UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

MATHEUS HENRIQUE DE ALMEIDA SOUZA

Classificação Automática de Textos em Língua Portuguesa Com Traços de Racismo no Twitter

Goiânia
2022

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos "reproduzir" e "publicar" conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Classificação Automática de Textos em Língua Portuguesa Com Traços de Racismo no Twitter

Autor(a): Matheus Henrique de Almeida Souza

Goiânia, 18 de Abril de 2022.

Matheus Henrique de Almeida Souza – Autor

Dra. Deborah Silva Alves Fernandes – Orientadora

Msc. Márcio Giovane Cunha Fernandes – Co-Orientador

MATHEUS HENRIQUE DE ALMEIDA SOUZA

Classificação Automática de Textos em Língua Portuguesa Com Traços de Racismo no Twitter

Trabalho de Conclusão apresentado à Coordenação do Curso de Sistemas de Informção do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informção.

Área de concentração: Sistemas de Informção.

Orientadora : Profa. Dra. Deborah Silva Alves Fernandes
Co-Orientador: Prof. Msc. Márcio Giovane Cunha Fernandes

Goiânia
2022

MATHEUS HENRIQUE DE ALMEIDA SOUZA

Classificação Automática de Textos em Língua Portuguesa Com Traços de Racismo no Twitter

Trabalho de Conclusão apresentado à Coordenação do Curso de Sistemas de Informção do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em Sistemas de Informção, aprovada em 18 de Abril de 2022, pela Banca Examinadora constituída pelos professores:

Profa. Dra. Deborah Silva Alves Fernandes
Instituto de Informática – UFG Presidente da Banca

Prof. Msc. Márcio Giovane Cunha Fernandes
Câmpus Central - Sede : Anápolis - CET – UEG

Prof. Dr. Fabrízzio Alphonsus Alves de Melo Nunes Soares
Instituto de Informática – UFG

Matheus Henrique de Almeida Souza

Graduando em Sistemas de Informação na UFG - Universidade Federal de Goiás.

Dedico este trabalho ao meu país, Brasil, e a todas pessoas que fazem parte dele, pois somos uma grande combinação de culturas e etnias, onde nossas maiores riquezas são justamente nossas singularidades, variedades e diferenças.

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

MATHEUS HENRIQUE DE ALMEIDA

Classificação automática de textos em língua portuguesa com traços de racismo

Trabalho de conclusão de curso apresentado à Universidade Federal de Goiás como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Profa. Dra. Deborah Silva Alves Fernandes

Aprovado em 18/04/2022.

BANCA EXAMINADORA

Profa. Dra. Deborah Silva Alves Fernandes
Universidade Federal de Goiás
Instituto de Informática

Prof. Me. Márcio Giovane Cunha Fernandes
Universidade Estadual de Goiás

Prof. Dr. Fabrízzio A. A. de M. N. Soares
Universidade Federal de Goiás
Instituto de Informática

Agradecimentos

Primeiramente a Deus, que fez com que meus objetivos fossem alcançados, que me deu sabedoria para entrar em contato com a Profa. Dra. Deborah Fernandes para ser minha orientadora, e por me permitir superar todos os obstáculos na realização deste trabalho.

De forma geral, aos meus pais, Dalton e Núbia, que sempre estiveram ao meu lado e pelo amor incondicional.

Especificamente à minha mãe, uma professora, que desde o ínicio da minha carreira acadêmica me instruiu e guiou, além de me mostrar a forma correta de estudar, que possibilitou este momento.

Aos meus avós, pois me ampararam durante minha trajetória universitária e me acolheram sob seus cuidados.

A Profa. Dra. Deborah Fernandes pela orientação e dedicação, por todos os conselhos, pela ajuda e pela paciência com a qual guiaram a realização deste trabalho.

Ao Governo Federal e à Universidade Federal de Goiás por me oferecerem essa oportunidade de ter acesso a um ensino de exímia qualidade. Uma formação que me colocou em diversas situações desafiadoras, mas através da superação das mesmas, quero acreditar que fui capaz de me tornar não apenas um profissional melhor, mas até mesmo crescer como pessoa e como cidadão.

A todos os envolvidos direta ou indiretamente no desenvolvimento deste trabalho, enriquecendo o meu processo de aprendizado.

O racismo se expressa concretamente como desigualdade política, econômica e jurídica.

Silvio Luiz de Almeida,
Trecho do livro "O que é racismo estrutural".

Resumo

Souza, Matheus. **Classificação Automática de Textos em Língua Portuguesa Com Traços de Racismo no Twitter**. Goiânia, 2022. 49p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Devido a um aumento na quantidade de dados gerados na internet, que cresce de forma exponencial, cria-se a necessidade de adoção de técnicas computacionais para extração de informações. Essa crescente demanda, especialmente nas redes sociais, traz consigo desafios para a área de classificação de texto automática, pois nesse ambiente qualquer pessoa é livre para fazer declarações de quaisquer tipos, incluindo textos com traços de racismo. Neste trabalho foi utilizado métodos de *Ensemble* para detecção de traços de racismo no Twitter na língua portuguesa. Além disso, foram aplicados cinco algoritmos de classificação autônomos, incluindo *Naïve Bayes*, *K-Nearest Neighbours*, *Logistic Regression*, *Random Forest* e *Support Vector Machines*, e dois métodos de *Ensemble*, *Stacking* e *Boosting*, em *datasets* compostos por *tweets* que possuem termos selecionados para este trabalho. O *Stacking* alcançou um resultado de F1 0.828, identificando ambas as classes de forma equilibrada, utilizando de um *dataset* balanceado e sem o uso de *stopwords*. Os resultados experimentais mostram que o desempenho da classificação pode ser melhorado usando o método *Ensemble*, mesmo que a melhoria não seja muito significativa, o uso do método *Ensemble* pode reduzir o risco de escolher um classificador que não irá se adequar a uma situação específica.

Palavras–chave

Aprendizado de Máquina, Racismo, Twitter, Classificação de Texto, Classificadores Automáticos, Ensemble.

Abstract

Souza, Matheus. **Automatic Classification of Portuguese Messages With Traces of Racism on Twitter**. Goiânia, 2022. 49p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Due to an increase in the amount of data generated on the internet, which grows exponentially, there is a need to adopt computational techniques for extracting information. This growing demand, especially on social networks, brings challenges to the area of automatic text classification, as in this environment anyone is free to make statements of any kind, including texts with traces of racism. In this work, Ensemble methods were used to detect traces of racism on Twitter in Portuguese. In addition, five autonomous classification algorithms were applied, including Naïve Bayes, K-Nearest Neighbors, Logistic Regression, Random Forest and Support Vector Machines, and two methods of Ensemble, Stacking and Boosting, in datasets composed of tweets that have terms selected for this work. Stacking achieved a result of F1 0.828, identifying both classes in a balanced way, using a balanced dataset and without the use of stopwords. Experimental results show that the classification performance can be improved using the Ensemble method, even if the improvement is not very significant, the use of the Ensemble method can reduce the risk of choosing a classifier that will not suit a specific situation.

Keywords

Machine Learning, Racism, Twitter, Text Classification, Automatic Classifiers, Ensemble

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Segundo Almeida, 2019 [2], a ideia proposta do racismo parte do princípio de que “o racismo é sempre estrutural, ou seja, integra a organização econômica e política da sociedade de forma inescapável”, sendo este “ a manifestação normal de uma sociedade” e não uma anormalidade. Logo, é possível inferir que o racismo assegura este sentido, tecnologia e lógica para a formação da desigualdade e violência que modelam nossa vida social contemporânea.

Analisando tecnologia e vida social contemporânea, existem as redes sociais. Uma rede social é uma estrutura social de pessoas ou organizações com valores e objetivos compartilhados, conectados por um ou vários tipos de relacionamentos. Uma das redes sociais online mais usadas no Brasil nos últimos anos é o Twitter, que permite enviar e receber atualizações pessoais e de outros usuários através de pequenos textos denominados "*tweets*". Um dos diferenciais do Twitter é sua rapidez e instantaneidade referentes a informações e a quantidade de perfis públicos, o que permite a coleta de mensagens sem restrições. Por seus diferenciais, destaca-se também o uso do Twitter durante manifestações políticas, e este é muito utilizado quando o assunto são tendências de pauta nacional e mundial. Outro aspecto importante para a realização deste trabalho é o tamanho dos *tweets*(máximo = 280 caracteres), por serem textos pequenos, forçam os usuários a reduzirem as informações que serão publicadas, contribuindo para uma maior objetividade com as palavras.

Nas últimas décadas, a quantidade de dados gerados na internet cresceu exponencialmente, contribuindo para a possibilidade e necessidade de adoção de técnicas computacionais para extração de informações. Essa crescente demanda, especialmente nas redes de mídia social, traz consigo desafios para a área de processamento de linguagem natural, pois no ambiente dessas redes qualquer pessoa é livre para fazer declarações de quaisquer tipos, incluindo discursos de ódio. A tecnologia favorece a expressão de discurso pelas pessoas, permitindo que estas se manifestem rapidamente sobre quaisquer temas, e se o assunto envolvido for de interesse popular, este é velozmente espalhado nas plataformas tornando-se viral, podendo até causar disputas entre grupos sociais [7].

A pesquisa apresentada neste projeto final de curso está focada em métodos para

abordar os desafios identificados por aplicações que envolvem classificação automática de texto e sua aplicação prática. Dentre esses desafios pode ser citada a parcialidade e imprecisão da linguagem natural, algo que varia de acordo com o julgamento de cada pessoa. No caso deste trabalho, o problema que se deseja resolver é a detecção de traços de racismo em textos da rede social Twitter. Para a realização de todo o experimento utilizou-se como *baseline* o desenho proposto por [7] que realizou detecção de discurso de ódio em tweets de língua indonésia.

O objetivo geral deste trabalho é a detecção automática de traços de racismo em tweets de língua portuguesa utilizando técnicas de aprendizado de máquina. Fazendo parte do propósito deste trabalho, está a utilização de classificadores automáticos de texto e aplicação de técnica de *Ensemble*, visando resultados mais precisos. Para alcançar este propósito podem ser elencados os seguintes objetivos específicos:

- Estudo sobre o racismo e suas formas de expressão em discursos em textos da internet;
- Estudo sobre classificadores automáticos de aprendizagem de máquina;
- Implementação de algoritmos classificadores independentes e em modo *ensemble* para detecção de traços de racismo em tweets;
- Realização de testes e avaliação dos modelos quanto à capacidade de aprendizado e identificação correta de classes.

Essa monografia possui ao total cinco capítulos, o primeiro a introdução, seguido pelo capítulo 2 são onde expostas definições gerais que compoê a fundamentação teórica necessária para a realização da pesquisa e são apresentados os trabalhos relacionados ao tema. Já no capítulo 3 é explicado o experimento apresentado nesta monografia, sua estrutura e metodologia de desenvolvimento. Em seguida, no capítulo 4 são tratados os resultados das diferentes abordagens empregadas neste trabalho, acompanhados com um comparativo aos resultados da pesquisa usada como *baseline* para a prática. As conclusões e chances de eventuais trabalhos futuros são detalhados no capítulo 5.

CAPÍTULO 2

Fundamentação Teórica

2.1 Racismo

2.1.1 Definições Gerais

Com ojetivo de alcançar um melhor entendimento sobre racismo, primeiro é necessário saber diferenciar o racismo de outras associações à ideia de raça, como por exemplo, preconceito e discriminação.

Segundo [2],"o racismo é uma forma sistemática de discriminação que tem a raça como fundamento, que se manifesta por meio de práticas conscientes ou inconscientes que culminam em desvantagens ou privilégios para indivíduos, a depender do grupo racial ao qual pertençam". Já sobre o preconceito racial, de acordo com [2], " é o juízo baseado em estereótipos acerca de indivíduos que pertençam a um determinado grupo racializado, e que pode ou não resultar em práticas discriminatórias". A discriminação racial, por sua vez, "é a atribuição de tratamento diferenciado a membros de grupos racialmente identificados", baseado em [2].

Conhecendo sobre a relação de tais conceitos, é possível classificar o racismo em três concepções diferentes: individualista, institucional e estrutural. O racismo individualista ressalta a natureza psicológica e ética do fenômeno, sendo ele individual ou coletivo, uma irracionalidade a ser combatida no campo jurídico [2]. Por outro lado, o racismo institucional é tratado como o resultado do funcionamento das instituições, as quais, ainda que indiretamente, atuam de forma que geram desvantagens e privilégios raciais [2]. Sob a perspectiva das instituições, há o racismo estrutural, que caracteriza a atuação das instituições condicionadas a uma estrutura social racista, ou seja, as instituições apenas materializam uma sociedade que tem o racismo como um de seus componentes [2].

Estendendo-se à leis brasileiras, em janeiro de 1989 foi sancionada a lei nº 7716 que pune os crimes resultantes de discriminação ou preconceito de raça, cor, etnia, religião ou procedência nacional. As penas previstas podem chegar até a 5 anos de reclusão. A lei tem como propósito preservar os objetivos fundamentais descritos na Constituição

Federal, de promoção do bem estar de todos, sem preconceitos de origem, raça, sexo, cor, idade e quaisquer outras formas de discriminação.

Além de crimes de racismo, temos ainda crimes de injúria racial, que se diferenciam baseados no direcionamento da conduta. De acordo com Tribunal de Justiça do Distrito Federal e dos Territórios[1], enquanto que na injúria racial os crimes geralmente estão relacionados ao uso de termos pejorativos referentes à raça ou cor com o objetivo de ofender a reputação da vítima, no crime de racismo, a ofensa é contra uma coletividade. Por exemplo, para toda uma raça, não há especificação do ofendido. O crime de injúria racial está inserido no capítulo dos crimes contra a honra, previsto no parágrafo 3º do artigo 140 do Código Penal. A pena é menos severa que a do crime de racismo, pode ir de 1 a 3 anos de reclusão.

2.1.2 Racismo na Internet

De acordo com [13], "mudanças críticas na tecnologia e na política impulsionaram o crescimento de discurso de ódio. A Web permite a criação quase instantânea de redes de associação, enquanto o isolamento e potencial anonimato dos perpetradores protege-os da responsabilidade". Apesar das adversidades do combate ao discurso de ódio, recorrentes ao avanço tecnológico, também surgem ferramentas por conta do mesmo, como por exemplo a prova digital, que são provas em processos judiciais. Como descrito em uma nótica publicada no site da Defensoria Pública do Estado do Ceará[2], essas provas digitais podem ser comunicações, informações, registros fotográficos, registros de áudios e de textos, que são mediados por equipamentos eletrônicos, tendo como plataformas redes sociais. Segundo o Tribunal Superior do Trabalho[3],"O uso dessas provas possui fundamentos nos artigos 369 e 370 do Código de Processo Civil. O primeiro autoriza as partes a empregarem todos os meios legais, bem como os moralmente legítimos, para provar a verdade dos fatos em que se funda o pedido ou a defesa e influir eficazmente na convicção do juiz.". As provas digitais acompanham a mudança de paradigma nas relações sociais, além disso, ampliam a própria produção, pois equipamentos eletrônicos deixam registros das comunicações.

Justamente por tais facilidades decorrentes da internet, diversas propostas de combate ao racismo “digital” surgiram, como descritos nos trabalhos relacionados no capítulo anterior deste documento. Por exemplo, no artigo [10], temos uma abordagem que incorpora dados demográficos na detecção de linguagem racista no Twitter. De acordo

[1] https://www.tjdft.jus.br/institucional/imprensa/campanhas-e-produtos/direito-facil/edicao-semanal/injuria-racial-x-racismo

[2] https://www.defensoria.ce.def.br/noticia/mensagens-de-texto-audio-e-fotos-em-redes-sociais-funcionam-como-provas-em-processos-judiciais/

[3] https://www.tst.jus.br/provas-digitais

com os pesquisadores desse trabalho, a linguagem racista não é uniforme, justamente por isso, os dados demográficos tornam-se necessários para um melhor entendimento das características contextuais e da comunidade alvo. Apesar de existirem várias propostas ao combate do racismo digital, boa parte delas são focadas e limitadas ao inglês, com o objetivo de resolver esse problema, no trabalho [6] foi criado um modelo que classifica *tweets* em sete linguagens diferentes.

2.2 Trabalhos Relacionados a Racismo nas Redes Sociais

A abordagem descrita por [20] em seu artigo ultrapassa a categorização binária de textos racistas e xenófobos. O trabalho utiliza de teorias das ciências sociais para desenvolver uma categoria quadridimensional ("*stigmatization*", "*offensiveness*", "*blame*", e "*exclusion*") para classificação, com o foco em crimes de ódio *anti-Asian* devido ao Covid-19. Utilizando um *dataset* composto por 247.153 *tweets* extraídos por meio do Tweepy API das dezoito *hashtags* racistas e xenófobas mais circuladas relacionadas à Covid-19 de 1º de janeiro a 30 de abril do ano de 2020. Os pesquisadores adotaram técnicas de modelagem de tópicos usando *Latent Dirichlet Allocation* de Gensim e *Latent Dirichlet Allocation* modelo de Mallet. O resultado foi uma metodologia que une métodos computacionais com teorias de ciências sociais, possibilitando ser transformada em sugestões de políticas construtivas.

Um classificador de conjunto para detectar discurso de ódio em pequenos textos foi proposto no trabalho de [21]. Neste, utilizaram um *dataset* de 16 mil *tweets*, associados aos rótulos de classes "*Neutral*" (N), "*Racist*" (R) and "*Sexist*" (S). A metodologia utilizou um conjunto de classificadores de Rede Neural Recorrente (RNN), e incorporou vários recursos associados a informações relacionadas ao usuário, com o objetivo de calcular a tendência dos usuários ao racismo ou sexismo. A tendência do usuário (T) é calculada baseada na proporção entre as mensagens neutras (Na), racistas (Ra) e sexistas (Sa) sobre todas as mensagens publicadas pelo usuário (M) (e.g: T(Ra) = RaM). Como resultado, obtiveram um esquema que é capaz de distinguir racismo e sexismo de textos normais com um classificação de qualidade maior que algoritmos de *state-of-the-art* em sua época de publicação. Outra característica interessante desse método é que o mesmo não é dependente de idioma.

O artigo [10] objetivou uma estratégia supervisionada de aprendizagem para detectar linguagem racista no Twitter com base em palavras que incorporavam dados demográficos (idade, sexo e localização). Neste trabalho, os dados demográficos possuem um papel muito importante, pois de acordo com os pesquisadores, a linguagem racista não é uniforme, alterando-se dependendo das características contextuais e da comunidade alvo. Utilizando a API de *streaming* do Twitter por um período de três meses (05/02/2015

a 05/05/2015), foram reunidos 17,2 milhões de *tweets* potencialmente racistas de 1,8 milhões de usuários. Para construir um *word embeddings* foi utilizado uma paradigma de aprendizagem autodidata, já para a detecção, usaram recursos sensíveis ao contexto de alto nível e baixa dimensão. Como resultado atingiram uma classificação razoável de precisão sobre um padrão ouro de conjunto de dados com medida F1 score de *76,3%*.

Um *framework* de tempo real para detectar agressão em mensagens do Twitter foi introduzido em [11]. Como solução utilizaram um paradigma de aprendizado de máquina de *streaming*, além disso, foi empregado classificadores de ML de forma incremental. Possuindo um *dataset* de 100 mil *tweets*, em todo o *pipeline* de processamento aplicaram o paradigma de *data streaming*.O *framework* proposto foi suficiente para detectar outros comportamentos relacionados como sarcasmo, racismo e sexismo em tempo real. O modelo foi capaz de atingir o mesmo desempenho de *machine learning* em lote, com mais de 90% de *accuracy*, *precision*, and *recall*.

O artigo [19] visou detectar automaticamente linguagem abusiva no Twitter através da aplicação de uma abordagem *two-step* para a classificação, além de comparar com uma abordagem *one-step*. A classificação *one-step* é dividida em rótulos: Nenhum, Racismo e Sexismo. Na *two-step*, o texto é primeiramente classificado em Nenhum ou Abusivo, detectada a linguagem abusiva, este é reclassificando em "racista" ou "sexista". O trabalho implementou três modelos baseados em CNN para classificar a linguagem abusiva sexista e racista: CharCNN, WordCNN e HybridCNN. A diferença entre esses modelos está nos recursos de entrada, se são caracteres, palavras ou ambos. Com um *dataset* de 20 mil *tweets*, a abordagem mostrou um desempenho promissor de 0,827 *F-measure* usando HybridCNN em *one-step* e 0,824 *F-measure* usando regressão logística em *two-step*.

O objetivo da pesquisa de [14] foi classificar *tweets* racistas, antes e depois do Covid-19 ser declarado uma pandemia global. Desejavam saber se o covid-19 influenciou de forma negativa o racismo e a discriminação, especialmente direcionados a indivíduos asiáticos. Foram utilizados dois *datasets* coletados pelos pesquisadores, ao total 40 mil *tweets*. Utilizaram um conjunto de técnicas de aprendizado de máquina, como *Linear Support Vector Classifiers*, *Logistic Regression models* e *Deep Neural Networks*. Como resultado da pesquisa, chegaram à conclusão que o racismo direcionado a certos grupos, não é exatamente causado pelo vírus, mas sim uma onda contínua de preconceito existente em relação à eles.

Com a necessidade de detecção e filtragem automática de discurso de ódio em redes sociais, o trabalho de [9] propõe uma abordagem de detecção para dados do Twitter. A metodologia aplicada é baseada em unigramas e padrões que são coletados automaticamente do conjunto de treinamento. É empregando aprendizado de máquina para realizar a classificação. Foi utilizado um *dataset* de 2010 *tweets* e alcançaram um

desempenho de 87.4% de *accuracy* na descoberta se o *tweet* é ofensivo ou não, e 78.4% de *accuracy* na percepção se o é odioso, ofensivo ou *clean*.

No artigo de [5] é verificado o uso de alguns algoritmos clássicos de aprendizado de máquina, com algumas alterações, na detecção de discurso de ódio em *tweets* escritos em português. Foram utilizados quatro modelos: *Support Vector Machine, Multilayer Perceptron Neural Network, Regressão Logística e Naive Bayes*. O *dataset* utilizado foi coletado por uma API do Twitter de janeiro à março de 2017. Ao total foram coletados 5668 *tweets*, sendo 1228 classificados como discurso de ódio. Como resultado, chegaram à conclusão de que esses algoritmos clássicos modificados obtiveram pontuações melhores do que o LSTM, sendo também uma alternativa competitiva aos modelos da literatura relacionada no próprio artigo.

Boa parte das vezes, a detecção de discurso de ódio é apenas focada no inglês. No artigo [6] foi criado um modelo que classifica *tweets* em sete linguagens diferentes, possuindo situações onde são utilizados vários idiomas simultaneamente em um mesmo *tweet*. As metodologias aplicadas foram Redes Neurais Convolucionais (CNN) e representação em nível de personagem. Utilizaram um *dataset* de 6136 *tweets* em árabe, 1827 em italiano, 5668 em português, 713 em indonésio, 24784 em inglês, 4.575 *tweets* em hindi-inglês e 9010 em alemão. Como resultado, obtiveram *accuracy* de 0.8893 no *dataset* contendo cinco idiomas diferentes e *accuracy* de 0.8300 para o *dataset* de sete idiomas.

O trabalho de [1] foi realizado com o objetivo de demonstrar que os modelos baseados em *deep learning* podem superar os três gargalos de *cyberbullying detection*. Sendo eles, primeiramente o fato de trabalhos existentes focarem em apenas uma plataforma de *social media*, segundo, a questão de abordarem apenas um tópico de *cyberbullying*, e, por último, o aspecto de contarem com recursos de dados elaborados cuidadosamente à mão. Para a pesquisa utilizaram três *datasets*: Formspring(12k postagens), Wikipedia(100k postagens) e Twitter(16k postagens). Nos experimentos foram aplicados quatro modelos baseados em *Deep Neural Network* (DNN) e comparados com quatro modelos tradicionais de *machine learning*, *logistic regression* (LR), *support vector machine* (SVM), *random forest* (RF) e and *naive Bayes* (NB). Como resultado, obtiveram que esses modelos de DNN, juntamente com aprendizagem por transferência, superaram os resultados de última geração para os três *datasets*.

O artigo [8] objetivou uma abordagem holística na detecção de abuso em plataformas de mídias sociais, que considera vários aspectos do comportamento abusivo, com o foco no Twitter. Propuseram neste trabalho uma arquitetura de *deep learning* que combinou os metadados com padrões coletados automaticamente dos *tweets*. O objetivo era detectar normas comportamentais abusivas. Foram utilizados múltiplos *datasets*, *cyberbullying*(6,091 *tweets*), *hateful*(16,059 *tweets*), *offensive*(24,783 *tweets*), *sarcasm*(61,075 *tweets*) e *abusive*(85,984 *tweets*) . Os resultados obtidos mostram valores de *AUC* de 92%

a 98%.

Com o crescimento da necessidade de detecção e filtragem de discurso de ódio, o trabalho [22] apresentou uma abordagem de detecção de tal discurso no Twitter. Nos experimentos foram aplicados OpenNLP5 para a execução de *Natural Language Processing* (NLP) e *Gate Twitter PoS Tagger* para o exercício de *Part-of-Speech* (PoS). Utilizando um *dataset* de 2010 *tweets*, obtiveram uma acurácia de 78.4% na classificação ternária, que identificou se o *tweet* era *hateful*, *offensive*, ou *clean*. Para uma classificação binária, se é ofensivo ou não, o resultado foi de *accuracy* de 87.4% .

2.3 Classificadores Automáticos

A classificação automática de texto é a prática onde atribui-se rótulos(*labels*,do inglês) à categorias predefinidas em textos de linguagem natural(*natural language*). Tais rótulos atribuídos são baseados em informações encontradas em um conjunto treinamento, que possui documentos rotulados [12]. Com a expansão do texto online, a tarefa torna-se crítica para o gerenciamento de informações [12].

Sem pormenorizar, *Text Classification* é um processo projetado para atribuir rótulos ou tags a unidades de texto, como *tweets*, comentários, parágrafos, frases e documentos. Possui uma ampla gama de aplicações, como detecção de *spam*, análise de sentimentos e novas classificações. Ela pode ser aplicada de acordo com níveis:

- Nível de documento : o algoritmo tem como finalidade classificar de forma relevante um documento. Considera o documento como uma unidade de informação.
- Nível de parágrafo : o algoritmo tem como finalidade classificar de forma relevante um único parágrafo, uma parte de um documento, considerando o parágrafo como uma unidade de fontes de dados [15].
- Nível de frase : o algoritmo tem como finalidade classificar de forma relevante uma única frase, parte de um parágrafo, considerando a frase como uma unidade de informação [15].
- Nível de sub-sentença : o algoritmo tem como finalidade classificar de forma relevante subexpressões dentro de uma frase, uma porção de uma frase, considerando a subexpressão como unidade de informação [15].

Apesar de um texto ser uma fonte rica de informações, classificá-las pode ser uma tarefa desafiadora. Tais fontes de dados podem ser obtidas em redes sociais, e-mails, avaliações e reclamações de usuários, como outros dados da *web*.

2.3.1 Tarefas de Classificação de Texto

Em conformidade com [17], as tarefas de classificação de texto normalmente incluem categorização de notícias, classificação de tópicos e análise de sentimento. Recentemente, pesquisadores mostram que é eficaz a utilização de classificadores de textos em tarefas de *natural language understanding* (NLU), como por exemplo resposta a perguntas extrativas [17] . Essa seção incorpora cinco exemplos de tarefas de TC(*Text classification*), sendo três delas típicas TC *task* e duas NLU *task*, estas classificações são baseadas em [17]:

- *Sentiment Analysis* : Tal tarefa tem como objetivo analisar opiniões e sentimentos sobre dados textuais, extraindo pontos de vista e polaridades. Podendo ser realizada de forma binária ou multiclasse.
- *News Categorization* : é uma tarefa que auxilia usuários na obtenção de informações em *real-time*, através de recomendações de informações relevantes baseadas em interesses do usuário e de identificação de tópicos emergentes de informações.
- *Topic Analysis* : Também conhecida como classificação de tópicos, tem como finalidade identificar tópicos ou temas de um dado textual.
- *Question Answering* (QA) : é uma tarefa que classifica as respostas em correta ou errada, após obter uma pergunta e um conjunto de possíveis respostas.
- *Natural language inference* (NLI) : Denominado *recognizing textual entailment* (RTE), reconhecimento da vinculação textual, pressupõe se o significado de um texto pode ser inferido de outro. Em um sistemas NLI, é necessário a atribuição de um par de dados textuais, para que possibilite a medição de semelhança semântica, podendo indicar a probabilidade de se uma frase é uma paráfrase da outra.

2.4 Técnicas de Classificação de Texto

A fase mais considerável de todo o processo de classificação de texto é a escolha do melhor classificador. Sem um entendimento completo de cada algoritmo, é inviável estabelecer efetivamente o classificador mais eficiente para uma aplicação de classificação de texto [15].

As técnicas de classificação de texto podem ser divididas em abordagem de aprendizado de máquina, abordagem baseada em lexicon e abordagem híbrida, além disso existem métodos de *Ensemble* que combinam vários classificadores para obter previsões mais precisas e corretas.

2.4.1 Aprendizado de Máquina

A abordagem de aprendizado de máquina(ML) utiliza algoritmos de ML e também recursos linguísticos. Podem ser divididos em métodos de aprendizagem supervisionados e não supervisionados. Os métodos supervisionados utilizam um grande número de dados rotulados para o treinamento [16]. Por outro lado, os métodos não supervisionados superam certas dificuldades dos métodos supervisionados, pois não são necessários esses documentos de treinamento rotulados [16]. Abaixo serão apresentados os classificadores frequentemente utilizados na abordagem de aprendizado de máquina.

- *k-nearest Neighbors* (KNN) é um algoritmo usado para aplicações de classificação de texto em muitos domínios de pesquisa nas últimas décadas. Trata-se de uma técnica não paramétrica usada para classificação, conforme [15]. O algoritmo KNN encontra os k vizinhos mais próximos x, sendo x um documento do teste , entre todos os documentos do conjunto de treinamento. Além disso, pontua os candidatos de acordo com a classe de k vizinhos mais próximos. Após categorizar os valores de pontuação, o algoritmo associa o candidato a classe com a maior pontuação de documento de teste x [15].

Probabilistic Classifiers

Segundo [16], classificadores probabilísticos (probabilistic classifiers) usam modelos de mistura(*mixture models*) para realizar a classificação. Nestes, o modelo entende que cada classe é um componente dessa tal mistura, também são conhecidos como classificadores generativos(*generative classifiers*). Um classificador probabilístico está apto a indicar um esquema de probabilidade em um conjunto de classes, ao contrário de apenas produzir a classificação mais provável [16]. Alguns dos classificadores probabilísticos mais conhecidos são o *Naıve Bayes Classifier* (NB), *Maximum Entropy Classifier* (ME) e *Bayesian Network* (BN) [16].

- *Naıve Bayes Classifier (NB)*: O classificador *Naive Bayes* é um dos classificadores mais simples e mais utilizados. Sua forma de classificação calcula a probabilidade de uma classe com base na distribuição de palavras no documento [16]. O modelo do classificador *Naïve Bayes* é baseado no teorema de Bayes, desenvolvido por Thomas Bayes entre 1701-1761 [15] . O modelo funciona extraindo recursos do *Bag of Words* ignorando as posições das palavras no documento [15]. A Figura 2.1 representa a formula do NB proposta em [16], onde P(*label*|*features*) refere-se à probabilidade de um determinado conjunto de características pertencenter a um rótulo específico. Em seguida, P(*label*) estende-se a probabilidade anterior de um rótulo ou a probabilidade de que um recurso aleatório definir o rótulo,

P(*features*|*label*) é a probabilidade de que um determinado conjunto de recursos seja classificado como um rótulo e P(*features*) diz respeito à probabilidade anterior de um determinado conjunto de características seja [16].

- *Maximum Entropy Classifier (ME)* (classificador de entropia máxima): é um classificador probabilístico, que converte o conjunto de características rotuladas em um vetor usando codificação [16]. Este vetor codificado é então usado para calcular pesos para cada característica. A Entropia Máxima não pressupõe que essas características sejam independentes umas das outras [3]. Essas características são combinadas para determinar o rótulo mais provável para o conjunto de características. Normalmente, o classificador *Maximum Entropy* é usado quando não há determinação para assumir a independência condicional dos recursos, tornando-se aplicável à classificação de texto, pois sabe-se que os recursos textuais dificilmente são independentes [3].

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)}$$

Figura 2.1: Fórmula do NB proposta no trabalho [16]

Linear Classifiers

No campo do aprendizado de máquina, o objetivo da classificação estatística é usar as características de um objeto para identificar a qual classe ele pertence. Os classificadores lineares fazem isso tomando decisões de classificação com base nos valores de uma combinação linear de recursos. Um dos classificadores lineares mais conhecidos é o *Support Vector Machine* (SVM), que é um classificador que tenta determinar um bom separador linear entre diferentes classes [16].

- *Support Vector Machines Classifiers (SVM)*: O SVM foi originalmente projetado para tarefas de classificação binária. No entanto, muitos pesquisadores utilizam esta técnica para estudar problemas multiclasse [15] . O princípio básico do SVM é identificar separadores lineares que melhor separam as diferentes classes. Devido à natureza do texto, os dados de texto são adequados para classificação SVM, na qual poucos recursos não são importantes e muitas vezes relacionados entre si, desse modo são organizados em categorias linearmente separáveis [16].
- *Logistic Regression*: A regressão logística é um classificador linear que se concentra em prever probabilidades em vez de prever classes, treinado na probabilidade de um recurso ser rotulado com base em alguns dados iniciais [15]. A regressão logística é

uma técnica estatística destinada a produzir um modelo a partir de um conjunto de observações que permite prever o valor tomado por uma variável categórica (geralmente binária) a partir de uma série de variáveis explicativas contínuas e/ou contínuas.

Decision Tree Classifiers

Os Classificadores de Árvores de decisão(DTCs) são usados com sucesso em diversas áreas para classificação, principalmente para text and data mining [15]. Os classificadores de árvore de decisão provêem uma divisão hierárquica dos dados de treinamento, no qual o valor do atributo é usado para dividir os dados, com a condição da presença ou ausência de uma ou mais palavras [16]. A divisão de dados é feita recursivamente até que um nó folha contenha um número mínimo de registros, que são usados para fins de classificação [16].

- *Random Forests* (florestas aleatórias) ou *Random Decision Forests technique* (técnica de florestas de decisão aleatória): é um *ensemble learning method*(método de aprendizado conjunto) para classificação de texto. A *Random Forest* agrupa árvores de preditores que, após treinadas, as previsões são atribuídas com base em uma votação. Florestas aleatórias (ou seja, conjuntos de árvores de decisão) são muito rápidas para treinar conjuntos de dados de texto em comparação a outras técnicas, mas bastante lentas para criar previsões uma vez treinadas [15]. Além do mais, construir tantas árvores pode ser computacionalmente exigente, na etapa de previsão, à medida que mais árvores na floresta aumentam, a complexidade do tempo também aumenta. Para obter uma estrutura mais rápida, o número de árvores na floresta deve ser reduzido [15].

2.4.2 Abordagem Baseada em Lexicon

A Abordagem baseada em Lexicon é fundamentada em encontrar uma coleção de termos de sentimentos conhecidos e pré-compilados, ou seja, um léxicon de sentimento, e usá-lo para analisar o texto. Segundo [16], esta abordagem pode ser baseada em dicionário ou em corpus.

- Baseada em Dicionário: Segundo [16], na abordagem baseada em dicionário, os termos de opinião são coletados e depois anotados de forma manual. Após isso, os sinônimos e antônimos desses termos, são pesquisados em conhecidos dicionários, com o objetivo de aumentar esse conjunto de palavras, WordNet[4] é um exemplo

[4]https://wordnet.princeton.edu

de tal dicionário. Uma grande desvantagem dessa estratégia é a incapacidade de encontrar palavras de opinião com diretrizes específicas de domínio e contexto [16].

- Baseada em Corpus: A abordagem baseada em corpus auxilia justamente na questão em que o procedimento descrito no item anteior, método baseado em dicionário, dispunha de problemas relacionados a certas incapacidades de encontrar palavras de opinião [16]. Sua técnica está sujeita à padrões sintáticos, juntamente com uma lista de palavras chaves de opinião. Tais elementos são utilizados para encontrar outras palavras de opinião em um grande corpus [16].

2.4.3 Abordagem Híbrida

A abordagem híbrida combina ambas as abordagens que foram esclarecidas anteriormente, ou seja, aprendizado de máquina e abordagem baseado em lexicon, e normalmente nos métodos híbridos os lexicons de sentimento desempenham um papel primordial, conforme [16].

2.4.4 Métodos de Ensemble

O *Ensemble learning*(aprendizado conjunto) é o processo pelo qual vários modelos, como classificadores, são gerenciados estrategicamente e combinados para resolver um problema de inteligência computacional específico [3]. O princípio básico do método *Ensemble* é combinar as previsões de vários classificadores distintos para obter previsões mais precisas e corretas. Para realizar a melhor classificação, os recursos dos melhores classificadores são utilizados neste método. Dentre os métodos *Ensemble*, temos : *Boosting*, *Stacking* e *Voting*:

- *Boosting*: é uma estratégia que combina múltiplos modelos fracos em um singular modelo forte. A ferramenta Orange Data Mining[5] implementa o *Boosting* por meio do método *Adaboost*, que pode ser usado com outros algoritmos de aprendizado para aumentar seu desempenho. Isso se faz ajustando os modelos fracos através de um sistema de pesos, com o objetivo de focar nas instâncias que são mais difíceis de classificar. Os métodos de *Boosting* do *AdaBoost* se assemelham à estrutura trabalhada no artigo [7], pois o algoritmo de classificação SAMME do *AdaBoost* utiliza dos resultados de classificação, similar ao *Hard Voting*, e o algoritmo de classificação SAMME.R utiliza das estimativas de probabilidade, similar ao *Soft Voting*.
- *Stacking*: é uma método *Ensemble* que calcula um meta modelo à partir de vários modelos básicos. Envolve o treinamento de um algoritmo de aprendizado para

[5] https://orangedatamining.com/

combinar as previsões de vários outros algoritmos de aprendizado. Em uma etapa inicial, os classificadores são treinados usando os dados disponíveis e, em seguida, um algoritmo agregador é treinado para usar as previsões dos outros algoritmos como base para fazer a previsão final.

- *Hard and Soft Voting*: Apoiado no trabalho realizado em [7] no *Hard Voting* cada classificador autônomo tem um voto e o resultado é selecionado por votação majoritária. Logo, a categoria selecionada é aquela que for a maioria, ou seja, pelo menos mais da metade dos votos. No entretanto, na *Soft Voting*, as médias das probabilidades de cada categoria são usadas como pontuação na realização da votação.

CAPÍTULO 3

Materiais e Métodos

O experimento apresentado trata-se de uma continuidade do trabalho de final de curso de [4] e uma reprodução do experimento realizado por [7]. Este último aborda métodos *Ensemble* na detecção de discurso de ódio no Twitter na Indonésia. A ideia principal do experimento a ser descrito, é reproduzir o modelo de classificação, mas adaptando-o para o contexto de detecção de traços de racismo em textos coletados do Twitter em língua portuguesa.

A Figura 3.1 apresenta o desenho do experimento realizado neste projeto final de curso. A detecção de traços de racismo consta de três etapas principais, sendo elas: pré-processamento; treinamento de classificadores de forma separada ; e a utilização de métodos *Ensemble*, combinação de classificadores.

As próximas subseções irão detalhar as etapas realizadas e ferramentas utilizadas no experimento conforme apresentados na Figura 3.1.

3.1 Dataset e Ferramentas

O *dataset* utilizado no experimento foi coletado pelo pesquisador de [4] em seu trabalho. A coleta de dados foi realizada no Twitter entre 04/06/2018 e 04/10/2018. Nesse período, foram coletados 106.739 *tweets*. Para realizar a coleta de dados, foram definidos previamente termos de busca, dentre eles: "senzala", "gorila", "cabelo de bombril", "cabelo de esfregão", "nariz de nego", "nariz de nega", "tinha que ser preto", "tinha que ser preta", "preto da senzala", "preta da senzala", "preto da macumba", "nega macaca", "preto macaco", "preta macaca", "preta nojenta", "preto nojento", "criola", "crioula", "crioulo". O levantamento deste conjunto de termos se deu através de uma pesquisa realizada pelo autor em textos de notícias e de casos sobre racismo na Internet (redes sociais e jornais) e também através de uma pesquisa na forma de questionário.

A ferramenta principal adotada na realização dos experimentos deste projeto final de curso foi o Orange Data Mining[1]. Trata-se de um kit de ferramentas *open source*

[1] https://orangedatamining.com/

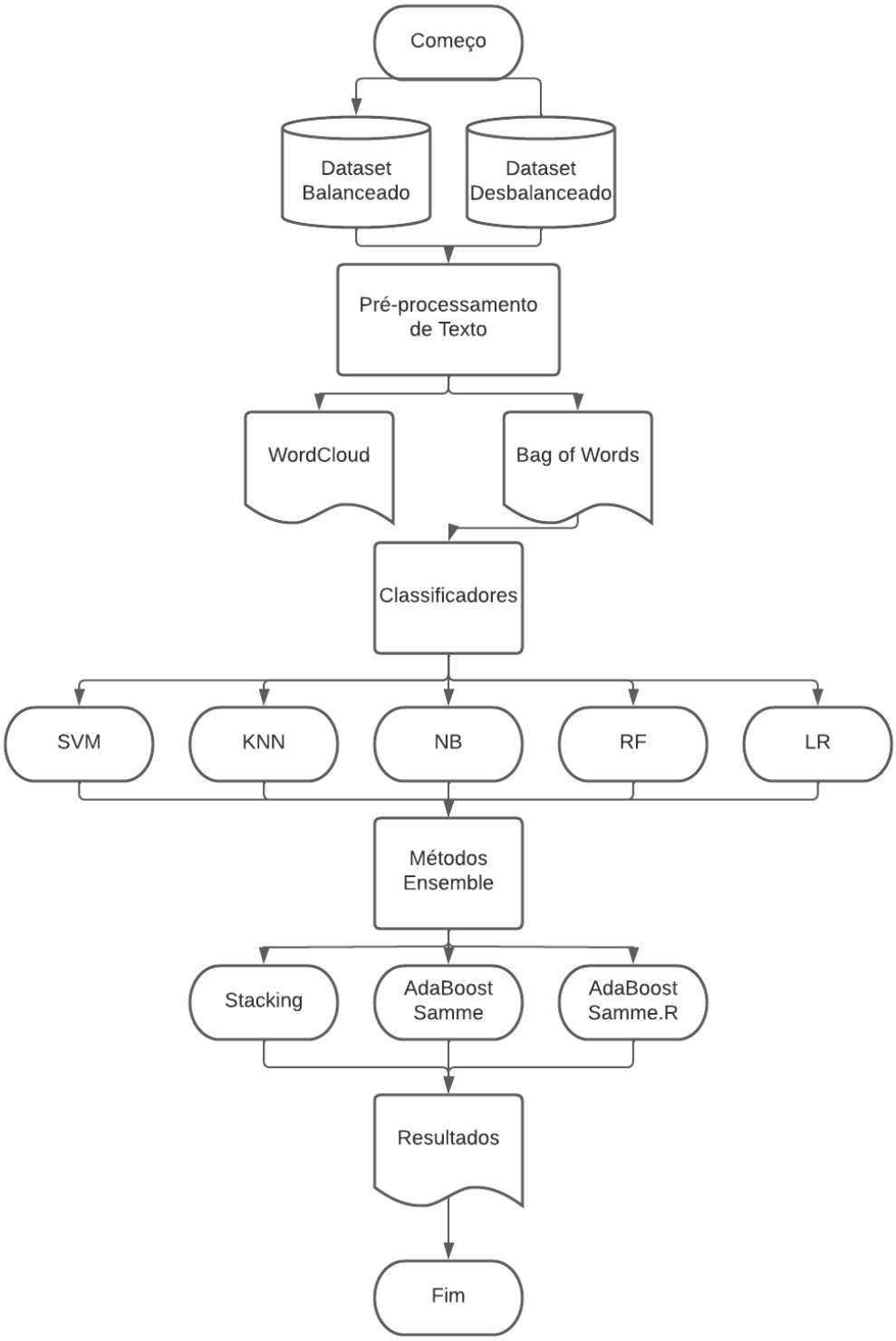

Figura 3.1: Desenho do experimento proposto e realizado neste trabalho de projeto final de curso.

para aprendizado de máquina e visualização de dados. Este possui um *front-end* de programação visual para análise de dados qualitativos exploratórios rápidos e visualização de dados interativa.

3.2 Pré-processamento

No pré-processamento realizado pelo pesquisador em [4], com objetivo de separar apenas o *tweets* de interesse, foram removidos *links* e nomes de usuários das mensagens. Para o experimento atual, além desses descritos, aplicou-se uma ferramenta do Orange, *Preprocess Text*[2] e em seus *preprocessors*: *transfomation*, *tokenization* e *filtering*.

No *preprocessor transformation*, que transforma o dado recebido, foram empregados também *lowercase*, *remove accents*, *parse html* e *remove urls* . Já em *tokenization*, que é um método que quebra os elementos em componentes menores, o texto foi manuseado utilizando o *regexp* de forma que foram separados em palavras, pegando apenas as com quatro ou mais caracteres. Por último, foi empregado o *filtering*, com o objetivo de filtrar as palavras, onde as *stopwords* foram inseridas.

No trabalho [4], foram apresentados pesquisas que apoiam a ideia de que as *stopwords* carregam nenhum valor semântico relevante ao classificador, assim como foram apontados trabalhos que defendem o oposto, que removê-las degrada a performance. Portanto, nos experimentos realizados neste trabalho foram utilizados *datasets* que possuíam e *datasets* que não possuíam *stopwords*, visando obter um resultado mais preciso.

Através do *widget WordCloud*[3], onde as palavras são listadas por sua frequência, foi possível monitorar os efeitos do pré-processamento, exibindo assim o estado atual do corpus. A *WordCloud* gerada neste trabalho, com o pré-processamento, pode ser vista na Figura 3.2 e a *WordCloud* gerada sem o pré-processamento é representada pela Figura 3.3.

[2] https://orange3-text.readthedocs.io/en/latest/widgets/preprocesstext.html
[3] https://orangedatamining.com/widget-catalog/text-mining/wordcloud/

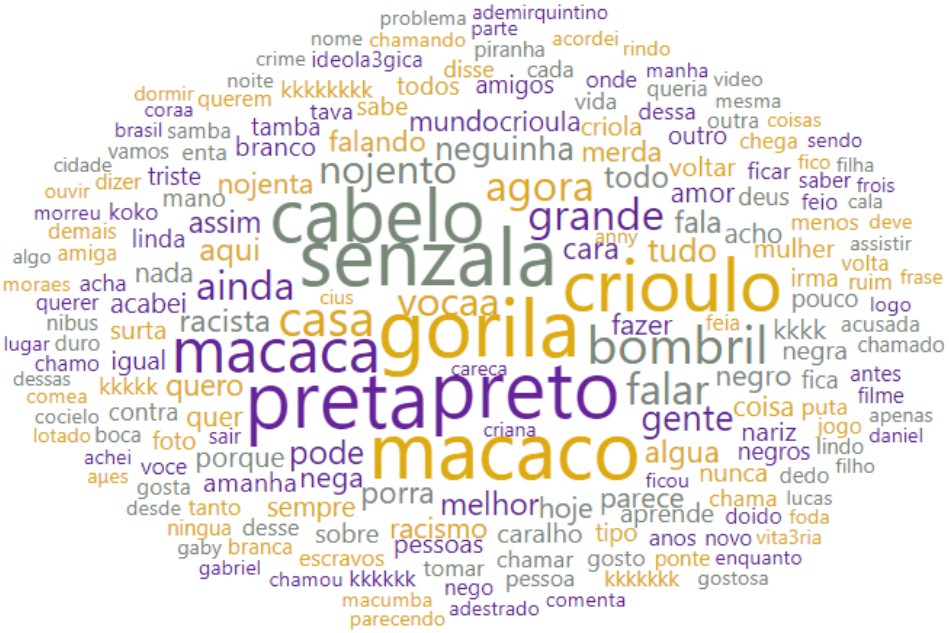

Figura 3.2: WordCloud do Experimento

Figura 3.3: WordCloud do Experimento sem pré-processamento

O último passo da etapa de pré-processamento foi a utilização da ferramenta *Bag of Words*[4] (BOW) do Orange, seguindo como base o artigo [7]. Essa ferramenta foi configurada de forma que combinasse *Term Frequency* (TF) e *Inverse Document Frequency* (IDF). Assim sendo, TF-IDF é uma estatística numérica projetada para refletir a importância das palavras para os documentos em uma coleção ou corpus. O TF é uma medida da frequência com que um termo aparece em um documento e IDF referindo-se à uma medida de quão importante o termo é.

3.3 Rotulação

Na etapa de rotulação, a base rotulada em [4] foi aproveitada. Nesta 349 *tweets* foram rotulados como “sim” e 1676 como “não” (sim = possui traços de racismo, o contrário, não). Para que fossem equilibradas as classes, foi necessária uma ampliação da mesma. Essa ampliação adicionou *tweets* com traços de racismo ao conjunto de treinamento, de forma que a base de dados passou a ser composta por 1200 *tweets*, sendo 600 rotulados como “sim” e 601 como “não”.

Durante a rotulação, surgiram diversos desafios, principalmente pelo fato de as mensagens conterem termos que normalmente seriam associados ao racismo mas, naquele momento, estavam sendo utilizados de forma “carinhosa” e “afetiva”, ou simplesmente foram absorvidos pelo dialeto popular, como pode ser vistos na Tabela 3.1:

Tweet	Traços de racismo
"Amo muito minha criola."	Não
"Neguinha, eu te amoo! To contigo."	Não

Tabela 3.1: Exemplos de *tweets* e seus rótulos

3.3.1 Problema de Classes Desbalanceadas

Uma questão de rotulação importante é o problema de classes desbalanceadas, que traduz-se em uma desproporção no número de dados de uma das classes do treinamento [4]. Como citado em [4],"modelos que são criados sob tais condições têm tendências de serem modelos com alta acurácia global, porém triviais, que predizem quase sempre a classe majoritária e não caracteriza um modelo bem representativo [18]".

Classificadores que usam *datasets* desequilibrados normalmente têm baixas taxas de erro para a classe majoritária e taxas de erro inaceitáveis para a classe minoritária

[4]https://orangedatamining.com/widget-catalog/text-mining/bagofwords-widget/

[18]. Normalmente, um classificador tem boa precisão de classificação para a classe majoritária, mas sua precisão para a classe minoritária é inaceitável, muitos classificadores não estão prontos para classificar com precisão a classe minoritária [18].

Em alguns cenários dos experimentos realizados neste trabalho foi utilizado de um *dataset* desbalanceado, logo, o comportamento esclarecido anteriormente foi reproduzido.

3.4 Implementação dos Modelos de Classificação Independentes

Na etapa de implementação dos modelos de classificação, a escolha dos classificadores e das técnicas foram baseadas no artigo [7], sendo eles *Naïve Bayes*, *K-Nearest Neighbours*, *Logistic Regression*, *Random Forest*, e *Support Vector Machines*. Esses classificadores são abordagens de *machine learning* supervisionada, sendo assim, necessitam da base rotulada de treinamento para a realização da classificação.

Para testar os classificadores foi aplicada a ferramenta *Test and Score*[5] do Orange, o *sampling method* empregado foi o *cross validation*, seguindo como base o artigo [7]. O *cross validation* divide os dados em um determinado número de *folds*, no experimento foram divididos em 10 *folds*. Em cada iteração de *cross validation*, 9 *folds* de *tweets* são usados como dados de treinamento e o *fold* restante é usado como dado de teste.

Os resultados gerados pelo *Test and Score* podem ser ultilizados por outros *widgets* do Orange, como por exemplo a *Confusion Matrix*[6], com o objetivo de analisar o desempenho de classificadores. A *Confusion Matrix* provê a proporção entre as classes previstas e as reais, podendo-se observar quais instâncias específicas foram mal classificadas.

3.5 Implementação do Ensemble

Na última etapa, a Implementação do *Ensemble*, foi realizada através de ferramentas do próprio Orange, tais como o método *Stacking*[7] e o método *AdaBoost*[8].

O método *Stacking*, é uma método ensemble que calcula um meta modelo a partir de vários modelos básicos, envolve treinar um algoritmo de aprendizado para combinar

[5] https://orangedatamining.com/widget-catalog/evaluate/testandscore/
[6] https://orangedatamining.com/widget-catalog/evaluate/confusionmatrix/
[7] https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/stacking.html
[8] https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/adaboost.html

as previsões de vários outros algoritmos de aprendizado. Já o método *Adaboost* pode ser usado com outros algoritmos de aprendizado para aumentar seu desempenho, ele faz isso ajustando os modelos fracos. Em geral, o *Boosting* é uma técnica que combina modelos fracos em um único modelo forte. Os métodos de *Boosting* do *AdaBoost* se assemelham à estrutura trabalhada no artigo [7], pois o algoritmo de classificação SAMME do *AdaBoost* utiliza dos resultados de classificação, similar ao *Hard Voting*, e o algoritmo de classificação SAMME.R utiliza das estimativas de probabilidade, similar ao *Soft Voting*.

CAPÍTULO 4

Resultados e Análise

Com o objeto de testar os algoritmos de aprendizagem em dados, foi utilizado o *widget* do Orange denominado *Test and Score*, o qual recebe como entrada um *dataset* e os algoritmos de aprendizagem, gerando como saída os resultados dos testes dos algoritmos de classificação.

Para o cálculo destes resultados, é necessário o entendimento de certo elementos, presentes em uma matriz de confusão4.1, sendo eles :

- **Verdadeiro Positivo** é um resultado que indica a classificação correta da classe Positivo.
- **Verdadeiro Negativo** é um resultado que indica a classificação correta da classe Negativo
- **Falso Positivo** é um resultado que indica um erro, no qual o modelo previu a classe Positivo quando o valor real era classe Negativo.
- **Falso Negativo** é em resultado que indica um erro, onde o modelo previu a classe Negativo quando o valor real era classe Positivo.

		Previsão	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo	Falso Positivo
	Não	Falso Negativo	Verdadeiro Negativo

Figura 4.1: Matriz de Confusão.

Uma das funcionalidades do *Test and Score* é mostrar uma tabela com diferentes medidas de desempenho do classificador, dentre elas:

- ***Classification accuracy*** é a proporção de exemplos classificados corretamente.
- ***F-1*** é uma média harmônica ponderada de precisão *Precision* e rechamada *Recall*;

- ***Precision*** é a proporção de verdadeiros positivos entre instâncias classificadas como positivas, por exemplo, a proporção de tweets com traços de racismo corretamente identificada como "contém traços".
- ***Recall*** é a proporção de verdadeiros positivos entre todas as instâncias positivas nos dados, por exemplo, o número de tweets com traços de racismo entre todos os rotulados como possuidores de traços.

No trabalho de [7], foi utilizada a média F-1 *Measure* como método de avaliação dos experimentos. Com o objetivo de fazer comparações, tal métrica também foi adotada neste trabalho, F-1, justamente por ser uma média harmônica entre *Precision* e *Recall*, mas também serão apresentadas outras diferentes medidas de desempenho.

Ao todo, foram realizados experimentos utilizando ambos *datasets*, sendo um desbalanceado e outro balanceado, com os classificadores independentes e com os métodos *Ensemble*, além do mais, foram efetuados experimentos com e sem *stopwords*. Sendo os classificadores independentes, SVM (*Support Vector Machines Classifiers*), KNN(*K-Nearest Neighbors*), NB(*Naive Bayes*), RF(*Random Forest*) e LR(*Logistic Regression*), e os métodos *Ensemble AdaBoost* e *Stacking*.

4.1 Resultados do Classificadores Independentes

As Tabelas 4.5, 4.6, 4.7,4.8 e 4.9 são os resultados alcançados pelos classificadores independentes. Começando com os experimentos realizados utilizando o *dataset* desbalanceado, com o uso das *stopwords*, destacam-se o SVM com F1 0.422, Tabela 4.5, com o pior resultado, mesmo seguindo as configurações de otimização apresentadas no trabalho [7], configurações que foram aplicadas em todos os cenários, e o RF com F1 0.871, Tabela 4.8, com o melhor desempenho como classificador independente.

Destacando o classificador KNN, Tabela 4.7, que apesar de sua métrica ser alta nesse experimento, F1 0.759, a Confusion matrix gerada mostra que na realidade uma grande parte de suas previsões estavam erradas, cerca de 339 situações onde a classificação correta seria “possui traço de racismo”, ele classificou como “ não possui traço de racismo”, como pode ser visto na Figura 4.2.

Passando para os experimentos realizados utilizando o *dataset* desbalanceado, mas sem o uso das *stopwords*, novamente com o SVM, Tabela 4.5, com o pior resultado, F1 0.445, e o RF, Tabela 4.8, com o melhor desempenho como classificador independente, F1 0.864.

A próxima etapa utilizou do *dataset* balanceado e das *stopwords*, obtendo alguns resultados muito próximos. A performance dos classificadores SVM, Tabela 4.5, e KNN, Tabela 4.7, foram similares e obtiveram as piores métricas, F1 0.554 e F1 0.563

respectivamente. Tiveram os melhores resultados os classificadores RF, Tabela 4.8, e LR, Tabela 4.9, F1 0.812 e F1 0.812 na devida ordem.

No passo seguinte, foi aplicado o *dataset* balanceado, mas sem as *stopwords*. Diferentes dos outros experimentos, o classificador KNN, Tabela 4.7, teve o pior desempenho, atingindo o resultado de F1 0.533. Novamente, o melhor desempenho foi do classificador RF, Tabela 4.8, alcançando F1 0.811.

Os classificadores individuais com o dataset desbalanceado atingiram resultados elevados em *Accuracy, F-1, Precision e Recall*, destacando os classificadores LR e RF. Devido ao uso do dataset desbalanceado, onde as classes "Positivo", Figura 4.1, compõem grande parte dos dados, as métricas tiveram um aumento significativo, mas os classificadores tiveram dificuldades de identificar ambas as classes, como podemos ver na Figura4.2. Onde tal comportamento está presente nos experimentos com os classificadores SVM e KNN, com um grande número de previsões erradas, Figura4.2, por conta de uma dificuldade em uma classificação equilibrada de ambos os rótulos.

4.2 Resultados dos Modelos Ensemble

Nessa seção estão presentes os resultados alcançados com as técnicas de *ensemble*, que consistem em combinar as previsões de vários classificadores distintos para obter previsões mais precisas e corretas, sendo tais técnicas aplicadas, o Adaboost e o Stacking. Seguindo como base a pesquisa realizada no trabalho [7], os métodos de *ensemble* foram utilizados em experimentos com todos os classificadores e em experimentos com apenas os três classificadores com melhores resultados, sendo eles NB(*Naive Bayes*), RF(*Random Forest*) e LR(*Logistic Regression*).

4.2.1 AdaBoost

Para o *dataset* desbalanceado, a Tabela 4.1 apresenta os resultados atingidos com o *Adaboost*, utilizando todos os classificadores no método *Ensemble*. Obteve-se métricas de F1 0.828 com o algoritmo de classificação SAMME.R e F1 0.869 para o algoritmo de classificação SAMME, ambos com *stopwords*. No mesmo cenário, mas sem as *stopwords*, o modelo *Adaboost* atingiu as métricas de F1 0.828 e F1 0.873 para os métodos de *Boosting* SAMME.R e SAMME, respectivamente.

Continuando com *dataset* desbalanceado, alterando apenas a questão dos classificadores, utilizando nesta situação apenas os três melhores algoritmos com o Adaboost, com o uso de *stopwords*, atingiu-se as métricas F1 0.877 e F1 0.869, com o algoritmo de classificação SAMME.R e com o algoritmo de classificação SAMME respectivamente, como pode ser visto na Tabela 4.2. Por outro lado, mantendo as mesmas configurações,

mas sem o uso de *stopwords*, atingiu-se os resultados presentes na mesma Tabela, sendo eles F1 0.877 e F1 0.873, para SAMME.R e SAMME por essa ordem.

Para o *dataset* balanceado, a Tabela 4.1 possui as métricas atingidas utilizando o *Adaboost*, com e sem *stopwords*, utilizando todos os classificadores no método *Ensemble*. Adquiriu-se as métricas de F1 0.783 com o algoritmo de classificação SAMME.R e F1 0.782 para o algoritmo de classificação SAMME, com *stopwords*. No cenário com todos os classificadores e sem *stopwords*, o modelo *Adaboost* atingiu as métricas de F1 0.796 e F1 0.800, para SAMME.R e SAMME por essa ordem.

Novamente, com a mesma configuração, alterando apenas a questão dos classificadores, utilizando nesta situação apenas os três melhores algoritmos com o *Adaboost* e com as *stopwords*, obteve-se as métricas presentes na Tabela 4.2, F1 0.783 e F1 0.801, com o algoritmo de classificação SAMME.R e com o algoritmo de classificação SAMME na devida ordem. Por outro lado, mantendo as mesmas configurações, mas sem o uso de *stopwords*, atingiu-se os resultados presentes nas mesmas figuras, sendo eles F1 0.796 e F1 0.800, para SAMME.R e SAMME por essa ordem.

Percebe-se que o *AdaBoost* obteve a métrica mais elevada com o *dataset* desbalanceado e sem a ultização das *stopwords*, destacando-se o cenário com os três melhores classificadores sendo utilizados no modelo *Ensemble* com o método de *Boosting*, que usa o algoritmo de classificação SAMME.R, atingindo F1 0.877.

4.2.2 Stacking

Para o *dataset* desbalanceado, as Tabelas 4.3 e 4.4 representam os resultados atingidos utilizando o *Stacking*, em seus diferentes cenários. Com a utilização das *stopwords*, alcançou-se métricas mais elevadas de F1 0.870 com todos os classificadores, e uma métrica de F1 0.859 com apenas os três melhores classificadores. Já sem as *stopwords*, atingiu- se métricas similares ao experimento anterior, onde os *Stacking* com todos os classificadores obteve o melhor resultado, com de F1 0.867, e a métrica de F1 0.864 com apenas os três melhores classificadores.

Para o dataset balanceado, as Tabelas 4.3 e 4.4 possuem os resultados atingidos utilizando o *Stacking*, com e sem *stopwords*. Pode ser visto, que com a utilização das *stopwords*, atingiu-se métricas de F1 0.813 para o *Stacking* com todos os classificadores e F1 0.810 com apenas os três melhores classificadores. No cenário sem as *stopwords*, com todos os classificadores e com apenas três, foram alcançadas as respectivas métricas F1 0.828 e F1 0.818 .

Com finalidade de acrescentar informações, as Figuras 4.3 e 4.4 representam as *Confusion Matrix* dos experimentos realizados com o método *Stacking* utilizando um *dataset* desbalanceado e um balanceado, na devida ordem, ambos com *stopwords*.

Nota-se que, apesar das métricas serem melhores no caso do *dataset* desbalanceado, o método *Stacking* no *dataset* balanceado realizou previsões mais equilibradas na questão das "classes" ou "rótulos", ou seja, ele desempenho melhor na tarefa de identificar ambas as classes.

4.3 Comparação dos Resultados dos Modelos Ensemble

Nesta seção serão comparados os resultados obtidos neste trabalho, com os resultados do experimento realizado no artigo [7], que por sua vez utilizaram de dois *datasets*, um balanceado e outro não, além do uso de *stopwords*. Logo, serão levados em conta os experimentos que mais se assemelham ao trabalho realizado em tal artigo. A fins de esclarecimento, neste trabalho foi realizado a detecção de traços de racismo no Twitter na língua portuguesa, enquanto no artigo a ser comparado, foi executado a detecção de discurso de ódio no Twitter na língua indonésia.

As Figuras 4.5 e 4.6 tratam dos resultados referentes aos dois experimentos realizados utilizando os diferentes *datasets*, no artigo [7]. Percebe-se que os três melhores resultados em ambos cenários são dos métodos de classificação de *Ensemble*. Com métricas de F1 0.789, F1 0.798 e F1 0.798 para o *dataset* desbalanceado, por meio dos modelos de classificação *Soft Voting*, *3-Best Hard Voting* e *3-Best Soft Voting*, respectivamente. No *dataset* balanceado, atingiu-se em destaque, métricas de F1 0.843 , F1 0.846 e F1 0.847 através dos respectivos modelos de classificação *Hard Voting*, *3-Best Soft Voting* e *Soft Voting*.

Já os resultados referentes aos modelos de *Ensemble* utilizados neste trabalho estão presentes nas Tabelas 4.11 e 4.10, sendo a primeira com o *dataset* desbalanceado e na segunda tabela o *dataset* balanceado. Repara-se que no primeiro cenário, o modelo *Ensemble AdaBoost*, com o algoritmo de classificação SAMME.R, similar ao *Soft Voting* [7], obteve o melhor desempenho, F1 0.877. Outra questão notada é que ambas as abordagens de *Ensemble* atingiram resultados melhores do que os classificadores independentes, exceto pelo classificador RF(*Random Forest*), que esteve entre os melhores em diversos testes.

Para os resultados do *dataset* balanceado, representados na Tabela 4.10, novamente o melhor resultado foi alcançado por um método *Ensemble*, *Stacking* com F1 0.813, e ambas as abordagens de *Ensemble* tiveram um ótimo desempenho, só não melhores do que os classificadores RF(*Random Forest*) e LR(*Logistic Regression*), que tiveram métricas de F1 0.812 e F1 0.812 respectivamente.

	AdaBoost - SaMME.R			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.840	0.834	0.900	0.900
CA	0.796	0.783	0.828	0.828
F1	0.796	0.783	0.828	0.828
Precision	0.797	0.783	0.828	0.828
Recall	0.796	0.783	0.828	0.828
	AdaBoost - SAMME			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.854	0.843	0.800	0.791
CA	0.800	0.782	0.875	0.871
F1	0.800	0.782	0.873	0.869
Precision	0.800	0.782	0.871	0.867
Recall	0.800	0.782	0.875	0.871

Tabela 4.1: Resultados do *Adaboost*, com todos os classificadores no método *Ensemble*.

	3- best-AdaBoost - SaMME.R			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.840	0.834	0.822	0.800
CA	0.796	0.783	0.880	0.880
F1	0.796	0.783	0.877	0.877
Precision	0.797	0.783	0.875	0.875
Recall	0.796	0.783	0.880	0.880
	3-best-AdaBoost - SAMME			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.854	0.869	0.800	0.791
CA	0.800	0.801	0.875	0.871
F1	0.800	0.801	0.873	0.869
Precision	0.800	0.802	0.871	0.867
Recall	0.800	0.801	0.875	0.871

Tabela 4.2: Resultados do *Adaboost*, com os três melhores classificadores no método *Ensemble*.

	Stacking			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.900	0.890	0.846	0.847
CA	0.828	0.813	0.876	0.878
F1	0.828	0.813	0.867	0.870
Precision	0.828	0.813	0.866	0.868
Recall	0.828	0.813	0.876	0.878

Tabela 4.3: Resultados do método *Ensemble Stacking*.

	3-best Stacking			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.897	0.892	0.846	0.840
CA	0.818	0.810	0.872	0.870
F1	0.818	0.810	0.864	0.859
Precision	0.818	0.810	0.862	0.858
Recall	0.818	0.810	0.872	0.870

Tabela 4.4: Resultados do método *Ensemble Stacking*, com apenas os três melhores classificadores.

	SVM			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.713	0.730	0.662	0.654
CA	0.583	0.596	0.403	0.385
F1	0.545	0.554	0.445	0.422
Precision	0.625	0.656	0.765	0.762
Recall	0.583	0.596	0.403	0.385

Tabela 4.5: Resultados do classificador individual SVM.

	NB			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.889	0.886	0.800	0.801
CA	0.799	0.799	0.682	0.690
F1	0.799	0.799	0.721	0.728
Precision	0.802	0.801	0.834	0.835
Recall	0.799	0.799	0.682	0.690

Tabela 4.6: Resultados do classificador individual NB.

	KNN			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.647	0.680	0.622	0.622
CA	0.570	0.598	0.832	0.831
F1	0.533	0.563	0.760	0.759
Precision	0.603	0.643	0.861	0.822
Recall	0.570	0.598	0.832	0.831

Tabela 4.7: Resultados do classificador individual KNN.

	RF			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.869	0.871	0.841	0.837
CA	0.811	0.812	0.876	0.878
F1	0.811	0.812	0.864	0.871
Precision	0.812	0.812	0.865	0.869
Recall	0.811	0.812	0.876	0.878

Tabela 4.8: Resultados do classificador individual RF.

	LR			
	Balanceado		Desbalanceado	
	s/ stopwords	c/ stopwords	s/ stopwords	c/ stopwords
AUC	0.881	0.877	0.823	0.823
CA	0.804	0.812	0.859	0.857
F1	0.804	0.812	0.844	0.843
Precision	0.804	0.812	0.842	0.841
Recall	0.804	0.812	0.859	0.857

Tabela 4.9: Resultados do classificador individual LR.

Métricas	AdaBoost SAMME	AdaBoost SAMME.R	3-Best-AdaBoost SAMME.R	3-Best AdaBoost SAMME	RF	LR	Stacking
AUC	0.843	0.834	0.834	0.869	0.871	0.877	0.890
CA	0.782	0.783	0.783	0.801	0.812	0.812	0.813
F1	0.782	0.783	0.783	0.801	0.812	0.812	0.813
Precision	0.782	0.783	0.783	0.802	0.812	0.812	0.813
Recall	0.783	0.783	0.782	0.801	0.812	0.812	0.813

Tabela 4.10: Melhores resultados alcançados com o *dataset* balanceado e com o uso das *stopwords*.

Métricas	AdaBoost SAMME.R	3-Best Stacking	3-Best AdaBoost SAMME	AdaBoost SAMME	Stacking	RF	3-Best AdaBoost SAMME.R
AUC	0.900	0.840	0.791	0.791	0.847	0.837	0.800
CA	0.828	0.870	0.871	0.871	0.878	0.878	0.880
F1	0.828	0.859	0.869	0.869	0.870	0.871	0.877
Precision	0.828	0.858	0.867	0.867	0.868	0.869	0.875
Recall	0.828	0.870	0.871	0.871	0.878	0.878	0.880

Tabela 4.11: Melhores resultados alcançados com o *dataset* desbalanceado e com o uso das *stopwords*.

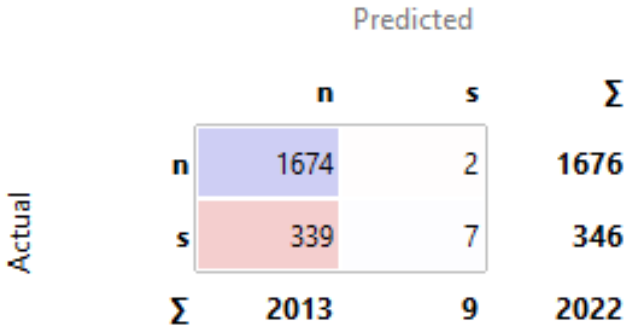

Figura 4.2: *Confusion Matrix* do KNN.

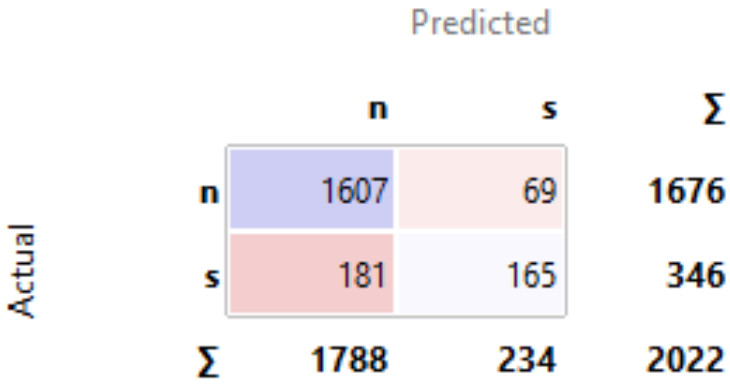

Figura 4.3: *Confusion Matrix* do *Stacking*, com um *dataset* desbalanceado.

Actual \ Predicted	n	s	Σ
n	489	112	601
s	116	484	600
Σ	605	596	1201

Figura 4.4: *Confusion Matrix* do *Stacking*, com um *dataset* balanceado.

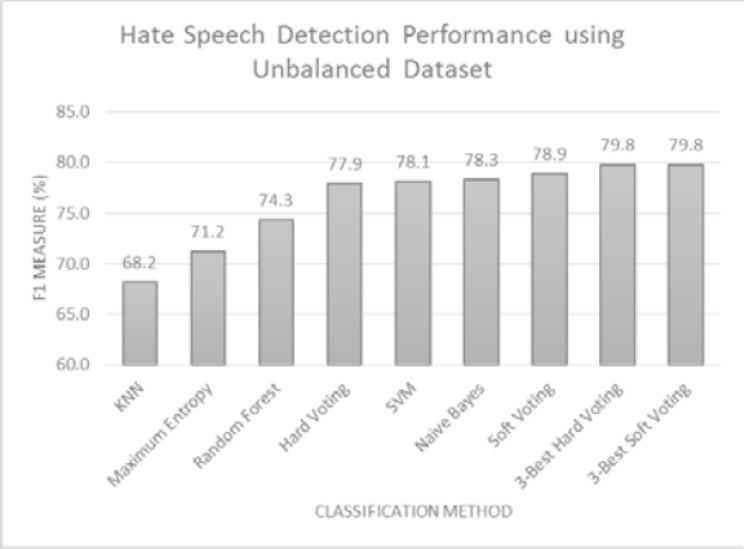

Figura 4.5: Imagem presente no trabalho[7], contendo os melhores utilizando o *dataset* balancedo com o uso de *stopwords*.

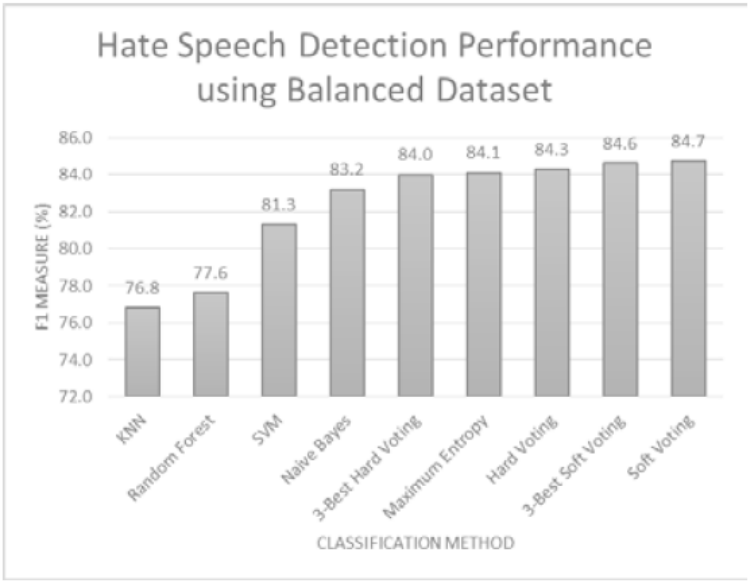

Figura 4.6: Imagem presente no trabalho[7], contendo os melhores utilizando o *dataset* desbalancedo com o uso de *stopwords*.

CAPÍTULO 5

Conclusão

Este trabalho apresentou uma visão geral da classificação automática de textos em língua portuguesa com o objetivo de detectar traços de racismo no Twitter, utilizando de cinco classificadores independentes e de dois métodos *Ensemble*.

Ao dispor de diferentes *datasets* e configurações, tal como o uso e desuso de *stopwords*, obteve-se os resultados experimentais onde o *Random Forest*, como classificador independente, atingiu o melhor resultado em grande parte dos cenários. Além disso, foi possível perceber que os métodos *Ensemble* podem produzir resultados mais precisos e corretos, como no caso do *Stacking*, alcançando F1 0.828, através do uso do conjunto com os cincos classificadores, sendo capaz de classificar ambos os rótulos de maneira proporcional.

Ambos experimentos que utilizaram os modelos de *Ensemble*, mostraram que o uso de tal técnica pode melhorar o desempenho da aplicação, mesmo que a melhoria não seja significativa comparada ao resultados de certo classificadores, o uso do método *Ensemble* pode reduzir o risco de escolher um classificador que não irá se adequar a uma situação específica.

Como exemplificado anteriormente, um dos grandes desafios na realização deste experimento, foi a questão da utilização do *dataset* desbalanceado, pelo fato das métricas serem geradas a partir de uma média sobre as categorias(*average over classes*), pois os resultados gerados com o mesmo, apresentam métricas mais altas do que as alcançadas com o *dataset* balanceado. Na realidade, com o *dataset* desbalanceado, os classificadores e até mesmo os modelos *Ensemble*, tinha uma grande dificuldade de classificar de forma equilibrada ambos os rótulos. Logo, as classificações pendiam para apenas um lado, sendo os incapazes os classificadores de identificar claramente cada categoria, mesmo possuindo métricas altas.

Em futuras oportunidades pode-se aprimorar o processo de configuração dos modelos, buscando recursos que possam levar em direções mais promissoras, com resultados mais precisos. Podendo-se, também, realizar um estudo mais aprofundado em relação a estas configurações, pois neste experimento, em seus diversos cenários, com diferentes formatos, nem sempre os resultados saíram conforme o esperado.

Em trabalhos futuros, pode-se ir além através de outros tipos de abordagens de classificação automática de texto, como por exemplo a metodologia descrita em [10], que por sua vez leva em conta palavras que incorporam dados demográficos. Tendo como objetivo, superar alguns desafios presentes no Processamento de Linguagem Natural, a subjetividade e a ambiguidade.

Referências Bibliográficas

[1] AGRAWAL, S.; AWEKAR, A. **Deep learning for detecting cyberbullying across multiple social media platforms**. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10772 LNCS, p. 141–153. Springer Verlag, 2018.

[2] ALMEIDA, S. **Racismo estrutural**. Pólen Produção Editorial LTDA, 2019.

[3] BEHERA, R. N.; ROY, M.; DASH, S. **Ensemble based hybrid machine learning approach for sentiment classification-a review**. *International Journal of Computer Applications*, 146(6):31–36, 2016.

[4] DA SILVA, R. C. C.; FERNANDES, D. S. A.; FERNANDES, M. G. C. **Caracterização de mensagens em língua portuguesa com traços de racismo no twitter**. In: *Anais da VI Escola Regional de Informática de Goiás*, p. 205–214. SBC, 2018.

[5] DOS, A.; DA SILVA, S. R.; ROMAN, N. T. **Hate Speech Detection in Portuguese with Na¨ıveNa¨ıve Bayes, SVM, MLP and Logistic Regression**. Technical report.

[6] ELOUALI, A.; ELBERRICHI, Z.; ELOUALI, N. **Hate speech detection on multilingual twitter using convolutional neural networks**. *Revue d'Intelligence Artificielle*, 34(1):81–88, 2020.

[7] FAUZI, M. A.; YUNIARTI, A. **Ensemble method for indonesian twitter hate speech detection**. *Indonesian Journal of Electrical Engineering and Computer Science*, 11(1):294–299, jul 2018.

[8] FOUNTA, A. M.; CHATZAKOU, D.; KOURTELLIS, N.; BLACKBURN, J.; VAKALI, A.; LEONTIADIS, I. **A unified deep learning architecture for abuse detection**. In: *WebSci 2019 - Proceedings of the 11th ACM Conference on Web Science*, p. 105–114. Association for Computing Machinery, Inc, jun 2019.

[9] GARCIA, K.; BERTON, L. **Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA**. *Applied Soft Computing*, 101, mar 2021.

[10] HASANUZZAMAN, M.; DIAS, G.; WAY, A. **Demographic word embeddings for racism detection on twitter**. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, p. 926–936, 2017.

[11] HERODOTOU, H.; CHATZAKOU, D.; KOURTELLIS, N. **A Streaming Machine Learning Framework for Online Aggression Detection on Twitter**. jun 2020.

[12] HIRSCH, L.; BRUNSDON, T. **A comparison of lucene search queries evolved as text classifiers**. *Applied Artificial Intelligence*, 32(7-8):768–784, 2018.

[13] JAKUBOWICZ, A.; DUNN, K.; MASON, G.; PARADIES, Y.; BLIUC, A.-M.; BAHFEN, N.; OBOLER, A.; ATIE, R.; CONNELLY, K. **Cyber Racism and Community Resilience Strategies for Combating Online Race Hate**. Technical report.

[14] JIA, B.; DZITAC, D.; SHRESTHA, S.; TURDALIEV, K.; SEIDALIEV, N. **An Ensemble Machine Learning Approach to Understanding the Effect of a Global Pandemic on Twitter Users' Attitudes**. *International Journal of Computers, Communications and Control*, 16(2):1–11, 2021.

[15] KOWSARI, K.; MEIMANDI, K. J.; HEIDARYSAFA, M.; MENDU, S.; BARNES, L.; BROWN, D. **Text classification algorithms: A survey**. *Information (Switzerland)*, 10(4):1–68, 2019.

[16] MEDHAT, W.; HASSAN, A.; KORASHY, H. **Sentiment analysis algorithms and applications: A survey**. *Ain Shams Engineering Journal*, 5(4):1093–1113, dec 2014.

[17] MINAEE, S.; KALCHBRENNER, N.; CAMBRIA, E.; NIKZAD, N.; CHENAGHLU, M.; GAO, J. **Deep learning–based text classification: a comprehensive review**. *ACM Computing Surveys (CSUR)*, 54(3):1–40, 2021.

[18] MONARD, M. C.; BATISTA, G. **Learning with skewed class distributions**. *Advances in Logic, Artificial Intelligence and Robotics*, 85:173–180, 2002.

[19] PARK, J. H.; FUNG, P. **One-step and Two-step Classification for Abusive Language Detection on Twitter**. Technical report.

[20] PEI, X.; MEHTA, D. **Beyond a binary of (non)racist tweets: A four-dimensional categorical detection and analysis of racist and xenophobic opinions on Twitter in early Covid-19**. jul 2021.

[21] PITSILIS, G. K.; RAMAMPIARO, H.; LANGSETH, H. **Effective hate-speech detection in Twitter data using recurrent neural networks**. *Applied Intelligence*, 48(12):4730–4742, dec 2018.

[22] WATANABE, H.; BOUAZIZI, M.; OHTSUKI, T. **Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection**. *IEEE Access*, 6:13825–13835, feb 2018.