Detecção de Discurso de Ódio em Textos Utilizando Redes Neurais Profundas

1st João Luís da Silva Marrocos
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
jlsm2@cin.ufpe.br

2nd Thomaz Cabral Corrêa de Araújo
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
tcca@cin.ufpe.br

3rd Welton Pereira da Luz Felix
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
wplf@cin.ufpe.br

***Abstract*—Considerando a crescente disseminação de conteúdos prejudiciais nas redes sociais, o presente trabalho analisa a eficácia de modelos de aprendizagem profunda, como Redes Neurais Recorrentes (RNNs) e Transformers, para a identificação desse tipo de discurso. Utilizamos um conjunto de dados real contendo comentários de plataformas como YouTube, Twitter e Reddit, e avaliamos diferentes arquiteturas de redes neurais. Os resultados demonstram que os modelos propostos são capazes de capturar nuances linguísticas e contextuais, contribuindo para a melhoria da moderação de conteúdos online e promovendo um ambiente digital mais seguro.**

***Index Terms*—Detecção de discurso de ódio, Redes Neurais Profundas, Redes Neurais Recorrentes, Transformers, Análise de Texto, Moderação de Conteúdo, Aprendizagem Profunda.**

I. Introdução

A capacidade de disseminação de conteúdo gerado pelos próprios usuários em redes sociais permite que qualquer pessoa, mesmo sem conhecimento técnico, possa expressar suas opiniões e visão de mundo em larga escala. Ambientes como o Twitter, Instagram, YouTube, TikTok e Reddit impulsionam as publicações dos usuários de forma ampla, podendo alcançar qualquer usuário da rede. Embora alguns fóruns tornem suas plataformas intencionalmente direcionadas ao discurso de ódio, o alcance das redes sociais é muito maior e a informação se propaga quase instantaneamente [1]. Em uma pesquisa feita em 16 países pela UNESCO e pela Ipsos [2], 67% dos entrevistados já observaram discursos de ódio sendo propagado online.

Dessa forma, são necessários mecanismos para identificação e remoção desse tipo de conteúdo dessas plataformas. Segundo Mullah e Zainon [3], a maior parte dos trabalhos em detecção de discursos de ódio utilizam modelos clássicos de *machine learning* em vez de modelos de *ensemble learning* ou *deep learning*. Assim, existe uma lacuna no que diz respeito à aplicação de modelos de aprendizagem profunda nessa área.

A utilização de redes neurais profundas representa uma oportunidade promissora para avançar na detecção de discurso de ódio por causa de sua capacidade de aprender representações complexas e hierárquicas dos dados, sendo bastante útil na análise de texto, onde nuances linguísticas e contextuais desempenham um papel importante.

Além disso, com o aumento da disponibilidade de grandes volumes de dados textuais, tornou-se viável treinar modelos de aprendizagem profunda com maior precisão e eficiência [4], fornecendo uma base complexa para a criação de modelos robustos que conseguem capturar nuances e variabilidades da linguagem humana.

II. Objetivo

O objetivo deste projeto é desenvolver modelos eficientes para a detecção de discurso de ódio em textos utilizando Redes Neurais Recorrentes (RNNs) e Transformers. Com isso, buscamos discutir a complexidade referente à identificação de conteúdos prejudiciais na linguagem natural, explorando a capacidade dessas redes de capturar dependências temporais e contextuais em textos.

Esse projeto também tem como objetivo explorar e implementar modelos de redes neurais profundas para a detecção de discurso de ódio em textos, por meio de uma análise comparativa de diferentes arquiteturas e seus desempenhos em um conjunto de dados real. Ao implementar e treinar essas redes neurais de forma robusta, espera-se inspirar a criação de sistemas que possam moderar e filtrar discursos de ódio na internet, contribuindo para uma navegação online mais segura e respeitosa.

Consequentemente, este trabalho também contribui para a diminuição da presença desse problema nas redes sociais, incentivando a criação de políticas e estratégias de combate ao discurso de ódio na internet. Acredita-se que os resultados obtidos possam melhorar a precisão das ferramentas de moderação de conteúdo e fornecer *insights* valiosos para o desenvolvimento de abordagens mais eficazes no combate ao discurso de ódio online.

III. Justificativa

Com a quantidade massiva de usuários, é inevitável que surjam pessoas mal-intencionadas que espalhem o ódio pelas plataformas digitais. Tendo isso em vista, é de extrema necessidade

que haja uma maneira de moderar o conteúdo compartilhado, porém, com o aumento exponencial do uso da internet, tornou-se inviável que essa moderação aconteça de forma manual.

Assim, surge a necessidade da criação de modelos que identifiquem de forma eficiente a ocorrência de discurso de ódio. Para isso, as redes profundas são escolhidas devido a sua capacidade de processar dados sequenciais, capturando contextos que são cruciais para a identificação correta desse discurso.

IV. Metodologia

A análise de discurso de ódio nas redes sociais é uma área essencial de pesquisa devido ao impacto significativo que esse tipo de conteúdo tem na sociedade. Nesse projeto, abordamos esse problema utilizando um conjunto completo de dados que inclui milhares de comentários de usuários de plataformas populares. O *dataset* foi originalmente descrito em estudos anteriores e contém, além dos textos, diversas características que ajudam a identificar e classificar o conteúdo inadequado, fornecendo uma visão detalhada das diferentes dimensões do discurso de ódio.

Para a modelagem, abordamos uma abordagem baseada em redes neurais recorrentes e transformers, projetados para capturar dependências temporais e contextuais em sequências de texto. A implementação dos modelos foi realizada com base nos modelos já existentes e consolidados na biblioteca *Pytorch*. O treinamento dos modelos envolveu a utilização de diversos parâmetros e diferentes combinações deles a fim de obter a melhor parametrização possível e, consequentemente, o melhor desempenho possível. Além disso, a avaliação dos modelos utilizados é realizada por meio de um conjunto robusto de métricas, as quais permitem uma análise completa e precisa do desempenho dos modelos, garantindo que eles sejam capazes de identificar corretamente o discurso de ódio.

A. Dataset

O *dataset* escolhido para a análise contém dados de cerca de 39 mil comentários do YouTube, Twitter e Reddit e foi originalmente descrito nos trabalhos de Kennedy et al. [5] e Sachdeva et al. [6]. Além dos texto dos comentários e da "taxa de discurso ódio" (onde um valor maior significa um texto mais abusivo), o conjunto de dados possui diversas *features* que indicam quais as características do comentário e quais grupos sociais ele foca. Algumas delas são:

- sentimento;
- respeito;
- insulto;
- humilhação;
- status;
- desumanização;
- violência;
- genocídio;
- ataque/defesa;
- discurso de ódio.

TABLE I
Exemplo de entrada da base de dados [5]

hate_speech_score	text	annotator_severity
$4,63$	we need to nuke [...]	$-0,39$
$-4,09$	Many of the families [...]	$-0,4$

A ideia é que a avaliação de cada uma dessas categorias fosse feita de forma subjetiva, deixando a cargo do humano que avaliou o comentário a interpretação do que cada uma delas significa. Junto a isso, são indicadas informações anônimas acerca do responsável pela avaliação do comentário, como o seu viés de interpretação, renda, escolaridade e idade. As redes sociais também foram anonimizadas mas, assim como aos avaliadores, foi atribuído um identificador a cada uma delas.

Além dos dados fornecidos pelos avaliadores, foi feita uma análise estatística dos dados coletados, resultando na inclusão dessas *features*. Ao todo, cada uma das 135.556 linhas da base de dados contém 131 *features*. O quadro I mostra dois exemplos de linhas com 3 features.

B. Tratamento dos Dados

As etapas de tratamento dos dados textuais são essenciais para preparar o *dataset* para utilização em modelos profundos de detecção de discurso de ódio. Cada etapa desempenha um papel essencial na limpeza e transformação dos dados, garantindo a qualidade e eficácia dos modelos:

- Análise do conjunto de dados: valores ausentes (NaN) foram identificados e removidos do dataset, uma vez que a presença de deles pode interferir nas análises subsequentes e no desempenho dos modelos. Outliers, que são valores atípicos que podem distorcer as análises [7], também foram detectados e removidos,ajudando a manter a integridade dos dados e evitando influências indevidas no treinamento do modelo;
- Remoção de Emoji e Caracteres Especiais: emojis e caracteres especiais foram removidos dos textos, já que esses elementos não contribuem significativamente para a análise semântica e podem introduzir ruído nos dados, prejudicando a performance dos modelos;
- Remoção de *stopwords*: *Stopwords*, palavras comuns que não carregam significado significativo (como "*the*", "*of*", "*a*", em inglês) [8], foram removidas dos textos, reduzindo a dimensionalidade dos dados e melhorando a eficiência do modelo ao focar em palavras mais relevantes;
- Lematização: a lematização foi aplicada para reduzir as palavras às suas formas base, normalizando os textos, agrupando diferentes formas de uma palavra (por exemplo, "*getting*" e "*got*" para "*get*"), o que ajuda a melhorar a consistência dos dados e a eficácia do modelo;
- Tokenização: os textos foram tokenizados, ou seja, divididos em unidades menores, como palavras ou subpalavras, sendo um passo fundamental que facilita a análise e

processamento dos textos pelos modelos de aprendizado profundo;

- *One-hot Encoding*: Aplicou-se a técnica de *one-hot encoding* para converter as palavras tokenizadas em vetores binários. Essa representação é crucial para que os modelos de aprendizado profundo possam interpretar e processar os dados textuais de forma eficiente.

Essas etapas de tratamento dos dados textuais são fundamentais para garantir que os modelos de detecção de discurso de ódio sejam treinados com dados de alta qualidade, livres de ruídos e inconsistências, contribuindo para a precisão e robustez dos modelos finais.

C. Redes Neurais Recorrentes

Redes Neurais Recorrentes (RNNs) são um tipo de rede neural eficaz para processamento de dados sequenciais, como textos e séries temporais. Elas são capazes de capturar dependências temporais ao manter uma memória de estados anteriores [9], tornando-as ideais para tarefas onde o contexto anterior é crucial para a interpretação atual dos dados, como a detecção do discurso de ódio.

Uma das limitações das RNNs tradicionais é o problema do desvanecimento do gradiente (vanishing gradient) [10], que dificulta o aprendizado de dependências de longo prazo. Para resolver esse problema, foram desenvolvidas variantes como a Gated Recurrent Unit (GRU) e a Long Short-Term Memory (LSTM), as quais são abordadas, discutidas e implementadas neste trabalho.

1) Gated Recurrent Unit: GRU é uma Rede Neural Recorrente (RNN) desenvolvida para resolver problemas de dependências de longo prazo e a questão do desvanecimento do gradiente (*vanishing gradient*), que são comuns em RNNs mais tradicionais. Ela é semelhante à Long Short-Term Memory (LSTM), mas possui uma estrutura mais simples e menos parâmetros [11].

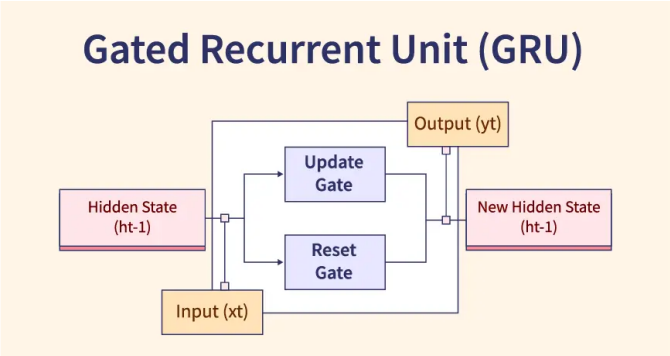

Fig. 1. Estrutura da GRU

A arquitetura de GRUs inclui duas portas principais: a porta de atualização e a porta de reinicialização, as quais permitem com que a rede controle o fluxo de informações de modo eficiente [12], preservando, assim, informações relevantes de longo prazo e descartando as informações irrelevantes. Isso ocorre devido à presença do *hidden state*, o qual armazena essas informações e carregam elas ao longo de todas as etapas da rede, o que pode ser observado na Figura 1. Isso faz com que esse modelo capture dependências temporais e contextuais importantes para a tarefa de detecção de ódio.

A escolha de GRUs é justificada pela sua eficiência computacional [13] e pela sua habilidade de lidar com dependências de longo prazo em sequências de texto, fazendo delas uma ferramenta importante para a tarefa de detecção de ódio.

2) Bidirectional Long Short-Term Memory: As redes Long Short-Term Memory (LSTMs) também foram desenvolvidas para superar as limitações das RNNs tradicionais, como o *vanishing gradient* e *exploding gradient*, permitindo a captura de dependências de longo prazo em sequências de dados. Entretanto, ela possui uma variação chamada de Bidirectional LSTM (BiLSTM), a qual estende essa capacidade de processamento da sequência de entrada em ambas as direções: da primeira para a última palavra e da última para a primeira [14], permitindo com que o modelo capture contextos passados e futuros concomitantemente e, consequentemente, oferecendo uma compreensão mais completa do texto.

A arquitetura de um bloco LSTM é composta por células de memória, onde cada célula é responsável por manter informações relevantes ao longo do tempo, controlando o fluxo de informações através de três portas principais: a porta de entrada, a porta de esquecimento e a porta de saída.

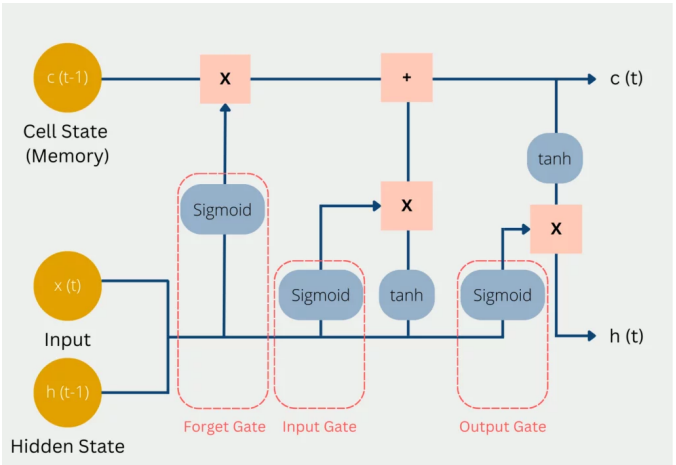

Fig. 2. Blocos LSTM

Na porta de entrada, uma função sigmoide decide quais valores serão atualizados, enquanto uma função tanh cria um vetor de novos valores candidatos para serem adicionados à célula de memória. Já a porta de esquecimento utiliza uma função sigmoide para determinar quais informações da célula de memória anterior serão descartadas. Por fim, a porta de saída, também através de uma função sigmoide decide quais informações da célula de memória serão enviadas

para a próxima unidade LSTM e para a saída final [15]. A utilização da *cell state* e da *hidden state* objetivam armazenas informações à longo e curto prazo, respectivamente. Esse esquema pode ser visto na Figura 2.

Em uma BiLSTM, a sequência de entrada é processada por duas camadas LSTM distintas: a camada *forward* e a camada *backward*, podendo ser visualizada na Figura 3. A camada *forward* processa essa sequência na direção natural, capturando as independências e informações contextuais que aparecem cronologicamente. Simultaneamente, a camada *backward* processa a mesma sequência de dados, mas na direção oposta, capturando contextos quando os dados são analisados em retrospectiva. Após esses processos, as saídas são combinadas, geralmente por concatenação, para formar uma representação completa em cada ponto da sequência [16].

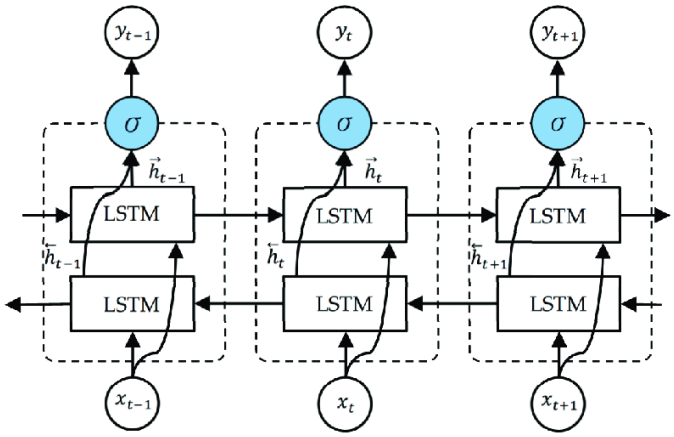

Fig. 3. Estrutura da BiLSTM

Para esse trabalho, a BiLSTM será útil na identificação de discurso de ódio ao capturar dependências complexas e contextos bidirecionais nos textos, melhorando a robustez do modelo na detecção de conteúdo prejudicial.

D. Transformers

Os Transformers revolucionaram o campo do Processamento de Linguagem Natural (PLN) ao oferecer uma maneira eficiente, escalar e diferenciada de modelar dependências de longo alcance em dados sequenciais. Diferentemente das Redes Neurais Recorrentes, os Transformers não processam dados de maneira sequencial, o que permite um paralelismo [17] importante para o treinamento.

A arquitetura dos Transformers é formada por camadas de atenção e *feed-forward*. A principal inovação dos Transformers é o mecanismo de atenção, especificamente a atenção multi-cabeça, a qual permite com que o modelo foque em diferentes partes da sequência de entrada simultaneamente. Cada camada de atenção é seguida por uma camada *feed-forward* totalmente conectada [18]. No codificador, a sequência de entrada é processada por várias dessas camadas de atenção e *feed-forward*, enquanto que no decodificador, a sequência de saída é gerada de forma auto-regressiva, uma palavra por vez, usando

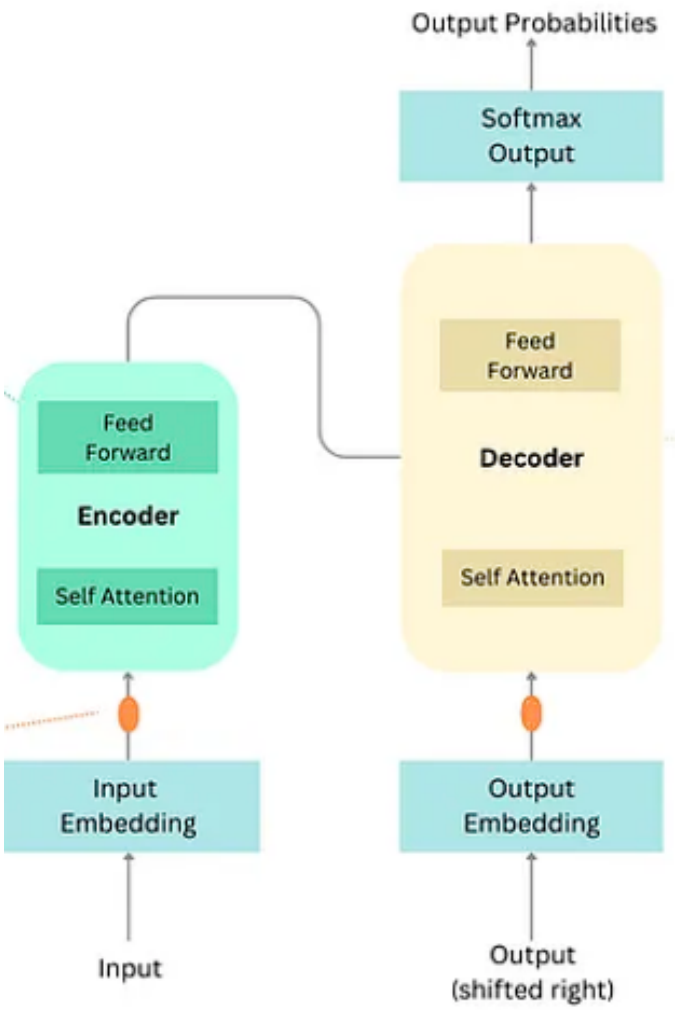

Fig. 4. Estrutura geral dos Transformers

informações do codificador através de mecanismos de atenção cruzada. Além disso, o uso de *embeddings* posicionais permite ao modelo incorporar a ordem das palavras na sequência, compensando a falta de processamento sequencial natural das Redes Neurais Recorrentes [19]. Isso pode ser facilmente visualizado na Figura 4.

A importância dos Transformers para a detecção de discurso de ódio reside em sua capacidade de capturar relações complexas e contextos longos em textos, sendo crucial para identificar nuances e implicitudes que frequentemente caracterizam discursos de ódio.

E. Parâmetros

Para a realização dos experimentos, vários parâmetros foram testados, incluindo diferentes otimizadores (Adam, SGD e RMSprop), taxas de aprendizado (*learning rate*), número de camadas ocultas (*hidden layers*), taxa de *dropout* e número de cabeças (*num heads*) nos modelos transformer.

Testar uma variedade de parâmetros é importante para identificar a combinação que resulta no melhor desempenho do modelo, permitindo ajustá-lo a fim de maximizar todas as métricas que analisam o desempenho. A escolha adequada de parâmetros também reduz o tempo de treinamento e melhora a capacidade de generalização do modelo. A seguir, apresentamos a definição de cada um desses parâmetros.

1) Otimizador: Os otimizadores são algoritmos usados para ajustar os pesos da rede neural com o objetivo de minimizar a função de perda. Três otimizadores foram testados: Adam, SGD e RMSprop. O otimizador Adam combina as vantagens de dois outros métodos de otimização: AdaGrad e RMSProp, proporcionando uma convergência rápida e eficiente. O SGD (*Stochastic Gradient Descent*) é bastante utilizado por sua simplicidade e eficiência, atualizando os pesos da rede com base em um subconjunto aleatório dos dados de treinamento. O RMSprop, por sua vez, ajusta a taxa de aprendizado para cada parâmetro, dividindo a taxa de aprendizado pelo valor médio dos gradientes recentes, ajudando a manter a estabilidade do treinamento [20].

2) Learning Rate: A taxa de aprendizado é um hiperparâmetro que controla o quanto os pesos da rede neural são ajustados com base no gradiente estimado. Valores muito altos podem fazer com que o modelo não converja, enquanto que valores muito baixos podem resultar em um treinamento muito lento e possivelmente em um mínimo local ruim [21]. Testar diferentes taxas de aprendizado ajuda a encontrar um equilíbrio entre velocidade e precisão de convergência.

3) Número de Camadas Ocultas: O número de camadas ocultas consiste na quantidade de camadas entre a camada de entrada e a camada de saída em uma rede neural. Mais camadas ocultas podem permitir que a rede capture padrões mais complexos nos dados, mas também podem aumentar o risco de *overfitting* e aumentar o tempo de treinamento [22]. Testar diferentes números de camadas ocultas ajuda a determinar a arquitetura ideal da rede.

4) Dropout: A taxa de *dropout* é uma técnica de regularização utilizada para prevenir *overfitting* em redes neurais, principalmente nos transformers. Durante o treinamento, unidades da rede são desligadas aleatoriamente com uma determinada probabilidade, impedindo que o modelo se torne excessivamente dependente de neurônios específicos, forçando a rede a aprender representações mais robustas e generalizáveis dos dados [23].

5) Número de Cabeças: O número de cabeças é um parâmetro específico para os transformers, que define quantas sub-representações independentes serão aprendidas em cada camada de atenção multi-cabeça. Mais cabeças permitem que o modelo se concentre em diferentes partes da entrada ao mesmo tempo, melhorando a capacidade da rede de capturar relações complexas entre os tokens de entrada [24].

F. Métricas

Com o intuito de avaliar a eficácia dos modelos de detecção de discurso de ódio em textos, é imprescindível utilizar um conjunto abrangente de métricas que ofereçam uma visão completa do desempenho obtido. Elas foram escolhidas com o objetivo de levar em consideração diversos aspectos dos modelos, como a capacidade do modelo de identificar corretamente as classes, minimizar a quantidade de falsos positivos e falsos negativos e manter um equilíbrio entre precisão e *recall*.

Dessa forma, as métricas selecionadas permitem uma avaliação robusta e confiável dos modelos, proporcionando uma compreensão ampla de seus potenciais e suas limitações em diversos cenários de uso. A seguir, as principais métricas utilizadas são detalhadas:

1) Acurácia: A acurácia mede a proporção de previsões corretas entre o total de previsões realizadas [25]. Em outros termos, é a fração das previsões que o modelo classificou corretamente, incluindo tanto as instâncias positivas quanto as negativas.

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN} \tag{1}$$

2) Precisão: A precisão avalia a fração de instâncias positivas classificadas corretamente em relação a todas as instâncias positivas [26], sejam elas verdadeiras ou falsas. Ou seja, mensura a capacidade do modelo específico de classificar corretamente as ocorrências positivas.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2}$$

3) *Recall*: Também conhecida como revocação ou sensibilidade, *recall* mede a proporção de instâncias positivas que foram classificadas corretamente em relação ao conjunto total de ocorrências positivas [27].

$$\text{Recall} = \frac{VP}{VP + FN} \tag{3}$$

4) F1-Score: F1-score consiste da média harmônica entre a precisão e o *recall*, significando que ele penaliza qualquer valor extremo encontrado, fornecendo, assim, uma medida balanceada do desempenho do modelo [28].

$$\text{F1-score} = \frac{2 \cdot (\text{precisão} \cdot \text{recall})}{\text{precisão} + \text{recall}} \tag{4}$$

$$\text{F1-score} = \frac{2 \cdot VP}{2 \cdot VP + FP + FN} \tag{5}$$

5) AUC-ROC: A área sobre curva ROC é uma métrica que avalia a capacidade do modelo em distinguir entre as classes positivas e negativas [29]. Não há uma fórmula simples e útil para calculá-la, uma vez que ela é calculada numericamente integrando a curva ROC. Entretanto, ela utiliza os parâmetros *True Positive Rate* (TPR) e *False Negative Rate* (FNR):

$$\text{TPR} = \frac{VP}{VP + FN} \tag{6}$$

$$\text{FNR} = \frac{FP}{FP + VN} \tag{7}$$

V. Resultados

Os resultados obtidos ao longo deste trabalho destacam a eficácia dos modelos de redes neurais profundas na detecção de discurso de ódio em textos. A análise detalhada dos dados coletados, combinada com a implementação e treinamento robustos dos modelos, permitiu alcançar um desempenho significativo na identificação de conteúdo prejudicial. Nesta seção, exploramos os resultados dos modelos de Redes Neurais Recorrentes (RNNs) e Transformers. As métricas de desempenho abordadas são discutidas para ilustrar a eficiência de cada modelo. Além disso, são apresentadas as implicações desses resultados para a moderação de conteúdo nas redes sociais e as possíveis melhorias para futuras pesquisas na área.

A. Gated Recurrent Unit

Primeiramente, é essencial analisar o desempenho dos modelos de Gated Recurrent Unit (GRU) treinados com base nos otimizadores para a detecção do discurso de ódio. O gráfico da Figura 5 apresenta a relação entre a acurácia e o F1-Score dos modelos. Essas duas métricas foram escolhidas para compor os gráficos da seção, pois esta envolve a precisão e o *recall*, outras métricas importantes utilizadas para a análise dos modelos, enquanto que aquela representa o desempenho geral dos modelos.

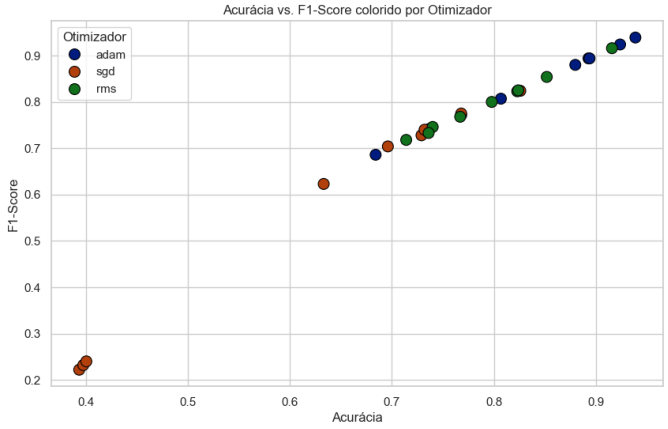

Fig. 5. Acurácia vs F1-Score por Otimizador

Analisando-o, é possível observar que os modelos otimizados com Adam geralmente apresentam melhores resultados, com acurácia e F1-Score mais elevados em comparação com SGD e RMSprop. A maioria dos pontos relacionados ao Adam está concentrada na parte inferior do gráfico, indicando um bom desempenho. Por outro lado, os modelos otimizados com SGD (*Stochastic Gradient Descent*) mostram maior variabilidade, com muitos pontos distribuídos na parte inferior do gráfico, sugerindo um desempenho menos consistente. Os modelos com RMSprop situam-se entre os outros, apresentando desempenho intermediário.

Através dessa análise de otimizadores, é possível concluir que Adam é o otimizador mais eficiente para os modelos GRU no contexto da detecção do discurso de ódio. Sua capacidade de ajustar os pesos de forma mais eficaz resulta em uma melhor performance global.

Além disso, também é possível realizar uma análise baseada nas taxas de aprendizado diferentes utilizadas nos modelos GRUs treinados, o que pode ser facilmente visualizado no gráfico da Figura 6.

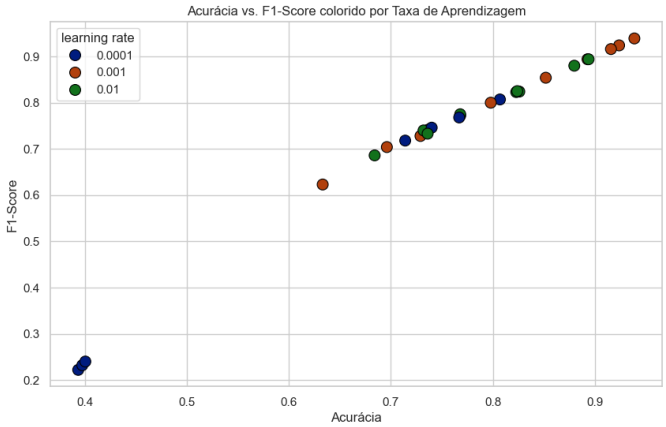

Fig. 6. Acurácia vs F1-Score por taxas de aprendizado (GRU)

Os resultados mostram que a taxa de aprendizado de 0,001 tende a ser a mais eficaz, resultando em modelos com métricas elevadas e consistentes. A taxa de 0,0001, embora tenha produzido alguns modelos de bom desempenho, apresenta maior variabilidade e alguns modelos com desempenho inferior, além de que, já que esse número é muito baixo, os modelos com essa *learning rate* demoram muito para serem treinados e, perceptivelmente, não é um bom investimento. A taxa de 0,001 também representou resultados similares, com alguns modelos com bom desempenho, mas também possuindo alta instabilidade, com vários modelos apresentando desempenho bem abaixo do esperado.

Isso indica que a taxa de aprendizagem de 0,001 é a mais adequada para o desenvolvimento de GRUs para a detecção do discurso de ódio, proporcionando um equilíbrio ideal entre a estabilidade do treinamento e a capacidade do modelo de aprender padrões complexos presentes nos dados.

Experimentos testando diferentes números de *hidden layers* também foram realizados, contudo nenhuma conclusão pode ser afirmada devido à enorme variabilidade de resultados quando esse fator foi levado em consideração, fazendo com que ele não seja tão crucial quanto os outros parâmetros para um bom desempenho do modelo.

Os resultados apresentados indicam que tanto a escolha do otimizador quanto a taxa de aprendizagem são fatores importantes que afetam significativamente o desempenho dos modelos de GRU treinados para a detecção do discurso de ódio. A combinação do otimizador Adam com uma *learning rate* de 0,001 resultou nos melhores desempenhos, fazendo com que o modelo treinado com base nessa configuração capture de forma eficaz os padrões complexos nos dados utilizados para a detecção do discurso de ódio.

B. Bidirectional Long Short-Term Memory

Primeiramente, é essencial analisar o desempenho dos modelos de Bidirectional Long Short-Term Memory (BiLSTM) treinados com base nos otimizadores para a detecção do discurso de ódio. O gráfico da Figura 7 apresenta a relação entre a acurácia e o F1-Score dos modelos. Essas duas métricas foram escolhidas para compor os gráficos da seção, pois esta envolve a precisão e o *recall*, outras métricas importantes utilizadas para a análise dos modelos, enquanto que aquela representa o desempenho geral dos modelos.

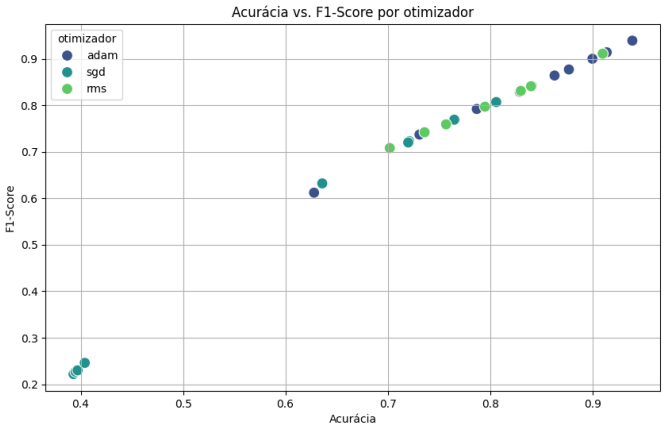

Fig. 7. Acurácia vs F1-Score por Otimizador (GRU)

Analisando-o, é possível observar que os modelos otimizados com Adam geralmente apresentam melhores resultados, com acurácia e F1-Score mais elevados em comparação com SGD e RMSprop. A maioria dos pontos relacionados ao Adam está concentrada na parte superior do gráfico, indicando um bom desempenho. Por outro lado, os modelos otimizados com SGD (*Stochastic Gradient Descent*) mostram maior variabilidade, com muitos pontos distribuídos na parte inferior do gráfico, sugerindo um desempenho menos consistente. Os modelos com RMSprop situam-se entre os outros, apresentando desempenho intermediário.

Através dessa análise de otimizadores, é possível concluir que Adam é o otimizador mais eficiente para os modelos BiLSTM no contexto da detecção do discurso de ódio. Sua capacidade de ajustar os pesos de forma mais eficaz resulta em uma melhor performance global.

Além disso, também é possível realizar uma análise baseada nas taxas de aprendizado diferentes utilizadas nos modelos BiLSTM treinados, o que pode ser facilmente visualizado no gráfico da Figura 6.

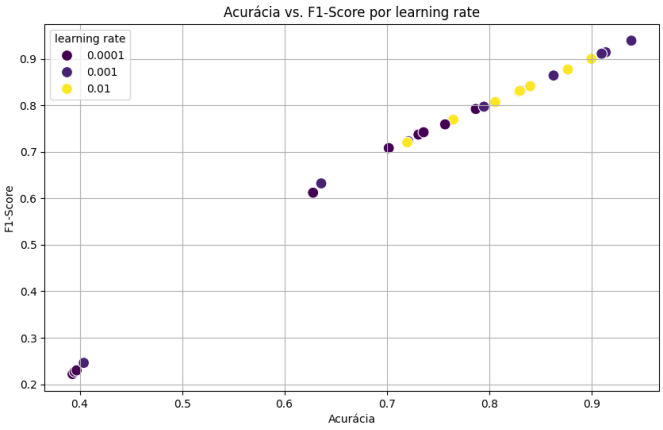

Fig. 8. Acurácia vs F1-Score por taxas de aprendizado (BiLSTM)

Os resultados mostram que a taxa de aprendizado de 0,001 tende a ser a mais eficaz, resultando em modelos com métricas elevadas e consistentes. A taxa de 0,0001, embora tenha produzido alguns modelos de bom desempenho, apresenta maior variabilidade e alguns modelos com desempenho inferior, além de que, já que esse número é muito baixo, os modelos com essa *learning rate* demoram muito para serem treinados e, perceptivelmente, não é um bom investimento. A taxa de 0,001 também representou resultados similares, com alguns modelos com bom desempenho, mas também possuindo alta instabilidade, com vários modelos apresentando desempenho bem abaixo do esperado.

Isso indica que a taxa de aprendizagem de 0,001 é a mais adequada para o desenvolvimento de BiLSTMs para a detecção do discurso de ódio, proporcionando um equilíbrio ideal entre a estabilidade do treinamento e a capacidade do modelo de aprender padrões complexos presentes nos dados.

Experimentos testando diferentes números de *hidden layers* também foram realizados, contudo nenhuma conclusão pode ser afirmada devido à enorme variabilidade de resultados quando esse fator foi levado em consideração, fazendo com que ele não seja tão crucial quanto os outros parâmetros para um bom desempenho do modelo.

Os resultados apresentados indicam que tanto a escolha do otimizador quanto a taxa de aprendizagem são fatores importantes que afetam significativamente o desempenho dos modelos de BiLSTM treinados para a detecção do discurso de ódio. A combinação do otimizador Adam com uma *learning*

rate de 0,001 resultou nos melhores desempenhos, fazendo com que o modelo treinado com base nessa configuração capture de forma eficaz os padrões complexos nos dados utilizados para a detecção do discurso de ódio.

C. Transformers

No caso dos modelos treinados utilizando a arquitetura Transformer, os resultados obtidos foram próximos aos das outras arquiteturas, com o otimizador Adam e a taxa de aprendizado de 0,001 apresentando os melhores resultados. A Figura 9 mostra a relação entre a acurácia e o F1-Score dos modelos treinados com base nos otimizadores.

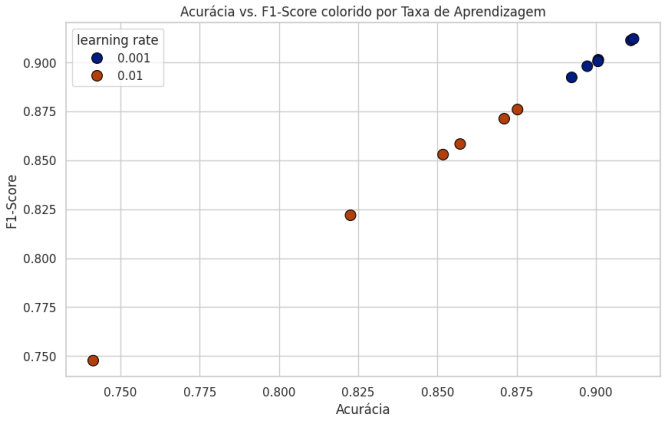

Fig. 9. Acurácia vs F1-Score por taxas de aprendizado (Transformer)

A taxa de aprendizado 0,01 apresenta resultados mais variáveis, com alguns modelos obtendo desempenho superior e outros inferior. A taxa de 0,001, por outro lado, resulta em modelos mais consistentes, com acurácia e F1-Score mais elevados. A taxa de 0,0001, embora tenha produzido alguns modelos com bom desempenho, apresenta maior variabilidade e alguns modelos com desempenho inferior.

Em relação ao número de cabeças, os resultados indicam que o número ideal é 8, com os modelos treinados com essa configuração apresentando os melhores desempenhos. O gráfico da Figura 10 mostra a relação entre a acurácia e o F1-Score dos modelos treinados com diferentes números de cabeças. O aumento do número de cabeças resulta no aumento da complexidade do modelo, o que, provavelmente devido ao conjunto de dados não ser tão grande, leva a um desempenho inferior.

Assim como nos modelos GRU e BiLSTM, os resultados dos experimentos com diferentes números de camadas ocultas não foram conclusivos.

Os resultados obtidos com os modelos Transformer confirmam a importância da escolha do otimizador e da taxa de aprendizado para o desempenho dos modelos, juntamente com o número de cabeças. A combinação do otimizador Adam com uma taxa de aprendizado de 0,001 e 8 cabeças resultou nos melhores desempenhos, destacando a eficácia dos Transformers na detecção do discurso de ódio. Porém, é importante

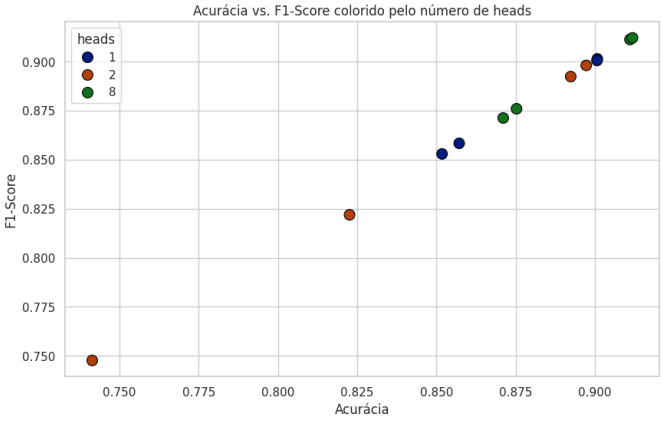

Fig. 10. Acurácia vs F1-Score por número de cabeças (Transformer)

ressaltar que, devido à complexidade dos Transformers, o tempo de treinamento desses modelos é significativamente maior em comparação com os modelos GRU e BiLSTM. Assim, por apresentarem resultados semelhantes (até mesmo um pouco piores), talvez os transformers não sejam a melhor escolha para a detecção de discurso de ódio em um conjunto de dados pequeno.

D. Discussão

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum tempor mauris, eu tempor quam. Vestibulum hendrerit scelerisque ante, non maximus dui ullamcorper vitae. Aliquam id lacus augue. Nullam sed convallis mauris. In convallis eros ut efficitur commodo. Mauris mollis odio sem, id finibus lectus maximus viverra. Nunc euismod imperdiet diam, nec gravida eros tincidunt non. In a velit sed ante tristique accumsan. Praesent gravida turpis vel velit lobortis rutrum. Praesent vitae ipsum at ante volutpat iaculis sit amet id purus. Suspendisse neque arcu, vehicula quis nulla non, rhoncus mattis velit. Mauris sit amet eros ut est elementum ullamcorper id ac erat. Vestibulum sodales quam velit, vitae fringilla metus cursus sagittis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras elementum tempor mauris, eu tempor quam. Vestibulum hendrerit scelerisque ante, non maximus dui ullamcorper vitae. Aliquam id lacus augue. Nullam sed convallis mauris. In convallis eros ut efficitur commodo. Mauris mollis odio sem, id finibus lectus maximus viverra. Nunc euismod imperdiet diam, nec gravida eros tincidunt non. In a velit sed ante tristique accumsan. Praesent gravida turpis vel velit lobortis rutrum. Praesent vitae ipsum at ante volutpat iaculis sit amet id purus. Suspendisse neque arcu, vehicula quis nulla non, rhoncus mattis velit. Mauris sit amet eros ut est elementum ullamcorper id ac erat. Vestibulum sodales quam velit, vitae fringilla metus cursus sagittis.

REFERENCES

[1] A. Guiora and E. A. Park, "Hate speech on social media," *Philosophia*, vol. 45, no. 3, pp. 957–971, Sep 2017. [Online]. Available:

https://doi.org/10.1007/s11406-017-9858-4

[2] S. Quétier-Parent, D. Lamotte, and M. Gallard, "Elections & social media: the battle against disinformation and trust issues," Nov. 2023. [Online]. Available: https://www.ipsos.com/en/elections-social-media-battle-against-disinformation-and-trust-issues

[3] N. S. Mullah and W. M. N. W. Zainon, "Advances in machine learning algorithms for hate speech detection in social media: A review," *IEEE Access*, vol. 9, pp. 88 364–88 376, 2021.

[4] A. Alkhudhayr, "Deep learning in the era of big data: Foundations, advances, applications, challenges, and future directions," *International Journal of Advanced Research*, vol. 12, pp. 549–552, 04 2024.

[5] C. J. Kennedy, G. Bacon, A. Sahn, and C. von Vacano, "Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application," 2020.

[6] P. Sachdeva, R. Barreto, G. Bacon, A. Sahn, C. von Vacano, and C. Kennedy, "The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism," in *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, G. Abercrombie, V. Basile, S. Tonelli, V. Rieser, and A. Uma, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 83–94. [Online]. Available: https://aclanthology.org/2022.nlperspectives-1.11

[7] A. H. Syed, "Dealing with outliers in data science: Techniques and best practices," Apr. 2023. [Online]. Available: https://syedabis98.medium.com/dealing-with-outliers-in-data-science-techniques-and-best-practices-a08172643b7a

[8] C. U. Press, "Dropping common terms: stop word," Jan. 2009. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html

[9] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 1, pp. 10–15, 07 2023.

[10] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, pp. 107–116, 04 2020.

[11] T.-T.-H. Le, J. Kim, and H. Kim, "Classification performance using gated recurrent unit recurrent neural network on energy disaggregation," 07 2016, pp. 105–110.

[12] Anishnama, "Understanding gated recurrent unit (gru) in deep learning," May 2023. [Online]. Available: https://medium.com/@anishnama20/understanding-gated-recurrent-unit-gru-in-deep-learning-2e54923f3e2

[13] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 32 Avenue of the Americas, New York, NY 10013-2473, USA: Cambridge university Press, 2023.

[14] A. Taparia, "Bidirectional lstm in nlp," Jun. 2023. [Online]. Available: https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/

[15] S. Arifin, A. WIJAYA, R. Nariswari, A. Yudistira, S. ., F. Faisal, and D. Wihardini, "Long short-term memory (lstm): Trends and future research potential," *International Journal of Emerging Technology and Advanced Engineering*, vol. 13, pp. 24–35, 05 2023.

[16] T. Mwata-Velu, J. G. Avina-Cervantes, J. M. Cruz-Duarte, H. Rostro-Gonzalez, and J. Ruiz-Pinales, "Imaginary finger movements decoding using empirical mode decomposition and a stacked bilstm architecture," *Mathematics*, vol. 1, no. 1, pp. 8–10, Dec 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/24/3297

[17] S. Ganesh, "Model parallelism using transformers and pytorch," Jan. 2021. [Online]. Available: https://medium.com/msakthiganesh/model-parallelism-using-transformers-and-pytorch-e751cc3e2303

[18] N. Nageswaran, "Transformers unleashed: A comprehensive study of transformer architectures and their applications," 08 2024.

[19] G. Giacaglia, "How transformers work," Mar. 2019. [Online]. Available: https://towardsdatascience.com/transformers-141e32e69591

[20] A. Gupta, "A comprehensive guide on optimizers in deep learning," Jan. 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/

[21] C. Igiri, A. Uzoma, and A. Silas, "Effect of learning rate on artificial neural network in machine learning," *International Journal of Engineering Research*, vol. 4, 06 2021.

[22] F. Arifin, H. Robbani, T. Annisa, and M. Ma'arof, "Variations in the number of layers and the number of neurons in artificial neural networks: Case study of pattern recognition," *Journal of Physics: Conference Series*, vol. 1413, p. 012016, 11 2019.

[23] B. Jabir and F. Noureddine, "Dropout, a basic and effective regularization method for a deep learning model: A case study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, p. 1009, 11 2021.

[24] S. Mahdavi, R. Liao, and C. Thrampoulidis, "Memorization capacity of multi-head attention in transformers," 06 2023.

[25] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning From Theory to Algorithms*. 32 Avenue of the Americas, New York, NY 10013-2473, USA: Cambridge university Press, 2014.

[26] I. da Silva Cardoso, "Técnicas de otimização e métricas de avaliação aplicadas a machine learning," *IF Goiano*, vol. 1, no. 1, pp. 35–37, Aug 2022. [Online]. Available: https://repositorio.ifgoiano.edu.br/handle/prefix/2712

[27] Google, "Classification: Precision and recall," Jul. 2022. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

[28] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Scientific Reports*, vol. 12, no. 1, p. 5979, Apr 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-09954-8

[29] V. Rodrigues, "Entenda o que é auc e roc nos modelos de machine learning," Oct. 2018. [Online]. Available: https://medium.com/bio-data-blog/entenda-o-que-%C3%A9-auc-e-roc-nos-modelos-de-machine-learning-8191fb4df772