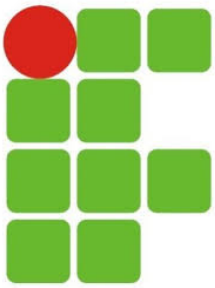


Instituto Federal de Alagoas
Campus Maceió
Coordenação de Informática
Curso Superior de Bacharelado em Sistemas de Informação

Investigação do Impacto dos Algoritmos de Aprendizado de Máquina na Detecção de Sentimentos em Avaliações de Produtos

Dielson Sales de Carvalho
Stewart Evangelista Gonçalves

Orientado por:
Prof. Dr. Leonardo Melo de Medeiros

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação.

Maceió, AL, outubro de 2023

Agradecimentos

Expresso minha profunda gratidão à minha mãe Eugênia, que sempre foi a base sólida da minha vida. Obrigado por todo o suporte e pelo amor infinito.

Agradeço também ao meu pai Luiz, que desde cedo me direcionou para a importância da educação e me incentivou a seguir o caminho acadêmico. Suas orientações e conselhos foram cruciais para meu desenvolvimento.

À minha querida namorada Sandy, que foi minha inspiração constante. Sua compreensão, paciência e amor me deram a motivação extra que eu precisava nos momentos mais desafiadores.

Ao meu orientador Dr. Leonardo Medeiros, todos os meus professores, colegas de classe, amigos e familiares que estiveram presentes ao longo desta jornada, o meu mais sincero agradecimento. Cada um de vocês contribuiram de maneira única para o meu crescimento e aprendizado.

Este TCC não é apenas uma conquista pessoal, mas também o resultado do apoio e encorajamento de todas as pessoas especiais que mencionei.

Stewart

À minha querida família, composta por minha mãe Cirlene e meu irmão Nilson, obrigado por sempre terem me apoiado e impulsionado nessa longa jornada, incentivando-me a tentar o meu melhor.

Ao meu pai Neilson, por sempre ter investido na minha educação acima de tudo.

À minha avó Luzanira, que sempre foi a pessoa mais feliz que conheci. Embora nunca tenha buscado me ensinar, tornou-se uma das maiores professoras.

Aos amigos que conquistei ao longo da minha trajetória acadêmica e profissional, muitos dos quais fizeram parte de ambos, obrigado pelo incentivo e inspiração constantes.

A todos os professores do IFAL que, apesar dos obstáculos burocráticos ocasionais, sempre pensaram na contribuição que deixariam em cada aluno, em especial meu orientador Dr. Leonardo Medeiros por acreditar no meu potencial.

Dielson

"Não tenho vergonha de mudar de ideia,
porque não tenho vergonha de pensar.
– Blaise Pascal

Resumo

A Internet revolucionou as interações humanas, servindo como uma plataforma para a livre troca de ideias e experiências. Em particular no comércio eletrônico, avaliações *online* podem ser consideradas uma fonte valiosa para diversos tipos de análises. O crescente volume de dados disponíveis abre caminho para o uso de modelos de Processamento de Linguagem Natural (PLN) na extração de *insights* automáticos a partir das avaliações de produtos vendidos *online*. Neste trabalho de conclusão de curso, fizemos uma pesquisa para selecionar os algoritmos de PLN mais empregados na Análise de Sentimentos nos últimos anos e comparamos seu desempenho com o BERT, um modelo de aprendizagem profundo publicado em 2018. Utilizamos um conjunto de dados com mais de 30 mil avaliações de produtos e testamos os algoritmos selecionados em diferentes situações. Os resultados obtidos evidenciam que o BERT apresenta um desempenho superior em comparação com os algoritmos convencionalmente empregados na análise de sentimentos, alcançando uma acurácia de 96,10% em comparação com 90,97% obtido pelo SVM no melhor caso. Esse mesmo destaque é notório quando se lida com uma quantidade limitada de texto para o treinamento, pois o BERT, mesmo com uma quantidade muito pequena de dados, consegue uma acurácia de 95,00% em comparação com 90,17% com o SVM no melhor caso. Embora seja importante observar que o BERT requer um custo computacional substancialmente mais elevado, este estudo o posiciona como uma alternativa viável em cenários onde a precisão é de extrema importância, independentemente dos custos. A pesquisa também contribui para a compreensão das aplicações práticas do PLN no contexto das avaliações de produtos em português na *web*, abrindo portas para melhorias em sistemas de análise de opiniões de consumidores.

Palavras-chave: análise de sentimentos, BERT, PLN, processamento de linguagem natural, *transformers*, redes neurais, aprendizagem profunda.

Abstract

The Internet has revolutionized human relationships, serving as a platform for the free exchange of ideas and experiences. Especially in e-commerce, online reviews can be considered a valuable source for different types of analysis. The growing volume of available data paves the way for the use of Natural Language Processing (NLP) models to extract automatic insights from online product reviews. In this study, we carried a research to select the NLP algorithms most used in Sentiment Analysis in recent years and compared their performance with BERT, a newer Deep Learning model published in 2018. We used a dataset with more than 30 thousand product reviews and tested the algorithms in different situations. The results obtained show that BERT presents superior performance compared to algorithms conventionally used in sentiment analysis, achieving an accuracy of 96.10% compared to 90.97% obtained by SVM in the best case. This same observation is noticeable when dealing with a limited amount of text for training, as BERT, even with a very small amount of data, achieves an accuracy of 95.00% compared to 90.17% with SVM in the best case. Although it is important to note that BERT requires a substantially higher computational cost, this study positions it as a viable alternative in scenarios where accuracy is of utmost importance, regardless of costs. The research also contributes to the understanding of the practical applications of PLN in the context of product reviews in Portuguese on the web, opening doors for improvements in consumer opinion analysis systems.

Keywords: sentiment analysis, BERT, NLP, natural language processing, transformers, neural networks, deep learning.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Esta pesquisa tem como objetivo realizar uma comparação sistemática entre o desempenho de algoritmos de aprendizado de máquina quando aplicados a avaliações de produtos na língua portuguesa a fim de identificar o sentimento expresso pelos usuários usando tanto os métodos convencionais de aprendizado de máquina quanto uma abordagem mais moderna, baseada em redes neurais.

Para isso, fizemos um levantamento dos algoritmos que têm sido tradicionalmente usados no campo de processamento de linguagem natural a partir da década de 1960 até a década de 2010. Chamamos de "métodos tradicionais" aqueles que são baseados em modelos estatísticos, como *Naïve Bayes* (NB), *K-Nearest Neighbor* (KNN) e *Support Vector Machine (SVM)* (LI et al., 2022) para compará-los com uma abordagem baseada em *Transformers*, uma vertente mais moderna e revolucionária do Processamento de Linguagem Natural (PLN). Os modelos baseados em *Transformers*, como o BERT (*Bidirectional Encoder Representations from Transformers*), são modelos de redes neurais que têm ganhado destaque recentemente por sua capacidade de capturar nuances contextuais e semânticas em texto, o que os torna particularmente promissores para a análise de sentimentos.

Essa comparação entre os modelos tradicionais e os modelos baseados em *Transformers*, no contexto das avaliações de produtos, é fundamental para avaliar se as abordagens mais modernas podem, de fato, aprimorar ou substituir eficazmente os métodos tradicionais na tarefa de análise de sentimentos na língua portuguesa.

1.1 Contextualização

A análise de sentimentos, também chamada de mineração de opiniões, tem sido um tópico de pesquisa bastante ativo nos últimos anos (HEMMATIAN; SOHRABI, 2019). De acordo com Pang e Lee (2008), podemos definir a análise de sentimento como o processo de identificação e extração de informações subjetivas de fontes não estruturadas, tais como opiniões, sentimentos e emoções expressos em textos. Essas informações podem ser úteis para diversas áreas, incluindo negócios, política e saúde.

No trabalho de Mubarok, Adiwijaya e Aldhi (2017) enfatizou-se a importância da extração de informações de análises de produtos, tanto para consumidores quanto para lojistas. Essa abordagem visa entender as reações do mercado e tomar as ações necessárias com base nas informações extraídas. No entanto, surgem problemas porque as avaliações podem conter informações incompletas, tendenciosas e diversificadas. A Internet transformou significativamente a maneira como as pessoas interagem, servindo como uma plataforma para a troca de ideias e experiências. No setor de *e-commerce*, cada

avaliação publicada online reflete o sentimento e a experiência únicos de cada pessoa. Esses sentimentos podem ser categorizados em três vias: favoráveis, desfavoráveis ou neutros. Caso uma pessoa tenha uma experiência boa com determinado produto e decida compartilhar uma opinião favorável publicamente, a tendência é que outros consumidores criem boas expectativas com base nessa e outras avaliações. Assim, cada opinião compartilhada contribui para a formação de uma percepção geral sobre o produto, reforçando a importância da análise de sentimentos para a compreensão das opiniões do público.

Existem diferentes técnicas de análise de sentimento disponíveis, incluindo abordagens baseadas em regras, técnicas de aprendizado de máquina e métodos híbridos que combinam essas duas abordagens. Uma revisão sistemática de literatura realizada por Hilario et al. (2021) identificou as técnicas mais utilizadas em análise de sentimento, que compreendem análise de polaridade, classificação de sentimento e análise de emoções. No estudo, destaca-se a aplicação de algoritmos como *Support Vector Machines* (SVM), Redes Neurais Artificiais (ANN) e Árvores de Decisão. Tais algoritmos de aprendizado de máquina têm se mostrado eficazes na realização da análise de sentimento, permitindo que as empresas analisem grandes volumes de dados de maneira eficiente e precisa. Porém, devido à natureza mais simples dessas técnicas, elas tendem a ter dificuldades para lidar com as sutilezas da linguagem, como ironia e sarcasmo, e podem não ser capazes de entender o contexto no qual as palavras são usadas.

A fim de superar algumas dessas limitações, surgiram os modelos de processamento de linguagem natural baseados em *Transformers*. Dentre esses, o BERT (*Bidirectional Encoder Representations from Transformers*) se destaca. Segundo Lutkevich (2020), o BERT foi desenvolvido com o objetivo de auxiliar os computadores a compreender o significado de linguagem ambígua nos textos. Para isso, esse algoritmo utiliza o contexto fornecido pelo texto circundante para estabelecer uma compreensão mais precisa. Os modelos baseados em *Transformers*, como o BERT, são treinados para entender o contexto das palavras em uma frase, o que os permite capturar melhor as sutilezas e ambiguidades da linguagem. Além disso, esses modelos não requerem a extração manual de recursos, o que pode simplificar o processo de análise de sentimentos e torná-los mais adaptáveis a diferentes conjuntos de dados e tarefas.

Em suma, a análise de sentimento tem se mostrado uma técnica cada vez mais valiosa para empresas e pesquisadores que desejam compreender melhor a opinião do público sobre produtos, serviços e outros assuntos. O uso de técnicas avançadas de aprendizado de máquina torna essa tarefa não só possível, como também mais rápida e eficiente.

1.2 Motivação

Conforme o comércio eletrônico continua a expandir sua presença no Brasil, as avaliações de produtos desempenham um papel cada vez mais significativo no processo de

tomada de decisão dos consumidores. Além de influenciar a escolha de outros compradores, essas avaliações fornecem valiosos *feedbacks* às empresas sobre a percepção de seus produtos.

O estudo conduzido por FILHO (2023), que aborda o desenvolvimento do comércio eletrônico no Brasil, revelou um crescimento notável no número de lojas *online*, com uma taxa anual de crescimento de 21,3% durante o período de 2014 a 2019. Esse crescimento foi ainda mais acentuado nos anos de 2020 e 2021, impulsionado pela pandemia e pelo fechamento de diversas lojas físicas. Em 2020, houve um aumento de 40% em relação ao ano anterior, e essa tendência continuou em 2021.

À medida que a quantidade de dados disponíveis nas plataformas de comércio eletrônico continua a se expandir, a análise de sentimentos pode ajudar a identificar tendências e padrões de comportamento dos consumidores. Isso permite que as empresas ajustem suas estratégias de marketing, aprimorem a qualidade de seus produtos e serviços, e, ao mesmo tempo, contribuam para uma gestão mais eficaz de perdas.

Entretanto, considerando a vasta quantidade de avaliações disponíveis, a análise manual desses dados pode ser considerada impraticável, além de ser suscetível à subjetividade na interpretação dos sentimentos expressos. É nesse ponto que a evolução tecnológica recente desempenha um papel crucial. Como afirmado por Junqueira e Fernandes (2018), "Os avanços tecnológicos permitem coletar, armazenar e processar grandes volumes de dados. Sites de notícias, blogs e redes sociais são fontes de dados que concentram grandes volumes de informação". Tudo isso tem impulsionado a crescente demanda pelo uso de técnicas mais avançadas de análise de sentimentos por empresas e organizações que buscam compreender a opinião e os sentimentos expressos pelos consumidores em relação aos seus produtos e serviços.

Um outro fator a se considerar é a recente popularização de modelos de aprendizado de máquina baseados em aprendizado profundo, que foram impulsionados pelos avanços e barateamento do *hardware* disponível para os consumidores (RAIAAN et al., 2023). Modelos de Processamento de Linguagem Natural (PLN) mais avançados têm surgido com uma velocidade cada vez maior como alternativas aos métodos tradicionais, sendo um deles o BERT e seus derivados. Mas apesar de sua crescente popularidade e do sucesso demonstrado em várias tarefas de PLN, ainda existe uma falta de estudos comparativos que avaliem a eficácia desses modelos em relação aos métodos convencionais, especialmente no contexto da língua portuguesa.

Este hiato na literatura motivou a condução deste trabalho, que se propõe a preencher parte dessa lacuna ao realizar uma análise comparativa entre um modelo de PLN mais avançado e os métodos tradicionais. Nosso foco principal está na análise de sentimentos expressos em avaliações de produtos em língua portuguesa, com o objetivo de fornecer uma compreensão mais sólida da eficácia dessas abordagens em nosso contexto linguístico.

1.3 Objetivo

O objetivo geral do estudo consiste em realizar uma comparação entre diversos algoritmos de processamento de linguagem natural aplicada à análise de sentimentos em avaliações de produtos no contexto da língua portuguesa. Além disso, iremos analisar o impacto de diferentes parâmetros e configurações de cada algoritmo para determinar como eles afetam o desempenho geral dos resultados. Por fim, apresentaremos nossas conclusões e recomendações.

1.3.1 Objetivos Específicos

- Efetuar uma Revisão sistemática de literatura para obter informações sobre as técnicas utilizadas em modelos tradicionais;
- Seleção de uma base de dados relevante para avaliar o desempenho de cada algoritmo;
- Analisar o impacto de diferentes parâmetros e configurações de cada algoritmo para determinar como eles afetam o desempenho geral em comparação com o BERT;
- Apresentar nossas conclusões e recomendações com base nos resultados obtidos.

1.4 Organização do Trabalho

Este estudo está estruturado da seguinte forma: o Capítulo 2 faz uma contextualização do assunto, com uma explicação dos conceitos usados no decorrer do trabalho.

No Capítulo 3 apresentamos trabalhos relacionados que contribuíram para a pesquisa aqui realizada, incluindo uma revisão sistemática de literatura sobre os algoritmos mais usados em processamento de linguagem natural.

As tecnologias e método usados neste trabalho são apresentados no Capítulo 4, que descreve também a base de dados usada, o pré-processamento dos dados e alguns detalhes sobre a implementação dos algoritmos. O Capítulo 5 apresenta e discute os resultados obtidos em contexto.

Por fim, no Capítulo 6, as conclusões do trabalho são apresentadas, relacionando-as aos objetivos e propondo ideias para trabalhos futuros.

2 Referencial Teórico

Este capítulo apresenta alguns conceitos importantes para o entendimento deste projeto. Para isso, faz-se necessário uma breve revisão de assuntos relacionados tanto a aprendizado de máquina e Processamento de Linguagem Natural (PLN) quanto os modelos atuais baseados em *Transformers*.

2.1 Aprendizado de Máquina

Conforme mencionado por Mitchell (1997), a pesquisa de aprendizado de máquina é fundamentada na ideia de desenvolver programas de computador capazes de "aprender" com a experiência. Em outras palavras, o desempenho desses programas em uma determinada tarefa é aprimorado à medida que eles interagem com dados e adquirem conhecimento através dessa experiência.

Aprendizado de máquina é um subcampo muito importante na pesquisa de Inteligência Artificial (IA), e inclui o estudo de métodos computacionais para a aquisição automática de conhecimento, bem como para a construção e acesso ao conhecimento existente. Mitchell (1997) aponta ainda que o conhecimento detalhado dos algoritmos de aprendizado de máquina também pode levar a compreender melhor a capacidade (e incapacidade) humana de aprender.

Os algoritmos de aprendizado de máquina podem ser divididos em dois grupos diretores, aprendizado *supervisionado* e aprendizado *não supervisionado*. O aprendizado supervisionado é usado para construir modelos preditivos, equanto que o não supervisionada é usado para criar modelos descritivos (SANTOS, 2021).

Modelos preditivos são aplicados a problemas envolvendo a previsão de variáveis a partir de outras variáveis adicionais do conjunto de dados. A variável preditora é geralmente chamada de variável de destino. O algoritmo tenta descobrir e modelar a relação entre a variável de destino e outras variáveis. Como o modelo preditivo recebe instruções do que precisa aprender, o processo de treinamento é chamado aprendizado supervisionado (LANTZ, 2013). Já no modelo descritivo, a ideia é analisar o conjunto de dados fornecido e tentar determinar correlações e ocorrências de alguma maneira, formando agrupamentos ou *clusters* (CHEESEMAN; STUTZ et al., 1996).

A técnica não supervisionada não requer a definição prévia de sentenças ou treinamento para criar o modelo. Um exemplo notável dessa técnica é a abordagem léxica, que se baseia em um tipo de dicionário de palavras. Nesse método, em vez de associar a cada palavra um significado textual, é atribuído a elas um valor numérico, geralmente entre -1 (indicando o sentimento mais negativo) e 1 (representando o mais positivo). A Figura 1 ilustra de forma geral o funcionamento de um método de análise de sentimentos léxico:

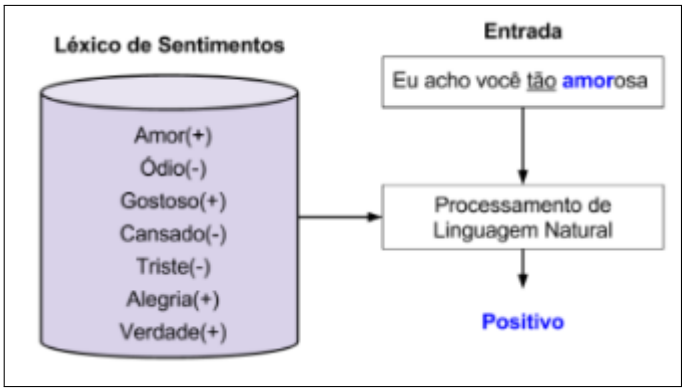


Figura 1 – Léxico de sentimentos
(BENEVENUTO; RIBEIRO; ARAÚJO, 2015)

A Figura 2 esboça de forma simplificada os conceitos subjacentes à criação de um classificador de aprendizado supervisionado. Nesse contexto, cada conjunto de dados Xi consiste em m atributos (palavras), ou seja, $Xi = (Xi1, ..., Xim)$. As variáveis Yi representam as classes correspondentes. Através dos exemplos e suas respectivas classes, o algoritmo de aprendizado de máquina é capaz de extrair informações e construir um classificador.

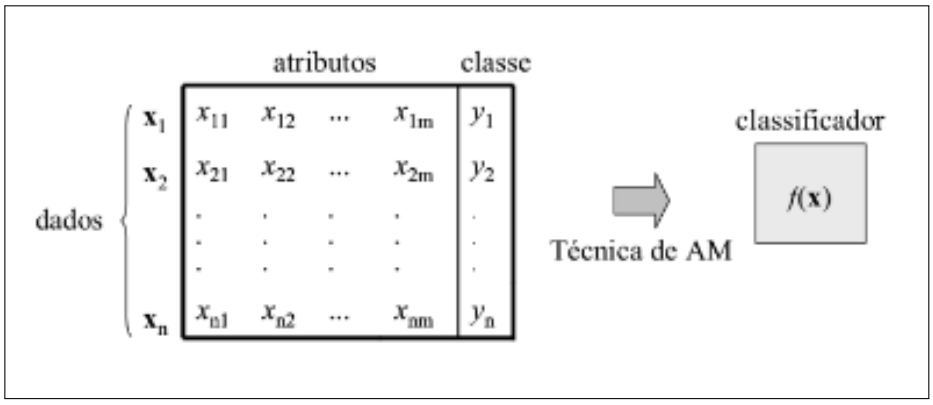


Figura 2 – Indução de classificador em aprendizado supervisionado
(LORENA; CARVALHO, 2007)

2.2 Processamento de Linguagem Natural

O processamento de linguagem natural, de forma simplista, tem como objetivo fazer o computador se comunicar em linguagem humana, levando em conta as diversas características que a compõem, como sons, palavras, sentenças e discursos (GONZALEZ; LIMA, 2003).

No decorrer da sua história, a área de processamento de linguagem natural (PLN) passou por diversas transformações, sendo três delas de destaque. A abordagem *simbólica*

foi a primeira a ser proposta e contribuiu para a elaboração das primeiras gramáticas que poderiam ser processadas por computadores e bases de conhecimento. Com avanços na disponibilização massiva de dados textuais, surgiu a oportunidade de aplicar técnicas estatísticas para o reconhecimento de padrões, marcando o surgimento dos primeiros *modelos estatísticos*. Nessa fase, foram empregados diversos algoritmos de aprendizado de máquina, destacando-se a primeira versão do Google Tradutor. Este modelo permaneceu em uso de 2006 a 2016, até ser substituído pela versão neural, que está em uso atualmente. A abordagem *neural* é hoje a mais utilizada quando se busca melhores resultados no estudo do processamento de linguagem natural (CASELI; FREITAS; VIOLA, 2022).

É identificado no processamento de linguagem humana a dificuldade de entendimento das variações linguísticas e ambiguidade por parte das máquinas, visto que objetos linguísticos de diferentes tipos, como frases e palavras, podem não ter o mesmo significado dependendo da ocorrência e do contexto inserido (KILGARRIFF, 1997).

Podemos definir, dentro do processamento de linguagem natural (PLN), dois tipos de ambiguidade: *sintática*, quando a mesma é encontrada já em nível sintático, e *semântica*, quando a ambiguidade aparece somente em nível semântico (JURAFSKY, 2000).

A complexidade da linguagem é ressaltada também por outros autores. Salton (1968), por exemplo, afirma que é relativamente fácil isolar palavras individuais em um texto, mas a interpretação do significado das palavras é bem mais difícil. Além disso, também é mencionado que não há um conjunto de regras bem definidas a partir das quais as palavras de uma linguagem podem ser combinadas em grupos ou sentenças com significado. É dito posteriormente que a correta identificação do significado de um grupo de palavras depende ao menos em parte do reconhecimento das ambiguidades sintáticas e semânticas, da correta interpretação dos homógrafos, do reconhecimento das equivalências semânticas, da detecção das relações entre palavras, dentre outros (SALTON, 1968).

2.3 Análise de Sentimentos

A análise de sentimentos busca identificar e extrair automaticamente as opiniões, sentimentos e emoções expressados em um texto (NARAYANAN; LIU; CHOUDHARY, 2009). Essa abordagem tem se tornado cada vez mais popular como uma ferramenta de mineração de dados, com diversas aplicações úteis. Através dessas técnicas, é possível revelar, por exemplo, como as pessoas se sentem em relação a um determinado assunto, o que é fundamental para que as empresas possam direcionar suas estratégias de marketing de maneira mais fundamentada. Esta análise é uma das áreas do processamento de linguagem natural (PLN) que passou a ser mais investigada a partir dos anos 2000 (LIU, 2022), já que o crescente número de dados disponíveis nos mais diversos contextos torna viável a utilização desses algoritmos em larga escala.

Existem vários níveis de granularidade disponíveis no campo da análise de sentimen-

tos (ZHANG; WANG; LIU, 2018). Dentre esses níveis, a análise de nível de *documento* admite que todo seu conteúdo é opinativo e apresenta uma polaridade geral. Já a análise de *sentença* determina a polaridade individual, supondo que as sentenças analisadas realmente expressam opiniões. Há muitas outras ramificações da análise de sentimento, e dado que é uma área vasta e pouco explorada, a tendência é de aumento.

2.4 Algoritmos mais comuns de aprendizado de máquina

Alguns dos algoritmos de aprendizado de máquina mais usados no contexto de análise de sentimento evidenciados pela Revisão Sistemática de Literatura (seção 3.4) foram o *Support Vector Machines* (SVM), *Naive Bayes*, Regressão Logística e *Random Forest*.

2.4.1 *Support Vector Machines*

Máquinas de Vetores de Suporte ou *Support Vector Machines* (SVM) é uma abordagem de aprendizagem introduzida por Vapnik em 1995 (CORTES; VAPNIK, 1995). Esta técnica busca minimizar o erro com relação ao conjunto de treinamento, assim como o erro com relação ao conjunto de teste, isto é, ela não emprega determinado conjunto de amostras no treinamento do classificador, essas são excluídas com base no critério linear da SVM.

SVM está entre os algoritmos mais conhecidos na área de aprendizado de máquina, pois possui utilidade em diversas aplicações, como categorização de textos e análise de imagens. Também possui alto índice de acurácia se comparado a outros algoritmos tradicionais (vide a seção 3.4). O mesmo apresenta aprendizado supervisionado linear, que é utilizado na classificação de um conjunto de pontos buscando uma linha de separação entre duas classes distintas.

Temos a figura 3 (a), a qual exibe um conjunto de duas classes de dados. Pode-se dizer que essas categorias são linearmente separáveis ao olhar a figura 3 (b), pois há uma linha capaz de separá-las. No entanto, é preciso ressaltar que apenas uma linha divide essas categorias de forma que a margem entre essas duas categorias seja a maior possível. O objetivo do SVM é conseguir a melhor separação entre as duas classes, conforme mostra a figura 3 (b).

Ao observar a figura 3 (b), nota-se que, para execução da linha que separa os conjuntos, não é necessário todo o conjunto de dados, apenas os dados mais próximos da linha como é exibido na figura 3 (c). Esses dados mais relevantes, que foram destacados são chamados de *support vectors* (SVs) ou vetores de suporte.

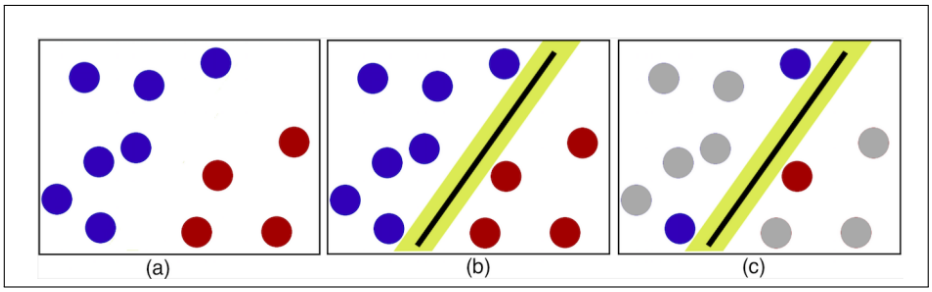


Figura 3 – Conjunto de dados linearmente separável
(NORONHA; FERNANDES, 2016)

2.4.2 *Naive Bayes*

Os classificadores *Naive Bayes* levam em consideração a existência de muitos atributos para fazer a previsão final (KOHAVI et al., 1996), uma propriedade que é útil em casos que não sejam capazes de gerar efeito rebote, onde a utilização de variáveis demais pode comprometer o resultado. Apesar de alguns pontos negativos, o *Naive Bayes* é um dos mais eficientes e eficazes algoritmos utilizados em aprendizado de máquina e mineração de dados (ZHANG, 2004).

O algoritmo de classificação *Naive Bayes* é baseado no Teorema de Bayes para fazer previsões. Apesar de o classificador se basear em um conceito que nem sempre é verdadeiro, apresenta boa acurácia em atividades de classificação (CHEN et al., 2009). A ideia básica é usar a junção das probabilidades das palavras e categorias para estimar as probabilidades das categorias de um novo documento.

Segundo Tan, Steinbach e Kumar (2009), a utilização do algoritmo de *Naive Bayes* em aplicações apresenta uma característica não determinística, onde o rótulo da classe de um registro não possui certeza em sua previsão, mesmo que seus atributos sejam idênticos a exemplos de treinamento.

No trabalho de Degasperi (2023) é exibida a estrutura linear do método *Naive Bayes*, temos com isso, a equação do Teorema Bayes:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Neste contexto, A e B são eventos, $P(A|B)$ é a probabilidade de A ocorrer visto que B ocorreu, $P(B|A)$ é a probabilidade de B ocorrer visto que A ocorreu, $P(A)$ é a probabilidade de A ocorrer e $P(B)$ é a probabilidade de B ocorrer e $P(B) \neq 0$.

2.4.3 Regressão Logística

A regressão logística é um algoritmo de classificação amplamente utilizado na área de estatística há bastante tempo, e mais recentemente, foi incorporado à área de aprendizado de máquina devido à sua relação com o SVM (Máquinas de Vetores de Suporte).

Esse algoritmo tem como objetivo analisar o relacionamento entre diversas variáveis independentes e uma variável dependente categórica. Sua principal aplicação é estimar a probabilidade de ocorrência de um evento ou resultado binário, ou seja, um evento que pode ter apenas duas categorias, como “sim” ou “não”, “positivo” ou “negativo”, “aprovado” ou “reprovado”, entre outros (SANTOS, 2013).

Na área do aprendizado de máquina supervisionado, existem dois tipos principais de problemas: *classificação* e *regressão* (MÜLLER; GUIDO, 2016). Na classificação, o objetivo é identificar objetos através de semelhanças entre várias categorias pré-definidas. Já na regressão, o objetivo é prever um número contínuo ou número de pontos flutuantes, em termos de programação.

A regressão logística estuda a relação entre uma variável resposta e uma ou mais variáveis independentes, que estão dispostas em categorias e a resposta é expressa por meio de uma probabilidade de ocorrência (SANTO; FILHO, 2020). Esse método consiste em uma técnica estatística que tem como objetivo produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica a partir de uma série de variáveis explicativas contínuas e/ou binárias, sendo muito útil para modelar a probabilidade de um evento ocorrer como função de outros fatores (MONTEIRO et al., 2019).

2.4.4 *Random Forest*

Random Forest é um algoritmo *ensemble* idealizado por Breiman (2001) que constrói classificadores do tipo árvore de decisão. Tais classificadores são construídos de forma que sua estrutura seja composta de forma aleatória. O método combina o resultado de várias árvores de decisão por meio do voto majoritário. Ao final, cada árvore gera uma classificação ou voto para uma classe e a classificação final é dada pela classe que recebeu o maior número de votos dentre todas as árvores da floresta (DINIZ et al., 2013).

Esse algoritmo possui características que lhe garantem um ótimo desempenho: pode ser utilizado quando há muito mais atributos do que exemplos; possui bom desempenho preditivo mesmo quando a maioria das variáveis preditivas são ruídos; não superajusta; pode lidar com uma mistura de atributos nominais e numéricos e há pouca necessidade de ajudar os parâmetros para alcançar um bom desempenho (DÍAZ-URIARTE; ANDRÉS, 2006).

Breiman (2001) define *Random Forest* como um classificador composto para uma coleção de árvores $hk(x), k = 1, 2, ..., L$, onde Tk são amostras aleatórias independentes e identicamente distribuídas e cada árvore vota na classe mais popular para a entrada x.

A figura 4 consegue exemplificar bem o funcionamento do método *Random Forest* (OSHIRO, 2013).

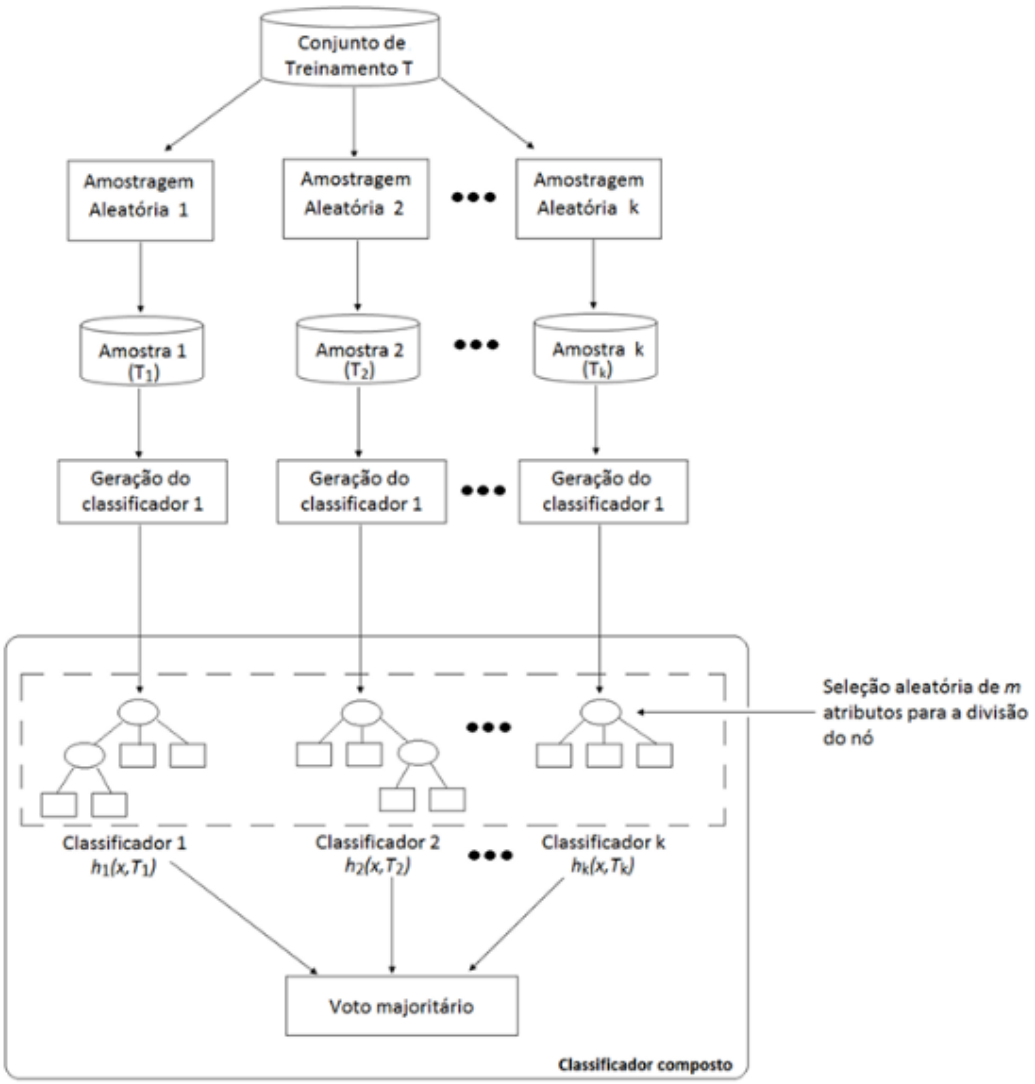


Figura 4 – Conjunto de dados linearmente separável
(OSHIRO, 2013)

2.5 Modelo de Linguagem BERT

Os modelos de linguagem têm a função de fornecer contexto para facilitar a interpretação de palavras ou frases com entonação semelhante. No campo do processamento de linguagem natural, utiliza-se modelos de representação de linguagem para realizar o aprendizado autônomo da interpretação de palavras (BOSCO; PILATO; SCHICCHI, 2018).

Os modelos para o processamento de linguagem natural eram voltados para funções específicas, como classificação, análise de sentimento e perguntas e respostas. No entanto, atualmente, surgiram modelos como o BERT, que são capazes de executar todas essas funções em um único algoritmo, permitindo uma abordagem mais abrangente e integrada (KENTON; TOUTANOVA, 2019).

2.5.1 *Transformers*

No contexto de Processamento de linguagem natural o *Transformer* é um tipo de arquitetura que realiza tarefas *sequence-to-sequence* (como dados textuais) enquanto atua com relacionamentos de longo alcance(KHAN et al., 2021), e foi inicialmente pensada para tarefas de tradução.

Como podemos visualizar na figura 5, baseada no artigo de Vaswani et al. (2017), a arquitetura da rede *Transformer* é composta por duas partes principais, o codificador e o decodificador, ambos consistem em vários blocos *Transformer*, todos com a mesma estrutura. O codificador é a parte primária da rede, e recebe uma representação da sequência de entrada. Essa sequência pode ser uma frase, um parágrafo ou qualquer forma de dados sequenciais. Seu papel é processar essa entrada para capturar quais partes dela são relevantes entre si. Esse processamento é feito através de mecanismos de atenção, onde a rede aprende a dar mais importância a certas partes da sequência.

Estes métodos de atenção tem como objetivo alcançar as partes mais importantes de um contexto, e trabalhar nos componentes de destaque, segundo Xu et al. (2015). Esse comportamento é inspirado no sistema biológico humano, onde o ser humano tende a ignorar informações irrelevantes e focar sua atenção em partes específicas. O primeiro trabalho integrando esse tema com processamento de linguagem natural foi feito por Bahdanau, Cho e Bengio (2014).

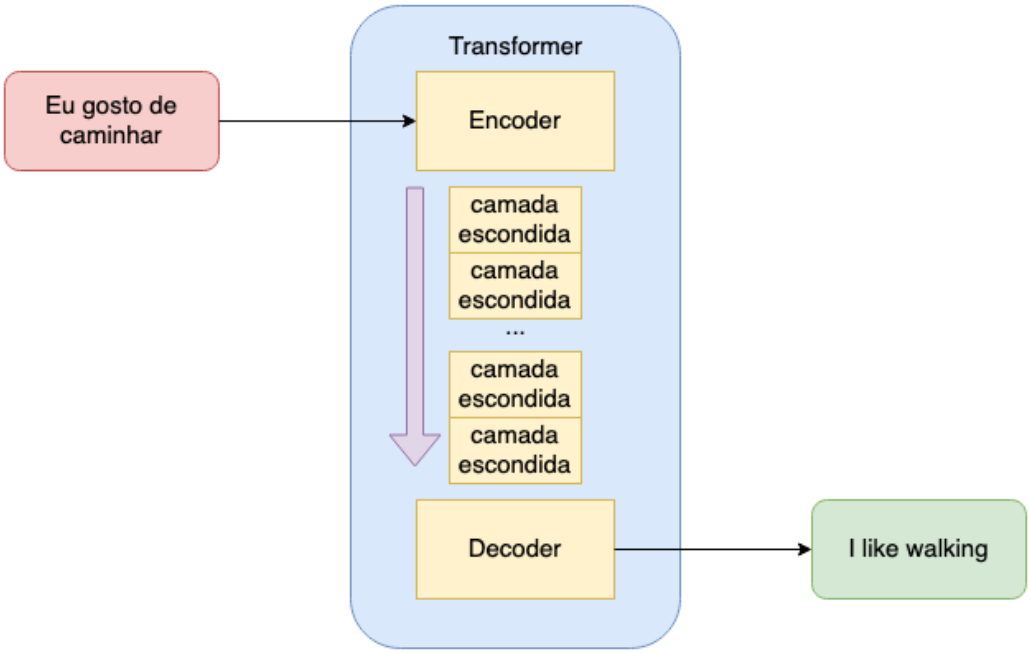


Figura 5 – Arquitetura do modelo Transformer, autoria própria.

Já o decodificador é a segunda parte da rede, com função de gerar as sequências de saída, ela usa as informações geradas do codificador para entender o contexto, após isso

é gerada a saída. Podemos ter um exemplo disso em mecanismos de tradução, onde você insere um idioma e tem como resultado uma sequência equivalente em outro idioma

Como demonstrado na imagem 5, cada bloco é formado por diversos componentes, e esses incluem um mecanismo de atenção multi-cabeça, que permite a rede aprender padrões de dependência entre as partes da sequência; uma rede neural *feed-forward*, que é uma camada densa que processa os dados. conexões residuais, que são conexões que permitem a passagem direta de informações através do bloco. e camadas de normalização, que ajudam a estabilizar o treinamento da rede.

2.5.2 BERT

O BERT (*Bidirectional Encoder Representations from Transformers*) é um modelo de aprendizado profundo pré-treinado e voltado para a representação contextual de linguagem que tem alcançado resultados estado-da-arte para diversas tarefas de PLN (KENTON; TOUTANOVA, 2019).

Este modelo é treinado de maneira não-supervisionada utilizando um grande corpus, a partir de uma abordagem que mascara (*masked language model* ou MLM) algumas palavras em uma sequência de palavras do corpus, ou seja, o modelo tem acesso a todas as palavras da sequência de entrada exceto às palavras mascaradas. O treinamento realizado é voltado a predizer quais as palavras mascaradas na sequência original e, para isso, a representação de cada palavra é calculada por um mecanismo de atenção que pondera a representação de todas as outras palavras da sequência (KENTON; TOUTANOVA, 2019).

O que diferencia o BERT de outros modelos de linguagem é a utilização de uma abordagem *bidirecional* em vez de ler a entrada de texto apenas sequencialmente, da esquerda para a direita (RADFORD et al., 2018). Para obter essa bidirecionalidade, a rede é treinada usando o MLM utilizando tanto as palavras que estão antes da palavra mascarada (contexto esquerdo) como as que estão depois dela (contexto direito). Posteriormente, o algoritmo busca prever a próxima frase em um texto. Após receber como entrada pares de sentenças, com o intuito de entender o relacionamento entre as duas, a sentença como um todo é alimentada ao modelo de transformação e gera como saída um vetor se utilizando de uma camada de classificação e, a partir disso, a probabilidade da próxima sentença é calculada por meio de uma função de normalização exponencial. Além disso, os modelos pré-treinados pelo BERT podem ser melhorados através de uma nova camada de saída de dados, sem que sejam necessárias modificações específicas em sua arquitetura (KENTON; TOUTANOVA, 2019).

O pré-processamento de dados com o BERT acontece da seguinte forma: inicialmente, o algoritmo aprende e utiliza o acondicionamento posicional de palavras em uma frase, fazendo com que o algoritmo compreenda a posição relativa ou absoluta de *tokens* em uma sequência; em seguida, a aprendizagem do acondicionamento pode ser utilizada para

juntar pares de sequências; por fim, através do acondicionamento de *tokens*, o texto de entrada para alimentação do modelo BERT é produzido (BARBOSA; MAAS; PEREIRA, 2019).

O algoritmo BERT usa *boosting* (MITCHELL; FRANK, 2017), procedimento que cria vários classificadores de forma iterativa, de acordo com a figura 6, adaptada de Pires, Martins e Sousa (2018).

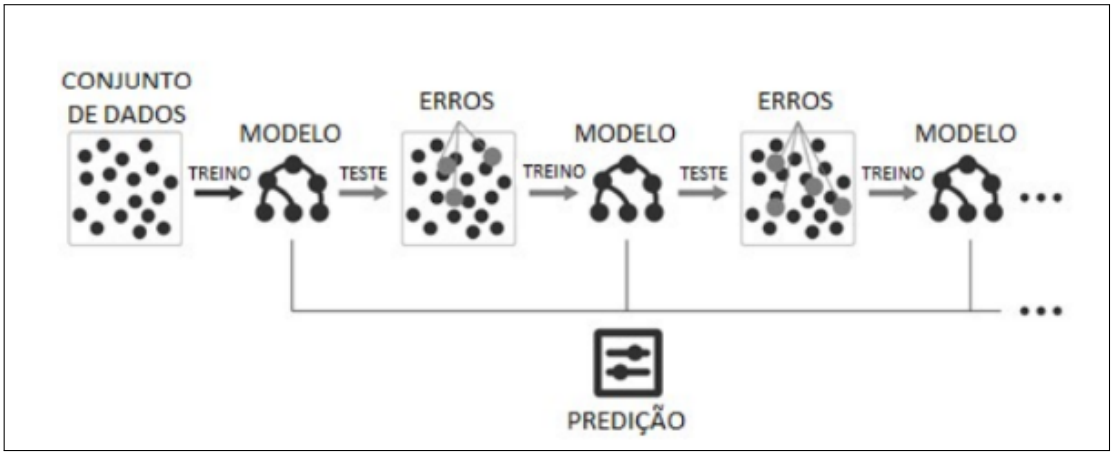


Figura 6 – Esquema de boosting
(PIRES; MARTINS; SOUSA, 2018)

Inicialmente, tem-se um conjunto de dados, ao qual são atribuídos pesos e a cada interação (de treino, modelo ou teste), esses pesos são modificados. Com isso, as amostras que são mais difíceis de se classificar recebem um peso maior e consequentemente se tornam o foco da iteração seguinte.

2.6 Métricas usadas na Análise de Sentimentos

Mikhail e Ackerman (1976) abordam os conceitos de acurácia e precisão especificamente no ramo da estatística. Em seu trabalho, definem a acurácia como a medida do quão próxima uma estimativa está do seu parâmetro real, enquanto a precisão é a medida da consistência das medições em relação a média da grandeza medida.

Uma matriz de confusão, também conhecida como tabela de confusão, é uma ferramenta usada em estatísticas e aprendizado de máquina para avaliar o desempenho de um modelo de classificação. Castro e Braga (2011) ressaltam que a maneira mais eficaz de se avaliar um dado classificador é através da distinção dos erros (ou acertos) cometidos para cada classe. Isso pode ser obtido descrevendo o desempenho a partir de uma matriz de confusão, que consiste em uma matriz interpor os dados reais e os valores preditos pelo modelo, conforme exemplo a seguir.

Conforme Castro e Braga (2011), ao longo da diagonal principal (em azul), estão representadas as predições corretas do modelo: verdadeiros positivos (TP) e verdadeiros

Matriz de confusão		Previsto pelo Modelo	
		Não *churner* (N)	*Churn*
Situação real	Não *churner* (N)	Verdadeiro Negativo (TN)	Falso Positivo (FP)
	Churner (P)	Falso Negativo (FN)	Verdadeiro Positivo (TP)

Figura 7 – Matriz de Confusão
(FRANCESCHI, 2019)

negativos (TN); os elementos fora dessa diagonal representam os erros cometidos pelo modelo: falsos positivos (FP) e falsos negativos (FN).

Importantes métricas que podem ser extraídas da matriz de confusão, conforme abaixo:

A acurácia (*Accuracy*) é uma métrica de desempenho intuitiva, que se baseia na razão entre as observações corretamente previstas (verdadeiros positivos e verdadeiros negativos) e o total de observações. Ela é calculada pela seguinte equação.

$$Accuracy = \frac{Tp - Tn}{Tp - Fp}$$

Onde:

Tp representa os verdadeiros positivos, Tn representa os verdadeiros negativos, Fp representa os falsos positivos, e Fn representa os falsos negativos. A precisão (*Precision*) é outra métrica importante, que considera a razão entre as observações positivas corretamente previstas (verdadeiros positivos) e o total de observações positivas previstas. Ela é definida pela equação:

$$Precision = \frac{Tp}{Tp - Fp}$$

O Recall, também conhecido como Sensibilidade (*Sensitivity*), é a razão das observações positivas corretamente previstas (verdadeiros positivos) para o total de observações na classe investigada. Sua fórmula é a seguinte:

$$Recall = \frac{Tp}{Tp - Fn}$$

A pontuação F1 (*F1 Score*) é uma métrica que combina a precisão e o recall, levando em consideração tanto os falsos positivos quanto os falsos negativos. É calculada pela fórmula:

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Essas métricas são essenciais na avaliação do desempenho de modelos de classificação, permitindo uma compreensão completa da qualidade das previsões em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

Ainda no contexto de análise de desempenho de modelos, a Curva ROC (*Receiver Operating Characteristic*) aparece como uma das mais utilizadas, por conta de sua representação gráfica da performance de um modelo de dados quantitativos segundo sua taxa de sensibilidade. (POLO; MIOT, 2020)

Os valores mencionados na figura 7 fornecem a base para calcular medidas amplamente utilizadas no contexto da curva ROC. Essas medidas incluem sensibilidade, especificidade e ponto de corte, todas desempenhando papéis essenciais na avaliação da capacidade de discriminação de um modelo de classificação. A sensibilidade representa a habilidade do modelo em identificar corretamente uma classificação positiva quando ela é, de fato, positiva. Por outro lado, a especificidade é a capacidade do modelo em identificar corretamente um resultado negativo quando ele é verdadeiramente negativo (MARTINEZ; LOUZADA-NETO; PEREIRA, 2003). O cálculo das medidas é definido da seguinte forma:

$$Sensibilidade = \frac{Tp}{Tp - Fn}$$

$$Especificidade = \frac{Vn}{Fp - Tn}$$

Já o ponto de corte representa um limiar crítico que separa e diferencia uma variável, categorizando se uma característica específica está presente ou ausente em indivíduos. O ponto de corte estabelece um valor para essa medida, delineando quais indivíduos se encontram acima ou abaixo desse limite (MORANA, 2004).

3 Trabalhos Relacionados

Neste capítulo são apresentados trabalhos relacionados aos escopos de análise de sentimento e algoritmos de Processamento de Linguagem Natural (PLN), além daqueles que fazem alguma comparação entre algoritmos de PLN tradicionais e o BERT.

3.1 Análise de sentimento

Os seguintes estudos apresentam diferentes abordagens para a classificação de emoções e sentimentos em diversos contextos, utilizando arquiteturas de classificação baseadas em redes neurais e outras estratégias de aprendizado de máquina.

No estudo conduzido por Avila (2017), foi observada uma grande quantidade de mensagens informais em forma de *feedback*, publicadas em fóruns, pesquisas de satisfação e redes sociais. Diante desse cenário, o autor optou por realizar a análise de sentimento em textos curtos, com o objetivo de identificar se uma mensagem específica tinha um sentimento positivo, negativo ou neutro. Os resultados obtidos foram uma acurácia de 86,8% em um modelo de três classes e 94,5% em um modelo de duas classes.

No trabalho de Souza et al. (2020), foi realizada uma pesquisa de análise de sentimento utilizando posts do Twitter. O objetivo era relacionar o sentimento expresso por investidores sobre empresas no Twitter com as variações correspondentes no mercado de ações. Utilizando a ferramenta *Google Cloud Natural Language* (GCNLP), a pesquisa identificou um padrão onde, em geral, um sentimento mais positivo expressado por investidores no Twitter estava associado a um maior retorno na IBOVESPA. No entanto, a pesquisa também sugeriu a possibilidade de um fenômeno de "efeito rebote", onde o mercado reagia exageradamente às notícias positivas, seguido por uma reversão que resultava em uma queda em relação aos valores anteriores.

A análise de sentimentos no nível de características foi abordada por Silva, Lima e Barros (2012), que desenvolveu um método composto por duas etapas distintas: extração de características, que identifica o contexto do texto com base em um corpus, e classificação, que envolve a polarização das características e pares identificados. Os resultados obtidos demonstraram uma alta precisão por meio desse processo, e os autores enfatizaram a eficácia das regras aplicadas para a língua portuguesa, sem encontrar quaisquer dificuldades significativas.

Outro estudo de análise de sentimento, focado em comentários em sites de *e-commerce* em português, foi realizado por Albuquerque (2022). O autor destacou os desafios da análise manual devido ao grande número e variedade de comentários disponíveis. Para superar esse desafio, foi desenvolvido um modelo de PLN capaz de realizar a análise de

sentimento de forma automática. Ao final dos testes, o modelo alcançou pelo menos 85% de acurácia em todos os testes realizados.

3.2 Algoritmos tradicionais de PLN

Alguns estudos focaram exclusivamente em comparar alguns dos algoritmos tradicionais de aprendizado de máquina no contexto de dados textuais.

O estudo conduzido por Varela (2012) concentra-se na classificação de sentimentos, com o propósito de avaliar métodos tradicionais. A pesquisa utilizou uma base de dados composta por avaliações relacionadas ao cinema em língua espanhola e portuguesa. Os resultados revelaram que a técnica de classificação *Naive Bayes* se destacou como uma abordagem apropriada para análises de textos curtos.

Já o estudo de Lin (2020) se voltou para a detecção de polaridade em avaliações de usuários em uma plataforma de *e-commerce* que comercializa roupas, sendo as avaliações em língua inglesa. O objetivo era comparar diversas técnicas de análise de sentimento. Após a comparação com algoritmos populares, como SVM e XGBoost, os resultados apontaram que essas técnicas alcançaram uma acurácia de 0,94. No entanto, o algoritmo Light GBM se destacou, obtendo uma acurácia ainda maior de 0,97.

Por fim, no trabalho de Tiburcio (2021), a pesquisa concentrou-se em cinco algoritmos de classificação de sentimento: *Random Forest*, Regressão Logística, *Naive Bayes*, *KNearest Neighbor* (KNN) e *Support Vector Machine* (SVM). O objetivo era analisar comentários em redes sociais. Para conduzir a análise, uma base de dados pública com mais de 50 mil registros de postagens de usuários foi utilizada, sendo obtida através de uma API (Interface de Programação de Aplicações) disponibilizada pelo Twitter em 2018. Os resultados da pesquisa indicaram que SVM e Regressão Logística obtiveram as maiores acurácias, atingindo 77% e 76,1%, respectivamente.

3.3 BERT

Apesar de ainda poucos, estudos sobre o BERT têm aparecido com cada vez mais frequência desde sua publicação em 2018.

O estudo conduzido por González-Carvajal e Garrido-Merchán (2020) realizou uma comparação entre o modelo baseado em redes neurais BERT e outros algoritmos de processamento de linguagem natural tradicionais no contexto de análise de sentimento em tarefas de classificação. Os resultados obtidos revelaram que o BERT apresentou um desempenho superior em diversas atividades, incluindo o processamento de textos em português e chinês.

No trabalho de Munikar, Shakya e Shrestha (2019), realizou-se uma pesquisa para entender a percepção de pessoas em relação a um produto ou serviço. Para esta análise

foram propostos muitos modelos de classificação de sentimentos. Como resultado, o BERT se mostrou superior a modelos mais populares para esta tarefa.

No trabalho de Hoang, Bihorac e Rouces (2019), O BERT foi capaz de melhorar os resultados em diversas tarefas de PLN, como a análise de sentimento e a resposta de perguntas, obtendo um desempenho próximo ao de humanos.

3.4 Revisão Sistemática

Com o intuito de identificar trabalhos relacionados a este trabalho de conclusão de curso, foi conduzida uma revisão sistemática com base nas etapas propostas em Kitchenham (2004). O objetivo foi identificar e selecionar artigos que realizaram comparações entre métodos tradicionais de PLN para, a partir dessa seleção, realizar uma comparação entre os algoritmos mais usados e o BERT.

1. **Questões de pesquisa:** para atingir o objetivo descrito anteriormente, foram definidas as seguintes questões de pesquisa:

   - Questão Secundária 1 (QS1): Quais são as linguagens de programação mais frequentemente utilizadas na implementação de análise de sentimentos?
   - Questão Secundária 2 (QS2): Quais são os algoritmos de Aprendizado de Máquina mais comumente utilizados na análise de sentimentos?
   - Questão Principal (QP): Dentre os algoritmos de machine learning mais utilizados no contexto de análise de sentimentos, quais apresentam melhores dados de acurácia em comparações entre si?

2. **Definição da *string* de busca:** O trabalho opta por utilizar termos que traduzem uma melhor significação das intenções desta pesquisa. Para maior especificação da relação com este trabalho, foi elaborada a seguinte *string* de busca: ("Machine Learning" or "Deep Learning" or NLP or "aprendizagem de máquina") and ("análise de sentimento") and (algorithms or algoritmo).

3. **Definição das fontes de pesquisa:** foram feitas buscas pelos portais de periódicos *Google Scholar* e Scielo para o período de 2018 e 2023.

4. **Definição dos critérios de inclusão e exclusão:** (a) O artigo deve deixar claro qual algoritmo de classificação está sendo utilizado; (b) O artigo deve mencionar e citar a fonte da base de dados utilizada.

5. **Seleção dos artigos:** A pesquisa inicial da *string* de busca resultou em 60 trabalhos provenientes de ambas as fontes utilizadas. Ao aplicar o primeiro critério de seleção de trabalhos publicados após 2018, o número de resultados reduziu para 42. Dessa

forma, prosseguiu-se com aplicação de outros critérios, e após a aplicação de todos CI e CE sobraram 22 estudos que se adequam ao objetivo final da RSL.

6. **Extração dos dados:** Os artigos selecionados foram trabalhados para extrair informações relevantes conforme as questões de pesquisa definidas na etapa 1.

A fim de responder à questão de pesquisa "Quais são as linguagens de programação mais frequentemente utilizadas na implementação de análise de sentimentos?" (QS1), foram selecionados trabalhos que apresentavam aplicações práticas de análise de sentimentos. Um pré-requisito fundamental para inclusão dos artigos na seleção foi a presença de exemplificações e trechos de códigos utilizados nos algoritmos, a fim de avaliar de fato a linguagem adotada, sem suposições.

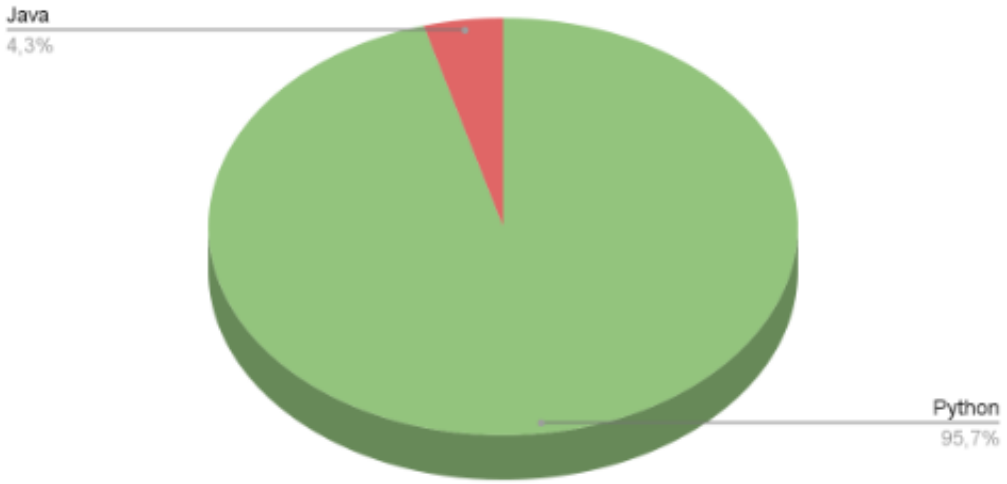


Figura 8 – Linguagens de programação mais utilizadas na análise de sentimentos.

A análise dos trabalhos selecionados revelou que a linguagem de programação Python está presente em 95,7% dos trabalhos, distribuição mostrada na figura 8. Isso confirma que Python é a linguagem mais utilizada na implementação de análise de sentimentos e ciência de dados.

Em relação à QS2, "Quais são os algoritmos de Aprendizado de Máquina mais comumente utilizados na análise de sentimentos?", a análise dos 22 artigos selecionados revelou que cada um apresentou suas particularidades e diversidade em técnicas de análise de sentimentos. Foram mapeados todos os algoritmos de classificação utilizados em todos os trabalhos, considerando que alguns trabalhos possuem mais de um algoritmo em seu corpo, em caso de existir no trabalho estudado algum tipo de comparação entre os modelos.

Conforme identificado na figura 9, os três algoritmos que possuem maior predominância são: *Naive Bayes, Support Vector Machine (SVM)* e *Random Forest*, respectivamente.

Para responder à QP, "Dentre os algoritmos de machine learning mais utilizados no contexto de análise de sentimentos, quais apresentam melhores dados de acurácia em

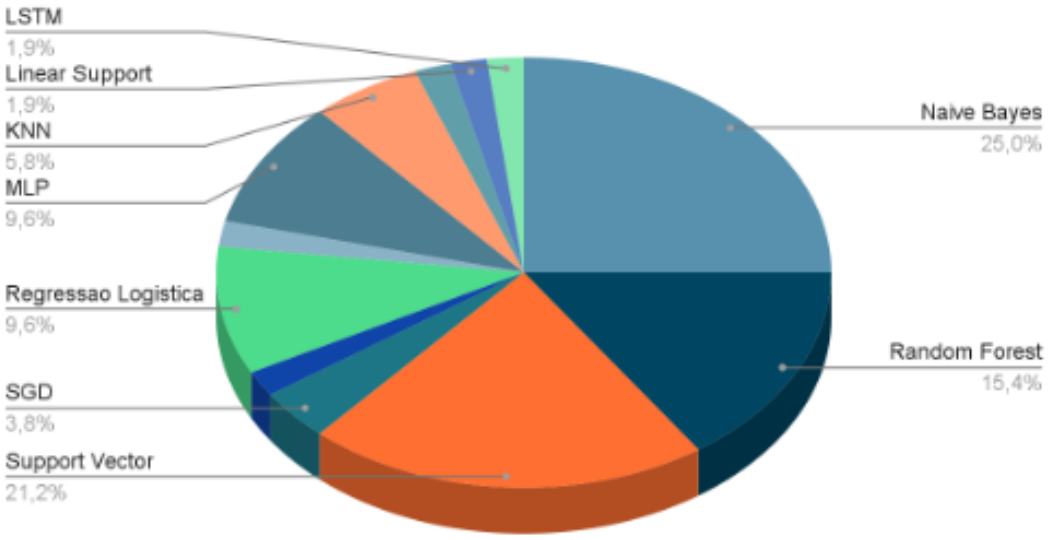


Figura 9 – Algoritmos de PLN mais utilizados para efetuar analise de sentimentos.

comparações entre si?", foi adicionado mais um critério de inclusão: apenas trabalhos que exercem comparação de algoritmos. Com isso, dos 22 trabalhos utilizados em questões anteriores, restaram apenas 11. Para encontrar a resposta, utilizou-se uma métrica de valor agregado sob a presença, ou seja, anota-se quantas vezes o algoritmo aparece em algum tipo de comparação, após isso, é necessário encontrar em quantos desses trabalhos o algoritmo foi tido como o de maior acurácia pelo pesquisador do estudo.

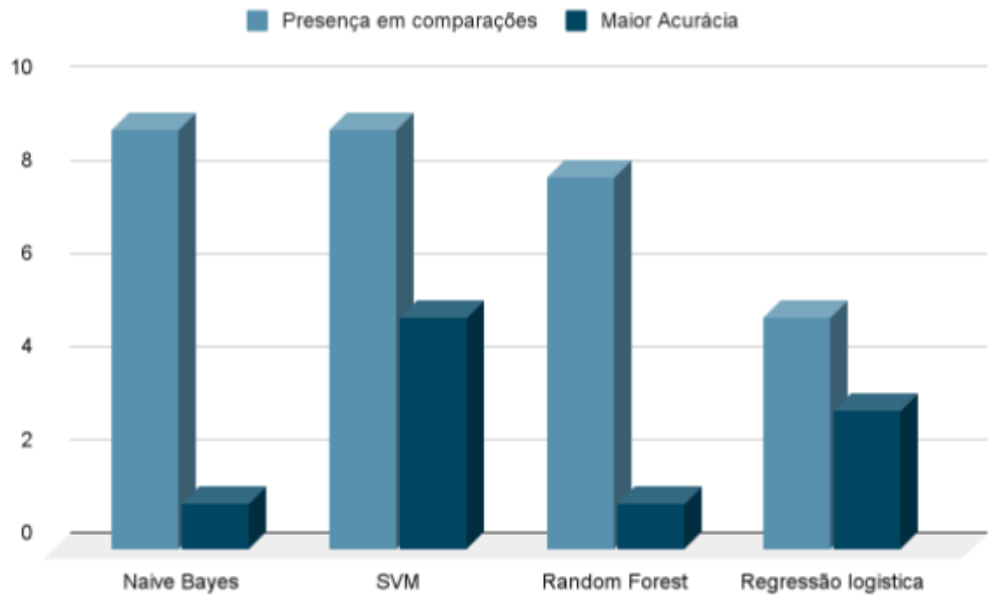


Figura 10 – Algoritmos de PLN com maior acurácia em comparações entre si.

Em conformidade a figura 10, nota-se que o algoritmo *Naive Bayes* aparece em nove dos 11 artigos, mas apesar de sua enorme participação, ele foi considerado o de melhor acurácia em apenas um dos artigos em que esteve presente. Realizando uma divisão de

presença/maior acurácia, ficou com uma porcentagem de 11,11%. Isso significa que em apenas 11,11% das comparações em que o *Naive Bayes* foi utilizado, ele apresentou a maior acurácia.

Já o algoritmo *Support Vector Machine (SVM)*, presente também em nove artigos, obteve maior acurácia em cinco dos artigos selecionados, sendo muito mais efetivo do que o *Naive Bayes*. Ao calcular a porcentagem de vezes em que o SVM apresentou maior acurácia em relação aos outros métodos, constatou-se que essa porcentagem foi de 55%.

O algoritmo *Random Forest* vem um pouco mais atrás, pois está presente em oito artigos e é considerado o mais efetivo em apenas um. Ao calcular a porcentagem de vezes em que o *Random Forest* obteve a maior acurácia em comparação com outros métodos, chegou-se a uma taxa de 12,5%.

O algoritmo de Regressão Logística aparece como uma grande surpresa, apesar de ser pouco utilizado se comparado aos outros, ele apresenta um cálculo de três vitórias em cinco participações, ou seja, em 60% das vezes em que está envolvido em comparações, ele lidera com a melhor acurácia.

4 Materiais e Métodos

Este capítulo apresenta os materiais e métodos utilizados durante este trabalho. O trabalho foi feito usando uma metodologia experimental de aspecto quantitativo (GIL, 1987), com o objetivo central de realizar a identificação de sentimentos positivos e negativos através da análise de comentários e avaliações em uma plataforma de *e-commerce* por diferentes algoritmos em diferentes situações. São apresentados aqui os métodos utilizados para a coleta de dados, pré-processamento e execução dos algoritmos selecionados. As tecnologias usadas para esse trabalho estão descritas na tabela 1.

4.1 Tecnologias utilizadas

A linguagem de programação adotada neste estudo foi o Python, escolhida devido à sua consolidação como a linguagem padrão para ciência de dados e aprendizado de máquina nos últimos anos, conforme confirmado pela Revisão Sistemática da Literatura (RSL) apresentada na seção 3.4. Além disso, o Python é conhecido por abrigar um extenso ecossistema de bibliotecas especializadas que desempenham um papel fundamental nesse campo, como o *Scikit-Learn*, o NLTK e a biblioteca *Transformers* do projeto *HugginFace*.

O *Scikit-Learn*[1] é uma biblioteca em Python especializada em aprendizado de máquina, fornecendo recursos como limpeza de dados, algoritmos, construção de *pipelines* e convenções padronizadas de nomenclatura (PEDREGOSA et al., 2011). Devido à ênfase do Python em facilidade de uso em vez de desempenho bruto, o *Scikit-Learn* e outras bibliotecas científicas frequentemente se apoiam na biblioteca *Pandas* para executar suas operações internas. O *Pandas*[2] é uma biblioteca otimizada para desempenho que oferece funcionalidades fundamentais para análise de dados.

O NLTK[3], abreviação de *Natural Language Toolkit* ou Conjunto de Ferramentas de Linguagem Natural, é uma biblioteca especializada na manipulação de linguagem humana. Ele disponibiliza uma ampla gama de funções, incluindo classificação, tokenização, *stemming* e marcação de classes gramaticais, sendo amplamente utilizado para tarefas de pré-processamento de texto que antecedem a aplicação de modelos de aprendizado de máquina em dados textuais.

A biblioteca *imbalanced-learn*[4] complementa o *Scikit-Learn*, expandindo suas capacidades para tratar conjuntos de dados desbalanceados. Ela oferece funções que permitem

[1] https://scikit-learn.org/
[2] https://pandas.pydata.org/
[3] https://www.nltk.org/
[4] https://imbalanced-learn.org/

Tabela 1 – Tecnologias utilizadas no trabalho

Ferramenta/Tecnologia	**Versão**	**Finalidade**
Python	3.10.12	Linguagem de programação
Google Colab	2023	Serviço de SaaS da Google para Jupyter notebooks
Scikit-Learn	1.2.2	Biblioteca em Python para aprendizado de máquina
Pandas	1.5.3	Biblioteca em Python para análise de dados
NLTK	3.8.1	Biblioteca em Python para PLN
imbalanced-learn	0.10.1	Biblioteca em Python para balanceamento de dados
Transformers	4.34.0	Biblioteca com modelos de aprendizado profundo
BERT	2023	Modelo de aprendizado profundo para PLN
BERTimbau	2023	Modelo pré-treinado em português

aumentar dados em conjuntos pequenos ou equilibrar diferentes classes por meio de diversas estratégias, como replicação ou redução aleatória dos dados.

A biblioteca *Transformers*[5], por sua vez, oferece uma ampla seleção de modelos pré-treinados que podem ser aplicados em tarefas variadas, incluindo não só o processamento de texto, mas também imagens e áudio. Sua flexibilidade permite o ajuste fino dos modelos para tarefas específicas, economizando tempo e recursos, uma vez que evita a necessidade de treinamento do zero, o que torna o *Transformers* uma biblioteca poderosa para acelerar o desenvolvimento de soluções em aprendizado de máquina.

Para este trabalho, foi escolhido um modelo pré-treinado do BERT com um vasto conjunto de textos em português chamado BERTimbau[6] (SOUZA, 2020) que também está disponível para ser usado através da biblioteca *Transformers*.

4.2 Plataforma de execução

O serviço usado para executar todos os códigos dessa pesquisa foi o *Google Colab*[7], ou *Google Colaboratory*. O *Google Colab* é um serviço *online* para disseminação de conhecimento em aprendizado de máquina e redes neurais. Ele oferece, de forma gratuita, um ambiente onde é possível codificar e executar códigos em Python na nuvem com recursos como CPU e GPU providos pela empresa, além de funcionalidades presentes no Google Docs como compartilhamento e edição em grupo. (CARNEIRO et al., 2018).

A máquina virtual padrão oferecida pela Google de forma gratuita possui em geral 12.7 GB de RAM e um processador Intel(R) Xeon(R) CPU, com 2,20 GHz. No caso do BERT, foi usada também uma GPU NVidia Tesla T4 com 16 GB de RAM.

Para esta pesquisa, foram criados dois projetos no *Google Colab*, um para os algoritmos convencionais presentes no *Scikit-Learn* e outro para o BERT, visto que tanto o código

[5] https://huggingface.co/transformers
[6] https://github.com/neuralmind-ai/portuguese-bert
[7] https://colab.google/

como o pré-processamento diferem em ambos. Todos os códigos estão disponíveis por completo no GitHub[8].

4.3 Conjunto de dados

Dado que o escopo deste trabalho tem como objetivo a investigação de sentimentos em avaliações de produtos, utilizou-se como fonte de dados o B2W-Reviews01 disponibilizado pelas Lojas Americanas no GitHub[9], que foi escolhido após uma breve pesquisa de quais opções estavam disponíveis. Este conjunto de dados possui mais de 130 mil avaliações de produtos, cada uma classificada em uma escala de um a cinco, juntamente com o respectivo texto da avaliação. Os dados foram coletados durante todo o período de janeiro até maio de 2018 e inclui também linguagem ofensiva que normalmente teria sido excluída do site da loja (REAL; OSHIRO; MAFRA, 2019).

A escolha do conjunto de dados B2W-Reviews01 foi motivada não apenas pela sua relevância para o propósito do estudo, mas também pelo formato dos dados. A disponibilização dos dados em CSV (*Comma-Separated Values*) tornou-o conveniente para manipulação e processamento, permitindo uma rápida iteração durante a etapa de exploração.

Além da nota atribuída à experiência de compra na loja (que varia de um a cinco) e o texto da avaliação em si, os dados presentes no conjunto possuem outras colunas com dados adicionais, como nome, marca e categoria do produto. Adicionalmente, o conjunto também engloba informações demográficas dos usuários que deixaram as avaliações, tais como data de nascimento, gênero e estado de origem.

Vale ressaltar que, embora o conjunto de dados contenha várias colunas com informações adicionais, foi observado que várias entradas apresentam valores nulos em algumas dessas colunas, e alguns poucos textos da avaliação são vazios.

Listing 4.1 – Código para obter informações sobre o conjunto de dados

```
import pandas as pd

df = pd.read_csv(
    "/content/drive/MyDrive/Datasets/B2W-Reviews01.csv",
    parse_dates=["submission_date"], low_memory=False
)
print(df.info())
```

Analisando a distribuição das avaliações por número de estrelas, temos o seguinte: 1 estrela com 20,68%, 2 estrelas com 6,24%, 3 estrelas como 12,33%, 4 estrelas com 24,43% e 5 estrelas com 36,23%.

[8] https://github.com/dielsonsales/TCC-2023
[9] https://github.com/americanas-tech/b2w-reviews01

Trabalhos anteriores na literatura recomendam o uso de uma e duas estrelas para avaliações negativas e apenas cinco estrelas para avaliações positivas (NOBRE et al., 2016), apesar de essa decisão ter se baseado mais na necessidade de balancear os dados, dado que o conjunto de dados possuía um número muito grande de avaliações com cinco estrelas (RAIN, 2013).

O conjunto de dados presente mostra que entre as avaliações com quatro estrelas, 98,42% dos usuários (ou seja, 31.837 de um total de 32.345) recomendariam o produto para um amigo. Logo, para este estudo, foram descartadas apenas as avaliações com três estrelas (consideradas neutras) e avaliações com uma e duas estrelas foram consideradas negativas, enquanto avaliações com quatro e cinco estrelas foram consideradas positivas.

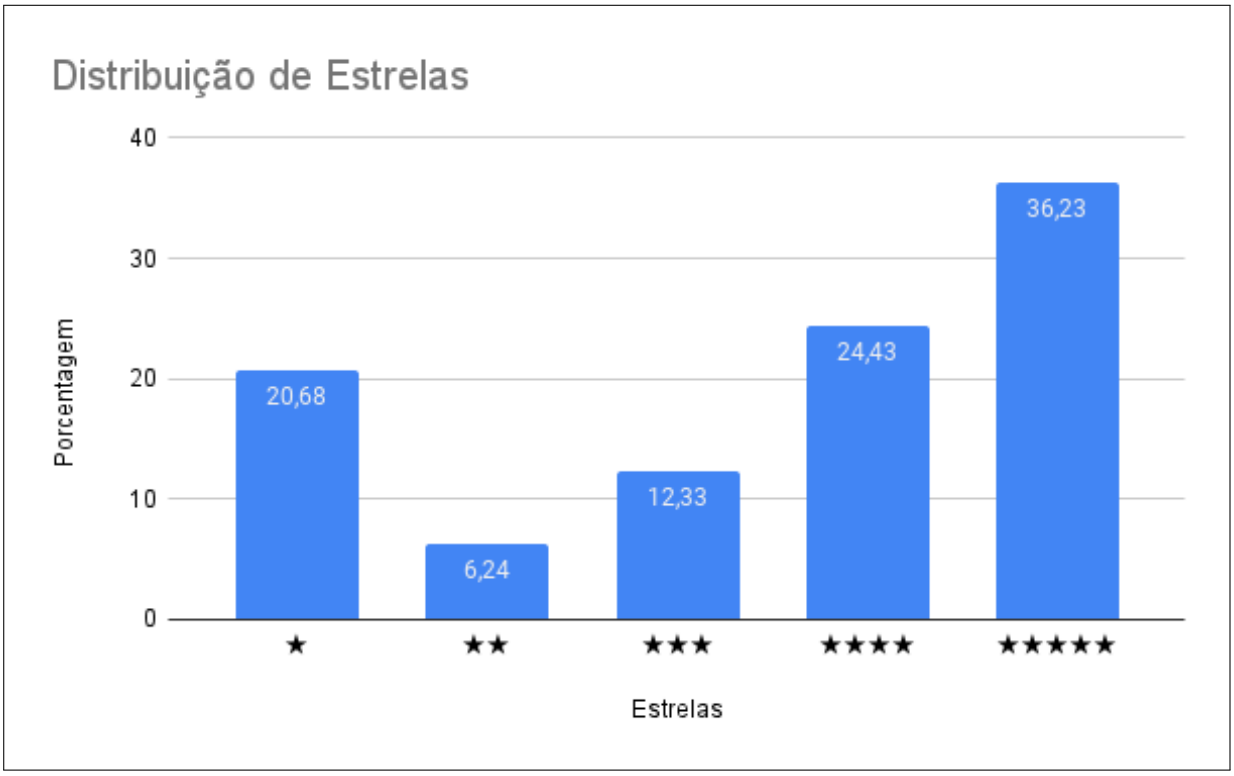


Figura 11 – Distribuição de estrelas no conjunto de dados B2W-Reviews01

4.4 Pré-processamento dos dados

Ao lidar com textos como dados de entrada, é uma prática comum na literatura, especialmente ao utilizar algoritmos estatísticos, realizar previamente uma etapa de "limpeza" dos dados. O principal objetivo é simplificar o texto e possivelmente reduzir a dimensionalidade dos dados.

Nesse trabalho, foram selecionadas duas etapas de pré-processamento dos dados, que estão disponíveis na biblioteca NLTK:

- Remoção de *stop words*;

Figura 12 – Nuvem de palavras para avaliações positivas

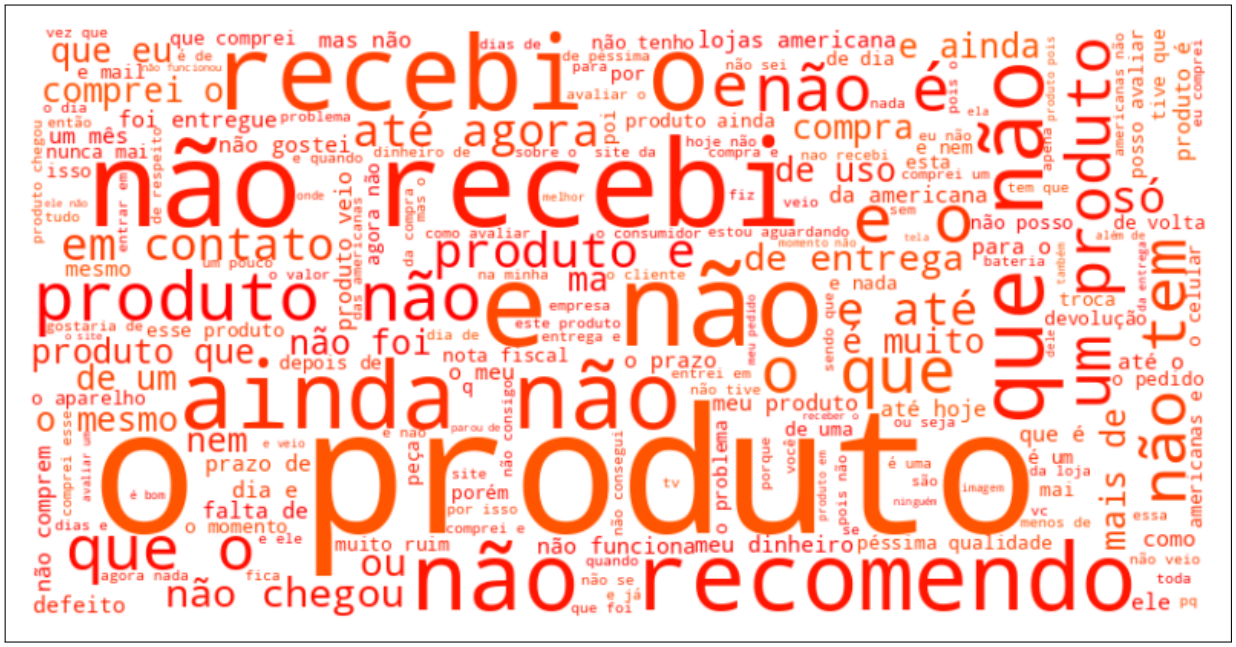


Figura 13 – Nuvem de palavras para avaliações negativas

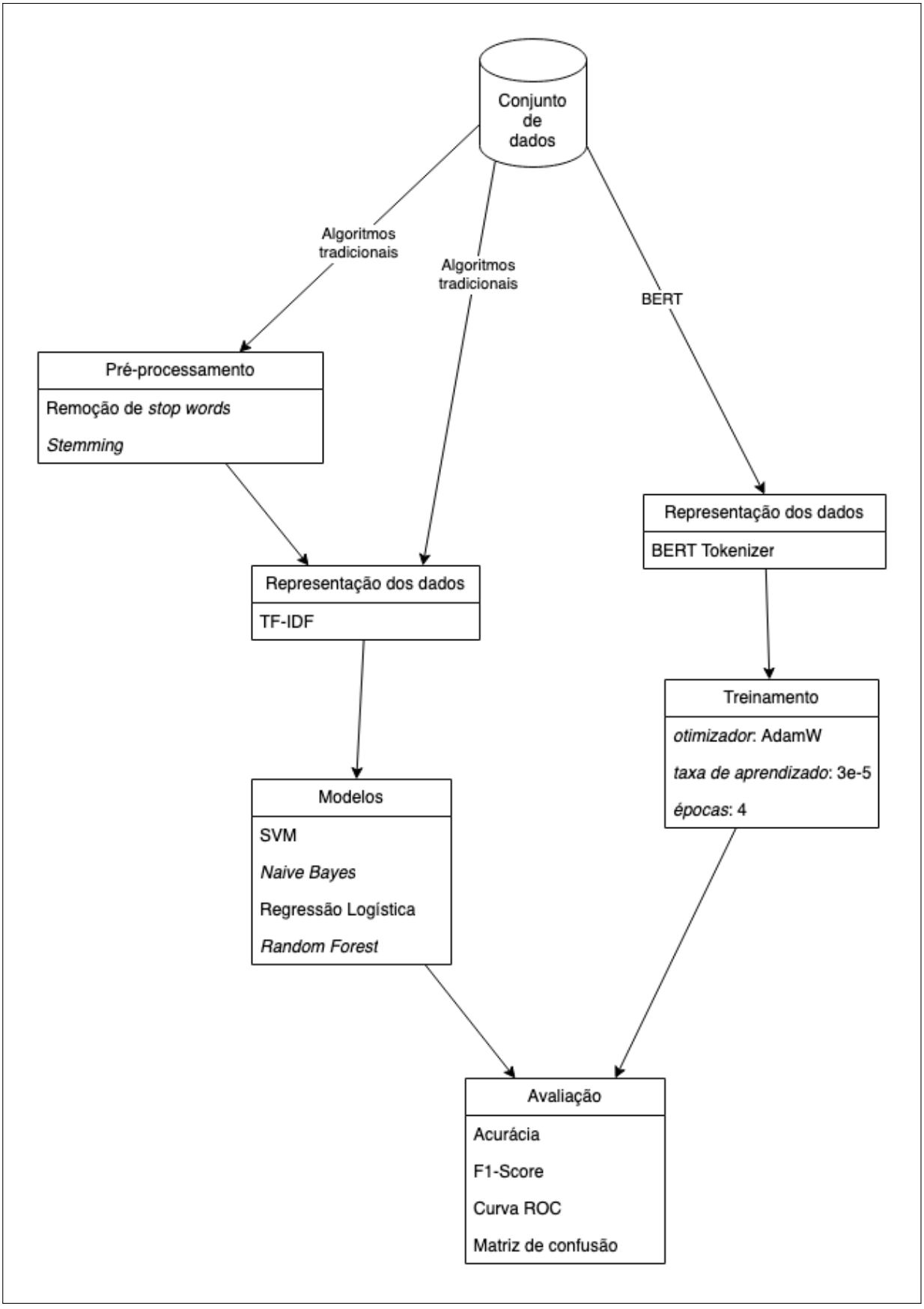


Figura 14 – Etapas do processamento

Tabela 2 – Exemplo de textos após pré-processamento

produt bom qq tip cabel
samsung 32 pol model 2008 séri 5 ach la excel ...
recom produt outr pesso excel qual entreg dent...
consider aparelh melhor cust benefíci
produt realiz propõ apresent mass panquec pass...
compr esp relogi lind poi ja pequen agor grand...
mor maranh produt flutu pq dia 30 mai cheg che...
refil filtr vei cabeçot limp compr outr loj en...
compr 3 vei lasc revest trê med faz escorreg c...
menos seman uso apresent problem trac digit ap...

- Aplicação de *stemming*.

A classe `RSLPStemmer` da biblioteca NLTK é responsável pela aplicação do algoritmo de *stemming*, que por sua vez se baseia no artigo de (ORENGO; HUYCK, 2001).

Listing 4.2 – Código para pré-processamento dos textos de avaliações

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import RSLPStemmer
from nltk.tokenize import RegexpTokenizer

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('rslp')

portuguese_stopwords = set(stopwords.words('portuguese'))

def preprocess(text):
  tokenizer = RegexpTokenizer(r'\w+')
  words = tokenizer.tokenize(text)
  stemmer = RSLPStemmer()
  filtered_words = [word for word in words if word.lower() not in
     portuguese_stopwords]
  stemmed_words = [stemmer.stem(word) for word in filtered_words]
  return ' '.join(stemmed_words)


samples[PROCESSED_TEXT] = samples[REVIEW_TEXT].apply(preprocess)
```

Como definido na seção 4.3, avaliações com uma e duas estrelas foram consideradas negativas, enquanto avaliações com quatro e cinco estrelas foram consideradas positivas. Para traduzir isso para que os algoritmos possam trabalhar de forma simplificada, foi utilizado o padrão de valor 0 para avaliações negativas e 1 para avaliações positivas.

Listing 4.3 – Código para mapeamento dos sentimentos

```
X = samples[PROCESSED_TEXT].values
Y = samples[OVERALL_RATING]

# Maps 1 and 2 stars as negative reviews, and 4 and 5 stars as
    positive reviews
Y = Y.map({1:0, 2:0, 4: 1, 5: 1}).values
```

4.5 Balanceamento dos dados

Após o pré-processamento dos dados como um todo, foi feita uma divisão da base de dados em treino e teste (ou validação), sendo 80% dos dados separados para treino e 20% para teste.

Um processo subsequente então fez uso da biblioteca *imbalanced-learn* para equilibrar a quantidade de amostras presentes em cada classe. Como o conjunto de dados em questão possui mais classes com sentimentos positivos do que negativos, foi usada a classe `RandomUnderSampler` para reduzir a quantidade de amostras positivas até que ambas tivessem a mesma quantidade. Isso evita que os modelos desenvolvam algum viés proveniente da quantidade de dados presentes em cada classe.

Listing 4.4 – Código para balancear o conjunto de dados de treino

```
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle

TEST_PERCENTAGE = 0.2

# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=
    TEST_PERCENTAGE, random_state=42)

reshaped_X_train = X_train.reshape(-1, 1)

undersampler = RandomUnderSampler(sampling_strategy="auto",
    random_state=42)
X_train_resampled, Y_train_resampled = undersampler.fit_resample(
    reshaped_X_train, Y_train)

X_train_resampled = X_train_resampled.flatten()

X_train_resampled = shuffle(X_train_resampled, random_state=42)
Y_train_resampled = shuffle(Y_train_resampled, random_state=42)
```

4.6 Configuração dos modelos

Além do objetivo explícito de fazer uma comparação com o modelo BERT, a seleção dos demais algoritmos de Processamento de Linguagem Natural (PLN) foi fundamentada na Revisão Sistemática de Literatura (RSL) detalhada na seção 3.4. A RSL resultou na escolha de quatro algoritmos comumente usados na análise de sentimentos para serem incluídos neste estudo: Máquinas de Vetores de Suporte (SVM), *Naive Bayes*, Regressão Logística e *Random Forest*. Constatou-se também que os algoritmos selecionados já estavam incluídos na biblioteca *Scikit-Learn* do Python.

Para que os algoritmos de aprendizado de máquina possam lidar com o texto como entrada, foi feita a conversão do texto em cada avaliação para vetores numéricos usando o algoritmo TF-IDF. No contexto da biblioteca *Scikit-Learn*, essa conversão é feita usando a classe `TfidfVectorizer`. A escolha do algoritmo TF-IDF se justifica pelo tamanho do conjunto de dados, uma vez que ele automaticamente ajusta os pesos das palavras, evitando inflacionar excessivamente o impacto de termos muito frequentes.

Em seguida, o método `evaluate_model` é testado individualmente com cada um dos algoritmos selecionados. Quando possível, foi usado o parâmetro `n_jobs=-1` para tirar proveito de todos os núcleos do processador disponíveis.

Listing 4.5 – Configuração e execução dos algoritmos tradicionais

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
    recall_score, f1_score, roc_curve, roc_auc_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC

vectorizer = TfidfVectorizer()

X_train_vectorized = vectorizer.fit_transform(X_train_resampled)
X_test_vectorized = vectorizer.transform(X_test)

def evaluate_model(model_name, model_instance):
  # Train the model
  model_instance.fit(X_train_vectorized, Y_train_resampled)
  # Make predictions on the test data
  Y_pred = model_instance.predict(X_test_vectorized)

  accuracy = accuracy_score(Y_test, Y_pred)
  recall = recall_score(Y_test, Y_pred, average="macro")
  f1 = f1_score(Y_test, Y_pred, average="macro")
```

```
    fpr, tpr, _ = roc_curve(Y_test, Y_pred)
    roc_auc = roc_auc_score(Y_test, Y_pred)

    plot_roc_curve(fpr, tpr, roc_auc)

    data = confusion_matrix(Y_test, Y_pred)
    data.astype(int)
    plot_confusion_matrix(data, labels=["negative", "positive"])


evaluate_model("Support Vector Machines", SVC(kernel="linear"))
evaluate_model("Naive Bayes", MultinomialNB())
evaluate_model("Logistic Regression", LogisticRegression(n_jobs=-1))
evaluate_model("Random Forest", RandomForestClassifier(n_jobs=-1))
```

4.7 Implementação do BERT

Esse trabalho fez uso de um modelo do BERT pré-treinado num corpus em português, denominado BERTimbau (SOUZA, 2020) e faz uso do da biblioteca *transformers* disponibilizada pela plataforma *Hugging Face*[10]. A implementação do código baseou-se largamente em exemplos disponíveis na plataforma *Kaggle* e em outros tutoriais *online*[11,12]. *Kaggle* é uma comunidade focada no compartilhamento de projetos de ciência de dados no geral, além de competições focadas em soluções para problemas que envolvem grandes quantidades de dados. O *Hugging Face*, por sua vez, é uma comunidade focada na disseminação de modelos de aprendizado de máquina que oferece funcionalidades análogas ao *GitHub* para modelos de aprendizado profundo. Os códigos usados neste trabalho foram modificado e adaptados para satisfazer as necessidades do estudo em questão.

O modelo sofreu um ajuste fino usando o conjunto de dados de avaliações de produtos para predizer o sentimento de cada avaliação. Para isso, foi usado o otimizador `AdamW` com uma taxa de aprendizado de `3e-5` e um total de quatro épocas, como recomendado pelos autores (KENTON; TOUTANOVA, 2019) e geralmente usado nos tutoriais.

O código completo da implementação do BERT encontra-se disponível no apêndice deste trabalho.

4.8 Treinamento dos modelos

Dois projetos foram desenvolvidos no ambiente do *Google Colab*, abordando distintas abordagens para a resolução do problema em estudo. O primeiro projeto adotou uma

[10] https://huggingface.co/
[11] https://skimai.com/fine-tuning-bert-for-sentiment-analysis/
[12] https://www.kaggle.com/code/hoshi7/bert-classify-news-sentiment/notebook

abordagem clássica, empregando a biblioteca *Scikit-Learn* para a implementação dos algoritmos tradicionais, enquanto o segundo projeto fez uso da biblioteca *transformers* para a implementação do BERT.

Com o intuito de avaliar o impacto da variação na quantidade de dados utilizados nos experimentos, foram realizadas duas configurações diferentes. A primeira configuração adotou uma seleção aleatória de 15 mil amostras, enquanto a segunda configuração reduziu ainda mais o tamanho da amostra, selecionando apenas 3 mil amostras. Para isso, foi usado a função `sample` da biblioteca *Scikit-Learn*.

Após a etapa inicial de divisão dos dados, na qual 80% das amostras foram destinadas ao treinamento dos modelos, procedeu-se ao balanceamento das classes positivas e negativas. Isso resultou em conjuntos de treinamento compostos por 7.060 amostras para a primeira configuração e 1.378 amostras para a segunda configuração.

Em ambas as configurações, os testes foram conduzidos tanto com o pré-processamento do texto quanto sem ele, no caso dos algoritmos tradicionais. Já o modelo BERT foi treinado sem qualquer modificação no texto, uma vez que os modelos baseados em *Transformers* se beneficiam da preservação de um maior contexto textual e não requerem simplificações no texto de entrada, conforme evidenciado por estudos anteriores com o BERT em outros idiomas (HUSAIN; UZUNER, 2022).

5 Resultados e Discussões

Este capítulo apresenta e discute os resultados obtidos nesse trabalho através da metodologia descrita no capítulo 4 para análise de sentimentos em avaliações de produtos *online* na língua portuguesa.

Avaliamos os modelos convencionais SVM, *Naive Bayes*, Regressão Logística e *Random Forest*, juntamente com o modelo de linguagem BERT. Nosso objetivo é fornecer uma visão completa de como esses modelos se comportam em termos de desempenho e eficiência quando aplicados ao contexto deste trabalho.

5.1 Resultados com 7.060 amostras

Nos experimentos com a seleção inicial de 15 mil amostras, o balanceamento das classes criou um conjunto de treinamento com 7.060 amostras. Os algoritmos foram testados com e sem pré-processamento dos textos, com exceção do BERT que apenas foi testado sem qualquer pré-processamento. A acurácia dos algoritmos é mostrada na tabela 3 e o gráfico com as comparações é mostrado na figura 15:

Assim como a acurácia, foi também computado o tempo de execução de cada algoritmo, levando em consideração que no caso do BERT foi computado o tempo de refinamento (*fine-tuning*) do modelo com os dados de treinamento. A tabela 5 apresenta o tempo de execução de cada algoritmo.

Tabela 3 – Acurácia dos algoritmos com 7.060 amostras

Modelo	**Sem pré-processamento**	**Com pré-processamento**
SVM	**90,97%**	**89,67%**
Naive Bayes	89,77%	89,43%
Regressão Logística	90,30%	89,43%
Random Forest	88,03%	89,40%
BERT	**96,10%**	

Tabela 4 – *F1-Score* dos algoritmos com 7.060 amostras

Modelo	**Sem pré-processamento**	**Com pré-processamento**
SVM	**89,80%**	**88,36%**
Naive Bayes	88,60%	88,13%
Regressão Logística	89,11%	88,19%
Random Forest	86,69%	88,03%
BERT	**95,46%**	

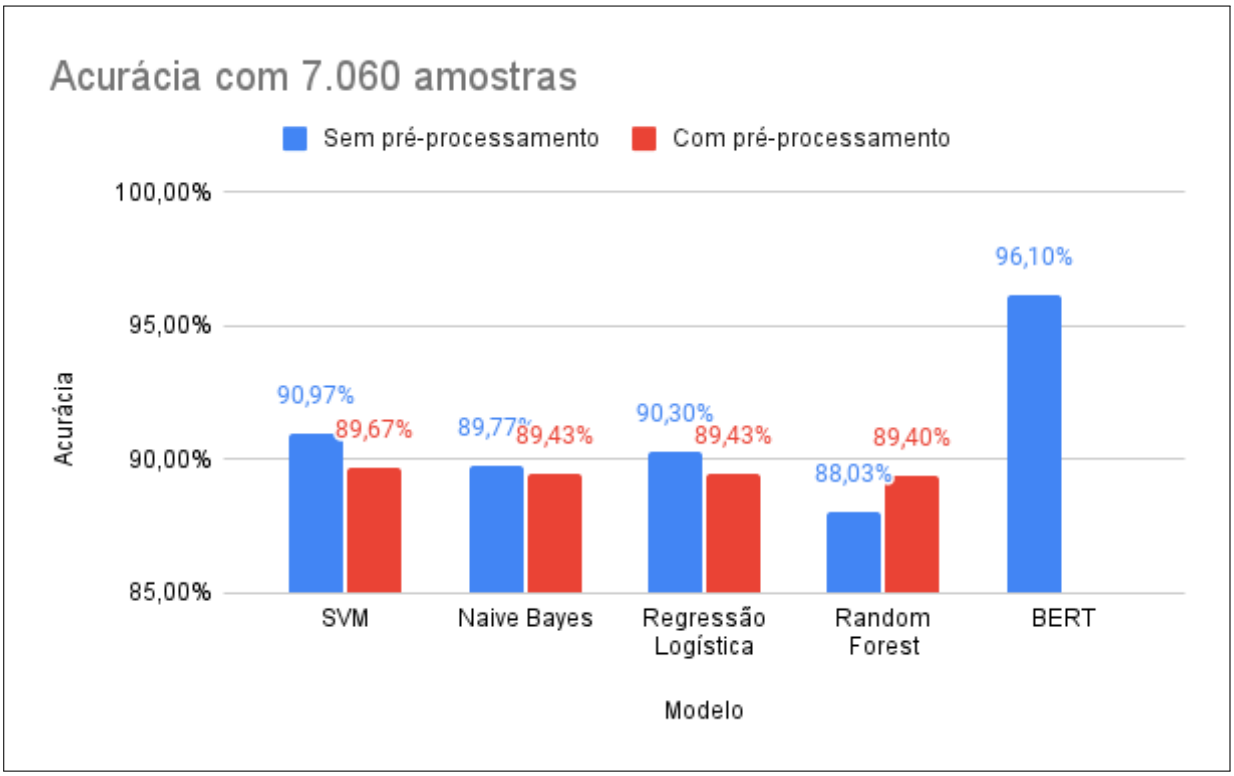


Figura 15 – Acurácia com 7.060 amostras

Tabela 5 – Tempo de treinamento e execução dos algoritmos com 7.060 amostras, em segundos

Modelo	**Sem pré-processamento**	**Com pré-processamento**
SVM	16,20	5,23
Naive Bayes	0,05	0,03
Regressão Logística	2,86	1,70
Random Forest	17,47	4,50
BERT	2909,91	

5.2 Resultados dos experimentos com 1.378 amostras

Nos experimentos com a seleção inicial de 3 mil amostras, o balanceamento das classes criou um conjunto de treinamento com 1.378 amostras. Assim como no experimento anterior, os algoritmos foram testados com e sem pré-processamento dos textos, e da mesma forma o BERT foi testado sem qualquer pré-processamento. A acurácia dos algoritmos é mostrada na tabela 6 e o gráfico com as comparações é mostrado na figura 16:

O tempo de execução de cada algoritmo é mostrado na tabela 8.

5.3 Acurácia dos modelos

Primeiramente, analisamos a acurácia alcançada pelos diferentes modelos em ambas as configurações de dados.

Tabela 6 – Acurácia dos algoritmos com 1.378 amostras

Modelo	Sem pré-processamento	Com pré-processamento
SVM	**90,17%**	**89,00%**
Naive Bayes	87,17%	88,67%
Regressão Logística	88,17%	88,67%
Random Forest	86,67%	88,83%
BERT	**95,00%**	

Tabela 7 – *F1-Score* dos algoritmos com 1.378 amostras

Modelo	Sem pré-processamento	Com pré-processamento
SVM	**89,03%**	**87,46%**
Naive Bayes	85,75%	87,15%
Regressão Logística	86,89%	87,34%
Random Forest	85,04%	87,26%
BERT	**94,14%**	

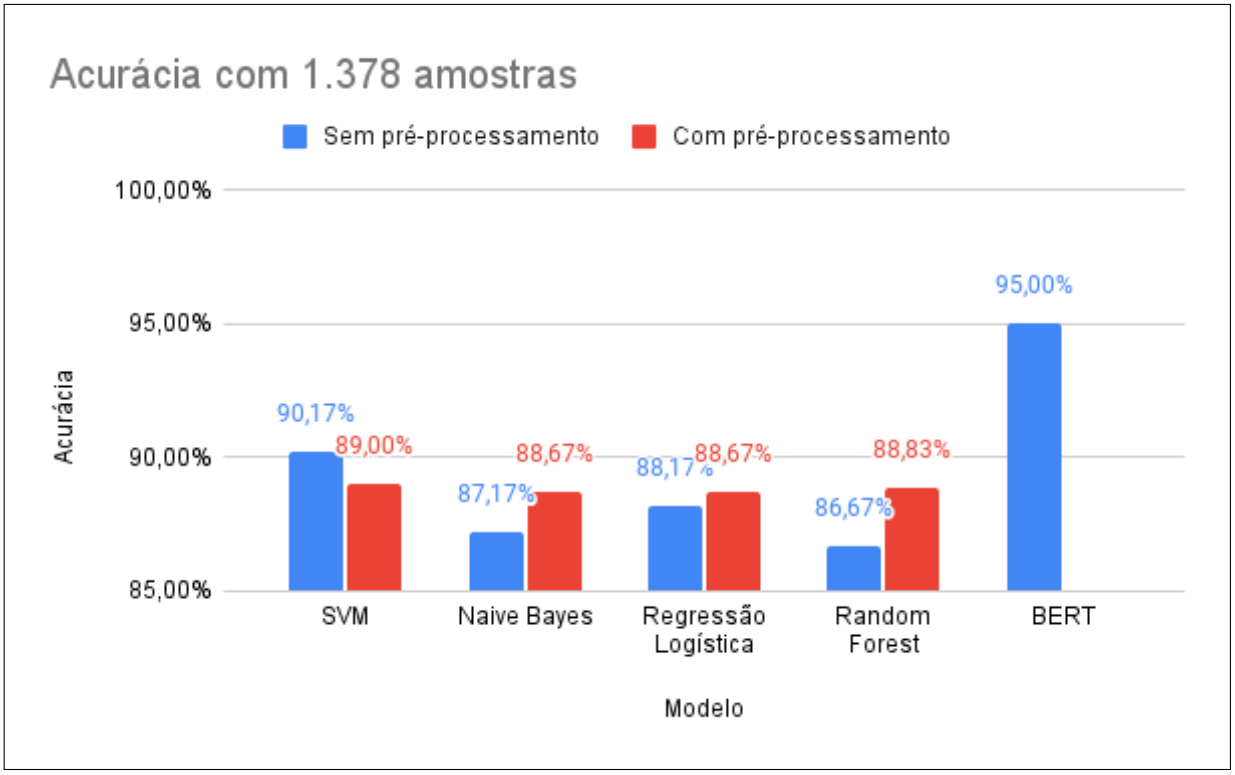


Figura 16 – Acurácia com 1.378 amostras

Tabela 8 – Tempo de treinamento e execução dos algoritmos com 1.378 amostras, em segundos

Modelo	Sem pré-processamento	Com pré-processamento
SVM	0,62	0,33
Naive Bayes	0,01	0,04
Regressão Logística	1,11	1,84
Random Forest	0,67	0,82
BERT	608,33	

Na configuração com 7.060 amostras, observamos que o modelo BERT superou os outros algoritmos em termos de acurácia, atingindo uma taxa de acerto de 96,10%. Entre os modelos tradicionais, o SVM apresentou a segunda melhor performance, com uma acurácia de 90,97% quando não houve pré-processamento de texto. É interessante notar que o pré-processamento de texto parece ter um impacto ligeiramente negativo na acurácia de todos os algoritmos, com exceção do *Random Forest*.

Na configuração com 1.378 amostras, o BERT novamente obteve uma acurácia muito acima dos outros, com 95,00%. O SVM foi novamente o segundo melhor modelo, com 90,17% de acurácia sem pré-processamento. Diferente da configuração anterior, o pré-processamento melhorou a performance dos algoritmos *Naive Bayes*, Regressão Logística e *Random Forest*.

5.4 *F1-Score*

O F1-Score, que combina precisão e *recall*, é uma métrica crítica para avaliar o desempenho de classificadores em problemas de classificação binária.

Os resultados mostram que o desempenho dos algoritmos varia em função do tamanho do conjunto de dados e da aplicação de pré-processamento. O BERT novamente se destacou com os maiores *F1-Scores*, indicando sua eficácia em tarefas de classificação de texto. Os modelos de aprendizado de máquina tradicionais, como SVM, *Naive Bayes* e Regressão Logística, também demonstraram desempenho sólido, com variações sutis em relação ao pré-processamento.

No geral, na configuração de 7.060 amostras, os algoritmos se saírem melhor sem qualquer forma de pré-processamento do texto, com exceção do *Random Forest*, que foi o único que apresentou melhoria após o pré-processamento em todas as configurações.

Na configuração de 1.378 amostras, todos os algoritmos apresentaram uma performance inferior, com o BERT novamente à frente, com 94,14%, e o SVM em segundo lugar, com 89,03% nos testes sem pré-processamento. Numa situação similar à acurácia, o pré-processamento aqui melhorou os algoritmos *Naive Bayes*, Regressão Logística e *Random Forest*.

5.5 Tempo de treinamento e execução

Avaliamos também o tempo necessário para treinar e executar cada algoritmo em ambas as configurações de dados.

Em ambas as configurações, com 7.060 amostras e 1.378 amostras, observamos que o BERT demandou um investimento significativamente maior de tempo tanto para o treinamento quanto para a execução, quando comparado aos modelos convencionais. Enquanto os modelos convencionais foram capazes de realizar essas tarefas em questão de segundos

ou milissegundos, o BERT demandou, respectivamente, 2.909,91 segundos (equivalente a aproximadamente 48 minutos) e 608,33 segundos (cerca de 10 minutos) para realizar o treinamento.

Vale destacar que uma particularidade do BERT é a possibilidade de salvar o estado do modelo, conhecido como "*check-point*", após o primeiro treinamento. Isso quer dizer que, em usos subsequentes, é possível carregar as configurações previamente treinadas do modelo sem a necessidade de retreiná-lo, a menos que sejam necessárias personalizações específicas para cenários particulares. Esse aspecto demonstra a flexibilidade e a eficiência potencial do BERT e outros modelos de aprendizado profundo em contextos nos quais os tempos de treinamento mais longos podem ser gerenciados com estratégias de otimização.

5.6 Pré-processamento

Observamos que o pré-processamento de texto, embora não tenha demonstrado uma melhora substancial no desempenho dos modelos tradicionais, pode ser benéfico em contextos específicos. Notavelmente, o algoritmo *Random Forest* apresentou um desempenho ligeiramente aprimorado quando aplicado ao conjunto de dados de 7.060 amostras com o pré-processamento.

Nas configurações com 1.378 amostras, os algoritmos *Naive Bayes*, *Random Forest* e Regressão Logística exibiram leves melhorias quando o pré-processamento foi aplicado.

Esses resultados são coerentes com achados em outros estudos recentes, que também indicavam a pouca eficácia do pré-processamento nos algoritmos tradicionais (AVILA, 2017).

Interessantemente, os resultados também sugerem uma possível tendência em que o pré-processamento tende a aprimorar o desempenho dos modelos quando o conjunto de dados é de tamanho mais limitado, possivelmente ajudando a reduzir a dimensionalidade dos dados e mitigar o impacto de ruídos. No entanto, à medida que o tamanho do conjunto de dados cresce substancialmente, o pré-processamento pode, de fato, prejudicar o desempenho dos modelos, uma vez que informações valiosas podem ser perdidas durante o processo de limpeza textual.

5.7 Limitações e desafios dos modelos analisados

A análise de sentimentos em avaliações de produtos é uma tarefa complexa e desafiadora, e apesar dos *insights* obtidos nesta pesquisa, é necessário reconhecer as limitações inerentes a esta abordagem.

Uma das principais limitações reside na dificuldade em estabelecer uma correspondência perfeita entre as notas em estrelas atribuídas pelos usuários e o conteúdo textual das avaliações. Como descrito no artigo de Real, Oshiro e Mafra (2019), sobre o conjunto

B2W-Reviews01, as avaliações por vezes contêm opiniões sutis que não são refletidas de forma precisa nas classificações numéricas, o que pode levar a desafios na rotulação dos dados para treinamento dos modelos.

A linguagem informal frequentemente presente nas avaliações de produtos apresenta desafios adicionais. As opiniões dos consumidores são frequentemente expressas de maneira subjetiva, com uso de gírias, sarcasmo e ironia, o que torna a interpretação automatizada uma tarefa complexa. Sutilezas na expressão de sentimentos podem levar a ambiguidades e desafios na atribuição de polaridade correta (positiva ou negativa) aos textos.

A implementação do modelo BERT, em particular, exige recursos computacionais substanciais, como GPUs e tempo de treinamento prolongado, o que pode ser uma limitação em cenários com restrições de recursos. Em aplicações práticas, como sistemas de recomendação em tempo real, a manutenção de modelos atualizados e eficazes é uma preocupação constante. À medida que as avaliações e preferências dos consumidores evoluem, é fundamental que os modelos se adaptem para continuar oferecendo análises precisas.

5.8 Escolha do melhor modelo com base nos resultados

Após uma análise dos resultados obtidos nesta pesquisa, fica claro que não existe um modelo que se destaque como o candidato perfeito na análise de sentimentos em avaliações de produtos na língua portuguesa. Diferentes algoritmos, como o SVM, *Naive Bayes*, Regressão Logística, *Random Forest* ou o BERT, possuem suas próprias vantagens e desvantagens. A escolha do modelo ideal depende das necessidades específicas da aplicação e das limitações de recursos disponíveis.

O SVM apresentou resultados ligeiramente superiores em termos de acurácia e *F1-Score* quando comparado com outros modelos tradicionais. No entanto, essa vantagem foi relativamente modesta, e em cenários onde a eficiência computacional é crucial, os modelos tradicionais podem ser preferíveis, oferecendo um bom equilíbrio entre desempenho e recursos disponíveis.

O BERT, por outro lado, demonstrou um desempenho notavelmente superior em termos de acurácia e *F1-Score* em relação a todos os outros modelos. No entanto, essa vantagem é acompanhada por um requisito muito maior de poder computacional e tempo de treinamento, além de exigir *hardware* especializado. Logo, a escolha de adotar o BERT deve ser ponderada em relação à disponibilidade de recursos e à urgência da análise.

O estudo demonstra que à medida que se aumenta o poder computacional, a melhoria na performance dos modelos se torna progressivamente menos significativa. A decisão sobre qual algoritmo usar depende de encontrar o equilíbrio adequado entre desempenho e eficiência de modo a assegurar que o projeto possa suportar os custos associados de maneira sustentável.

Vale sempre ressaltar que, à medida que a tecnologia avança e os recursos computacionais se tornam mais acessíveis, as escolhas ideais podem evoluir ao longo do tempo. Portanto, esta decisão deve ser revisada periodicamente à medida que as condições e os objetivos da aplicação mudam.

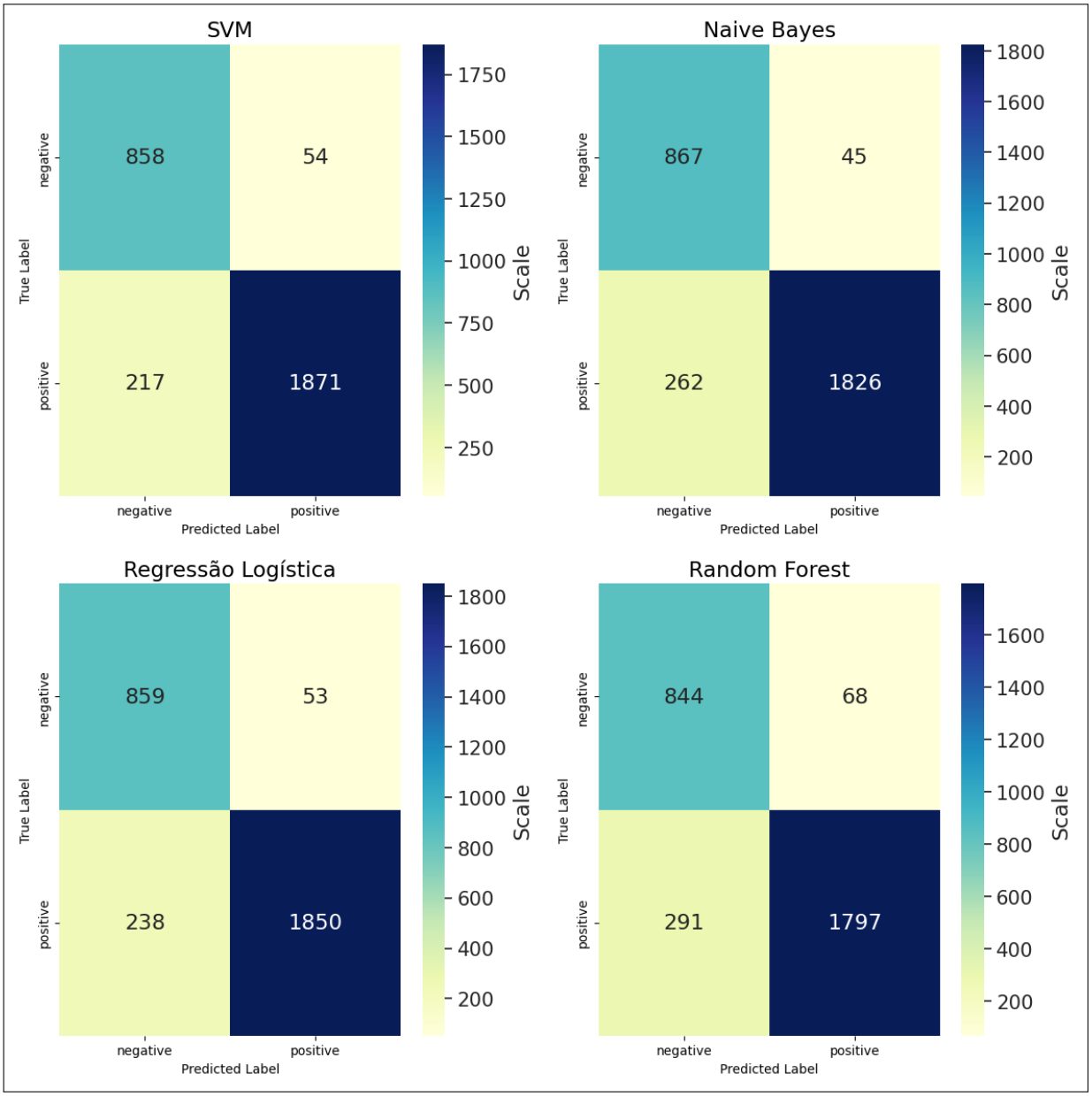


Figura 17 – Matriz de confusão dos modelos com 7.060 amostras sem pré-processamento

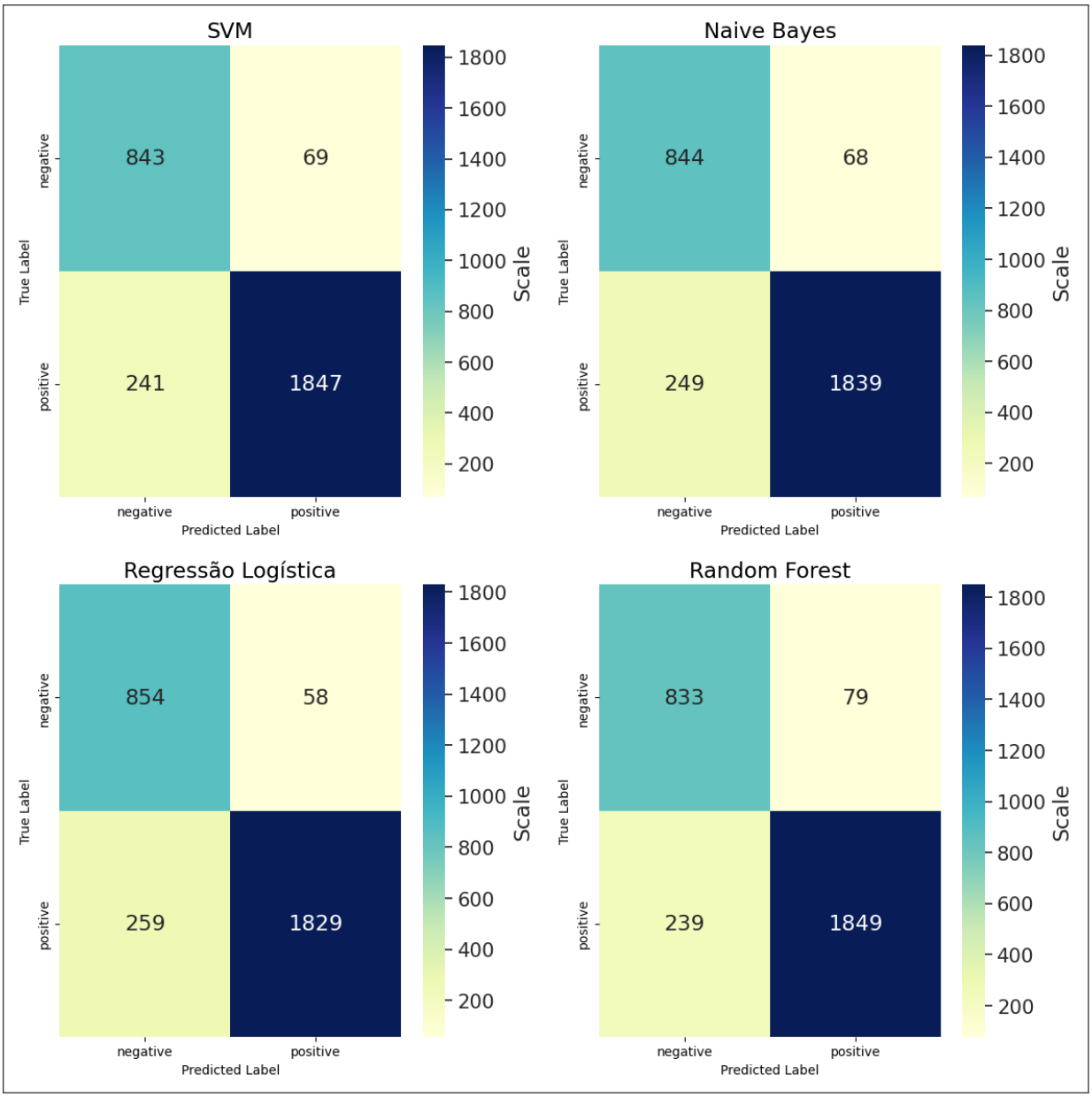


Figura 18 – Matriz de confusão dos modelos com 7.060 amostras com pré-processamento

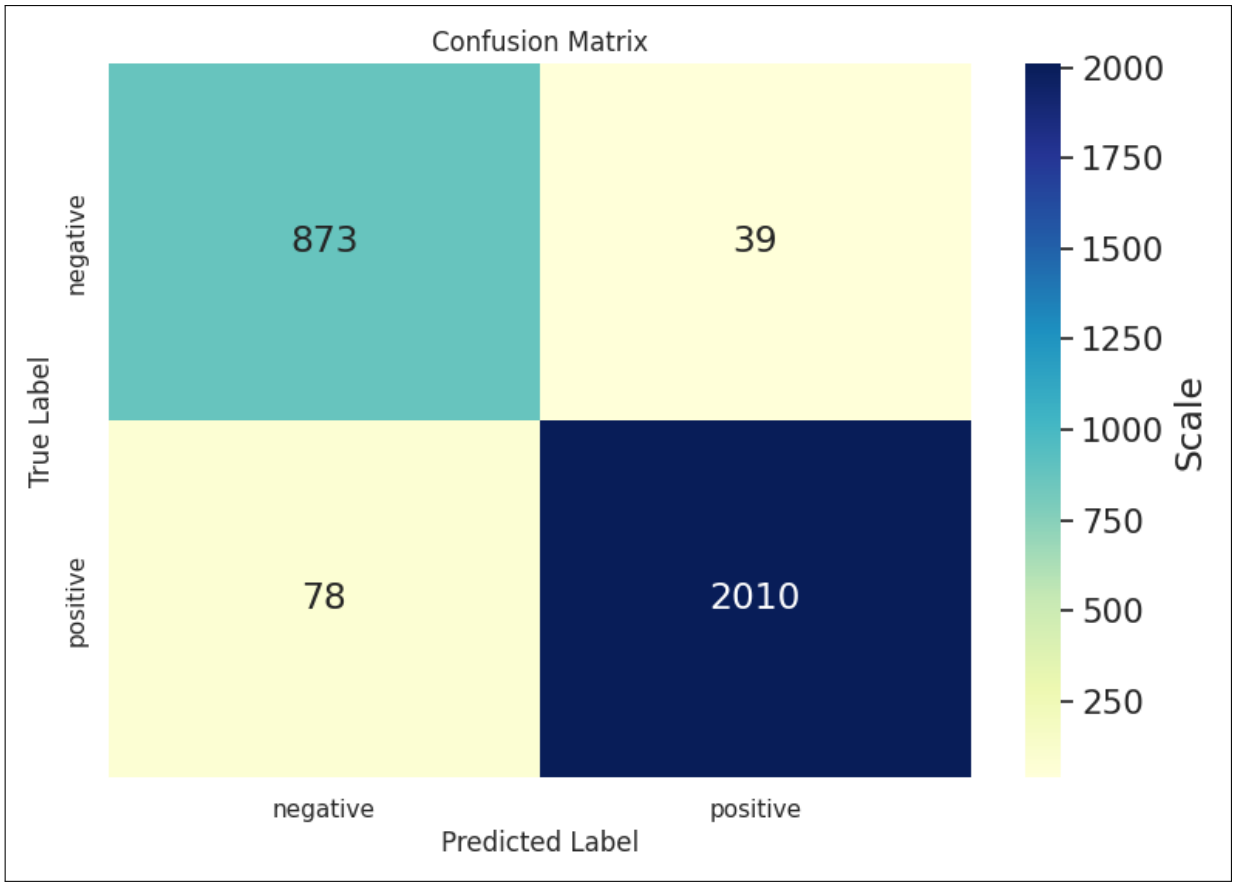


Figura 19 – Matriz de confusão do BERT com 7.060 amostras

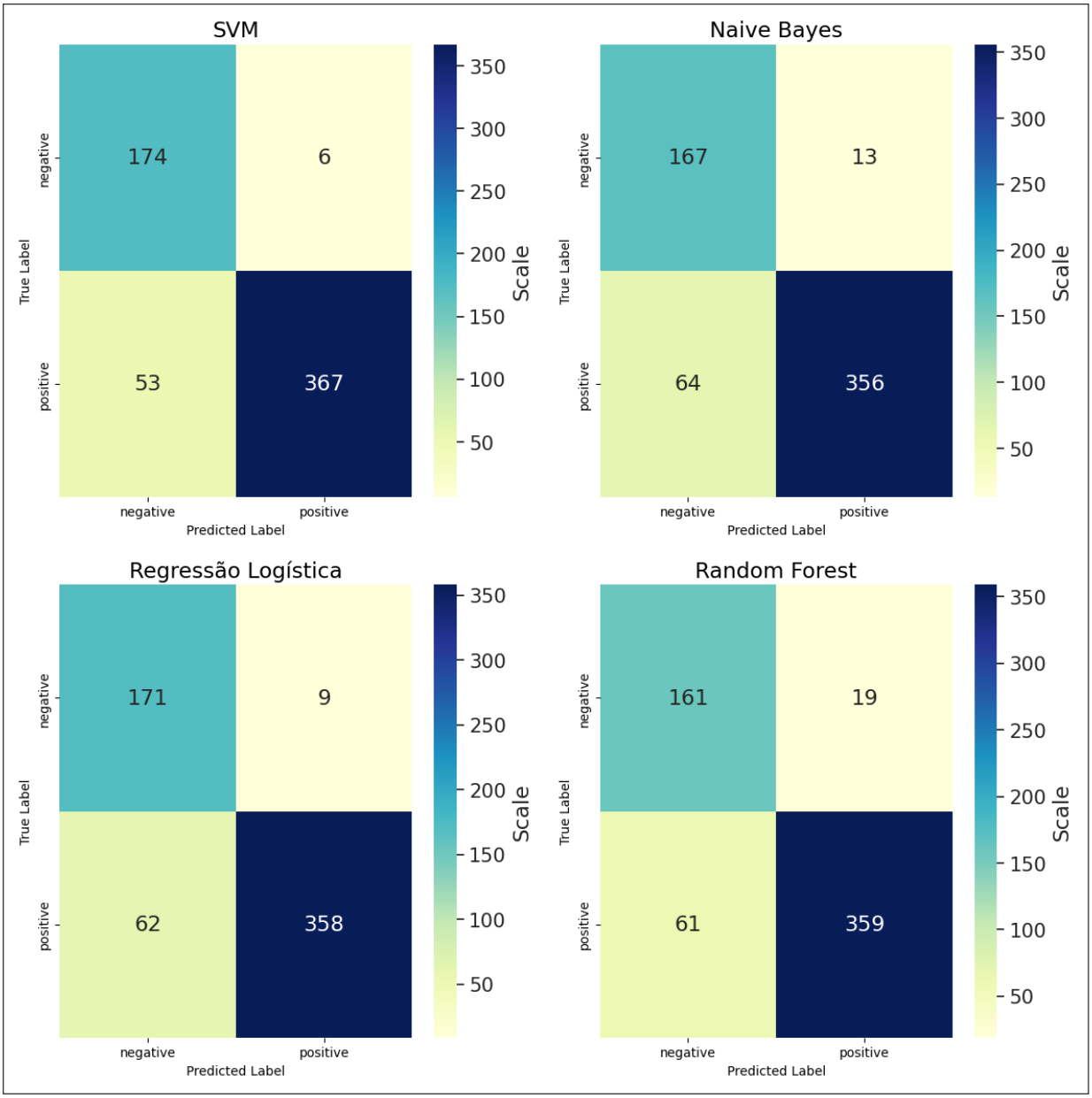


Figura 20 – Matriz de confusão dos modelos com 1.378 amostras sem pré-processamento

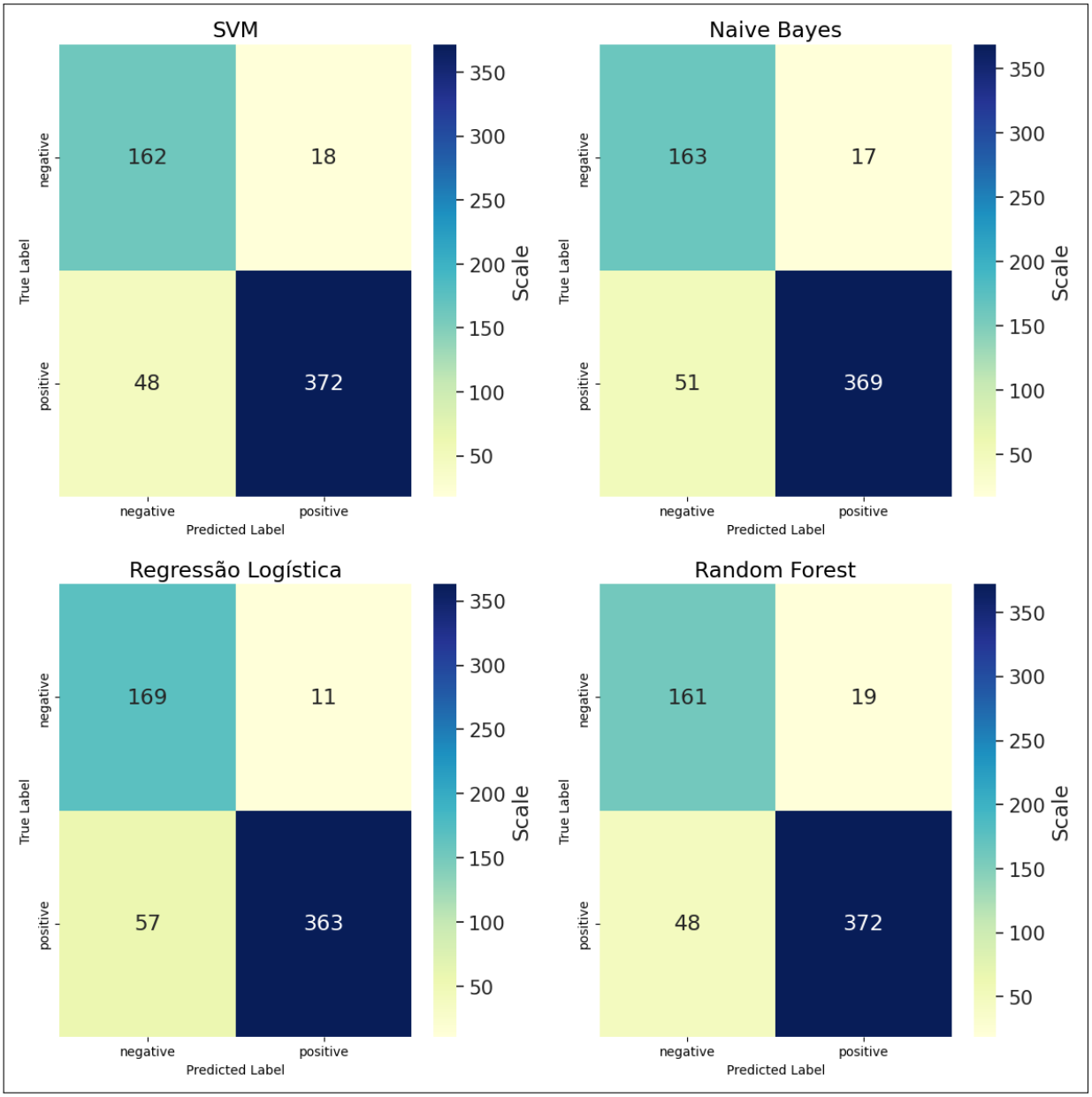


Figura 21 – Matriz de confusão dos modelos com 1.378 amostras com pré-processamento

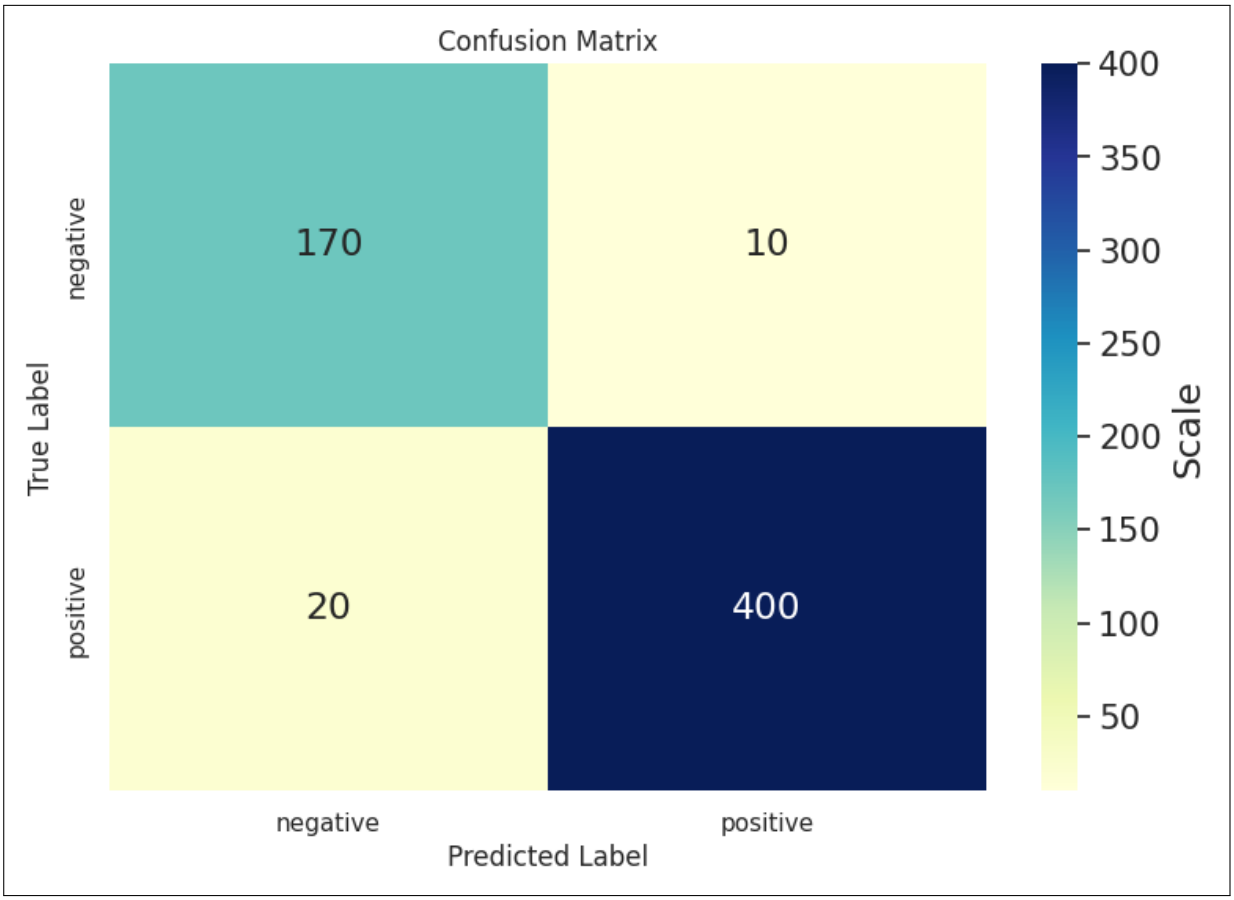
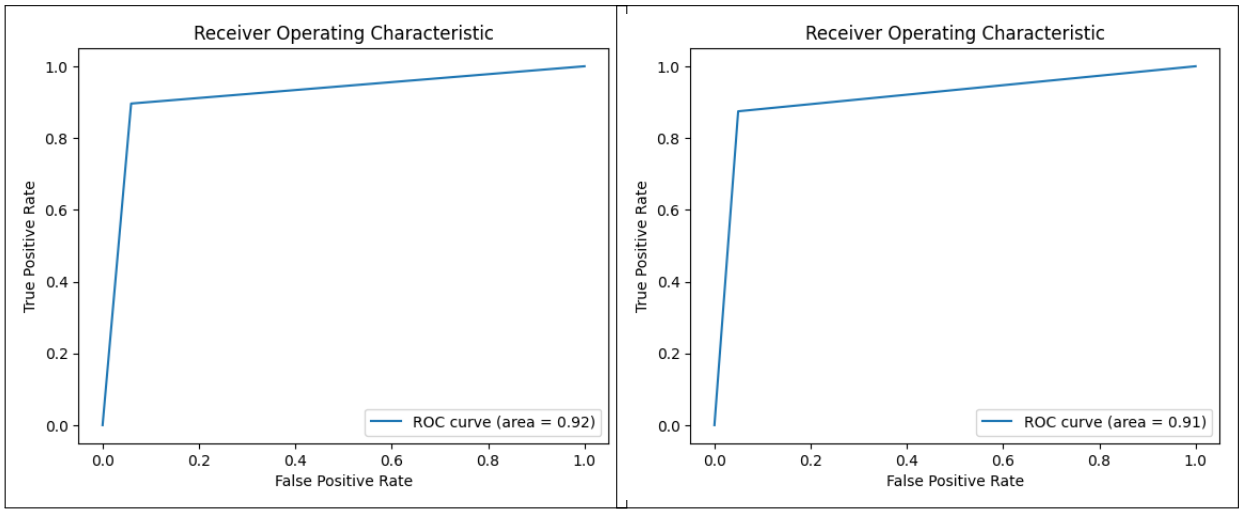
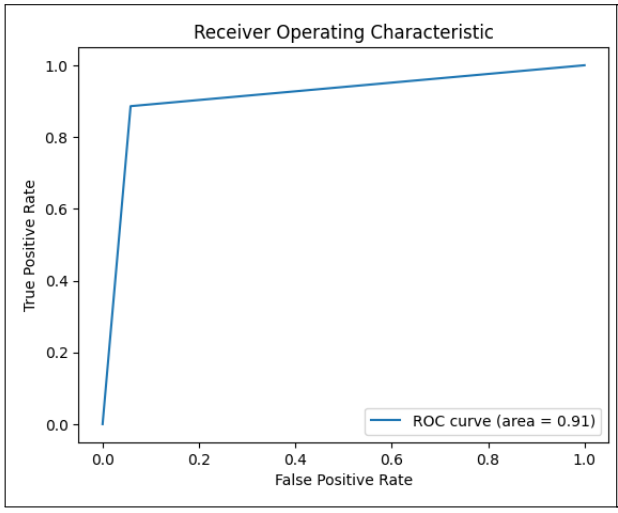


Figura 22 – Matriz de confusão do BERT com 1.378 amostras

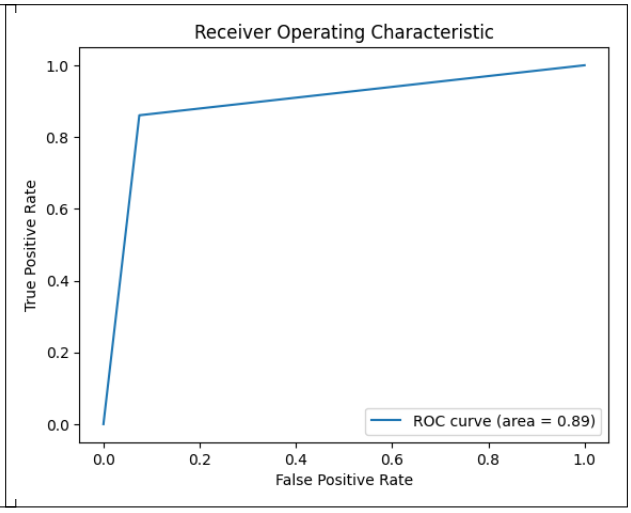


(a) SVM

(b) *Naive Bayes*

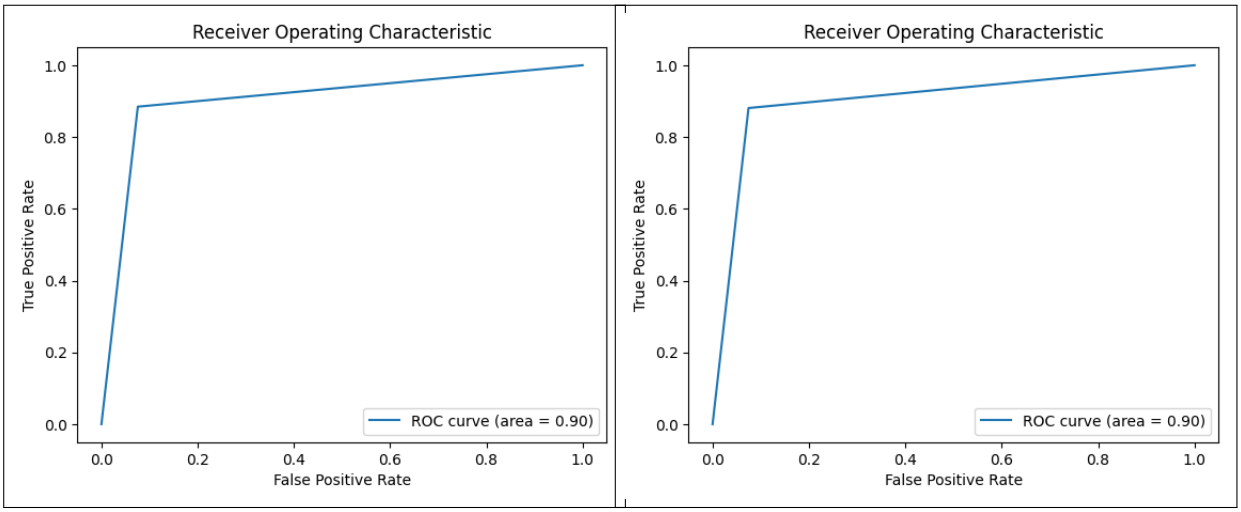


(c) Regressão Logística

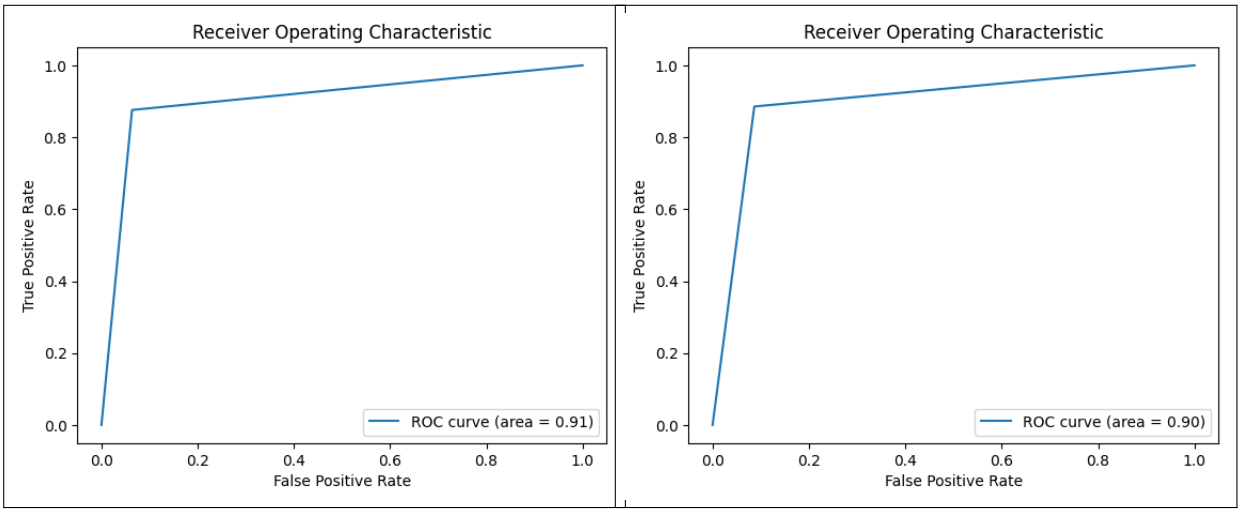


(d) *Random Forest*

Figura 23 – Curva ROC com 7.060 amostras e sem pré-processamento



(a) SVM (b) *Naive Bayes*



(c) Regressão Logística (d) *Random Forest*

Figura 24 – Curva ROC com 7.060 amostras e com pré-processamento

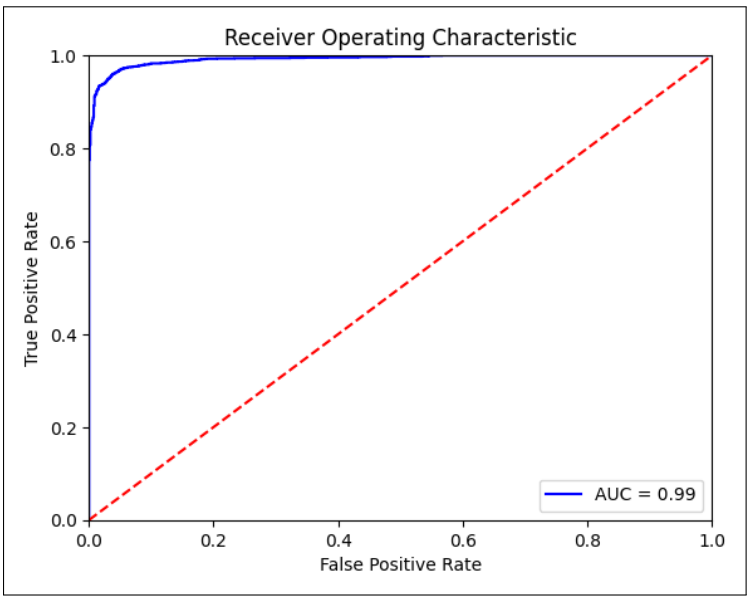
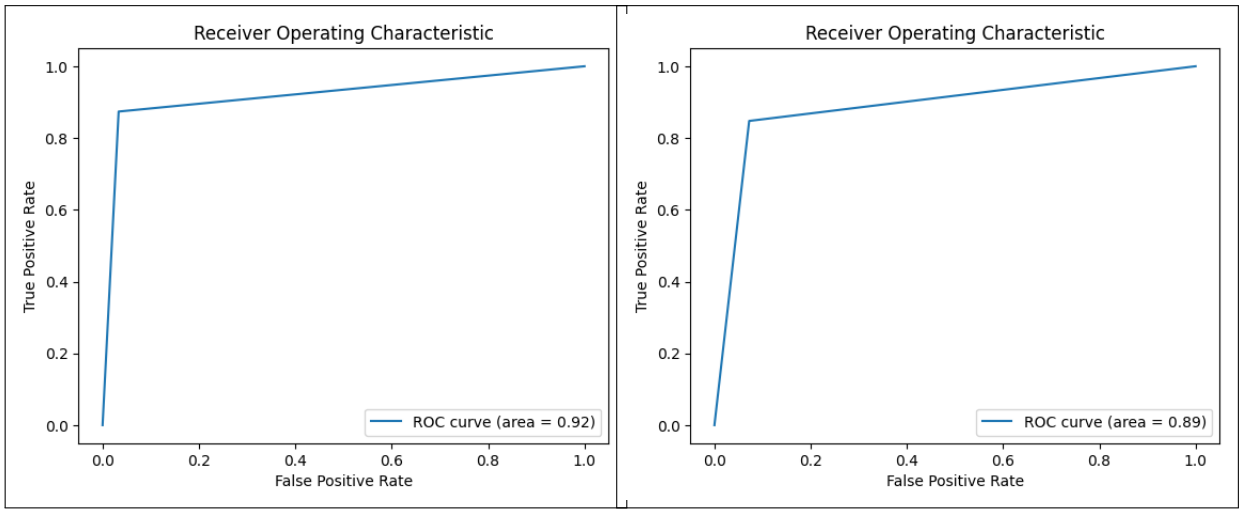
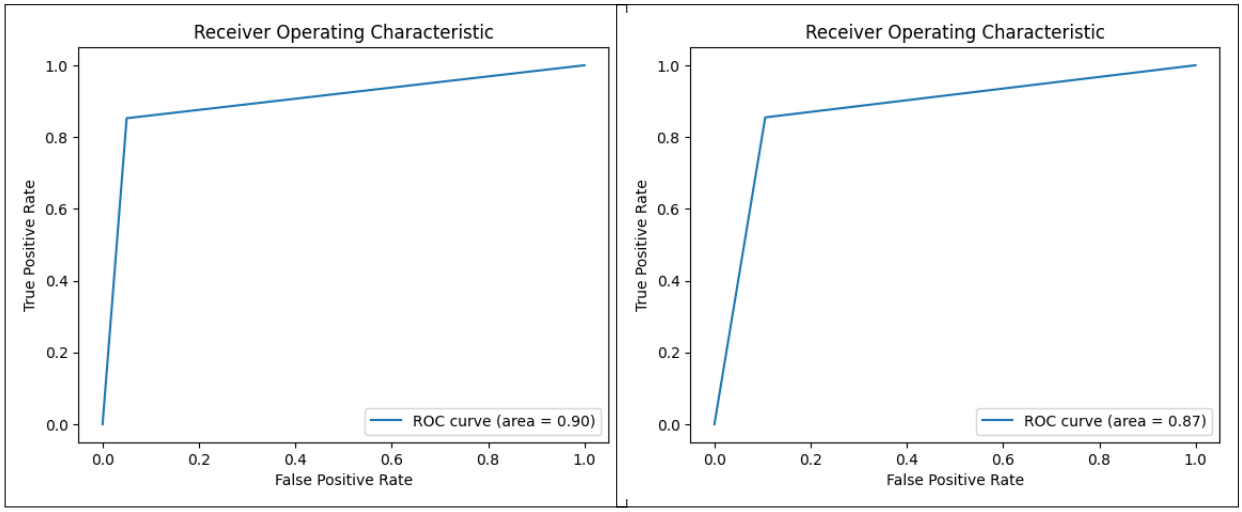


Figura 25 – Curva ROC do BERT com 7.060 amostras

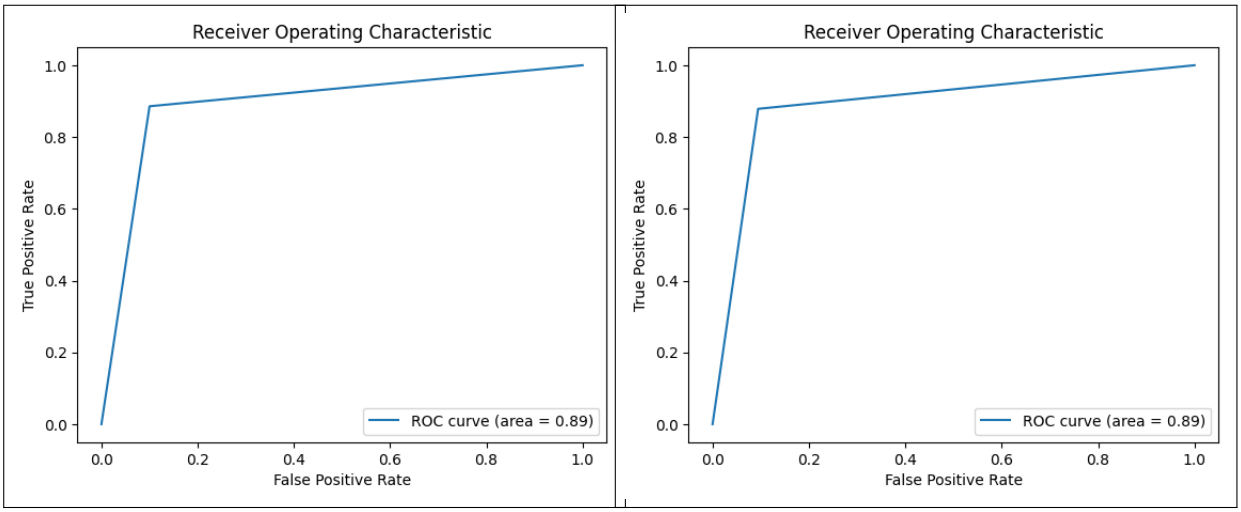


(a) SVM (b) *Naive Bayes*

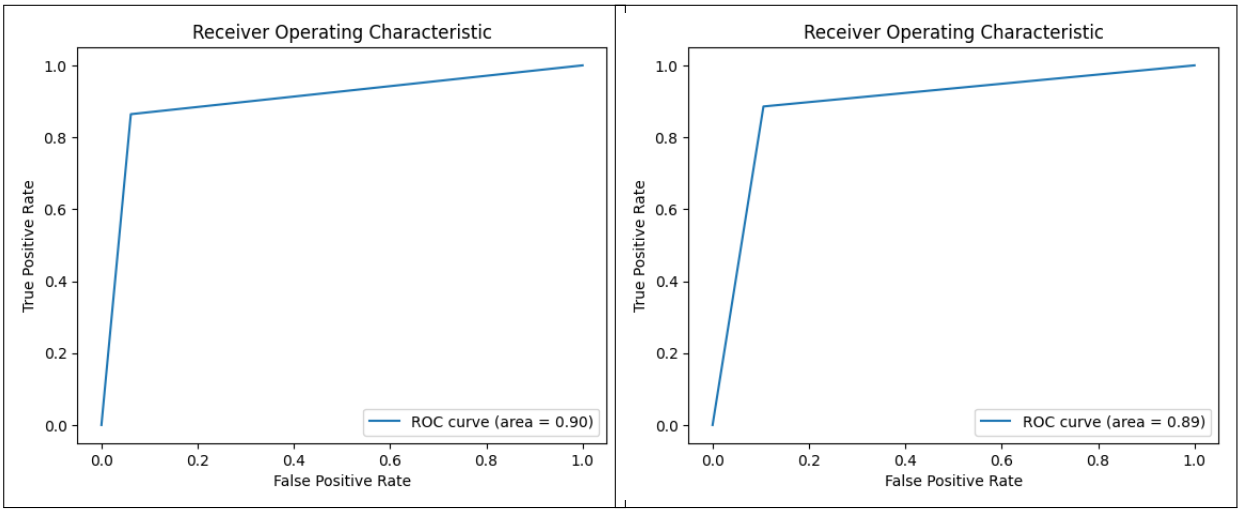


(c) Regressão Logística (d) *Random Forest*

Figura 26 – Curva ROC com 1.378 amostras e sem pré-processamento



(a) SVM (b) *Naive Bayes*



(c) Regressão Logística (d) *Random Forest*

Figura 27 – Curva ROC com 1.378 amostras e com pré-processamento

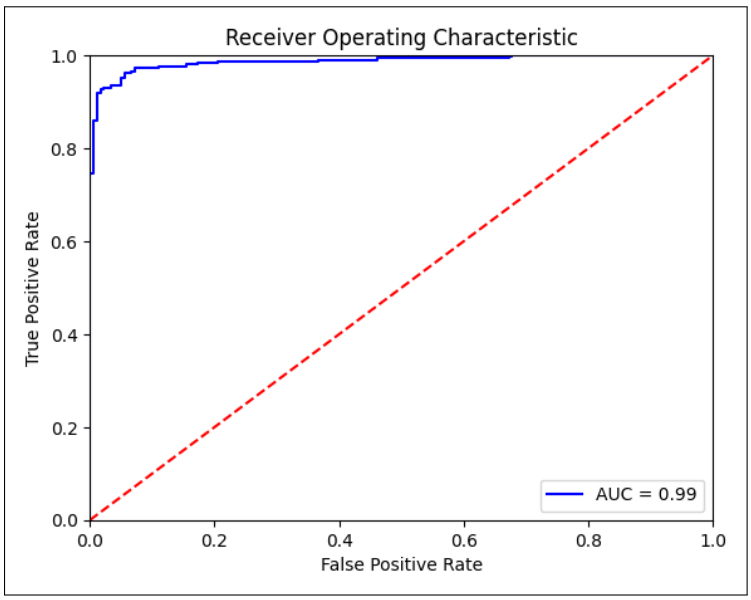


Figura 28 – Curva ROC do BERT com 1.378 amostras

6 Conclusão

Esta pesquisa concentrou-se na análise de sentimentos em avaliações de produtos em língua portuguesa, explorando o desempenho desses algoritmos na tarefa de decifrar as nuances das opiniões expressas pelos consumidores.

Os resultados que obtivemos têm implicações significativas, especialmente para a aplicação prática em língua portuguesa. Constatou-se que o BERT é realmente capaz de alcançar as mais altas taxas de acurácia, o que representa um avanço significativo na capacidade dos modelos de linguagem em compreender nuances contextuais em texto não processado. Isso é particularmente relevante em aplicações onde a compreensão precisa das opiniões dos clientes falantes de português é crucial, como análise de *feedbacks* dos usuários e o monitoramento de avaliações de serviços.

Contudo, essa vantagem é contrabalanceada pelo requisito de um tempo de treinamento e execução substancialmente maior, bem como pela necessidade de utilização de uma GPU, questões cruciais que podem restringir a viabilidade do uso do BERT em diversos cenários de pesquisa.

Além disso, este estudo constatou que o pré-processamento de texto não apresenta um aprimoramento significativo no desempenho dos modelos tradicionais, ao menos nos textos em português. Ainda assim, a decisão sobre o modelo de classificação de texto a ser adotado deve considerar as necessidades específicas da aplicação. Se a busca pela máxima acurácia for o objetivo central da pesquisa e os recursos computacionais não forem uma limitação, o BERT se apresenta como uma ferramenta poderosa e disruptiva. Em contrapartida, para cenários onde a eficiência computacional é fundamental, modelos tradicionais como o SVM e a Regressão Logística ainda permanecem como escolhas preferenciais, oferecendo um equilíbrio sólido entre desempenho e recursos disponíveis.

As conclusões deste estudo, portanto, enriquecem e atualizam o campo da pesquisa em processamento de texto na língua portuguesa, proporcionando um guia valioso para a seleção de modelos em conformidade com as demandas de pesquisa e outras aplicações comerciais.

6.1 Limitações do estudo e sugestões para trabalhos futuros

Embora tenhamos obtido *insights* sobre o desempenho de diferentes algoritmos de classificação de texto, é importante reconhecer algumas limitações que afetaram a pesquisa.

Uma das principais limitações encontradas foi a dificuldade de treinar o modelo BERT no ambiente do *Google Colab*. A necessidade de monitorar a sessão e interagir periodicamente com a página da web para evitar a desconexão ou destruição da máquina virtual

(VM) pode ser inconveniente e exigiu bastante tempo livre, por vezes uma tarde inteira para um único experimento.

Nossos experimentos utilizaram os algoritmos de classificação com seus parâmetros padrão. Para obter um desempenho ainda melhor, poderíamos realizar uma busca de hiperparâmetros para ajustar os modelos e encontrar possíveis configurações ótimas.

Uma possível direção para pesquisas futuras poderia envolver a realização de experimentos incrementais com conjuntos de dados de tamanhos variados, a fim de investigar mais profundamente o impacto do pré-processamento na performance dos algoritmos. Essa abordagem possibilitaria uma análise mais detalhada da possível tendência que sugere se o pré-processamento se torna mais benéfico à medida que o conjunto de dados diminui em escala ou se, por outro lado, sua utilização pode realmente prejudicar os algoritmos em cenários com volumes de dados muito grandes.

Além de tudo, o conjunto de dados utilizados nos treinamentos foram artificialmente equilibrados em termos de distribuição de classes. Futuros estudos poderiam explorar o desempenho dos modelos em conjuntos de dados altamente desbalanceados, que são comuns em muitas aplicações do mundo real.

Referências

ALBUQUERQUE, C. L. D. S. Análise de sentimento sobre comentários em sites de e-commerce no idioma português/br fortaleza 2022. 2022. Citado na página 33.

AVILA, G. V. *Análise de sentimento para textos curtos*. Tese (Doutorado), 2017. Citado 2 vezes nas páginas 33 e 55.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. Citado na página 28.

BARBOSA, M. E.; MAAS, H.; PEREIRA, B. H. Análise de sentimento por meio de redes neurais artificiais em sistemas de avaliação de cursos de graduação. 2019. Citado na página 30.

BENEVENUTO, F.; RIBEIRO, F.; ARAÚJO, M. Métodos para análise de sentimentos em mídias sociais. *Sociedade Brasileira de Computação*, 2015. Citado na página 22.

BOSCO, G. L.; PILATO, G.; SCHICCHI, D. A neural network model for the evaluation of text complexity in italian language: a representation point of view. *Procedia computer science*, Elsevier, v. 145, p. 464–470, 2018. Citado na página 27.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, p. 5–32, 2001. Citado na página 26.

CARNEIRO, T. et al. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, IEEE, v. 6, p. 61678, 2018. Citado na página 40.

CASELI, H.; FREITAS, C.; VIOLA, R. Processamento de linguagem natural. *Sociedade Brasileira de Computação*, 2022. Citado na página 23.

CASTRO, C.; BRAGA, A. Supervised learning with imbalanced data sets: An overview. *Sba: Controle Automação Sociedade Brasileira de Automatica*, v. 22, p. 441–466, 10 2011. Citado na página 30.

CHEESEMAN, P. C.; STUTZ, J. C. et al. Bayesian classification (autoclass): theory and results. *Advances in knowledge discovery and data mining*, Philadelphia, PA, USA, v. 180, p. 153–180, 1996. Citado na página 21.

CHEN, J. et al. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, Elsevier, v. 36, n. 3, p. 5432–5435, 2009. Citado na página 25.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, p. 273–297, 1995. Citado na página 24.

DEGASPERI, M. M. Classificação de relevância de atendimentos de uma base de help desk por meio de técnicas de processamento de linguagem natural. In: IFES. [S.l.], 2023. Citado na página 25.

DÍAZ-URIARTE, R.; ANDRÉS, S. Alvarez de. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, Springer, v. 7, p. 1–13, 2006. Citado na página 26.

DINIZ, F. A. et al. Redface: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces. *Revista Brasileira de Computação Aplicada*, v. 5, n. 1, p. 42–54, 2013. Citado na página 26.

FILHO, J. I. d. S. *O desenvolvimento do e-commerce durante a pandemia.* Dissertação (B.S. thesis), 2023. Citado na página 19.

FRANCESCHI, P. R. d. Modelagens preditivas de churn: o caso do banco do brasil. Universidade do Vale do Rio dos Sinos, 2019. Citado na página 31.

GIL, A. C. *Como elaborar projetos de pesquisa.* 4. ed. São Paulo: Atlas, 1987. Citado na página 39.

GONZÁLEZ-CARVAJAL, S.; GARRIDO-MERCHÁN, E. C. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020. Citado na página 34.

GONZALEZ, M.; LIMA, V. L. S. Recuperação de informação e processamento da linguagem natural. In: SN. *XXIII Congresso da Sociedade Brasileira de Computaçao.* [S.l.], 2003. v. 3, p. 347–395. Citado na página 22.

HEMMATIAN, F.; SOHRABI, M. K. A survey on classification techniques for opinion mining and sentiment analysis. *Artificial intelligence review*, Springer, v. 52, n. 3, p. 1495–1545, 2019. Citado na página 17.

HILARIO, M. et al. Systematic literature review of sentiment analysis techniques. *Journal of Contemporary Issues in Business and Government Vol*, v. 27, n. 1, 2021. Citado na página 18.

HOANG, M.; BIHORAC, O. A.; ROUCES, J. Aspect-based sentiment analysis using bert. In: *Proceedings of the 22nd nordic conference on computational linguistics.* [S.l.: s.n.], 2019. p. 187–196. Citado na página 35.

HUSAIN, F.; UZUNER, O. Investigating the effect of preprocessing arabic text on offensive language and hate speech detection. *Transactions on Asian and Low-Resource Language Information Processing*, ACM New York, NY, v. 21, n. 4, p. 1–20, 2022. Citado na página 49.

JUNQUEIRA, K. T.; FERNANDES, A. M. da R. Análise de sentimento em redes sociais no idioma português com base em mensagens do twitter. *Anais do Computer on the Beach*, p. 681–690, 2018. Citado na página 19.

JURAFSKY, D. *Speech & language processing.* [S.l.]: Pearson Education India, 2000. Citado na página 23.

KENTON, J. D. M.-W. C.; TOUTANOVA, L. K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of naacL-HLT.* [S.l.: s.n.], 2019. v. 1, p. 2. Citado 3 vezes nas páginas 27, 29 e 48.

KHAN, S. et al. Fahad shahbaz khan and mubarak shah. transformers in vision: A survey. 2021. Citado na página 28.

KILGARRIFF, A. I don't believe in word senses. *Computers and the Humanities*, Springer, v. 31, p. 91–113, 1997. Citado na página 23.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, Citeseer, v. 33, n. 2004, p. 1–26, 2004. Citado na página 35.

KOHAVI, R. et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: *Kdd*. [S.l.: s.n.], 1996. v. 96, p. 202–207. Citado na página 25.

LANTZ, B. *Machine Learning with R*. [S.l.]: Packt Publishing, 2013. ISBN 1782162143. Citado na página 21.

LI, Q. et al. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM New York, NY, v. 13, n. 2, p. 1–41, 2022. Citado na página 17.

LIN, X. Sentiment analysis of e-commerce customer reviews based on natural language processing. In: *Proceedings of the 2020 2nd International Conference on Big Data and Artificial Intelligence*. [S.l.: s.n.], 2020. p. 32–36. Citado na página 34.

LIU, B. *Sentiment analysis and opinion mining*. [S.l.]: Springer Nature, 2022. Citado na página 23.

LORENA, A. C.; CARVALHO, A. C. D. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado na página 22.

LUTKEVICH, B. *BERT language model*. 2020. Disponível em: <https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model>. Citado na página 18.

MARTINEZ, E. Z.; LOUZADA-NETO, F.; PEREIRA, B. d. B. A curva roc para testes diagnósticos. *Cad. saúde colet.,(Rio J.)*, p. 7–31, 2003. Citado na página 32.

MIKHAIL, E.; ACKERMAN, F. *Observations and least squares, New York, IAP*. 1976. Citado na página 30.

MITCHELL, R.; FRANK, E. Accelerating the xgboost algorithm using gpu computing. *PeerJ Computer Science*, PeerJ Inc., v. 3, p. e127, 2017. Citado na página 30.

MITCHELL, T. M. *Machine learning*. 1997. Citado na página 21.

MONTEIRO, R. et al. Classificação de fake news com textos de notícias em língua portuguesa integrando data warehousing e machine learning. 10 2019. Citado na página 26.

MORANA, H. C. P. *"Identificação do ponto de corte para a escala PCL-R (Psychopathy Checklist Revised) em população forense brasileira: caracterização de dois subtipos de personalidade transtorno global e parcial"*. Tese (Doutorado), 2004. Disponível em: <https://doi.org/10.11606/t.5.2004.tde-14022004-211709>. Citado na página 32.

MUBAROK, M. S.; ADIWIJAYA; ALDHI, M. D. Aspect-based sentiment analysis to review products using naïve bayes. In: AIP PUBLISHING LLC. *AIP conference proceedings*. [S.l.], 2017. v. 1867, n. 1, p. 020060. Citado na página 17.

MÜLLER, A. C.; GUIDO, S. *Introduction to machine learning with Python: a guide for data scientists*. [S.l.]: "O'Reilly Media, Inc.", 2016. Citado na página 26.

MUNIKAR, M.; SHAKYA, S.; SHRESTHA, A. Fine-grained sentiment classification using bert. In: IEEE. *2019 Artificial Intelligence for Transforming Business and Society (AITB)*. [S.l.], 2019. v. 1, p. 1–5. Citado na página 34.

NARAYANAN, R.; LIU, B.; CHOUDHARY, A. Sentiment analysis of conditional sentences. In: *Proceedings of the 2009 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2009. p. 180–189. Citado na página 23.

NOBRE, G. et al. Booviews: Aspect-based sentiment analysis on product reviews combining svm and crf in portuguese. *Student Research Workshop - PROPOR*, p. 2, 2016. Citado na página 42.

NORONHA, D. H.; FERNANDES, M. A. Implementaçao em fpga de máquina de vetores de suporte (svm) para classificaçao e regressao. *XIII Encontro Nacional de Inteligência Artificial e Computacional-ENIAC*, 2016. Citado na página 25.

ORENGO, V.; HUYCK, C. A stemming algorithm for the portuguese language. In: IEEE. *Proceedings Eighth Symposium on String Processing and Information Retrieval*. [S.l.], 2001. p. 186–193. Citado na página 45.

OSHIRO, T. M. *Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado 2 vezes nas páginas 26 e 27.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, New York, v. 2, n. 1, p. 1–135, maio 2008. Citado na página 17.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <http://jmlr.org/papers/v12/pedregosa11a.html>. Citado na página 39.

PIRES, J. C.; MARTINS, W. S.; SOUSA, D. X. de. Impulsionando árvores extremamente aleatórias ensacadas em paralelo para classificação de textos. In: SBC. *Anais da VI Escola Regional de Informática de Goiás*. [S.l.], 2018. p. 325–330. Citado na página 30.

POLO, T. C. F.; MIOT, H. A. Aplicações da curva ROC em estudos clínicos e experimentais. *Jornal Vascular Brasileiro*, FapUNIFESP (SciELO), v. 19, 2020. Disponível em: <https://doi.org/10.1590/1677-5449.200186>. Citado na página 32.

RADFORD, A. et al. Improving language understanding with unsupervised learning. Technical report, OpenAI, 2018. Citado na página 29.

RAIAAN, M. A. K. et al. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. TechRxiv, 2023. Citado na página 19.

RAIN, C. Sentiment analysis in amazon reviews using probabilistic machine learning. *Swarthmore College*, v. 42, p. 2–3, 2013. Citado na página 42.

REAL, L.; OSHIRO, M.; MAFRA, A. B2w-reviews01-an open product reviews corpus. In: *the Proceedings of the XII Symposium in Information and Human Language Technology*. [S.l.: s.n.], 2019. p. 200–208. Citado 2 vezes nas páginas 41 e 55.

SALTON, G. *Automated language processing*. [S.l.], 1968. Citado na página 23.

SANTO, J. do E.; FILHO, J. de O. Classificação e contagem de bovinos em imagens aéreas utilizando visão computacional e aprendizagem de máquina. In: *Anais da VIII Escola Regional de Computação do Ceará, Maranhão e Piauí*. Porto Alegre, RS, Brasil: SBC, 2020. p. 165–172. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/ercemapi/article/view/11481>. Citado na página 26.

SANTOS, A. E. M. Classificação de maciços rochosos por meio de técnicas da estatística multivariada e inteligência artificial. 2021. Citado na página 21.

SANTOS, F. L. d. Mineração de opinião em textos opinativos utilizando algoritmos de classificação. 2013. Citado na página 26.

SILVA, N. R.; LIMA, D.; BARROS, F. Sapair: Um processo de análise de sentimento no nível de característica. In: *4nd International Workshop on Web and Text Intelligence (WTI'12), Curitiba*. [S.l.: s.n.], 2012. p. 2. Citado na página 33.

SOUZA, D. M. S. d. et al. Efeito do sentimento do investidor manifesto via twitter sobre os retornos e o volume negociado no mercado acionário brasileiro. Universidade Federal da Paraíba, 2020. Citado na página 33.

SOUZA, F. C. de. *BERTimbau: pretrained BERT models for brazilian portuguese= BERTimbau: modelos BERT pré-treinados para português brasileiro*. Tese (Doutorado) — [sn], 2020. Citado 2 vezes nas páginas 40 e 48.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introdução ao datamining: mineração de dados*. [S.l.]: Ciência Moderna, 2009. Citado na página 25.

TIBURCIO, G. V. Avaliação experimental de classificadores para análise de sentimentos em dados de redes sociais. Universidade Federal de Uberlândia, 2021. Citado na página 34.

VARELA, P. d. N. B. L. *Sentiment Analysis*. Dissertação (Mestrado), dez. 2012. Citado na página 34.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 28.

XU, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 2048–2057. Citado na página 28.

ZHANG, H. The optimality of naive bayes. *Aa*, v. 1, n. 2, p. 3, 2004. Citado na página 25.

ZHANG, L.; WANG, S.; LIU, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, v. 8, n. 4, p. e1253, 2018. Citado na página 24.