Relato de Experiência na Avaliação em Larga Escala de Estratégias de Undersampling para Redução de Viés em Classificação de Texto Baseada em SLMs/LLMs

Guilherme Fonseca[1], Gabriel Prenassi[2], Washington Cunha[3], Marcos André Gonçalves[1] , Leonardo Rocha[2]

[1]Universidade Federal de Minas Gerais (UFMG), Brasil

[2]Universidade Federal de São João del Rei (UFSJ), Brasil

[3]Universidade Estadual de Campinas (Unicamp), Brasil

{guilhermefonseca,mgoncalv}@dcc.ufmg.br, wcunha@unicamp.br,

prenassigabriel@aluno.ufsj.edu.br, lcrocha@ufsj.edu.br

Resumo. *Este artigo apresenta um relato de experiência sobre o uso da infraestrutura em nuvem da AWS para viabilizar uma ampla avaliação de métodos de undersampling em Classificação Automática de Texto (CAT) com SLMs e LLMs. O protocolo experimental envolveu 21 técnicas de undersampling, 13 bases de dados, com até 1,3 milhão de instâncias, e modelos como RoBERTa e Llama 3.1, impondo demandas computacionais massivas e heterogêneas. A solução adotou o Amazon S3 como data lake e instâncias EC2 especializadas: c6a.8xlarge para balanceamento de dados e g4dn.xlarge/g5.xlarge para fine-tuning e inferência. A nuvem permitiu padronizar o ambiente experimental, paralelizar execuções e assegurar rigor metodológico na análise de desempenho e tempo.*

Abstract. *This paper presents an experience report on using AWS cloud infrastructure to enable a broad evaluation of undersampling methods for Automated Text Classification (ATC) with SLMs and LLMs. The experimental protocol involved 21 undersampling techniques, 13 datasets, with up to 1.3 million instances, and models such as RoBERTa and Llama 3.1, imposing massive and heterogeneous computational demands. The solution used Amazon S3 as a data lake and specialized EC2 instances: c6a.8xlarge for data balancing and g4dn.xlarge/g5.xlarge for fine-tuning and inference. Cloud adoption enabled standardized experimental environments, parallel execution, and methodological rigor in performance and time analysis.*

1. Introdução

A Classificação Automática de Texto (CAT) tem sido amplamente utilizada como ferramenta essencial para a análise e organização de grandes volumes de documentos, com grande importância em diversos cenários desafiadores, incluindo relevance feedback, análise de sentimentos, avaliações de produtos, entre outros [Cunha et al. 2023]. A área de CAT passou por uma rápida evolução nos últimos anos com o advento e a popularização de modelos de aprendizagem profunda baseados em Transformers [Vaswani et al. 2017], tanto Pequenos Modelos de Linguagem (SLMs – como BERT e RoBERTa) quanto Grandes Modelos de Linguagem (LLMs – como GPT e Llama).

Apesar dos enormes avanços de efetividade obtidos com o uso dessa nova arquitetura em CAT, estas abordagens ainda podem ser afetadas negativamente por problemas tradicionais de classificação, como o viés de aprendizado decorrente de coleções de dados desbalanceadas. Nesses cenários, as classes minoritárias podem ser sub-representadas no processo de aprendizado, resultando em modelos com baixa capacidade de generalização e enviesados em favor das classes majoritárias. Além de questões éticas relacionadas a modelos enviesados [Ferrer et al. 2021], existem diversos cenários em que a classe minoritária é a de interesse, como no exemplo do cenário de predição de doenças [Zanotto et al. 2021].

Existem duas principais abordagens para lidar com o desbalanceamento de dados. Uma delas é *oversampling*, que consiste em criar amostras da classe minoritária para igualá-las à classe majoritária [Chawla et al. 2002]. Porém, para a CAT, essa estratégia recai sobre diversos problemas que tornam sua aplicação menos direta do que em outros domínios, como a garantia da qualidade linguística e da fidelidade semântica das instâncias geradas, e o aumento de custos computacionais no treinamento dos modelos. A outra solução existente, e foco de nossos estudos, é o *undersampling* (US), que visa reduzir o número de instâncias da classe majoritária para alcançar o equilíbrio entre as classes [Fonseca et al. 2024b, Fonseca et al. 2025]. Devido à falta de estudos sobre *undersampling* no cenário de CAT, nossa pesquisa teve como foco principal investigar como esses métodos interagem com SLMs e LLMs aplicados a essa tarefa. Fizemos essa análise focando em avaliar os algoritmos de *undersampling* pela capacidade deles alcançarem cinco objetivos simultaneamente: (i) reduzir o viés do classificador em direção à classe majoritária; ao mesmo tempo em que (ii) mantêm (ou melhoram) a efetividade e são (iii) eficientes; (iv) escaláveis para grandes conjuntos de dados; e (v) consistentes quando aplicados tanto a SLMs quanto a LLMs.

A realização dessa avaliação exigiu um amplo desenho experimental, que incluiu a execução de 21 algoritmos de *undersampling* (19 métodos tradicionais e 2 propostos por nós), em conjunto com classificadores baseados em SLM (RoBERTa) e em LLM (Llama 3.1). Esse tipo de investigação, embora necessária do ponto de vista metodológico, impõe demandas computacionais muito elevadas, requerendo servidores com diferentes perfis de processamento: maior capacidade de CPU para os métodos de *undersampling* e necessidade de GPU para os modelos de classificação.

Nesse cenário, este artigo relata a experiência de concepção, planejamento e execução de uma extensa investigação sobre o comportamento de técnicas de *undersampling* aplicadas a SLMs e LLMs. O objetivo central é evidenciar o papel fundamental da computação em nuvem (via infraestrutura da AWS) na viabilização de estudos de alta complexidade metodológica, tornando possível a realização de experimentos em larga escala que seriam inexequíveis em infraestruturas locais típicas de grupos de pesquisa. Para tanto, o artigo traz três contribuições principais: (i) descreve a estruturação da nuvem para suportar mais de 5.000 execuções experimentais com demandas heterogêneas de processamento operando paralelamente; (ii) discute as decisões práticas de alocação de recursos, ponderando restrições de tempo, custo e particularidades dos algoritmos; e (iii) sintetiza os principais aprendizados obtidos, servindo de guia para projetos acadêmicos de experimentação intensiva em Inteligência Artificial e Ciência de Dados.

2. Contexto e motivação do projeto

O projeto teve início com a constatação de que a literatura recente sobre CAT, apesar dos enormes avanços advindos de modelos baseados na arquitetura Transformers, carece de investigações aprofundadas sobre como o viés de aprendizado, oriundo de coleções de dados desbalanceadas, afeta tais modelos e sobre como métodos de undersampling se comportam quando aliados a SLMs e LLMs para minimizar esse viés. A adoção crescente de algoritmos de aprendizado de máquina em cenários reais críticos, como a análise de sentimentos em avaliações de produtos ou pontos de interesse, exige modelos que não apenas atinjam alta eficácia geral, mas que também sejam resilientes aos vieses gerados por distribuições assimétricas.

A partir dessa constatação inicial, formulamos duas perguntas de pesquisa (PP) para guiar nosso trabalho: **PP1)** *Como os modelos no estado da arte, baseados em SLMs e LLMs, são afetados pelo desbalanceamento de classes em tarefas de análise de sentimentos, e há espaço para melhorias?* **PP2)** *A aplicação de métodos de undersampling em SLMs e LLMs para a CAT é capaz de mitigar o viés de classificação decorrente do desbalanceamento de dados? Qual é o impacto dessa abordagem na eficácia e eficiência dos modelos? Observa-se consistência nos resultados obtidos entre SLMs e LLMs? As técnicas são escaláveis para bases de dados de médio e grande porte?*

Responder a essas questões exigia um desenho experimental rigoroso e abrangente. Sendo necessário trabalhar com diversos métodos de undersampling, diversos modelos de classificação e tudo isso em um conjunto amplo de bases de dados de diferentes níveis de desbalanceamento e escalas. A execução de tal protocolo experimental mostrou-se um desafio, pois, além de bastante ampla, as execuções necessitavam de máquinas com perfis de execução muito distintos entre si. De um lado, os algoritmos de *undersampling* tradicionais apresentam perfis de processamento altamente heterogêneos e exigentes em CPU e em memória RAM. De outro lado, a etapa subsequente de fine-tuning e inferência dos modelos SLMs e LLMs demanda instâncias com aceleração por GPU.

A conciliação desses dois perfis distintos de processamento, para milhares de execuções variadas, em um prazo viável, ultrapassou a capacidade da infraestrutura local do laboratório de pesquisa. Nesse contexto, **a adoção da computação em nuvem mostrou-se a única forma de viabilizar todo o protocolo experimental do projeto empírico.**

3. Caracterização da experiência

3.1. Escopo experimental

A metodologia consistiu em um estudo comparativo abrangente para avaliar o impacto do *undersampling* na CAT. O protocolo experimental englobou 13 conjuntos de dados textuais clássicos da literatura, selecionados por sua diversidade de escala e proporção de classes. Dez desses conjuntos são de pequeno a médio porte (até cerca de 30.000 instâncias) [Ribeiro et al. 2016], enquanto três são de grande escala [Fonseca et al. 2025], superando a marca de 1,3 milhão de documentos. O volume de dados variou de 752 (SentiStrength BBC) a mais de 1,3 milhão de documentos (CDS Reviews). Além disso, as bases apresentam diferentes níveis de desbalanceamento, com a razão entre a classe majoritária e a minoritária variando de 1,4:1 a 39:1.

Para responder à PP1, conduziu-se, inicialmente, um estudo aplicando diversos classificadores aos conjuntos de dados, com o objetivo de avaliar a eficácia e o viés de desbalanceamento de cada abordagem. O escopo da avaliação incluiu: (i) algoritmos tradicionais, como SVM [Joachims 1998], Regressão Logística [LaValley 2008], Random Forest [Biau and Scornet 2016], XGBoost [Chen and Guestrin 2016] e LightGBM [Ke et al. 2017]; (ii) modelos de linguagem de pequeno porte (SLMs), como RoBERTa [Liu et al. 2019], BART [Lewis et al. 2020] e BERT [Devlin et al. 2018]; e (iii) um representante dos LLMs, o Llama 3.1 8B [Dubey et al. 2024].

Para responder à PP2, o projeto implementou e analisou 21 métodos de undersampling distintos. Esse conjunto incluiu 19 técnicas estabelecidas na literatura (como Tomek Links [Tomek 1976], Near Miss [Mani and Zhang 2003] e OSS [Kubat et al. 1997], entre outras) e duas novas abordagens propostas neste trabalho: E2SC-US e UBR, ambas inspiradas em estratégias de Seleção de Instâncias. Em parte dos experimentos, utilizamos os classificadores RoBERTa e Llama 3.1, que foram, respectivamente, o melhor SLM e o melhor LLM encontrados na etapa anterior. Para garantir a robustez estatística dos resultados, todos os experimentos foram executados a partir do procedimento de validação cruzada (k-fold cross-validation) com 10 folds.

Todo o conjunto de experimentos propostos para responder à PP1 e à PP2 resultou em um desenho experimental com milhares de execuções distintas de pipelines completos (pré-processamento, undersampling, fine-tuning e inferência). Além do volume, o desafio central residia na heterogeneidade computacional do processo: enquanto os métodos de undersampling exigiam uma alta capacidade de CPU dos servidores, a etapa de treinamento e inferência dos Transformers dependia estritamente do uso de instâncias com aceleração por GPU. O volume experimental e a diferença nos requisitos de processamento de cada etapa do protocolo tornaram a execução em uma infraestrutura local inviável.

4. Infraestrutura em nuvem adotada

Diante da escala do projeto, a infraestrutura local mostrou-se insuficiente, motivando a migração para a computação em nuvem. Para isso, a solução foi estruturada em dois componentes principais da AWS: armazenamento escalável de dados (Amazon S3) e servidores de processamento (Amazon EC2). Neste último, utilizamos tanto instâncias focadas em CPU quanto em GPU.

4.1. Armazenamento e Gerenciamento de Dados com Amazon S3

A condução do amplo protocolo experimental exigiu o gerenciamento dos dados textuais referentes aos 13 conjuntos de dados utilizados (incluindo coleções de larga escala com mais de 1,3 milhão de instâncias, como a base CDS Reviews), além das novas bases balanceadas pelas 21 técnicas de undersampling para cada um dos datasets. Para suportar essa quantidade de dados, o Amazon S3 foi adotado como data lake do projeto. Todo o processo de sincronização de dados, desde o envio das partições de validação cruzada até as novas bases rebalanceadas, foi realizado por meio da interface de linha de comando da AWS (AWS CLI). Essa estratégia permitiu automatizar a transferência ágil de dados entre os buckets do S3 e os volumes Amazon EBS anexados às diferentes instâncias EC2 utilizadas. Como resultado, foi possível minimizar gargalos de entrada e saída de dados.

4.2. Instâncias otimizadas para computação (CPU)

A etapa de undersampling exigiu processamento exclusivo de CPU, especialmente para métodos tradicionais que dependem do cálculo de k-vizinhos mais próximos (KNN) ou de algoritmos iterativos que são otimizados para ambientes de CPU. Para a execução eficiente dessa fase, o projeto alocou instâncias do tipo **c6a.8xlarge**. Esse tipo de instância apresenta 32 vCPUs e 64 GB de memória RAM, o que o torna ideal para a execução dos métodos de undersampling. Os métodos tradicionais de classificação (SVM, Regressão Logística, Random Forest, XGBoost e LightGBM) também foram executados nesse mesmo ambiente, pois suas implementações são mais otimizadas para execução em CPU e não se beneficiariam de instâncias aceleradas por GPU.

4.3. Instâncias aceleradas por GPU

Para a etapa de classificação, que englobava o fine-tuning e a inferência dos modelos de linguagem nas bases de dados, o uso de hardware especializado foi obrigatório. Para os modelos SLMs (RoBERTa, BART e BERT), os experimentos foram conduzidos utilizando instâncias **g4dn.xlarge**, que têm uma GPU NVIDIA T4 com 16 GB de memória. Essas instâncias provaram ser o ponto de equilíbrio ideal entre custo e desempenho para arquiteturas do porte dos SLMs. Para os LLMs, por sua vez, devido à sua quantidade substancialmente maior de parâmetros (8 bilhões), foi necessário utilizar uma máquina mais robusta, e foram escolhidas instâncias do tipo **g5.xlarge**, que têm uma GPU NVIDIA A10G com 22,4 GB de memória.

4.4. Execuções Paralelas e Padronização de Ambiente

Além da adequação dos perfis de hardware (CPU vs. GPU), a adoção da nuvem viabilizou a paralelização massiva das execuções experimentais. O escopo do projeto exigia milhares de testes distintos que, se executados de forma sequencial em um único servidor, consumiriam meses de processamento. A elasticidade da AWS permitiu o uso simultâneo de múltiplas instâncias idênticas, escalando horizontalmente a execução dos pipelines.

Esse paralelismo padronizado foi um fator crítico para responder à PP2, que demandava a medição do tempo de execução de cada método de undersampling e do respectivo fine-tuning para a avaliação dos ganhos de eficiência advindos das técnicas de undersampling. Para que as comparações de tempo entre os 21 algoritmos fossem metodologicamente justas, era necessário que os testes ocorressem em ambientes iguais. A infraestrutura em nuvem garantiu essa uniformidade através do uso de imagens padronizadas, assegurando que todas as instâncias paralelas rodassem sob a exata mesma configuração de hardware, sistema operacional, drivers e versões de bibliotecas.

Em uma arquitetura local típica de laboratórios de pesquisa, replicar esse cenário seria praticamente inviável. Clusters acadêmicos frequentemente sofrem com a heterogeneidade de hardware (máquinas de gerações e especificações diferentes) e com a concorrência de recursos por múltiplos usuários, fatores que inserem ruídos inaceitáveis nas medições de tempo de execução. Dessa forma, a capacidade de subir máquinas idênticas sob demanda não apenas reduziu drasticamente o tempo total do calendário da pesquisa, tornando a execução muito mais rápida, mas também conferiu o rigor científico e o isolamento necessários para a validade das métricas de eficiência avaliadas no estudo.

5. Execução do projeto e resultados alcançados

O uso da infraestrutura em nuvem da aws permitiu a execução do plano experimental do estudo em sua totalidade, não sendo necessário fazer qualquer limitação no estudo por questões de hardware. Com isso, foi possível realizar nosso objetivo de avaliar como métodos de undersampling se comportam quando aplicados junto a classificadores baseados em SLMs e LLMs.

Após a execução e a análise de todos os experimentos, obtivemos achados relevantes para o campo. A partir dos experimentos da PP1, chegamos à conclusão de que os modelos baseados em transformers, além de mais efetivos, possuem um viés menor quando comparados a modelos gerados por classificadores tradicionais. No entanto, nossos resultados também apontam que ainda há espaço considerável para melhorias, principalmente em bases de dados com elevado grau de desbalanceamento (razão entre o número de documentos da classe majoritária e o da classe minoritária superior a 5).

Os resultados da PP2 mostram que apenas as duas estratégias de undersampling propostas por nós, UBR e E2SC-US, obtiveram bons resultados em todos os critérios avaliados. Elas foram as únicas a conseguir, simultaneamente, reduzir o viés do modelo em relação à classe majoritária e diminuir o tempo total de treinamento e classificação, mantendo a sua eficácia em comparação ao modelo treinado sem undersampling. Tudo isso de maneira escalável para as bases de dados grandes e apresentando desempenhos consistentes entre SLMs e LLMs.

Em termos de produção científica, a experiência resultou em três artigos diretos já publicados, sendo eles (i) *Análise Comparativa de Métodos de Undersampling em Classificação Automática de Texto Baseada em Transformers*, publicado em Revista Eletrônica de Iniciação Científica em Computação, em 2024 [Fonseca et al. 2024a]; (ii) *Estratégias de Undersampling para Redução de Viés em Classificação de Texto Baseada em Transformers*, publicado no WebMedia, em 2024 [Fonseca et al. 2024b]; (iii) *Instance-Selection-Inspired Undersampling Strategies for Bias Reduction in Small and Large Language Models for Binary Text Classification*, publicado na ACL, em 2025 [Fonseca et al. 2025].

6. Lições aprendidas

A execução do projeto trouxe lições valiosas e um conjunto de aprendizados práticos e metodológicos altamente relevantes para pesquisas em nuvem.

6.1. A nuvem como um auxílio metodológico

A principal lição aprendida foi perceber que a nuvem da AWS atuou como base viabilizadora do rigor científico da pesquisa, sendo uma etapa fundamental e indispensável para a execução integral do plano metodológico do projeto. Para responder de forma robusta às PP1 e PP2, foi necessário cruzar 21 técnicas de undersampling com 13 bases de dados (algumas com mais de 1,3 milhão de documentos), utilizando validação cruzada de 10 folds. Executar esse volume de testes sequencialmente em modelos do porte do Llama 3.1 (8 bilhões de parâmetros) seria impraticável em um laboratório físico em tempo hábil. Com a AWS, pudemos executar os testes sem reduzir o escopo do projeto.

6.2. Desacoplamento de hardware e perfis de processamento

A experiência evidenciou que tentar resolver todos os gargalos com um único tipo de instância, com altas capacidades de CPU e GPU, é um equívoco técnico e financeiro. Os métodos de undersampling baseiam-se em cálculos intensivos que demandam e são otimizados exclusivamente para CPU e memória RAM, enquanto o treinamento e a inferência dos SLMs/LLMs necessitam quase que exclusivamente do uso de aceleradores de GPU. Ao desacoplar o fluxo de trabalho, utilizando instâncias de CPU (c6a.8xlarge) apenas para gerar o balanceamento de dados e instâncias isoladas de GPU (g4dn.xlarge e g5.xlarge) apenas para o fine-tuning dos Transformers, o projeto pôde buscar instâncias focadas em um desses dois aspectos, o que resultou em um custo menor do que se tivéssemos buscado um único tipo de instância que fosse robusta nos dois aspectos.

6.3. Paralelismo e Padronização de ambiente para medições de eficiência

Um dos objetivos centrais da PP2 consistiu em avaliar o impacto dos métodos de undersampling na eficiência dos modelos. Em clusters locais, a heterogeneidade do hardware e a concorrência de recursos com outros pesquisadores introduzem variações de tempo que inviabilizam medições precisas. O ambiente em nuvem, por sua vez, permitiu a criação de Imagens de Máquina (AMIs) padronizadas, assegurando que todos os experimentos paralelos fossem executados sob idênticas condições de sistema, de drivers e de capacidade de processamento. Essa padronização viabilizou a paralelização dos experimentos, reduzindo drasticamente o tempo total de execução em comparação a uma abordagem estritamente sequencial.

7. Desafios e limitações

Apesar de resolver muitos problemas, o uso da infraestrutura de nuvem da aws também apresenta desafios próprios. O primeiro deles foi o controle financeiro e de créditos, instâncias com aceleração por GPU, em especial as do tipo g5.xlarge, têm um custo por hora significativamente elevado quando em execução. Qualquer falha em um script ou descuido que deixasse uma máquina ociosa, aguardando um processo travado ou não a desligasse após a conclusão da tarefa, poderia resultar em um desperdício orçamentário considerável. O projeto exigiu uma rotina de monitoramento e de encerramento automático das instâncias EC2 logo após a exportação dos resultados para o Amazon S3.

Outro desafio enfrentado foi a maior necessidade de organização para a execução dos experimentos em paralelo. Esse cenário tornou necessário um controle mais rigoroso sobre quais experimentos já foram executados, quais ainda faltam e em quais instâncias cada processo está alocado.

8. Conclusão

Este artigo apresentou um relato de experiência sobre o planejamento, a orquestração e o uso de infraestrutura em nuvem na execução de um projeto de pesquisa voltado à avaliação em larga escala de estratégias de undersampling em Modelos de Linguagem (SLMs e LLMs) para a Classificação Automática de Texto. A condução do rigoroso protocolo experimental, que exigiu a utilização de 21 métodos de undersampling, 13 conjuntos de dados com diferentes níveis de desbalanceamento e escala e o uso de modelos de classificação baseados em SLMs e LLMs, gerou uma demanda computacional massiva e heterogênea.

A experiência demonstrou que a adoção da computação em nuvem (via AWS) foi o fator decisivo para transformar um desenho metodológico extenso em uma investigação efetivamente realizável e executável em tempo hábil. A estratégia de particionar e desacoplar o fluxo de trabalho experimental, alocando a tarefa de undersampling em servidores otimizados para CPU e o fine-tuning e a inferência dos Transformers em instâncias aceleradas por GPU, garantiu o isolamento necessário para utilizar máquinas específicas para cada etapa. Além disso, a capacidade de paralelizar os experimentos em instâncias de mesma configuração foi fundamental para a execução de todo o protocolo experimental em um tempo muito inferior ao que seria gasto executando o processo em uma infraestrutura local.

Do ponto de vista científico, o suporte dessa infraestrutura paralela e padronizada viabilizou a comprovação empírica de que as novas estratégias propostas (UBR e E2SC-US) são capazes de mitigar de forma consistente o viés dos classificadores em direção à classe majoritária, sem degradar a eficácia geral (Macro-F1) e sem aumentar o tempo de treinamento. Tudo isso, enquanto se mostram escaláveis para bases de dados grandes e mantêm a consistência dos resultados entre SLMs e LLMs. Por fim, como trabalhos futuros, pretendemos continuar utilizando a estrutura de nuvem da AWS para aprofundar os estudos relatados no presente trabalho.

Agradecimentos

Este trabalho foi apoiado por: CNPq, CAPES, Instituto Nacional de Ciência e Tecnologia em Inteligência Artificial Responsável para Linguística Computacional, Tratamento e Disseminação de Informação (INCT-TILD-IAR), FAPEMIG, AWS, Google, NVIDIA, CIIASaúde e FAPESP.

Declaração sobre uso de Inteligência Artificial

Os autores utilizaram uma ferramenta de inteligência artificial generativa como apoio à organização inicial do manuscrito e à revisão textual. Todas as decisões relacionadas ao conteúdo, à estrutura argumentativa, à seleção das informações relatadas e à redação final do artigo foram realizadas e revisadas pelos autores, que se responsabilizam integralmente pelo conteúdo submetido.

Referências

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2):197–227.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd KDD*.

Cunha, W., França, C., Fonseca, G., Rocha, L., and Gonçalves, M. A. (2023). An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In *the 46th ACM SIGIR*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ferrer, X., Nuenen, T. v., Such, J. M., Coté, M., and Criado, N. (2021). Bias and discrimination in ai: A cross-disciplinary perspective. *IEEE Technology and Society Magazine*.

Fonseca, G., Cunha, W., Prenassi, G., Gonçalves, M. A., and Da Rocha, L. C. D. (2025). Instance-selection-inspired undersampling strategies for bias reduction in small and large language models for binary text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9323–9340.

Fonseca, G., Cunha, W., and Rocha, L. (2024a). Análise comparativa de métodos de undersampling em classificaçao automática de texto baseada em transformers. *Revista Eletrônica de Iniciação Científica em Computação*, 22:1–10.

Fonseca, G., Prenassi, G., Cunha, W., Gonçalves, M. A., and Rocha, L. (2024b). Estratégias de undersampling para reduçao de viés em classificaçao de texto baseada em transformers. In *Brazilian Symposium on Multimedia and the Web (WebMedia)*, pages 144–152. SBC.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in NeurIPS*.

Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*. Citeseer.

LaValley, M. P. (2008). Logistic regression. *Circulation*, 117(18):2395–2399.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mani, I. and Zhang, I. (2003). knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*. ICML.

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., and Benevenuto, F. (2016). Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ DS*.

Tomek, I. (1976). Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*.

Zanotto, B. S., Beck da Silva Etges, A. P., Ruschel, R., Luiz, W., et al. (2021). Stroke outcome measurements from electronic medical records: cross-sectional study on the effectiveness of neural and nonneural classifiers. *JMIR Med. Infor.*