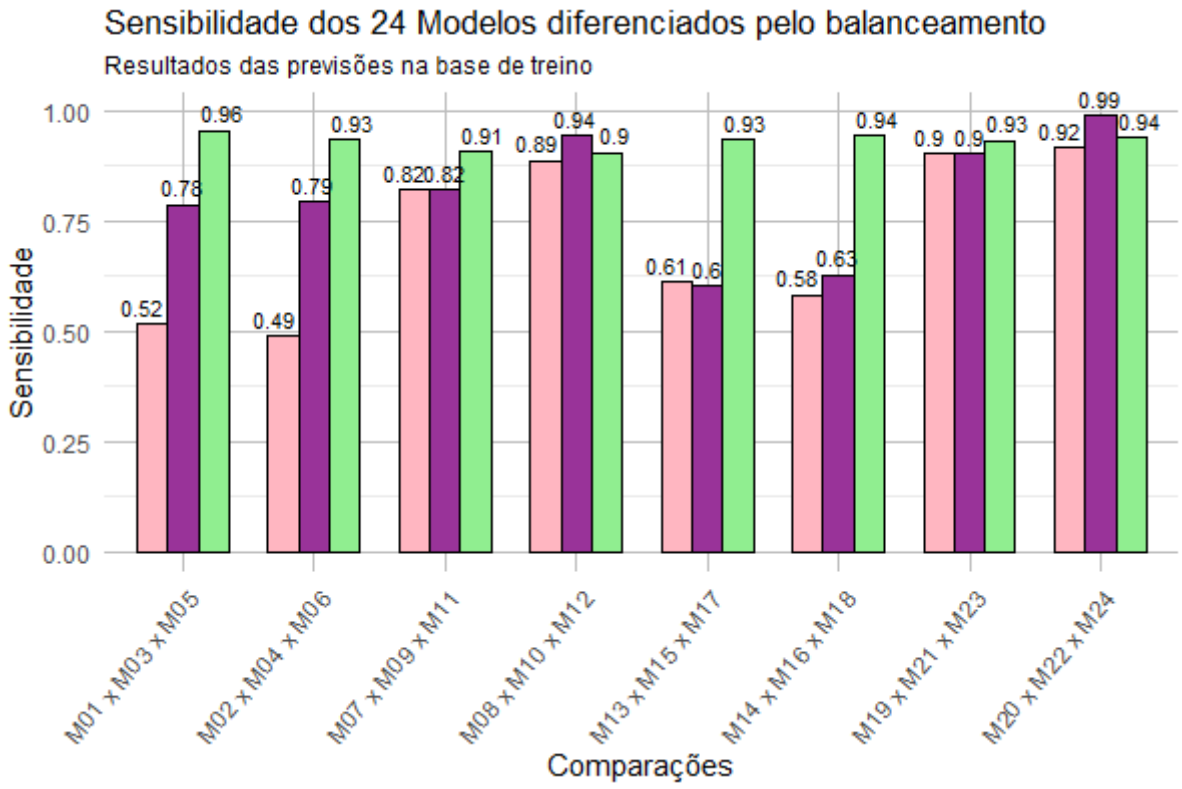

Figura 3: Sensibilidade dos 24 modelos de previsão em suas respectivas bases de treino diferenciados pelo balanceamento da base de treino.

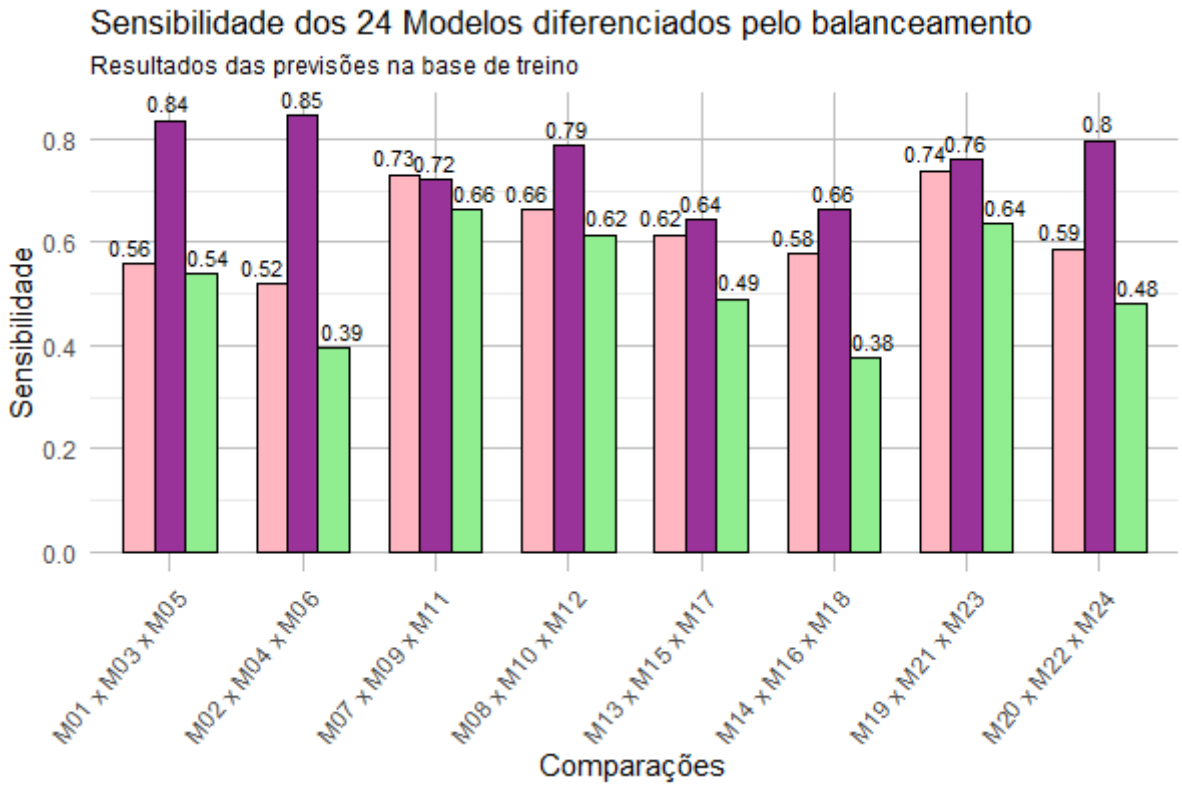

Figura 4: Sensibilidade dos 24 modelos de previsão na base de teste diferenciados pelo balanceamento da base de teste.

Explorando minuciosamente os gráficos, é evidente que, ao analisar os resultados da sensibilidade na base de teste, as bases de treinamento balanceadas por *undersampling* apresentam uma notável superioridade em termos de qualidade de ajuste quando comparadas às outras duas configurações. Por outro lado, é possível constatar que os modelos treinados com a base de dados balanceada por *SMOTEENN*, ou seja, 1000 observações para cada rótulo, exibem desempenhos insatisfatórios, manifestando resultados desfavoráveis em todas as instâncias analisadas. Esse contraste marcante entre as diferentes quantidades de observações sublinha a importância crítica da escolha adequada do tamanho da amostra no processo de treinamento dos modelos, influenciando diretamente na eficácia e robustez de suas previsões.

Fazendo uma análise agora olhando ambos os gráficos, vê-se que os modelos com a base balanceada por *SMOTEENN* possuem uma grande vantagem na base de treino, porém o desempenho cai bastante ao passar para a base de teste, o que indica que há um grande sobreajuste, o chamado *overfitting*. Isto é, os modelos com a base balanceada pela junção do *undersampling* e *oversampling* contendo 1000 observações de cada rótulo, estão se adequando muito à base na qual estes estão sendo treinados mas não estão sendo

capazes de classificar corretamente para observações fora do treinamento.

Com os gráficos apresentados, vê-se que foi importante balancear a base de dados para conseguir um resultado melhor. A primeira tentativa de balanceamento, por *undersampling*, surpreendentemente, possuiu resultados superiores e relevantes comparado com as outras configurações de base de treino. Este resultado pode ser considerado inesperado pois foram utilizadas apenas 494 observações na base de treino. Por outro lado, os modelos treinados com uma base balanceada por *SMOTEENN* pareciam poder dar mais resultados, por haver mais observações. Em praticamente todas as estruturas de dados utilizadas, o balanceamento por *undersampling* obteve altos níveis de qualidade de ajuste, na base de teste.

Portanto, há indícios que modelos balanceados por *undersampling* apresentam melhores resultados no contexto do estudo, mas ainda não há evidências para o restante das discussões. Portanto, agora, o foco será nos modelos apenas com o balanceamento visto como melhor, para que se possa chegar a uma conclusão acerca das outras discussões.

Estes resultados podem ser melhores especificados a seguir. Na Figura 5 pode-se ver os valores da sensibilidade dos modelos que utilizaram a base balanceada por *undersampling*, no treino e no teste.

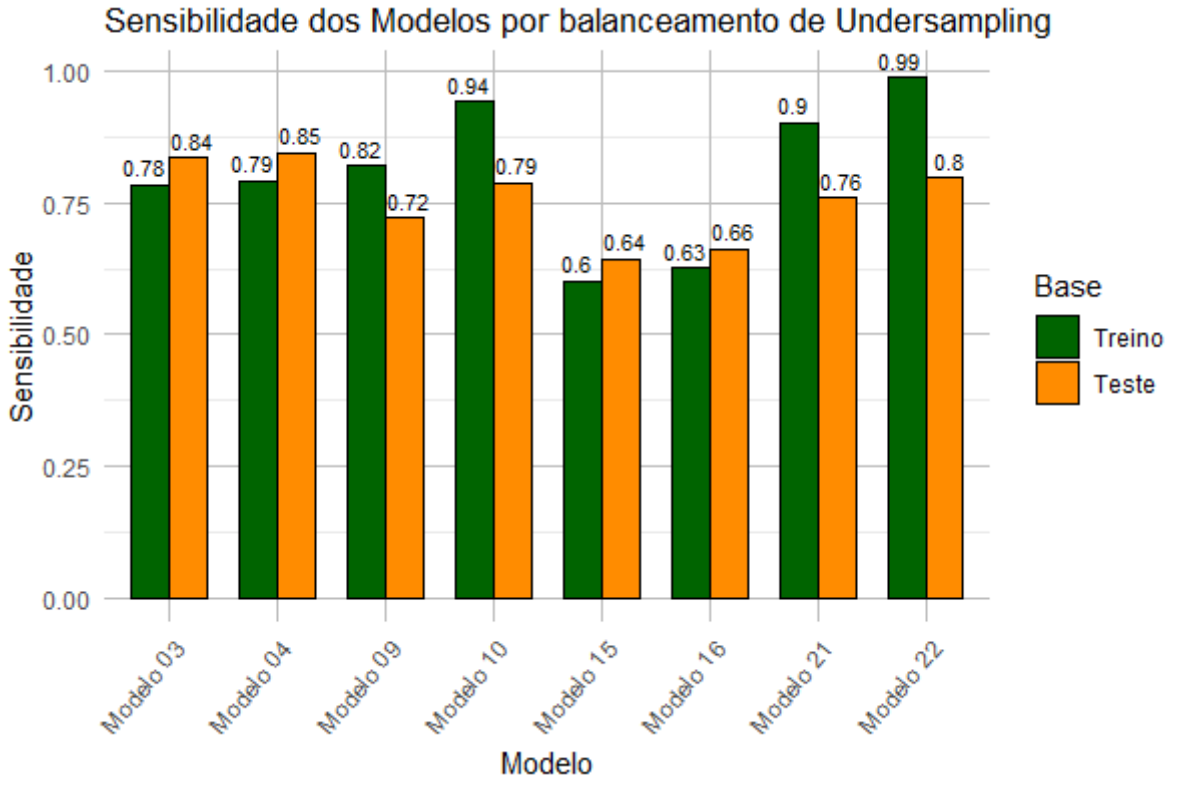

Figura 5: Sensibilidade dos 8 modelos treinados com base balanceada por *undersampling*.

Analisando o gráfico acima, consegue-se ver que, em alguns casos, parece haver, também, um sobreajuste *(overfitting)*. Porém, nos modelos 3 e 4, é possível ver que há uma leve melhora no teste em comparação ao treino e bons resultados de sensibilidade.

Pode-se também analisar outra medida de qualidade de ajuste. Por exemplo, na Tabela 6 temos os resultados da acurácia e da sensibilidade na base de treino dos modelos com a base de treino balanceada por *undersampling*. Já na Tabela 7 temos os resultados na base de teste.

Tabela 6: Medidas de Acurácia e Sensibilidade dos modelos treinados com uma base de treino balanceada por *undersampling* na base de treino

Modelo	**Acurácia**	**Sensibilidade**
Modelo 3	0,7465	0,7846
Modelo 4	0,7262	0,7927
Modelo 9	0,8276	0,8211
Modelo 10	0,8925	0,9431
Modelo 15	0,6998	0,6016
Modelo 16	0,6714	0,6260
Modelo 21	0,8763	0,9024
Modelo 22	0,9067	0,9878

Tabela 7: Medidas de Acurácia e Sensibilidade dos modelos treinados com uma base de treino balanceada por *undersampling* na base de teste

Modelo	**Acurácia**	**Sensibilidade**
Modelo 3	0,6599	0,8365
Modelo 4	0,6476	0,8462
Modelo 9	0,7016	0,7212
Modelo 10	0,6287	0,7885
Modelo 15	0,7739	0,6442
Modelo 16	0,7453	0,6635
Modelo 21	0,6500	0,7596
Modelo 22	0,5920	0,7981

Agora, somente com os modelos selecionados, é possível fazer uma melhor comparação entre eles.

A acurácia de modelos mede o quanto um modelo foi capaz de prever corretamente. Comparando os 8 modelos, temos que o modelo 15 é o que apresentou maior acurácia na base de teste (77,39%), ou seja, este modelo previu corretamente o maior número de observações como ódio ou não ódio. O modelo que teve o pior resultado foi o modelo 22 (59,20%).

A sensibilidade, que é a medida a qual este trabalho está tendo o seu foco, mostra quantas observações verdadeiramente classificadas como a classe positiva foram, de fato, classificadas como ela. Nesta medida, alguns modelos apresentaram bons resultado, e o modelo 4 é o que se destacou (84,62%). Nesta medida, o modelo que teve o pior resultado foi o modelo 15 (64,42%). Esta medida foi escolhida pois viu-se uma necessidade para se atentar àqueles modelos que estivessem classificando mais acertadamente os casos de ódio como realmente de ódio.

Vistos estes resultados, nota-se que alguns resultados de sensibilidade são expressivos dentre os 8 modelos em que houve balanceamento através da técnica de *undersampling* ao se atentar a capacidade destes modelos para classificar de forma correta o rótulo positivo (discurso de ódio). Com isso, alguns os modelos parecem estar próximos de atender ao objetivo de acertar o máximo possível de casos de ódio.

Observando as características dos modelos escolhidos para esta última análise, pode-se observar que os modelos 3 e 4 que foram vetorizados a partir de Matriz Termo Documentos e treinados por Floresta Aleatória, apresentaram maior média na sensibilidade, com 84,14%. Já na acurácia, os modelos de TF-IDF, com Floresta Aleatória, tiveram melhor resultado, com uma média de 75,96%. Notou-se que nesses 4 modelos com resultados melhores não houve uma diferença significativa na inserção de novas variáveis, no contexto do estudo.

Para ilustrar um pouco, pode ser mostrado, graficamente, como se comportou a curva ROC dos modelos balanceados por *undersampling* e de Floresta Aleatória, com vetorização de Matriz Termo Documento nas Figuras 6 e 7. Neles, vemos a medida AUC. Vê-se que suas medidas estão bem próximas, porém o Modelo 3, que é o modelo com apenas *tokens*, leva uma pequena vantagem.

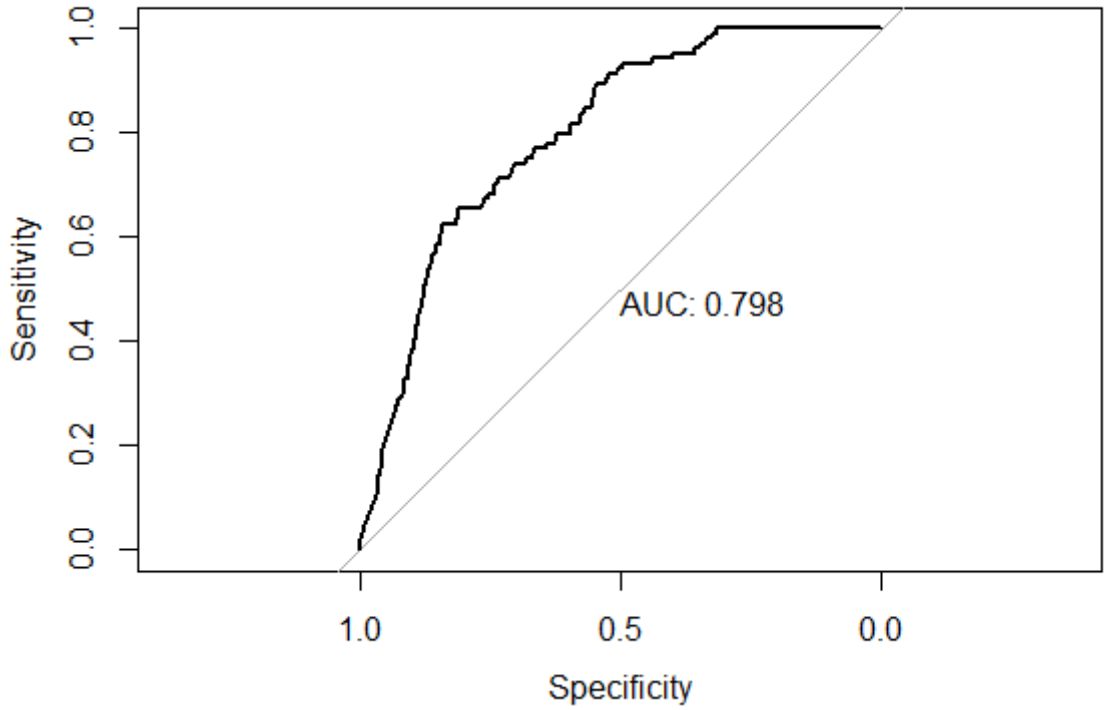

Figura 6: Curva ROC do Modelo 3 (Base somente com *tokens*, balanceamento por *undersampling*, por algoritmo de Floresta Aleatória e vetorização de Matriz Termo Documento)

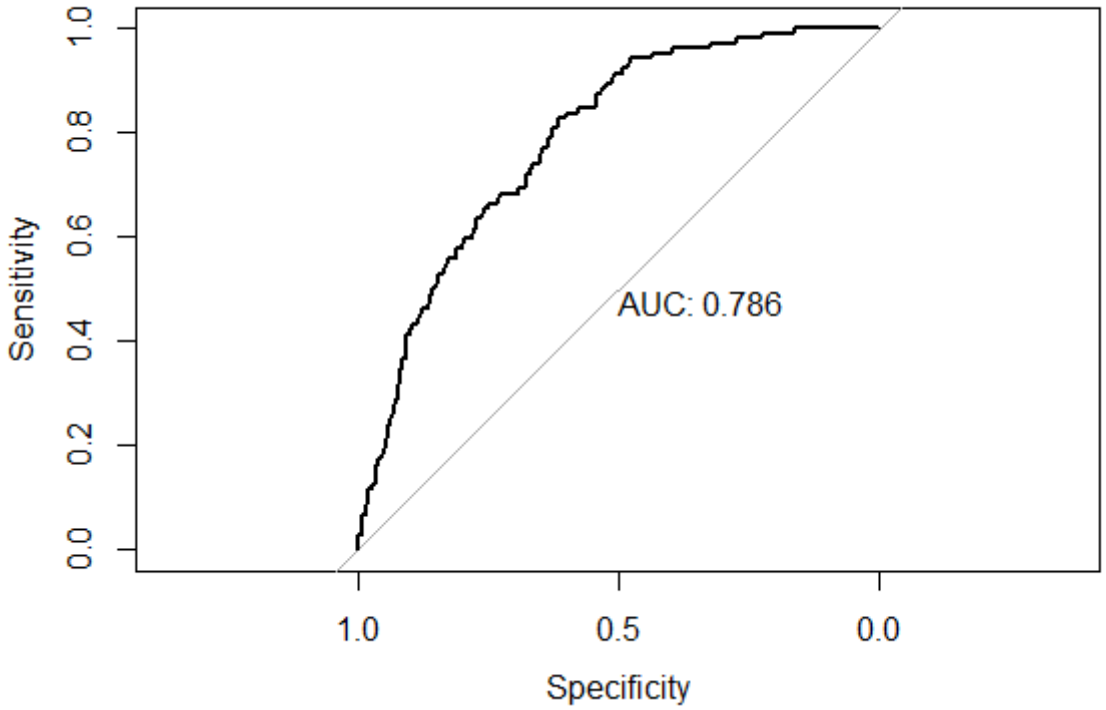

Figura 7: Curva ROC do Modelo 4 (Base com todas as variáveis, balanceamento por *undersampling*, por algoritmo de Floresta Aleatória e vetorização de Matriz Termo Documento)

4 Conclusões

Tendo em vista a importância das redes sociais para as intercomunicações humanas e como isto afeta o dia a dia de todos, foi proposto um trabalho que testasse formas para identificar algo que, infelizmente, acontece o tempo todo na internet: o preconceito. O objetivo específico deste trabalho foi analisar *tweets* relacionados à grupos LGBTQIA+ e identificar se estes são um discurso de ódio ou não.

Para que este estudo funcionasse, foi preciso utilizar muitas técnicas de mineração e manipulação de dados a fim que fosse produzida uma base de dados que pudera ser utilizada para análises estatísticas. Nestas etapas, foi visto que era necessário uma manipulação mais enxuta da base de dados de treino pois haviam muito menos observações rotuladas como discurso de ódio do que observações não identificadas como tal. Para isso, foi preciso fazer duas técnicas diferentes de balanceamento de dados: *undersampling* e *SMOTEENN*. Estas abordagens foram feitas para que, no processo de treinamento, não houvesse um viés para o rótulo predominante.

Após todas as técnicas mostradas anteriormente, foram utilizados os métodos de Floresta Aleatória e *XGBOOST* para treinar esses classificadores. Com isso, pôde ser visto que, realmente, há desfechos bem melhores ao balancear a base. No caso deste estudo, os resultados que foram treinados após a técnica de balanceamento *undersampling* foram significativamente melhores que as configurações de não balanceamento e de *SMOTEENN*.

Dentre os modelos que utilizaram a base feita com o método de *undersampling*, vetorização por Matriz Termo Documento e algoritmo de Floresta Aleatória apresentaram uma média de 84,14% na sensibilidade, isto é, a capacidade de classificar corretamente o discurso de ódio. Já os que utilizaram *XGBOOST*, que obtiveram, uma média de 75,96% para a sensibilidade. Isto é um indicador de que os modelos de Floresta Aleatória estão se adequando mais a base de dados e ao contexto do estudo.

Os modelos de *undersampling* surpreenderam ao apresentar resultados melhores, por haver um pequenos número de observações em sua amostra, enquanto os modelos de *SMO-*

TEENN possuem maior quantidade de observações e, viu-se que, apenas se adaptaram a classificar dentro da própria base de treino, o que leva a crer que ocorreu um *overfitting*.

Portanto, vê-se uma necessidade de outras tentativas de balanceamento para estudos futuros. Há a possibilidade de variar de maneira diferente as classes menos e mais frequentes da base de dados de treino. Logo, trabalhos posteriores podem ser focados em diferentes níveis de balanceamento, até que se chegue em um que apresente um patamar alto para a métrica de sensibilidade, que, no contexto do estudo, é a capacidade de classificar acertadamente o discurso de ódio.

Por fim, com os resultados atingidos neste estudo, é factível sugerir que, ao se chegar à um grau de balanceamento que seja considerado bom para o trabalho, seja replicado para outros diferentes temas de preconceito encontrados na mesma base de dados estudada no presente trabalho. Com isso feito, a análise pode ser promovida a prever, usando toda a base de dados, o tipo de preconceito à qual as postagens se referem.

Referências

ALMEIDA, P.; KUBRUSLY, J. Criação de um banco de tweets rotulado como não ofensivo, ofensivo e discursos de ódio. In: SEMEXT UFF. [S.l.], 2022.

AYYADEVARA, V. K.; AYYADEVARA, V. K. Gradient boosting machine. *Pro machine learning algorithms: A hands-on approach to implementing algorithms in python and R*, Springer, p. 117–134, 2018.

BATISTA, G. E.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, v. 6, n. 1, p. 20–29, 2004.

BENOIT, K.; MUHR, D.; WATANABE, K. stopwords: Multilingual stopword lists. *R package version 2.3*, 2021. Disponível em: ⟨https://CRAN.R-project.org/package=stopwords⟩.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, p. 5–32, 2001.

CAPPI, J. et al. Internet, big data e discurso de ódio: reflexões sobre as dinâmicas de interação no twitter e os novos ambientes de debate político. Pontifícia Universidade Católica de São Paulo, 2017.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794.

CHEN, T. et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, v. 1, n. 4, p. 1–4, 2015.

CHEN, T. et al. xgboost: Extreme gradient boosting. *R package version 3.3*, 2023. Disponível em: ⟨https://CRAN.R-project.org/package=xgboost⟩.

COUTINHO, V. M. d. M. S.; MALHEIROS, Y. Detecçao de mensagens homofóbicas em português no twitter usando análise de sentimentos. In: SBC. *Anais do IX Brazilian Workshop on Social Network Analysis and Mining*. [S.l.], 2020. p. 1–12.

FEINERER, I.; HORNIK, K.; MEYER, D. Text mining infrastructure in r. *Journal of Statistical Software*, v. 25, n. 5, p. 1–54, March 2008.

HVITFELDT, E.; SILGE, J. *Supervised machine learning for text analysis in R*. [S.l.]: CRC Press, 2021.

JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112.

JUNIOR, J. R. C. Desenvolvimento de uma metodologia para mineração de textos. *Pontificia Universidad Catolica de Rio de Janeiro: Rio de janeiro, Brasil*, 2007.

KUBRUSLY, J.; NEVES, A. L.; MARQUES, T. L. A statistical analysis of textual e-commerce reviews using tree-based methods. *Open Journal of Statistics*, Scientific Research Publishing, v. 12, n. 3, p. 357–372, 2022.

Kuhn; Max. Building predictive models in r using the caret package. *Journal of Statistical Software*, v. 28, n. 5, p. 1–26, 2008. Disponível em: ⟨https://www.jstatsoft.org/index.php/jss/article/view/v028i05⟩.

LIAW, A.; WIENER, M. Classification and regression by randomforest. *R News*, v. 2, n. 3, p. 18–22, 2002. Disponível em: ⟨https://CRAN.R-project.org/doc/Rnews/⟩.

LLOYD, C. J. Using smoothed receiver operating characteristic curves to summarize and compare diagnostic systems. *Journal of the American Statistical Association*, Taylor & Francis, v. 93, n. 444, p. 1356–1364, 1998.

MAIMON, O.; ROKACH, L. *Data mining and knowledge discovery handbook*. [S.l.]: Springer, 2005. v. 2.

MAIMON, O. Z.; ROKACH, L. *Data mining with decision trees: theory and applications*. [S.l.]: World scientific, 2014. v. 81.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014.

MOREO, A. et al. Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications*, Elsevier, v. 39, n. 10, p. 9166–9180, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2014. Disponível em: ⟨http://www.R-project.org/⟩.

ROBIN, X. et al. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, v. 12, p. 77, 2011.

SILGE, J.; ROBINSON, D. *Welcome to text mining with R*. [S.l.]: O'Reilly, Sebastopol, 2017.

SILVA, L. R. L. et al. A gestão do discurso de ódio nas plataformas de redes sociais digitais: um comparativo entre facebook, twitter e youtube. *Revista ibero-americana de ciência da informação*, v. 12, n. 2, p. 470–492, 2019.

TABOADA, M. et al. Lexicon-based methods for sentiment analysis. *Computational linguistics*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 37, n. 2, p. 267–307, 2011.

ANEXO 1 – Lista de Stopwords

a	as	desse	e’	estar
à	às	desta	é	estas
agora	assim	deste	eh	estava
ah	até	deve	ela	estavam
ai	através	devem	elas	estávamos
aí	c	devendo	ele	este
ainda	cada	dever	eles	esteja
alguém	coisa	deverá	em	estejam
alguem	com	deverão	enquanto	estejamos
algum	como	deveria	entao	estes
alguma	contra	deveriam	então	esteve
ampla	contudo	devia	entre	estive
amplo	d	deviam	era	estivemos
ante	da	disse	eram	estiver
antes	daí	disso	éramos	estivera
após	das	disto	essa	estiveram
ao	de	dito	essas	estivéramos
aos	dela	diz	esse	estiverem
aquela	delas	dizem	esses	estivermos
aquelas	dele	do	esta	estivesse
aquele	deles	dos	está	estivessem
aqueles	depois	duma	estamos	estivéssemos
aquilo	dessa	e	estão	estou

etc	hajamos	ir	não	ô
eu	hão	isso	nas	oq
fazendo	havemos	isto	ne	os
fazer	hei	já	né	ou
feita	hein	lhe	nem	outra
feito	houve	lhes	nenhum	outras
ficar	houvemos	lo	nessa	outro
foi	houver	mais	nessas	outros
fomos	houvera	mas	nesta	p
for	houverá	me	nestas	para
fora	houveram	mesma	ninguém	pela
foram	houvéramos	mesmo	nn	pelas
fôramos	houverão	meu	no	pelo
forem	houverei	meus	nos	pelos
formos	houverem	minha	nós	pequena
fosse	houveremos	minhas	nossa	pequenas
fossem	houveria	mt	nossas	pequeno
fôssemos	houveriam	mto	nosso	pequenos
fui	houveríamos	muita	nossos	per
grande	houvermos	muito	num	perante
há	houvesse	n	numa	pode
haja	houvessem	na	nunca	pôde
hajam	houvéssemos	nao	o	podendo

poder	que	sou	teria	
poderia	quem	sua	teriam	todos
poderiam	são	suas	teríamos	tu
podia	se	ta	teu	tua
podiam	seja	tá	teus	tuas
pois	sejam	talvez	teve	tudo
por	sejamos	também	ti	última
porém	sem	tampouco	tido	últimas
porque	sempre	tao	tinha	último
posso	sendo	tão	tinham	últimos
pouca	ser	tava	tínhamos	um
poucas	será	tb	tive	uma
pouco	serão	tbem	tivemos	umas
poucos	serei	tbm	tiver	uns
pq	seremos	te	tivera	vai
pra	seria	tem	tiveram	vc
primeiro	seriam	tém	tivéramos	vcs
primeiros	seríamos	temos	tiverem	vendo
própria	seu	tendo	tivermos	ver
próprias	seus	tenha	tivesse	vez
próprio	si	tenham	tivessem	vindo
próprios	sido	tenhamos	tivéssemos	vir
q	sim	tenho	to	você
quais	so	ter	tô	vocês
qual	só	terá	toda	vos
quando	sob	terão	todas	vós
quanto	sobre	terei	todavia	
quantos	somos	teremos	todo	