CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO

Pedro Otávio Soriano Jandre

ANÁLISE DE SENTIMENTOS APLICADA AO SISTEMA DE AVALIAÇÃO DE CLIENTES EM UMA LOJA VIRTUAL

Timóteo

2024

Pedro Otávio Soriano Jandre

ANÁLISE DE SENTIMENTOS APLICADA AO SISTEMA DE AVALIAÇÃO DE CLIENTES EM UMA LOJA VIRTUAL

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Ms. Odilon Corrêa da Silva

Timóteo

2024

Pedro Otávio Soriano Jandre

ANÁLISE DE SENTIMENTOS APLICADA AO SISTEMA DE AVALIAÇÃO DE CLIENTES EM UMA LOJA VIRTUAL

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 11 de fevereiro de 2025:

Prof. Me. Odilon Corrêa da Silva
Orientador

Prof. Dr. Maurílio Alves Martins da Costa
Professor Convidado

Prof. Dr. Marcelo de Sousa Balbino
Professor Convidado

Timóteo
2025

Resumo

Através de avaliações, os usuários têm a oportunidade de compartilhar suas experiências e opiniões sobre produtos adquiridos com outros consumidores. Essas avaliações desempenham um papel crucial para lojistas e produtores, pois podem fornecer percepções valiosas sobre o desempenho de um produto, ajudando a melhorar a oferta e a experiência do cliente. No entanto, a grande quantidade de dados gerados torna difícil para os comerciantes interpretar essas opiniões manualmente. Nesse contexto, a análise de sentimentos surge como uma solução, permitindo que essas informações sejam processadas automaticamente para extrair percepções. O objetivo deste estudo foi realizar uma análise comparativa entre modelos de análise de sentimentos, focando na relação entre comentários textuais e as notas atribuídas pelos consumidores. Para isso, foi realizada uma coleta automatizada de dados de avaliações da plataforma de *e-commerce* Mercado Livre, seguida pelo processamento, balanceamento da base de dados e aplicação de algoritmos de aprendizado de máquina, como Regressão Logística, Floresta Aleatória e Árvore de Decisão. Os dados textuais foram vetorizados utilizando técnicas como Count Vectorizer e TF-IDF, empregando unigramas e bigramas para otimizar o desempenho dos modelos. Os resultados foram validados por meio de testes rigorosos em bases balanceadas, bem como em bases contendo exclusivamente avaliações positivas e negativas. A Regressão Logística apresentou a maior acurácia em todos os cenários de validação, alcançando 96% de acurácia quando combinada com unigramas e o método de vetorização Count Vectorizer. O estudo conclui que analisadores de sentimentos são ferramentas valiosas para compreender as percepções dos consumidores, permitindo que plataformas de comércio eletrônico aprimorem suas estratégias e ofereçam uma melhor experiência aos usuários, com base em conclusões orientadas por dados. A pesquisa evidenciou que o desempenho desses analisadores varia conforme o modelo de aprendizado de máquina (Regressão Logística, Floresta Aleatória ou Árvore de Decisão), o tipo de vetorização (Count Vectorizer ou TF-IDF) e o uso de unigramas ou bigramas, com a Regressão Logística e unigramas apresentando os melhores resultados.

Palavras-chave: análise de sentimentos, comércio eletrônico, processamento de linguagem natural, vetorização de textos, regressão logística, floresta aleatória, árvore de decisão, mineração de dados.

Abstract

Through reviews, users have the opportunity to share their experiences and opinions about purchased products with other consumers. These reviews play a crucial role for retailers and producers, as they can provide valuable insights into a product's performance, helping to improve offerings and customer experience. However, the large volume of data generated makes it difficult for merchants to manually interpret these opinions. In this context, sentiment analysis emerges as a solution, allowing this information to be automatically processed to extract insights. The objective of this study was to conduct a comparative analysis of sentiment analysis models, focusing on the relationship between textual comments and the ratings assigned by consumers. To achieve this, an automated data collection of reviews from the e-commerce platform Mercado Livre was performed, followed by data processing, dataset balancing, and the application of machine learning algorithms such as Logistic Regression, Random Forest, and Decision Tree. The textual data was vectorized using techniques such as Count Vectorizer and TF-IDF, employing unigrams and bigrams to optimize model performance. The results were validated through rigorous testing on balanced datasets, as well as on datasets containing only positive and negative reviews. Logistic Regression achieved the highest accuracy in all validation scenarios, reaching 96% accuracy when combined with unigrams and the Count Vectorizer method. The study concludes that sentiment analyzers are valuable tools for understanding consumer perceptions, enabling e-commerce platforms to refine their strategies and provide a better user experience based on data-driven insights. The research highlighted that the performance of these analyzers varies according to the machine learning model (Logistic Regression, Random Forest, or Decision Tree), the type of vectorization (Count Vectorizer or TF-IDF), and the use of unigrams or bigrams, with Logistic Regression and unigrams delivering the best results.

Keywords: sentiment analysis, e-commerce, natural language processing, text vectorization, logistic regression, random forest, decision tree, data mining

Lista de ilustrações

Lista de tabelas

Sumário

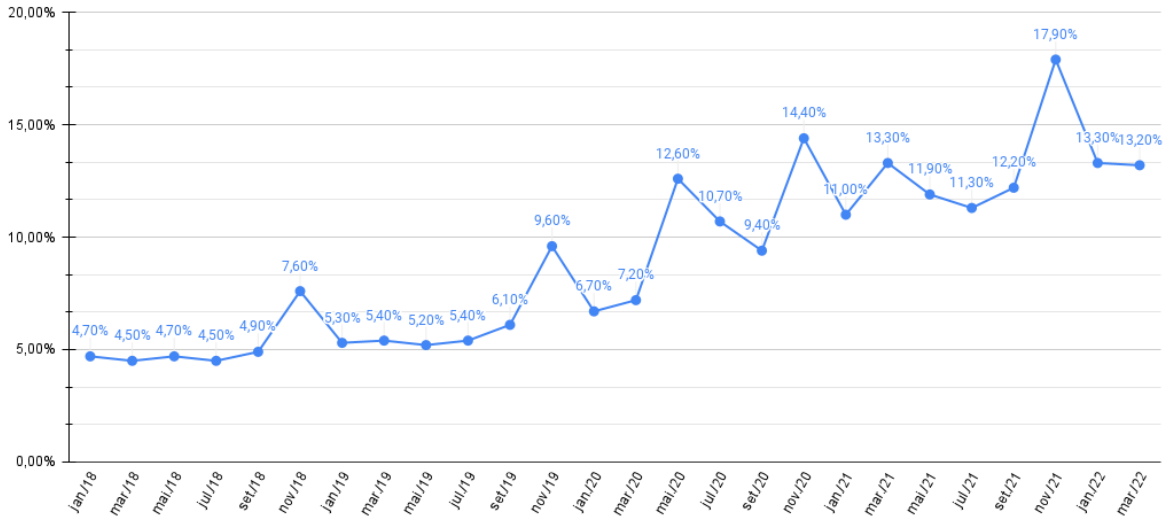
1 Introdução

De acordo com Tomé (2021), o comércio eletrônico foi definido como o processo de compra e venda de produtos e serviços pela Internet. Ele pode ocorrer por meio de lojas virtuais, *marketplaces* ou até mesmo por redes sociais.

"Em 2020, houve um aumento significativo no comércio eletrônico, impulsionado pelo *lockdown* decorrente da pandemia de COVID-19. Com o fechamento do atendimento presencial, muitos negócios buscaram no comércio eletrônico uma alternativa viável, tornando-o o principal canal de vendas para diversos comerciantes", Tomé (2021) .

O gráfico apresentado na Figura 1 demonstra a participação que o comércio eletrônico possui em relação ao comércio varejista. Visualizando o gráfico, pode-se perceber o que foi supracitado: o comércio eletrônico já estava tendo um crescimento, e isso se acentuou ainda mais com a pandemia no ano de 2020.

Figura 1 – Gráfico participação do *E-Commerce* no comércio varejista



Fonte: MCC-ENET, 2022

Pode-se citar como exemplos de e-commerce as empresas *Mercado Livre*, *Amazon* e o *Submarino* e outras grandes lojas varejistas que entraram para o mundo virtual como *Magalu* e as *Lojas Americanas*. Todos esses serviços citados, bem como a maioria das lojas *e-commerce*, possuíam algumas funcionalidades em comum: o registro de usuários, carrinhos de compras e um sistema de avaliação de produtos com comentários dos usuários sobre os produtos.

Através dos comentários, os usuários têm a oportunidade de compartilhar suas experiências e opiniões sobre os produtos com outros consumidores. Além disso, esses comentários fornecem ao lojista novas visões sobre a percepção do público em relação a um determinado

produto.

Figura 2 – Resenha de um produto

Fonte: Mercado Livre, 2024

Como pode ser observado na Figura 2, temos um exemplo de como as opiniões dos usuários podem ser usadas para extrair percepções gerais sobre um produto. A partir deste exemplo, vemos que pode-se extrair diretamente informações sobre as características gerais, como a duração da bateria e a qualidade dos materiais e, também, com a ajuda das resenhas, pode-se formar uma opinião geral sobre o produto. Além disso, é possível observar a coerência dessa avaliação em que a opinião reflete a nota que o produto recebeu, como no caso do exemplo acima em que a nota geral que o produto recebeu foi 4.4, que faz sentido ao comparar com as opiniões em destaque, que foram geradas se baseando nas opiniões dos consumidores.

1.1 Motivação

A multiplicação das opiniões, avaliações e recomendações online constrói um extenso conjunto de dados que ultrapassa a capacidade humana de interpretar. Isso torna necessário o uso de técnicas avançadas voltadas para a extração e tratamento desses dados, a fim de aproveitá-los para identificar novas tendências e oportunidades de negócios.

Uma das técnicas existentes é a análise de sentimentos, que utiliza o processamento de linguagem natural para extrair o sentimento expresso em um texto analisado.

Segundo uma pesquisa desenvolvida pela empresa Lett em parceria com a OpinionBox sobre as opiniões do consumidor e os aspectos mais importantes de uma página de produtos em serviços de e-commerce, é apontado que: “As avaliações são as informações mais importantes na página de produto. Em seguida, aparecem descrições, comentários, imagens e categorização e busca” (LETT; OPINIONBOX, 2019).

Dessa forma, é possível perceber que as opiniões de um consumidor sobre um produto são de suma importância para seu negócio virtual, e isso faz dos comentários deixados pelo consumidor uma área possível para utilizar a análise de sentimentos.

Agora, visando explorar os benefícios da análise de sentimento, este estudo procura responder à seguinte pergunta de pesquisa: Como a aplicação de modelos de análise de sentimentos em avaliações de usuários em plataformas de *e-commerce* pode auxiliar na identificação da coerência entre comentários textuais e notas atribuídas?

1.2 Objetivos

O objetivo geral deste estudo é propor e avaliar a relação entre os comentários textuais e as notas atribuídas por usuários de plataformas de *e-commerce* por meio de um modelo de análise de sentimentos.

1.2.1 Objetivos específicos

Afim de alcançar o objetivo geral, foram traçados os seguintes objetivos específicos:

1. Coletar, organizar e disponibilizar uma base de dados composta por avaliações de usuários em plataformas de comércio eletrônico, utilizando um método automatizado para criar uma base de dados consistente para analise.
2. Implementar e demonstrar o uso de três técnicas de aprendizado de máquina para classificar os sentimentos expressos nas avaliações da base de dados.
3. Validar a eficácia do modelo de análise de sentimentos por meio da avaliação da acurácia das classificações realizadas, verificando a correspondência entre os resultados das análises textuais e as avaliações.

1.3 Justificativa

A análise de sentimentos é uma técnica da área de Processamento de Linguagem Natural (PLN) que permite identificar e interpretar emoções expressas em textos. Um de seus objetivos é compreender a opinião dos usuários sobre determinados produtos, serviços, eventos ou até mesmo personalidades públicas. Essa abordagem tem sido amplamente utilizada em diversos setores, desde o aprimoramento da experiência do cliente no e-commerce até a análise de tendências em redes sociais e estudos de comportamento do consumidor.

Comentários e avaliações de usuários são recursos valiosos para empresas que buscam oferecer produtos e serviços mais alinhados às expectativas do público. Como destacado por Vanaja e Belwal, (2018) “as avaliações dos produtos são importantes para os proprietários de negócios, pois podem ajudá-los a tomar decisões com base nos resultados da análise das opiniões dos clientes sobre os produtos”. Além disso, ao classificar sentimentos em categorias como positivo, negativo e neutro, é possível gerar recomendações mais precisas, melhorando a experiência de compra e aumentando a satisfação do consumidor.

"Extrair novas percepções, entender melhor o comportamento do usuário e redefinir estratégias de marketing são alguns benefícios da análise de sentimentos. A técnica pode servir para entender como os clientes enxergam seu negócio. A ferramenta é abrangente e pode ser utilizada para diversas finalidades", (MJVTEAM, 2020). A seguir, algumas dessas finalidades:

- Análises de mensagens em fóruns e plataformas de ensino online visam identificar possíveis lacunas no aprendizado dos alunos como visto no trabalho de Vivian et al, (2022).
- A análise sobre a troca de informações entre clientes sobre produtos e serviços oferecidos pelas empresas, visando compreender melhor as necessidades do cliente e aprimorar o atendimento ao cliente de Carneiro, (2023).
- Uma análise em cima da popularidade dos candidatos à presidência do Brasil em 2018, utilizando dados textuais do Twitter durante o período eleitoral, e relacionar posteriormente o apoio demonstrado nas mídias sociais com o desempenho eleitoral como apresentado por Pereira, (2019).

Outra aplicação relevante da análise de sentimentos está na avaliação de comentários e notas atribuídas por consumidores em plataformas de e-commerce. Essa abordagem permite compreender como as avaliações textuais refletem as percepções dos usuários e influenciam suas classificações, possibilitando a melhoria de sistemas de recomendação e a adaptação de estratégias empresariais para melhor atender às expectativas do público. Além de impulsionar o engajamento dos consumidores, essa aplicação contribui para um entendimento mais profundo da experiência do usuário e da reputação dos produtos no mercado digital.

2 Referencial Teórico

2.1 Análise de sentimento

Segundo Medhat, 2014, “O estudo computacional de opiniões, atitudes e emoções das pessoas direcionadas a uma entidade é denominado análise de sentimento. Essa entidade pode representar indivíduos, eventos ou assuntos. Essas opiniões deixadas por usuários devem passar por um estudo classificatório para serem utilizadas.”

A análise de sentimento se trata de um processo classificatório que busca identificar, extrair e classificar o sentimento de um indivíduo sobre uma entidade. Segundo Igor (2017), a análise de sentimento é um campo interdisciplinar que envolve o uso de processamento de linguagem natural e inteligência artificial para identificar, extrair e medir informações subjetivas de maneira estruturada.

De acordo com Kumar, 2020, a análise de sentimentos pode ser dividida em três principais frentes de classificação que se diferenciam pela dimensão do texto a ser analisado: ao nível de documento, sentença ou aspecto. A análise de documentos procura classificar uma opinião considerando o conteúdo do documento na totalidade, partindo do pressuposto de que ele aborda apenas um único assunto. A análise ao nível de sentença, por sua vez, foca na classificação do sentimento expresso em cada sentença individualmente no documento. Por fim, a análise ao nível de aspecto visa classificar o texto em relação a aspectos específicos das entidades mencionadas no texto. Com base no exemplo da Figura 3, é possível realizar a seguinte classificação:

Figura 3 – Resenha de produto

Fonte: Mercado livre

- Ao nível do documento:
  - Neste caso, o documento é o comentário na totalidade.
- Ao nível da sentença

  - "Ótima compra."
  - "Fácil de usar e de limpar."
  - "Um bom tamanho para família pequena."
  - "Recomendo utilizar embalagens de papel manteiga para forrar o cesto."

- Ao nível de aspecto
  - Facilidade de uso e limpeza
  - Tamanho adequado para família pequena
  - Recomendação de uso de embalagens de papel manteiga

No presente estudo é realizada uma classificação ao nível de documento, por cada resenha ser considerada um único documento e não ser separada em sentenças ou aspectos. O documento é então tratado como um conjunto de palavras nas quais as palavras com significados semelhantes são tratadas da mesma forma e, a partir disso, é descoberta a sua polaridade.

A análise de sentimento também pode ser classificada quanto às técnicas de classificação, podendo seguir uma abordagem através do aprendizado de máquinas ou uma abordagem baseada em léxico.

- Abordagem por Aprendizado de Máquina
  - Segundo Igor, (2017) a abordagem por aprendizado de máquina é feita por meio do treinamento supervisionado em que o algoritmo analisa os dados marcados como positivo, negativo ou neutro e extrai as características que modelam as diferenças entre as classes e infere uma função de classificação que pode ser usada para avaliar novos exemplos.
- Abordagem Baseada em Léxico
  - De acordo com Chiappe, 2010, a abordagem baseada em léxico calcula o sentimento do texto a partir de uma média da polaridade das frases. Uma frase é considerada positiva se a polaridade das palavras que a compõem forem positivas. Para essa abordagem ser feita é necessário ter anexado um dicionário de palavras com suas polaridades pré-calculadas.

Este trabalho adotou a abordagem por aprendizado de máquina.

2.2 Regressão Logística

De acordo com Hosmer e Lemeshow (1989), a regressão logística consiste em um modelo que relaciona um conjunto de p variáveis independentes $X1, X2, ..., Xp$, a uma variável dependente Y que representa a presença $(Y = 1)$ ou a ausência $(Y = 0)$ de uma característica.

Figura 4 – Curva de regressão logistica

Fonte: adaptado de Vanaja; Belwal, 2018

A regressão logística não prevê apenas valores absolutos (0 ou 1), mas a probabilidade de ocorrência que vai se encaixar dentro desses valores. Por se tratar de uma função logarítmica, a relação entre as variáveis independentes e a variável dependente terá esta curva em forma de “S” como pode ser vista na Figura 4

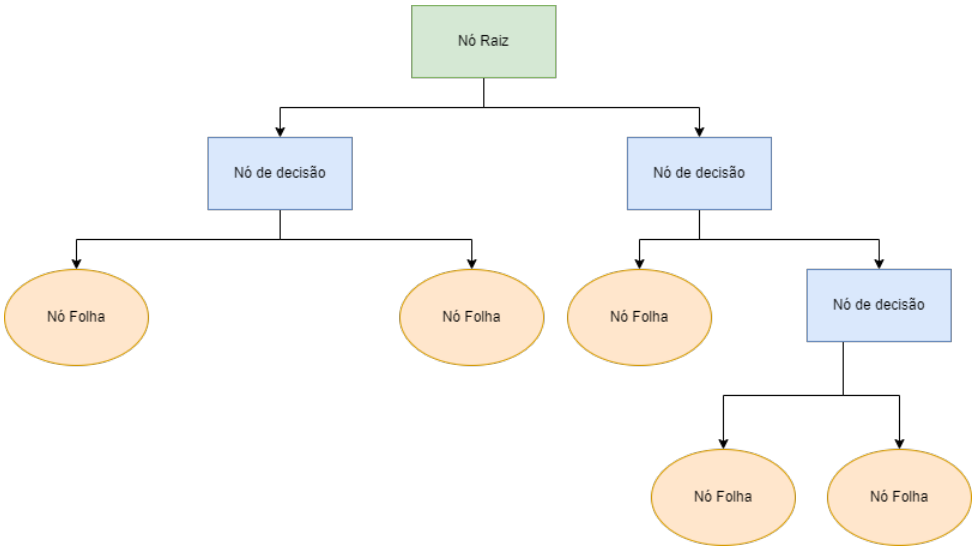
A literatura da área traz alguns exemplos de aplicação da regressão logística, como, por exemplo, no estudo de Palmuti e Piacchiai (2012) onde a regressão logística é usada para montar um modelo de mensuração de risco de crédito. Neste modelo foi utilizada como variável dependente a qualidade do crédito que possui duas categorias: adimplente e inadimplente. Como variáveis independentes, o estudo considerou: gênero, grau de formalidade do empreendedor, garantia oferecida, setor de atuação, escolaridade, valor do crédito, taxa de juros, idade, prazo de pagamento, renda declarada e valor da prestação. Com isso, o modelo final teve taxa de acertos de 87,4%. E dessa forma, o modelo de Palmuti e Picchiai (2012) observou que as variáveis do perfil não possuem influência na qualidade do crédito, mas, sim, variáveis relacionadas diretamente ao risco (juros, valor da prestação, renda, etc.).

2.3 Árvore de decisão

Segundo Myles, (2004), a análise por árvore de decisão é um método baseado na abordagem de *“dividir para conquistar”* para classificação; ela é amplamente utilizada na extração de padrões e identificação de características em grandes bases de dados. Sua capacidade de discriminação e modelagem preditiva, aliada à interpretação intuitiva, faz com que esse modelo seja aplicado extensivamente tanto para análise exploratória de dados quanto para tarefas de previsão. Um exemplo genérico da estrutura de uma árvore de decisão pode ser

visto na Figura 5 abaixo.

Figura 5 – Esquema de uma Árvore de decisão



Fonte: Adaptado de Santana, 2020

De acordo com Santana, (2020), para a tomada de decisão utilizando um modelo baseado em árvore de decisão, as amostras são divididas conforme os critérios estabelecidos. O processo de seleção de uma variável e a divisão das amostras continuam em cada novo nó até que uma regra específica seja atendida:

1. **Início no Nó Raiz:** O algoritmo começa no nó raiz e examina a melhor variável para dividir os dados, com base em uma métrica de seleção chamada Medida de Seleção de Atributo (ASM, em inglês, Attribute Selection Measure). Essa métrica ajuda a determinar qual atributo é o mais informativo e eficiente para dividir o conjunto de dados e criar nós subsequentes.

2. **Divisão dos Dados:** A partir da variável escolhida, o nó raiz é dividido em ramos que levam a novos nós. Cada ramo representa uma possível decisão ou valor do atributo selecionado.

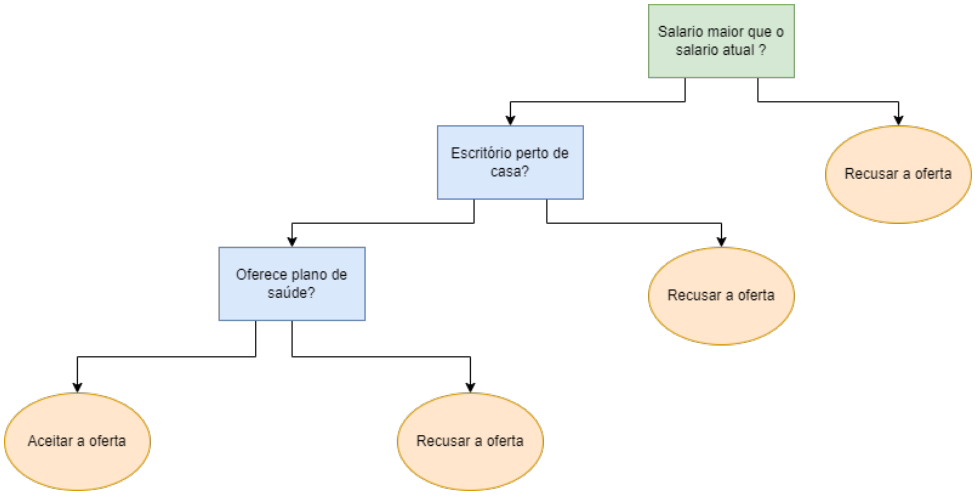
3. **Processo Recursivo:** O algoritmo continua a dividir o conjunto de dados em cada nó subsequente. Para cada novo nó de decisão, o processo de seleção de atributo é repetido, buscando a melhor variável para dividir o subconjunto atual de dados. Esse processo recursivo continua até que não seja mais possível dividir os dados, seja porque os subconjuntos são suficientemente pequenos, seja porque uma regra de parada foi atingida (como um número mínimo de amostras por folha).

4. **Nós Folha e Decisão Final:** Os nós finais são chamados de nós folha e representam o resultado da árvore para uma dada sequência de decisões. Cada nó folha indica uma classificação ou decisão específica.

Um exemplo prático, ilustrado na Figura 6, apresenta o caso de um candidato a emprego que precisa decidir sobre aceitar ou não uma oferta de trabalho. Para resolver esse problema com uma árvore de decisão, o algoritmo começa com o nó raiz que representa o atributo mais importante, por exemplo, “salário”. Este nó raiz é então dividido em novos ramos, com base nas diferentes faixas salariais que o candidato pode considerar.

Em seguida, cada uma dessas categorias salariais pode levar a um novo nó de decisão, como “distância até o trabalho”. Dependendo do valor desse atributo (por exemplo, “próximo” ou “distante”), o nó pode se dividir novamente, talvez considerando se há “benefícios de transporte” oferecidos pela empresa. A árvore continua dividindo os dados até que seja possível chegar a uma decisão final em um nó folha, como “aceitar a oferta” ou “recusar a oferta”.

Figura 6 – Exemplo de uma Árvore de decisão

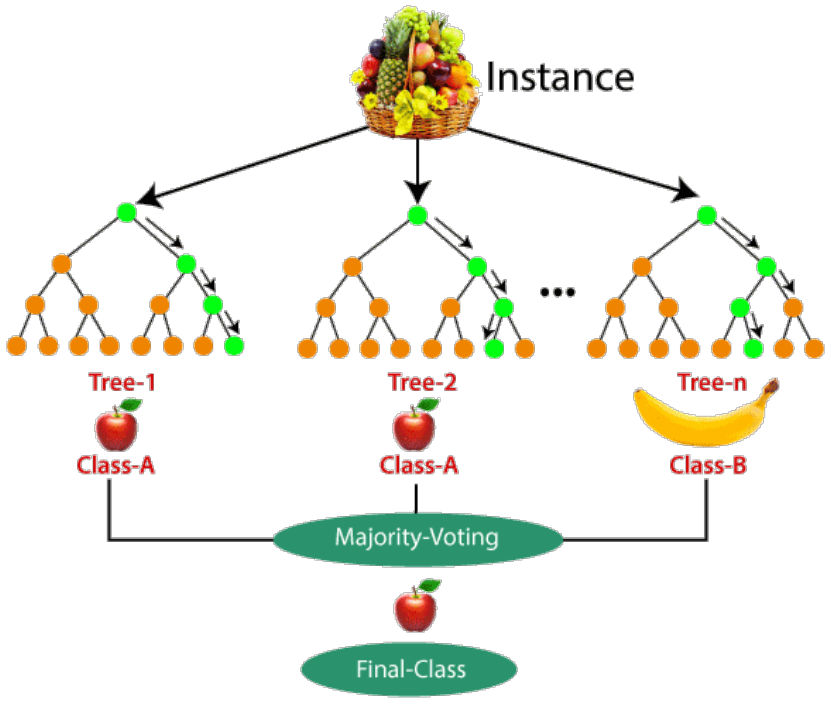


Fonte: Adaptado da página javatpoint, 2024a

2.4 Floresta Aleatória

Segundo Júnior (2018), floresta aleatória é um modelo baseado em árvores de decisão, no qual é utilizada uma combinação de preditores de árvores. Uma floresta aleatória consiste em um conjunto de múltiplas árvores de decisão individuais, cada uma construída de forma independente a partir de diferentes subconjuntos dos dados de treinamento. Esse processo ajuda a reduzir problemas como o *overfitting* (quando um modelo fica muito ajustado aos dados de treinamento e perde a capacidade de generalizar) e aumenta a precisão do modelo.

Figura 7 – Método da Floresta Aleatória.



Fonte: Adaptado da página Javatpoint, 2024b

A Figura 7 apresenta um exemplo prático de como ocorre a seleção utilizando uma floresta aleatória. Suponha que haja um conjunto de dados composto por várias imagens de frutas, como maçãs, laranjas e bananas. Esse conjunto de dados é fornecido a um classificador de floresta aleatória, que aprende a identificar o tipo de fruta em novas imagens.

1. **Divisão do conjunto de dados:** O classificador de floresta aleatória começa dividindo o conjunto de dados em diversos subconjuntos. Em vez de entregar todas as imagens para uma única árvore de decisão, o modelo distribui diferentes subconjuntos para várias árvores, de forma que cada uma receba apenas uma amostra dos dados. Isso aumenta a diversidade de perspectivas entre as árvores e ajuda a reduzir o *overfitting*.

2. **Treinamento das Árvores de Decisão:** Com cada árvore recebendo um subconjunto específico dos dados, o modelo começa a fase de treinamento. Nesse processo, cada árvore aprende a fazer previsões com base nas informações que recebeu. Por exemplo, uma árvore pode identificar certas características que distinguem maçãs, como a forma redonda e a cor avermelhada. Outra árvore pode se especializar em laranjas, considerando aspectos como cor alaranjada e superfície texturizada. Assim, cada árvore de decisão desenvolve uma habilidade única ao identificar padrões específicos nas imagens de frutas.

3. **Processo de Votação e Decisão Final:** Após o treinamento, o modelo está pronto para fazer previsões sobre novas imagens. Quando uma nova imagem de fruta é apresentada ao modelo, cada árvore de decisão a analisa individualmente e fornece uma previsão

sobre o tipo de fruta que acredita estar vendo. A previsão final do modelo de floresta aleatória é então determinada por meio de uma votação. Cada árvore "vota" em sua previsão, e a classificação final é escolhida com base na maioria dos votos. Por exemplo, se a maioria das árvores classificar a nova imagem como "maçã", essa será a decisão final do classificador.

Além de ser utilizada para classificação, a floresta aleatória pode ser aplicada em tarefas de regressão, análise de importância de variáveis e até mesmo na detecção de *outliers*. Ela é amplamente empregada em áreas que envolvem grandes volumes de dados e requisitos de precisão, como diagnóstico médico por imagem, sensoriamento remoto, análise de *big data* e detecção de falhas. A capacidade do modelo de combinar várias árvores de decisão em uma estrutura unificada e robusta torna a floresta aleatória uma das ferramentas mais poderosas e flexíveis no campo do aprendizado de máquina, Júnior (2018).

2.5 Vetorização

Antes de aplicar os algoritmos de aprendizado de máquina é necessário transformar os dados textuais em uma representação numérica para os algoritmos poderem processar. Essa etapa é chamada de vetorização dos dados e, no presente estudo, foram aplicadas duas técnicas de vetorização, *Count Vectorize* e *Term Frequency – Inverse Document Frequency* (TF-IDF) para comparar a eficácia de cada uma.

1. *Count Vectorize*

    - De acordo com Turki (2022), o método de Count Vectorizer representa os textos por meio de um vetor com base na frequência das palavras no vocabulário. Nesse processo, cada palavra encontrada aumenta a contagem correspondente e expande a dimensionalidade do vetor. Para isso o algoritmo realiza as seguintes etapas.
        - Tokenização do texto: utilizando por padrão os espaços em branco como delimitador, divide o texto em palavras individuais.
        - Contagem de frequência: conta a frequência de cada palavra da base de texto e cria um vetor de contagem para cada documento em que cada elemento do vetor representa a contagem de uma palavra específica.
        - Construção do vocabulário: cria um vocabulário único de todas as palavras (*bag of words*) cada palavra única do vocabulário é atribuída a uma posição específica em um vetor.
        - Transforma em vetores numéricos: converte os documentos em vetores numéricos. Cada vetor possui um tamanho igual ao tamanho do vocabulário e contém as contagens das palavras correspondentes ao documento.

2. *Term Frequency – Inverse Document Frequency*

    - Como destacado por Eshan (2017), o **Tfidf Vectorizer** tenta determinar a importância de uma característica para que o classificador não perca características que

são menos frequentes, mas que continuam sendo importantes. Dessa forma, ele atribui pesos mais altos às palavras mais relevantes no contexto do documento e pesos mais baixos às palavras comuns, utilizando a métrica TF-IDF para equilibrar a frequência do termo no documento e em toda a base de dados.

3 Trabalhos correlatos

3.1 Mineração de Dados e Análise de Sentimentos em Ambientes Educacionais

O estudo realizado por Vivian et al. (2022) emprega a análise de sentimentos na esfera educacional, especificamente em ambientes virtuais de aprendizagem (AVA). Seu propósito é realizar um mapeamento sistemático da literatura referente à Mineração de Dados Educacionais e Análise de Sentimentos em AVAs. Esse levantamento aborda técnicas, métodos, algoritmos, bibliotecas e ferramentas de mineração de dados educacionais usados para analisar os sentimentos e emoções dos estudantes nos AVAs.

Os dados utilizados para a análise foram extraídos principalmente de mensagens em fóruns e chats das plataformas AVA e a classificação ficou a cargo do estudo utilizado, mas a principal divisão utilizada foi em neutro, positivo e negativo. O estudo ainda respondeu às perguntas de pesquisa propostas:

1. Quais são as técnicas, métodos, algoritmos, bibliotecas e ferramentas de mineração de dados educacionais para análise de base de dados referentes a sentimentos e emoções do estudante no processo de aprendizagem em AVA?
2. Com qual objetivo foi empregada a análise de sentimentos em AVA?
3. Quais tipos de emoções e sentimentos são considerados nos estudos?

A fim de selecionar os melhores artigos envolvendo o assunto também foi necessário definir regras de inclusão e exclusão dos artigos em que, para o artigo ser incluído no mapeamento eles precisavam respeitar as seguintes regras:

- Estudo primário;
- Estudos que abordam Mineração de Dados Educacionais, Análise de Sentimentos e Ambientes Virtuais de Aprendizagem

Já os critérios para excluir os artigos incluíam as seguintes características:

- Estudos secundários;
- Livros, capítulos de livros, teses e dissertações de pós-graduação, manuais, relatórios;
- Estudos duplicados;
- Estudos a que não foi possível ter acesso;
- Estudos não escritos na língua inglesa;

- Estudos que não abordam Mineração de Dados Educacionais, Análise de Sentimentos e Ambientes Virtuais de Aprendizagem (fora do escopo)
- Estudos irrelevantes para a pesquisa, considerando as questões de pesquisa;
- Estudos publicados antes de janeiro de 2010.

E as etapas para o estudo dos artigos foram:

1. Ler os Títulos, resumos e palavras-chave dos artigos e, em seguida, excluir aqueles considerados irrelevantes para as questões;
2. Ler, na íntegra, os artigos selecionados na etapa anterior;
3. documentar os artigos selecionados em um formulário predefinido.

Com a seleção primária dos artigos utilizando apenas as strings de busca, foram encontrados 323 artigos; em seguida, com a execução da primeira etapa, 48 artigos foram pré-selecionados para a leitura na íntegra e, após a leitura na íntegra, apenas 20 estudos permaneceram para a análise em profundidade.

Agora os 20 artigos selecionados foram usados como base para responder às perguntas de pesquisa propostas:

Quais são as técnicas, métodos, algoritmos, bibliotecas e ferramentas de mineração de dados educacionais para análise de base de dados referentes a sentimentos e emoções do estudante no processo de aprendizagem em AVA?

As técnicas incluem principalmente algoritmos de Aprendizado de Máquina como Support Vector Machine, Naive Bayes e K-Nearest Neighbours e a ferramenta R para manipulação e visualização de dados. Observa-se que não há uma biblioteca predominante para análise de dados referentes a sentimentos e emoções do estudante em AVA.

Com qual objetivo foi empregada a análise de sentimentos em AVA?

Os pesquisadores empregam a análise de sentimentos com diferentes finalidades: avaliar cursos e professores, verificar a eficácia do AVA para melhorar o processo de ensino e, também, fornecer aprendizagem personalizada. Contudo, observa-se a predominância por investigar as opiniões dos alunos sobre cursos, professores e material didático.

Quais tipos de emoções e sentimentos têm sido considerados nos estudos?

A maioria dos estudos considera a polaridade dos sentimentos: positiva, negativa e neutra. Alguns estudos usam escalas mais detalhadas como muito negativo, negativo, neutro, positivo e muito positivo. As emoções mais comuns analisadas são raiva, nojo, medo, alegria, tristeza e surpresa.

3.2 Análise de Sentimento em Redes Sociais

Neste trabalho Nascimento, Osiek e Xexeo, 2015 utilizaram a análise de sentimento para determinar a polaridade de textos coletados do serviço Twitter. Segundo os autores, “Este trabalho visa a avaliar a reação das pessoas em relação às notícias compartilhadas na mídia por meio da análise de publicações feitas no Twitter. O objetivo é concluir, a partir dos sentimentos expostos nos tweets, se a população achou um determinado fato positivo ou negativo.”

Para isso, foram escolhidas três categorias de assunto a serem investigadas e, para cada uma delas, foi verificado se as notícias daquele dado tópico são vistas de modo positivo ou negativo.

Além disso, o estudo também busca realizar uma comparação entre três diferentes classificadores de linguagem para verificar qual deles se adequa melhor às características das mensagens coletadas através do Twitter e consegue obter melhor resultado ao tratar textos em português brasileiro.

Para a classificação dos textos foi usado modelos N-grama permitem prever a probabilidade de um grupo de palavras aparecerem em uma determinada sequência, a partir das N-1 palavras do N-grama-sequência de N palavras.

Por funcionar de acordo com um paradigma conhecido por treinamento e teste, este modelo utiliza uma base de treinamento que terá a função de ensinar ao classificador quais sequências de palavras estão associadas a uma determinada classificação, e uma outra base diferente para ser usada como teste para não mascarar o real desempenho dos classificadores.

Os modelos utilizados para classificar a base de dados foram:

- **Trigrama:** O classificador leva em conta o modelo N-grama e calcula a probabilidade de cada texto ser positivo ou negativo, analisando sequencias de 3 palavras.
- **Hexagrama:** O classificador leva em conta o modelo N-grama e calcula a probabilidade de cada texto ser positivo ou negativo, analisando sequencias de 6 palavras.
- **NAIVE_BAYES:** É um classificador em que os textos são representados como *bag of words*, ou seja, suas posições exatas são ignoradas e o classificador é montado com base em um modelo probabilístico baseado no teorema de Bayes, assumindo independência entre as variáveis, e também calcula a probabilidade do texto ser positivo.

A coleta dos dados para o experimento foi feita durante os meses de agosto a outubro de 2011 de maneira manual, focando em três categorias: Entretenimento, Política e Policial.

Durante a janela de tempo considerada, foram selecionadas notícias que não apenas tiveram grande repercussão entre os usuários do microblog, mas que também foram bastante divulgadas pelos jornais e revistas escritos e televisionados. Para cada categoria escolhida, foram selecionadas entre 2 e 3 notícias de cada categoria.

Após a seleção manual das notícias e das postagens relacionadas às noticias foi feito um levantamento de 400 *tweets* de cada notícia e os próprios pesquisadores fizeram uma pré-análise das postagens, os tweets como positivo e negativo, para que depois esta categorização manual fosse utilizada como base para a ferramenta de classificação automática.

Após este processo, foi criada uma base com cerca de 925 documentos, divididos em cerca de 50% positivos e 50% negativos.

A extração dos dados foi usada para criar uma ferramenta de classificação automática capaz de estabelecer para uma dada postagem a polaridade da opinião contida no texto analisado.

Para a construção desta ferramenta foi utilizada uma biblioteca para processamento de texto usando linguística computacional chamada *LingPipe*.

Por fim, a ferramenta foi utilizada para classificar as postagens, dividindo-se a base utilizada em 10 partes, tomando a proporção de uma parte para teste e as outras 9 para treino.

O processo de classificação foi realizado para cada um dos classificadores e tópicos de notícias escolhidos, levando em conta os seguintes parâmetros:

- **Acurácia:** Calcula a corretude do processo de classificação
- **Intervalo de confiança:** Estabelece o valor que, somado ou subtraído do parâmetro considerado, oferece um intervalo no qual estamos 95% certos de ter o valor verdadeiro. Quanto mais próximo de 0, maior é a confiança de que o resultado obtido está correto.

Observando a tabela 1 abaixo, podemos observar o desempenho dos classificadores para as categorias testadas e, dessa forma, observar que a classificação utilizando trigramas foi o que obteve o melhor desempenho:

Tabela 1 – Medidas dos classificadores para cada uma das categorias testadas.

	TRIGRAMA		**HEXAGRAMA**		**NAIVE_BAYES**	
	Acur.	IC	Acur.	IC	Acur.	IC
Entreterimento	0.5895	0.1448	0.5710	0.1457	0.5458	0.1457
Policial	0.8363	0.1660	0.8272	0.1830	0.7727	0.2210
Política	0.7368	0.1347	0.7421	0.1369	0.7342	0.1402

Fonte: Adaptado de (NASCIMENTO; OSIEK; XEXÉO, 2015)

3.3 Análise de sentimento em avaliações de comércio eletrônico

Assim como proposto no presente estudo, Vanaja e Belwal (2018) aplicam a análise de sentimentos em sites de comércio virtual. O intuito de realizar este tipo de estudo é auxiliar os serviços de lojas virtuais a entender melhor seus consumidores para poder melhorar sua experiência e aumentar as vendas.

Neste caso a análise de sentimento foi usada para classificar as opiniões dos usuários em positiva, negativa ou neutra, utilizando uma base de dados retirada dos comentários da loja virtual *Amazon* e foi utilizada a análise ao nível de aspecto para realizar esta classificação.

Além disso, o estudo citado se propôs a comparar dois algoritmos de análise de sentimentos aplicados a esta base de dados: *Naïve Bayes* e *Support Vector Machine*.

O autor dividiu a aplicação da análise de sentimento ao nível de aspecto em três etapas:

1. **Identificação:** Identificar os pares de alvo de emoção nas frases;
2. **Classificação:** Classificar os pares de alvo de emoção identificados como positivo, negativo ou neutro;
3. **Agregação:** Agregar os valores classificados para obter uma visão geral, dependendo dos requisitos específicos de uma aplicação.

Para realizar a análise, o autor utilizou um sistema e tratou os seguintes passos:

1. **Coleta de avaliações de clientes:** A entrada que consistiu nas avaliações de produtos da Amazon;
2. **Tagging de partes do discurso:** As avaliações em forma de texto são analisadas para categorizar cada palavra em substantivo, verbo, adjetivo, etc. Isso é feito para identificar principalmente adjetivos e substantivos que expressam emoções e opiniões sobre os produtos
3. **Extração das características:** Foi utilizado um algoritmo para extrair os termos mais frequentes e relevantes das avaliações.
4. **Remoção de características:** Etapa de remoção de características irrelevantes como as *stop words*
5. **Classificação:** Nesta etapa é realizada a classificação dos dados em positivo, negativo ou neutro utilizando os algoritmos Naïve Bayes e Support Vector Machine.

Na tabela 2 abaixo está o resultado da comparação feita entre os dois algoritmos:

Tabela 2 – Comparação dos algoritmos Naïve Bayes e SVM

	Naïve Bayes	Support Vector Machine
Acurácia	90.423	83.423
Precisão	0.947	0.852
Recall	0.959	0.83
F-Score	0.952	0.841

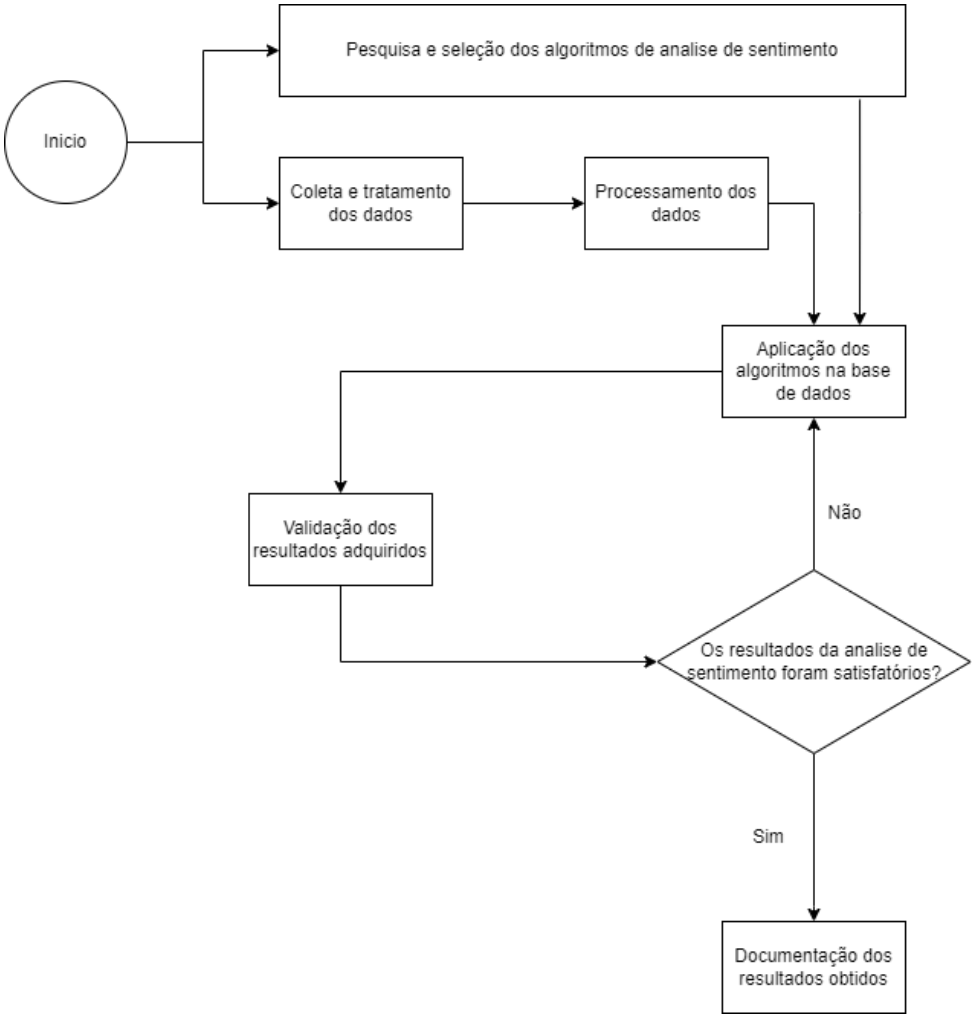
Fonte: adaptado de (VANAJA; BELWAL, 2018)

4 Materiais e Métodos

O presente estudo aplicou a análise de sentimentos em um banco de avaliações, portanto, ele pode ser classificado quanto aos objetivos como uma pesquisa de caráter exploratório. Segundo Fátima et al. (2011), os métodos empregados em uma pesquisa exploratória envolvem levantamentos em fontes secundárias, levantamentos de experiências, estudos de casos selecionados e observação informal. De acordo com Selltiz, (1965), a pesquisa exploratória visa descobrir ideias e intuições na tentativa de adquirir maior familiaridade com o fenômeno pesquisado e tais características vão ao encontro dos objetivos desta pesquisa.

A figura 8 abaixo representa o diagrama de fluxo dos procedimentos utilizados para alcançar o objetivo deste estudo, partindo da coleta e seleção dos algoritmos de análise de sentimentos até alcançar os resultados e sua documentação:

Figura 8 – Diagrama de fluxo dos procedimentos metodológicos.



Fonte: autor

A base de dados e a implementação dos três métodos/algoritmos utilizados neste estudo estão disponíveis na plataforma Google Colab através do link: bit.ly/3PWymn9

4.1 Coleta e tratamento dos dados

Os dados para realizar a análise de sentimento foram coletados a partir das avaliações da plataforma de comércio digital *Mercado Livre*. Esta plataforma foi escolhida pela facilidade de se realizar a coleta, uma vez que ela disponibilizava esses dados gratuitamente por meio de uma interface de programação de aplicações (API), e isso facilitou a coleta de uma abundância de dados de maneira automática através do código que vai ser apresentado nos algoritmos 5.1 e 5.2 na seção 5.1.

Após a coleta, a base de dados passou por um processo de tratamento que abrange tanto a base na totalidade quanto cada avaliação individualmente. O tratamento incluiu a conversão das notas dos usuários, variando originalmente de 1 a 5, para um sistema binário, no qual os valores 1 e 2 que representam as avaliações negativas foram convertidos para *"neg"* e os valores 4 e 5 que representam avaliações positivas foram convertidos para *"pos"*. Já as notas de valor 3 que representam análises neutras foram excluídas da base, uma vez que a avaliação de análises neutras não seria considerada nesse estudo. Ademais, foi acrescentado um cabeçalho ao arquivo para facilitar a leitura, e cada avaliação recebeu uma identificação (ID) única.

4.2 Separação da base de dados

De modo a buscar uma base equilibrada entre avaliações negativas e positivas, foi realizado um balanceamento da base, deletando parte dos dados para deixar o mesmo número de avaliações negativas e positivas.

Após esse balanceamento, a base de dados resultante foi dividida em quatro bases, sendo uma base com as avaliações balanceadas contendo 80% da base original, que foi responsável por fazer o treinamento e o teste dos algoritmos, e outras três bases para realizar a validação dos modelos treinados, sendo uma base balanceada contendo 10% da base original, uma base apenas de avaliações positivas contendo 5% da base original e uma base apenas de avaliações negativas contendo 5% da base original. Isso foi feito com o intuito de observar como cada uma das bases se comporta comparada com os resultados obtidos utilizando a base balanceada.

4.3 Processamento da base de dados

O procedimento de análise de sentimento compreende uma fase inicial de processamento dos dados, seguida pela aplicação dos algoritmos de análise de sentimento. O processamento dos dados é conduzido por meio das seguintes etapas:

1. Exclusão de *Stopwords*: Eliminação de palavras comuns e pouco informativas, visando concentrar a análise nas expressões mais significativas.

2. Exclusão de Pontuações: Remoção de sinais de pontuação, promovendo uniformidade e consistência no conteúdo textual.

3. Exclusão da Acentuação das Palavras: Normalização das palavras pela remoção de acentos, assegurando consistência na representação textual.

4. Padronização das Palavras: Uniformização das palavras para um formato comum, facilitando comparações e análises subsequentes.

5. *Stemização* das Frases: Implementação do algoritmo de stemming especializado para a língua portuguesa para remover sufixos das palavras. A técnica de stemming visa reduzir cada palavra à sua forma raiz ou radical, eliminando sufixos. Esse processo simplifica o texto e aprimora a correspondência de palavras em pesquisas ou análises de texto.

4.4 Vetorização da base de dados

Após o processamento da base de dados, é necessário realizar a *vetorização* do texto. Este processo transforma as avaliações em um vetor de valores numéricos para poderem ser processados pelos modelos de análise de sentimento. Para isso, foram utilizados dois modelos de vetorização, o *CountVectorizer* e o *Term Frequency– Inverse Document Frequency*. Ainda foram comparados ambos os métodos de vetorização para descobrir qual teria a melhor acurácia. Após passar pela vetorização, a base de dados resultante foi dividida em conjuntos de treino e teste, utilizando, respectivamente, 80% e 20% da base de dados.

4.5 Análise e aplicação dos algoritmos de aprendizado de máquina

Após a vetorização do texto, foi realizada a análise de sentimentos utilizando três algoritmos de aprendizado de máquina: Regressão Logística, Árvore de Decisão e Floresta Aleatória. Cada um desses modelos foi escolhido por suas características específicas, que oferecem vantagens distintas para a tarefa de classificação de sentimentos.

4.6 Análise dos resultados

Após o treinamento e validação dos dados, realizou-se a análise comparativa dos resultados, buscando identificar qual modelo apresentou melhor desempenho na classificação dos sentimentos, considerando tanto a base de treinamento quanto os dados de validação. Essa abordagem permitiu verificar a eficácia dos modelos em identificar corretamente os sentimentos, bem como explorar o impacto da vetorização e da divisão da base de dados nos resultados.

5 Desenvolvimento

5.1 Coleta e tratamento dos dados

Como mencionado na seção 4.1, a coleta de dados foi feita por meio de uma *API* disponibilizada pelo Mercado Livre. A coleta dos dados foi realizada em três etapas. Na primeira etapa, foram obtidos os identificadores únicos de cada categoria do produto de forma manual através da plataforma; na segunda etapa, essas categorias foram usadas para coletar uma lista dos identificadores únicos dos produtos. No Algoritmo 5.1 esquematizado abaixo, pode-se observar como essa coleta foi feita.

```
1 def buscarItens(categoriaId, offset):
2     x = requests.get(
3         'https://api.mercadolibre.com/sites/MLB/search?category='+categoriaId+
    '&limit=50&offset='+str(offset))
4     retornoJson = x.json()
5     return retornoJson['results']
```

Algoritmo 5.1 – função para buscar os itens

Na terceira etapa, esses identificadores dos produtos foram utilizados para coletar os dados das avaliações propriamente ditas, utilizando o Algoritmo 5.2 apresentado abaixo:

```
1 def buscarReviews(itemId):
2     x = requests.get(
3         'https://api.mercadolibre.com/reviews/item/'+itemId+'?limit=200')
4     retornoJson = x.json()
5     return retornoJson['reviews']
```

Algoritmo 5.2 – função para buscar reviews

Abaixo é exemplificado como as frases coletadas estavam no formato em que foram coletadas:

- *O original Ótimo! e igual o zé vaqueiro original :).;5*
- *Ruim Das que já tive,esta é a pior. O ajuste do jato é uma piada. E a pressão muito fraca. Só não troquei ,pela burocracia.;2*
- *Enganado... Estão agindo de má fé na descrição. Ruim, pois a propaganda está enganosa.;1*
- *Lindo!! Muito fofo e lindo! os vasos são pequenos o que deixa o ambiente mais delicado e charmoso.;5*
- *Ótimo Produto muito bom ,veio direitinho recomendo.;2*

- *Bom Achei boa,mas não tem o jato fino só o jato leque se for preciso tirar uma sujeira mais agarrado não tem como sem o jato fino.;3*

Após a coleta e o armazenamento dos dados, inicia-se o tratamento dos dados em que foi feita uma limpeza geral nos dados para remover entradas inválidas que não tivessem avaliação, retirar os espaços em branco no começo e no fim das frases e adicionar um identificador para as avaliações.

Também foi feito um tratamento para excluir as avaliações de valor neutro (avaliações com a nota 3) já que podem prejudicar a análise de sentimentos que será aplicada, visto que esse tipo de avaliação não é considerado nem positivo nem negativa; ela foge do objetivo da análise que está sendo feita nos dados. Por fim, foi adicionada a esta parte do tratamento a alteração das notas para "pos"nas avaliações com nota 4 e 5 para indicar que são avaliações positivas e "neg"para avaliações com notas 1 e 2 que indicam serem negativas. Isso pode ser conferido na íntegra no algoritmo 5.3 abaixo:

```
1  def modify_lines(input_filename, output_filename):
2      with open(input_filename, 'r', encoding='utf-8') as input_file, open(
       output_filename, 'w', encoding='utf-8') as output_file:
3          line_number = 1
4          for line in input_file:
5              line = line.strip()
6              parts = line.rsplit(";", 1)
7              if len(parts) != 2 or not parts[1].isdigit() or parts[1] == "3":
8                  continue
9              content, rating = parts
10             content = content.strip()
11
12             if rating in {"4", "5"}:
13                 rating = "pos"
14             elif rating in {"1", "2"}:
15                 rating = "neg"
16
17             modified_line = f'"{line_number}","{content}","{rating}"\n'
18             output_file.write(modified_line)
19             line_number += 1
20
21 input_filename = 'base.csv'
22 output_filename = 'output.csv'
23
24 modify_lines(input_filename, output_filename)
25 print("Linhas modificadas e escritas no arquivo de saida.")
```

Algoritmo 5.3 – função para tratamento dos dados

Após este tratamento dos dados, temos uma nova base de dados que se assemelha ao apresentado abaixo:

- *"116", "O original Ótimo! e igual o zé vaqueiro original :)."";"pos"*

- *"522","Ruim Das que já tive,esta é a pior. O ajuste do jato é uma piada. E a pressão muito fraca. Só não troquei ,pela burocracia.","neg"*
- *"22243","Enganado... Estão agindo de má fé na descrição. Ruim, pois a propaganda está enganosa.","neg"*
- *"22281","Lindo!! Muito fofo e lindo! os vasos são pequenos o que deixa o ambiente mais delicado e charmoso.","pos"*
- *"450","Otimo Produto muito bom ,veio direitinho recomendo.","neg"*

A coleta e o tratamento de dados resultaram em uma base de 57.830 registros, sendo 50.803 avaliações positivas e 7.027 avaliações negativas.

5.2 Separação da base de dados

No Algoritmo 5.4 abaixo pode ser observado como foi feita a exclusão de parte da base de dados para termos uma base mais balanceada entre avaliações positivas e negativas:

```
1  def create_partial_dataset(input_filename, output_filename, stop_condition):
2      with open(input_filename, 'r', encoding='utf-8') as input_file, \
3           open(output_filename, 'w', encoding='utf-8') as output_file:
4
5          line_number = 1
6          neg_count = 0
7          pos_count = 0
8          for line in input_file:
9              line = line.strip()
10             parts = line.split(',')
11             if len(parts) == 3:
12                 content = parts[1]
13                 label = parts[2]
14
15                 if "neg" in label and neg_count <= stop_condition:
16                     output_file.write(f'"{line_number}",{content.strip()},{
       label}\n')
17                     neg_count += 1
18                     line_number += 1
19
20
21                 if "pos" in label and pos_count <= stop_condition:
22                     output_file.write(f'"{line_number}",{content.strip()},{
       label}\n')
23                     pos_count += 1
24                     line_number += 1
25
26
27 input_filename = 'output.csv'
28 output_partial_filename = 'base_mae.csv'
29 stop_condition = 7259
```

```
30 create_partial_dataset(input_filename, output_partial_filename, stop_condition
       )
31 print(f"Novo arquivo parcial criado: {output_partial_filename}")
```

Algoritmo 5.4 – função para tratamento dos dados

E isso resultou em uma base com 14.287 avaliações igualmente divididas entre avaliações positivas e negativas. A fim de fazer uma validação posterior, essa base foi redividida entre quatro bases:

- **Base balanceada**: A base principal e utilizada para o treinamento e os testes iniciais, foi construída com 80% da base original (11.429 avaliações), mantendo o equilíbrio entre avaliações positivas e negativas. A Figura 9 mostra a nuvem de palavras gerada a partir dessa base, destacando os termos mais frequentes nas avaliações.

Figura 9 – Nuvem de palavras para base balanceada.

Fonte: autor

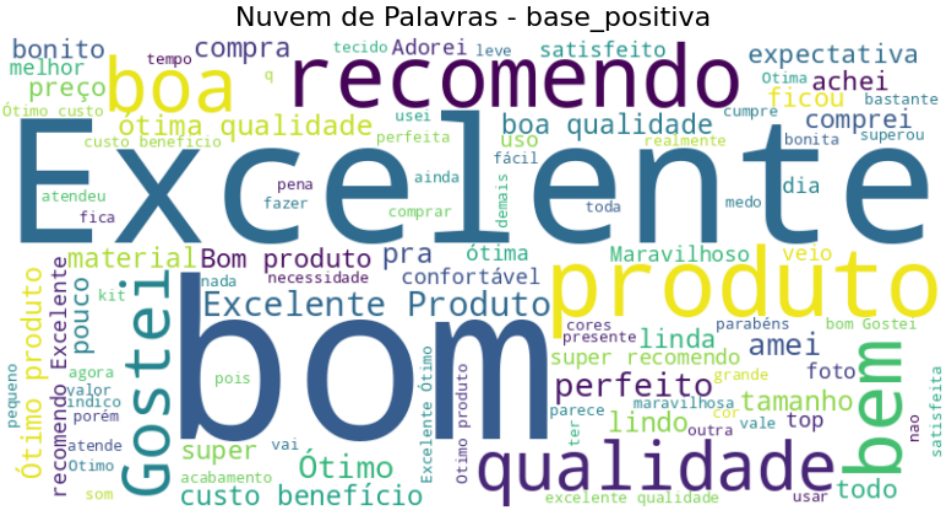
- **Base balanceada validação**: utilizada exclusivamente para validar o modelo já treinado, essa base foi construída com 10% da base original (1.428 avaliações), mantendo o equilíbrio entre avaliações positivas e negativas. A Figura 10 apresenta a nuvem de palavras correspondente a essa base, evidenciando a distribuição dos termos mais relevantes.

Figura 10 – Nuvem de palavras para base balanceada de validação.

Fonte: autor

- **Base positiva validação**: utilizada exclusivamente para validar o modelo já treinado, essa base foi construída com 5% da base original (714 avaliações) e contém apenas avaliações positivas. A Figura 11 ilustra a nuvem de palavras gerada a partir dessa base, destacando os termos mais associados a sentimentos positivos.

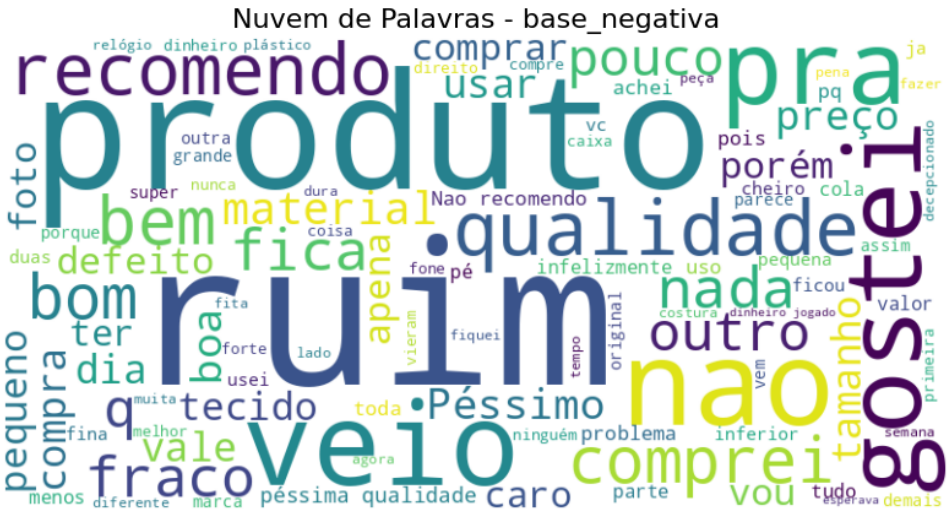
Figura 11 – Nuvem de palavras para base positiva.



Fonte: Autor

- **Base negativa validação**: utilizada exclusivamente para validar o modelo já treinado, essa base foi construída com 5% da base original (714 avaliações) e contém apenas avaliações negativas. A Figura 12 exibe a nuvem de palavras correspondente a essa base, com foco nos termos mais relacionados a sentimentos negativos.

Figura 12 – Nuvem de palavras para base negativa.



Fonte: Autor

Para cada uma das bases, foi gerada uma nuvem de palavras que representa visualmente a frequência dos termos mais relevantes presentes nas avaliações. O tamanho de cada palavra na nuvem é proporcional à sua frequência no conjunto de dados correspondente, permitindo uma rápida visualização dos termos mais significativos em cada base. É possível observar que, apesar das diferentes proporções das bases, os padrões de palavras mais frequentes se mantêm relativamente consistentes na base balanceada e na base balanceada de validação, enquanto apresentam variações características nas bases positiva e negativa de validação, refletindo a natureza específica das avaliações em cada uma delas.

Essa divisão pode ser observada no algoritmo 5.5 abaixo:

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3
4 # Ler os dados do arquivo CSV
5 df = pd.read_csv('base_mae.csv', header=None, names=['id', 'review', '
      sentiment'])
6
7 # Separar os dados positivos e negativos
8 pos_df = df[df['sentiment'] == 'pos']
9 neg_df = df[df['sentiment'] == 'neg']
10
11 # Dividir os dados em 80%, 10%, 5% e 5%
12 # Parte 1: 80% balanceada
13 pos_80, pos_remaining = train_test_split(pos_df, test_size=0.2, random_state
      =42)
14 neg_80, neg_remaining = train_test_split(neg_df, test_size=0.2, random_state
      =42)
15
16 part1 = pd.concat([pos_80, neg_80]).sample(frac=1, random_state=42).
      reset_index(drop=True)
```

```
17
18 # Parte 2: 10% balanceada
19 pos_10, pos_remaining = train_test_split(pos_remaining, test_size=0.5,
       random_state=42)
20 neg_10, neg_remaining = train_test_split(neg_remaining, test_size=0.5,
       random_state=42)
21
22 part2 = pd.concat([pos_10, neg_10]).sample(frac=1, random_state=42).
       reset_index(drop=True)
23
24 # Parte 3: 5% apenas positivas
25 part3 = pos_remaining.sample(frac=0.5, random_state=42).reset_index(drop=True)
26
27 # Parte 4: 5% apenas negativas
28 part4 = neg_remaining.sample(frac=0.5, random_state=42).reset_index(drop=True)
29
30 part1.to_csv('base_balanceada.csv', index=False, header=False, quoting=1)
31 part2.to_csv('base_balanceada_validacao.csv', index=False, header=False,
       quoting=1)
32 part3.to_csv('base_positiva.csv', index=False, header=False, quoting=1)
33 part4.to_csv('base_negativa.csv', index=False, header=False, quoting=1)
```

Algoritmo 5.5 – função para tratamento dos dados

5.3 Processamento de texto

Para realizar o processamento da base, os dados são carregados a partir do arquivo armazenado e, em seguida, os valores categóricos “neg” e “pos” são convertidos em valores binários: “0” para resenhas negativas e “1” para positivas. O resultado desse processamento é armazenado em um *DataFrame* denominado ’resenha’, que serve como base para todo o restante do processo. Esse processamento pode ser conferido no Algoritmo 5.6 abaixo:

```
1 resenha = pd.read_csv("balanced_output.csv")
2 classificacao = resenha["sentiment"].replace(["neg", "pos"], [0, 1])
3 resenha["classificacao"] = classificacao
```

Algoritmo 5.6 – leitura da base de dados

Baseado nos itens descritos na seção 4.3 foi feita uma função para fazer o processamento da base de dados. Ele realiza os seguintes passos:

1. Conversão das letras para minúsculo
2. Remoção dos acentos das palavras
3. Eliminação das pontuações
4. Aplicação da função de remoção de duplicatas
5. *Tokenização* do texto

6. Remoção das *stopwords*
7. Aplicação do *stemmer* nas palavras

O código do tratamento das resenhas pode ser conferido no Algoritmo 5.7 abaixo:

```
def tratamento_unico(resenha):
    frase_processada = list()
    stemmer = nltk.RSLPStemmer()
    palavras_irrelevantes = nltk.corpus.stopwords.words("portuguese")

    for opiniao in resenha.text_pt:
        nova_frase = list()
        opiniao = opiniao.lower()
        opiniao = unidecode.unidecode(opiniao)
        opiniao = re.sub(r'nao', 'n o', opiniao)
        opiniao = re.sub(f"[{re.escape(punctuation)}]", " ", opiniao)
        opiniao = remover_letras_duplicadas(opiniao)
        palavras_texto = tokenize.WhitespaceTokenizer().tokenize(opiniao)
        for palavra in palavras_texto:
            if palavra not in palavras_irrelevantes:
                nova_frase.append(stemmer.stem(palavra))
        frase_processada.append(' '.join(nova_frase))
    resenha["tratamento_unico"] = frase_processada
    return resenha
```

Algoritmo 5.7 – tratamento da base da dados

O resultado do processamento pode ser visualizado abaixo:

- *orig otim igual ze vaqu orig*
- *ruim ja pi ajust jat pi presa frac so nao troq burocrac*
- *engan esta agind ma fe descrica ruim poi propagand engan*
- *lind fof lind vas sao pequen deix ambi delic charm*
- *otim produt bom vei direit recom*

5.4 Vetorização dos textos

Após o processamento dos dados, foi realizada a vetorização dos textos, onde os termos das avaliações são convertidos em um valor numérico para poderem ser processados. Para isso, foi considerada a divisão dos tokens em unigramas e bigramas, e também duas formas diferentes de realizar a vetorização: o *Count Vectorize* e o *TF-IDF*. Isso foi feito para podermos comparar qual o tipo de vetorização trará um melhor resultado.

No Algoritmo 5.8 descrito abaixo pode-se ver como a vetorização foi realizada:

```
# Unigrama CountVectorizer
vect_uni_cv = CountVectorizer(
    ngram_range=(1,1),
    stop_words=stop_words,
    min_df=2,
    max_features=5000
)
vect_uni_cv.fit(resenha.tratamento_unico)
text_vect_uni_cv = vect_uni_cv.transform(resenha.tratamento_unico)

# Unigrama IDF
vect_uni_idf = TfidfVectorizer(
    ngram_range=(1,1),
    use_idf=True,
    norm='l2',
    stop_words=stop_words,
    min_df=2,
    max_features=5000
)
vect_uni_idf.fit(resenha.tratamento_unico)
text_vect_uni_idf = vect_uni_idf.transform(resenha.tratamento_unico)

# Bigrama CountVecrotizer
vect_bi_cf = CountVectorizer(
    ngram_range=(2,2),
    stop_words=stop_words,
    min_df=2
)
vect_bi_cf.fit(resenha.tratamento_unico)
text_vect_bi_cf = vect_bi_cf.transform(resenha.tratamento_unico)

# Bigrama IDF
vect_bi_idf = TfidfVectorizer(
    ngram_range=(2,2),
    use_idf=True,
    norm='l2',
    stop_words=stop_words,
    min_df=2
)
vect_bi_idf.fit(resenha.tratamento_unico)
text_vect_bi_idf = vect_bi_idf.transform(resenha.tratamento_unico)
```

Algoritmo 5.8 – Preparação da base para análise

As funções receberam como parâmetro o ***ngram_range*** que determina quantas palavras serão usadas para a divisão dos *tokens*. Foram informados os valores *(1,1)* para os unigramas, considerando palavras individuais, e *(2,2)* para bigramas, analisando pares de palavras consecutivas. Também foram informadas as ***stop_words*** para remover as palavras pré-definidas anteriormente como termos que não fariam diferença significativa na análise, como artigos e preposições.

Além disso, temos também o parâmetro ***use_idf*** que controla se o componente *Inverse Document Frequency* (IDF) será usado no cálculo. Quando ativado, este parâmetro penaliza palavras que aparecem com muita frequência em diversos documentos, dando maior peso a termos mais distintivos. O parâmetro ***norm*** controla como os vetores serão normalizados, sendo que a normalização ***L2*** (euclidiana) foi escolhida para ajustar a escala dos valores, permitindo que documentos de diferentes tamanhos sejam comparados de forma mais eficaz.

Esta combinação de parâmetros é considerada uma boa prática pois considera a importância relativa das palavras no corpus completo, normaliza os vetores removendo o viés do tamanho do documento e torna os documentos mais comparáveis entre si, contribuindo para melhorar o desempenho do algoritmo de classificação. A normalização também ajuda a focar no conteúdo real dos textos, em vez de ser influenciada pelo tamanho dos documentos.

5.5 Separação em conjuntos de treinamento e teste

Agora, após o processamento, é feita a divisão em conjuntos de treinamento e teste para poder realizar a análise de sentimento da base de dados, como pode ser vista no algoritmo 5.9 abaixo:

```
1
2 # Unigrama CV
3 X_trainUCV, X_testUCV, y_trainUCV, y_testUCV = train_test_split(
      text_vect_uni_cv, resenha["classificacao"], test_size=0.2, random_state
      =123)
4
5 # Unigrama IDF
6 X_trainUIDF, X_testUIDF, y_trainUIDF, y_testUIDF = train_test_split(
      text_vect_uni_idf, resenha["classificacao"], test_size=0.2, random_state
      =123)
7
8 # Bigrama CV
9 X_trainBCV, X_testBCV, y_trainBCV, y_testBCV = train_test_split(
      text_vect_bi_cf, resenha["classificacao"], test_size=0.2, random_state
      =123)
10
11 # Bigrama IDF
12 X_trainBIDF, X_testBIDF, y_trainBIDF, y_testBIDF = train_test_split(
      text_vect_bi_idf, resenha["classificacao"], test_size=0.2, random_state
      =123)
```

Algoritmo 5.9 – Separação da base em treino e teste

O parâmetro *test_size* escolhido foi 20%, o que corresponde a 2.285 avaliações, e o valor do *random_state* foi escolhido de maneira arbitrária apenas para ter a certeza de que a divisão dos dados é fixa, garantindo a reprodutibilidade da análise.

5.6 Treinamento dos modelos

Com os dados pré-processados e devidamente vetorizados, com o auxílio da biblioteca *sklearn* foi feito o processamento dos dados com cada tipo de vetorização feito na etapa 5.4.

No algoritmo 5.10 abaixo, é feita a configuração dos parâmetros do modelo árvore de decisão.

```
dt_params = {
    'max_depth': 15,
    'min_samples_split': 10,
    'min_samples_leaf': 5,
    'random_state': 123,
    'class_weight': 'balanced'
}
```

Algoritmo 5.10 – Parametrização do modelo Árvore de decisão

dt_params: parâmetros usados para o algoritmo de árvore de decisão, como profundidade máxima da árvore (*max_depth*), número mínimo de amostras necessárias para dividir um nó (*min_samples_split*), número mínimo de amostras em um nó folha (*min_samples_leaf*), estado aleatório (*random_state*) para garantir reprodutibilidade, e *class_weight* ajustado para '*balanced*' para lidar com classes desbalanceadas.

As configurações do modelo de florestas aleatórias podem ser visualizadas no Algoritmo 5.11 abaixo:

```

rf_params = {
    'n_estimators': 200,
    'max_depth': 15,
    'min_samples_split': 10,
    'min_samples_leaf': 5,
    'random_state': 123,
    'class_weight': 'balanced',
    'n_jobs': -1
}
```

Algoritmo 5.11 – Parametrização do modelo Florestas aleatórias

rf_params: parâmetros para o algoritmo de florestas aleatórias, incluindo número de árvores na floresta (*n_estimators*), além dos parâmetros comuns com o modelo de árvore de decisão, e *n_jobs* configurado para -1 para usar todos os processadores disponíveis. As configurações feitas podem ser visualizadas no Algoritmo 5.12 abaixo:

```
lr_params = {
    'random_state': 123,
    'max_iter': 1000,
    'class_weight': 'balanced',
    'C': 1.0
```

```
}
```

Algoritmo 5.12 – Parametrização do modelo Regressão Logística

lr_params: Parâmetros para Regressão logística, como o número máximo de iterações (*max_iter*), peso da classe (*class_weight*) para balancear as classes, e "C" que controla a força da regularização.

Após serem feitas as configurações, é feito de fato o treinamento dos modelos de análise de sentimentos, respectivamente, da árvore de decisão (Algoritmo 5.13), floresta aleatória (Algoritmo 5.14) e regressão logística (Algoritmo 5.15):

```
treeUCV = DecisionTreeClassifier(**dt_params)
treeUCV.fit(X_trainUCV, y_trainUCV)
models['Unigrama DT CV'] = (vect_uni_cv, treeUCV)

treeUIDF = DecisionTreeClassifier(**dt_params)
treeUIDF.fit(X_trainUIDF, y_trainUIDF)
models['Unigrama DT TFIDF'] = (vect_uni_idf, treeUIDF)

treeBCV = DecisionTreeClassifier(**dt_params)
treeBCV.fit(X_trainBCV, y_trainBCV)
models['Bigrama DT CV'] = (vect_bi_cf, treeBCV)

treeBIDF = DecisionTreeClassifier(**dt_params)
treeBIDF.fit(X_trainBIDF, y_trainBIDF)
models['Bigrama DT TFIDF'] = (vect_bi_idf, treeBIDF)
```

Algoritmo 5.13 – Aplicação do algoritmo Árvore de Decisão

```

rfUCV = RandomForestClassifier(**rf_params)
rfUCV.fit(X_trainUCV, y_trainUCV)
models['Unigrama RF CV'] = (vect_uni_cv, rfUCV)

rfUIDF = RandomForestClassifier(**rf_params)
rfUIDF.fit(X_trainUIDF, y_trainUIDF)
models['Unigrama RF TFIDF'] = (vect_uni_idf, rfUIDF)

rfBCV = RandomForestClassifier(**rf_params)
rfBCV.fit(X_trainBCV, y_trainBCV)
models['Bigrama RF CV'] = (vect_bi_cf, rfBCV)

rfBIDF = RandomForestClassifier(**rf_params)
rfBIDF.fit(X_trainBIDF, y_trainBIDF)
models['Bigrama RF TFIDF'] = (vect_bi_idf, rfBIDF)
```

Algoritmo 5.14 – Aplicação do algoritmo da Floresta Aleatória

```

```

```
lrUCV = LogisticRegression(**lr_params)
lrUCV.fit(X_trainUCV, y_trainUCV)
models['Unigrama LR CV'] = (vect_uni_cv, lrUCV)

lrUIDF = LogisticRegression(**lr_params)
lrUIDF.fit(X_trainUIDF, y_trainUIDF)
models['Unigrama LR TFIDF'] = (vect_uni_idf, lrUIDF)

lrBCV = LogisticRegression(**lr_params)
lrBCV.fit(X_trainBCV, y_trainBCV)
models['Bigrama LR CV'] = (vect_bi_cf, lrBCV)

lrBIDF = LogisticRegression(**lr_params)
lrBIDF.fit(X_trainBIDF, y_trainBIDF)
models['Bigrama LR TFIDF'] = (vect_bi_idf, lrBIDF)
```

Algoritmo 5.15 – Aplicação do algoritmo da Regressão Logística

5.7 Teste dos modelos

Com a aplicação dos algoritmos de análise de sentimentos, são extraídas as acurácias de cada um, e elas podem ser comparadas para descobrir, dentre os modelos apresentados, qual apresenta a melhor acurácia. O algoritmo 5.16 apresentado abaixo foi usado para calcular a acurácia de cada modelo, utilizando a base de teste para fazer o cálculo:

```
print("/\nAvaliacao dos modelos:")
for nome, (vectorizer, model) in models.items():
    if 'Unigrama' in nome:
        if 'CV' in nome:
            acc = accuracy_score(y_testUCV, model.predict(X_testUCV))
        else:
            acc = accuracy_score(y_testUIDF, model.predict(X_testUIDF))
    else:
        if 'CV' in nome:
            acc = accuracy_score(y_testBCV, model.predict(X_testBCV))
        else:
            acc = accuracy_score(y_testBIDF, model.predict(X_testBIDF))
    print(f"{nome}: {acc:.4f}")
```

Algoritmo 5.16 – Avaliação dos modelos

A tabela 3 apresenta o resultado da acurácia de cada método de análise de sentimento:

Tabela 3 – Acurácia dos métodos para as bases de treinamento e teste:

Método analisado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	0.9339	0.9146	0.9305	0.9121
Floresta Aleatória	0.9546	0.9413	0.9425	0.9396
Regressão Logística	**0.9623**	0.9602	0.9575	0.9586

Fonte: autor

Esta acurácia apresentada é um indicador da proporção de previsões corretas feitas pelo modelo, refletindo sua eficiência em classificar corretamente as instâncias de dados de teste.

Após o treinamento dos modelos, utilizamos frases para testar e poder avaliar a eficiência dos modelos utilizando o Algoritmo 5.17 abaixo:

```
frases_teste = [
    "O original  timo ! e igual o z  vaqueiro original :).",
    "Ruim Das que j  tive,esta  a pior. O ajuste do jato  uma piada. E a
    press o muito fraca. S  n o troquei ,pela burocracia.",
    "Enganado... Est o agindo de m  f  na descri  o. Ruim, pois a
    propaganda est  enganosa.",
    "Lindo!! Muito fofo e lindo! os vasos s o pequenos o que deixa o ambiente
     mais delicado e charmoso.",
    "Otimo Produto muito bom ,veio direitinho recomendo.",
]

print("\nTestando frases de exemplo:")
for frase in frases_teste:
    resultado = analisar_sentimento_completo(frase, models)
    print(f"\nFrase: {resultado['frase_original']}")
    print("Resultados:")
    for r in resultado['resultados']:
        print(f"Modelo: {r['modelo']}")
        print(f"Sentimento: {r['sentimento']}")
        print(f"Probabilidade: {r['probabilidade']}%")
```

Algoritmo 5.17 – Teste com frases de exemplo:

A partir das frases trabalhadas no exemplo da figura , avaliando uma a uma, foram criadas as seguintes tabelas de resultados da análise de sentimentos:

- **Frase:** "O original Ótimo! e igual o zé vaqueiro original :)."
- **Avaliação do usuário:** Positiva

Tabela 4 – Resultado da avaliação da frase 1

Modelo utilizado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	P	P	P	P
Floresta Aleatória	P	P	P	P
Regressão logística	P	P	P	P

- **Frase:** "Ruim Das que já tive,esta é a pior. O ajuste do jato é uma piada. E a pressão muito fraca. Só não troquei ,pela burocracia."
- **Avaliação do usuário:** Negativa

Tabela 5 – Resultado da avaliação da frase 2

Modelo utilizado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	N	P	N	P
Floresta Aleatória	N	N	N	N
Regressão logística	N	N	N	N

- **Frase:** "Enganado... Estão agindo de má fé na descrição. Ruim, pois a propaganda está enganosa."
- **Avaliação do usuário:** Negativa

Tabela 6 – Resultado da avaliação da frase 3

Modelo utilizado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	N	P	N	P
Floresta Aleatória	N	N	N	N
Regressão logística	N	N	N	N

- **Frase:** "Lindo!! Muito fofo e lindo! os vasos são pequenos o que deixa o ambiente mais delicado e charmoso."
- **Avaliação do usuário:** Positiva

Tabela 7 – Resultado da avaliação da frase 4

Modelo utilizado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	P	P	P	P
Floresta Aleatória	P	P	P	P
Regressão logística	P	P	P	P

- **Frase:** "Otimo Produto muito bom ,veio direitinho recomendo."

- **Avaliação do usuário:** Negativa

Tabela 8 – Resultado da avaliação da frase 5

Modelo utilizado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	P	P	P	P
Floresta Aleatória	P	P	P	P
Regressão logística	P	N	P	P

5.8 Validação dos modelos

Utilizando as bases de validação divididas na seção 5.2 foi feita uma validação para atestar a acurácia do modelo treinado, usando todas elas como base de treino da seguinte forma:

```
def validar_bases_de_teste(bases_de_teste, models):
    resultados_validacao = {}

    for i, base in enumerate(bases_de_teste):
        df_teste = pd.read_csv(base)
        df_teste = tratamento_unico(df_teste)

        if 'sentiment' in df_teste.columns:
            df_teste['classificacao'] = df_teste['sentiment'].replace(['neg',
    'pos'], [0, 1])

        resultados_modelos = {}
        for nome_modelo, (vectorizer, model) in models.items():
            vetor_teste = vectorizer.transform(df_teste['tratamento_unico'])
            predicoes = model.predict(vetor_teste)
            acuracia = accuracy_score(df_teste['classificacao'], predicoes)
            resultados_modelos[nome_modelo] = acuracia

        resultados_validacao[f'Base de Teste {i+1}'] = resultados_modelos

    for base, resultados in resultados_validacao.items():
        print(f"\nResultados para {base}:")
        for modelo, acuracia in resultados.items():
            print(f"Modelo: {modelo} - Acur cia: {acuracia:.4f}")

    return resultados_validacao

path = "/content/drive/MyDrive/Colab Notebooks/"
bases_de_teste = [path+"base_balanceada_validacao.csv", path+"base_positiva.
    csv", path+"base_negativa.csv"]
resultados_validacao = validar_bases_de_teste(bases_de_teste, models)
```

Algoritmo 5.18 – Validação dos modelos

O Algoritmo 5.18 exibido acima mostra como foi feita a validação utilizando as bases auxiliares. Utilizando este código, obtivemos o seguinte resultado:

Tabela 9 – Acurácia dos métodos para a Base de validação balanceada:

Método analisado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	0.9305	0.9177	0.9288	0.9161
Floresta Aleatória	0.9518	0.9457	0.9426	0.9440
Regressão Logística	0.9633	0.9623	0.9587	0.9642

Fonte: autor

Tabela 10 – Acurácia dos métodos para a Base de validação positiva:

Método analisado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	0.9409	0.9921	0.9315	0.9906
Floresta Aleatória	0.9610	0.9728	0.9476	0.9681
Regressão Logística	0.9681	0.9807	0.9598	0.9772

Fonte: autor

Tabela 11 – Acurácia dos métodos para a Base de validação negativa:

Método analisado	**Count Vectorizer**		**Tfidf Vectorizer**	
	Unigramas	**Bigramas**	**Unigramas**	**Bigramas**
Árvore de decisão	0.9261	0.3864	0.9318	0.3835
Floresta Aleatória	0.9432	0.7699	0.9432	0.7955
Regressão Logística	0.9347	0.8381	0.9489	0.8778

Fonte: autor

5.9 Discussão dos Resultados

A análise dos resultados obtidos com os diferentes modelos de aprendizado de máquina para a tarefa de análise de sentimentos revela percepções importantes sobre o desempenho e a consistência de cada método. Para facilitar a visualização e interpretação desses resultados, os gráficos abaixo foram gerados com base nas tabelas de acurácia aplicadas a cada base de dados.

Os resultados de acurácia apresentados na Tabela 3 indicam que o modelo de Regressão Logística obteve as maiores taxas de acurácia em todas as configurações de vetorizadores, tanto para unigramas quanto para bigramas, atingindo a melhor acurácia ao utilizar o modelo de vetorização Count Vectorizer com unigramas. Isso sugere que a Regressão Logística é mais eficiente em capturar a estrutura subjacente dos dados de sentimentos em comparação com os modelos de Árvore de Decisão e Floresta Aleatória. Essa tendência pode ser observada de forma clara nos gráficos de desempenho (Figuras 13 14 15 16), que comparam a acurácia dos modelos em diferentes bases de dados.

Os modelos Floresta Aleatória e Árvore de Decisão também demonstraram um bom desempenho, especialmente quando aplicados a dados vetorizados com unigramas. No entanto, entre esses dois modelos, sua performance foi ligeiramente inferior à Regressão logística, particularmente com bigramas, o que indica que os modelos podem não capturar tão bem a importância relativa dos termos em documentos longos.

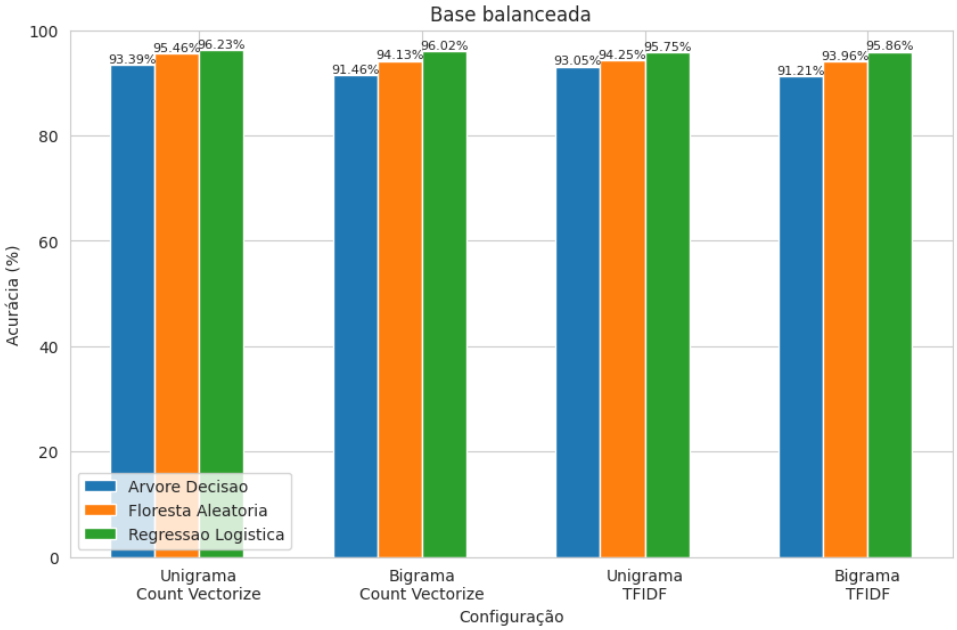
Os resultados da validação cruzada com diferentes bases de teste reafirmaram as tendências observadas durante a fase de treinamento. Como pode ser observado com as acurácias presentes nas Tabelas 9, 10 e 11 a Regressão Logística manteve-se como o modelo de maior acurácia, com desempenhos superiores em bases balanceadas, positivas e negativas. Notou-se também que a Floresta Aleatória teve um desempenho consistente, embora inferior à Regressão Logística, enquanto a Árvore de Decisão permaneceu como o modelo com pior desempenho, principalmente ao lidar com as avaliações negativas. Esses padrões são visualmente representados nos gráficos, que permitem uma comparação direta entre os modelos em diferentes cenários.

A análise das frases individuais apresentadas nas Tabelas 4, 5, 6, 7 e 8 demonstrou a consistência da Regressão Logística em classificar corretamente tanto sentimentos positivos quanto negativos. Este modelo foi o mais confiável em suas previsões, especialmente com unigramas e utilizando Tfidf Vectorizer, conforme visto na Tabela 3 de resultados de avaliação das frases.

Por outro lado, a Árvore de Decisão frequentemente falhou ao lidar com frases negativas, como observado na avaliação da frase *"Enganado... Estão agindo de má fé na descrição. Ruim, pois a propaganda está enganosa."*, onde houve erros ao classificar sentimentos negativos como positivos ao usar bigramas. Essa limitação também é evidenciada no gráfico presente na Figura 16, que mostra uma queda significativa na acurácia desse modelo em bases com avaliações negativas.

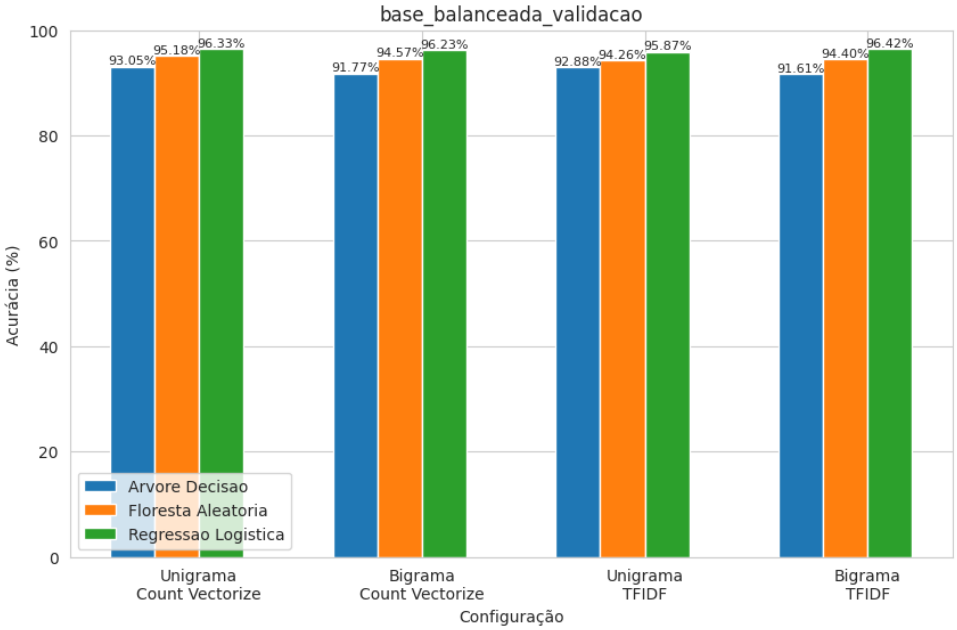
Conclui-se que, para aplicações práticas de análise de sentimentos, a Regressão Logística, especialmente quando combinada com Count Vectorizer e unigramas, oferece uma solução consistente e eficaz. Os resultados também sugerem que, embora outros métodos como a Floresta Aleatória possam oferecer vantagens em certos cenários, sua complexidade adicional pode nem sempre se traduzir em ganhos significativos de desempenho em relação a outros métodos como a Regressão Logística. A visualização desses resultados por meio dos gráficos reforça a clareza e a confiabilidade das conclusões obtidas.

Figura 13 – Gráfico acurácia dos modelos - Base balanceada



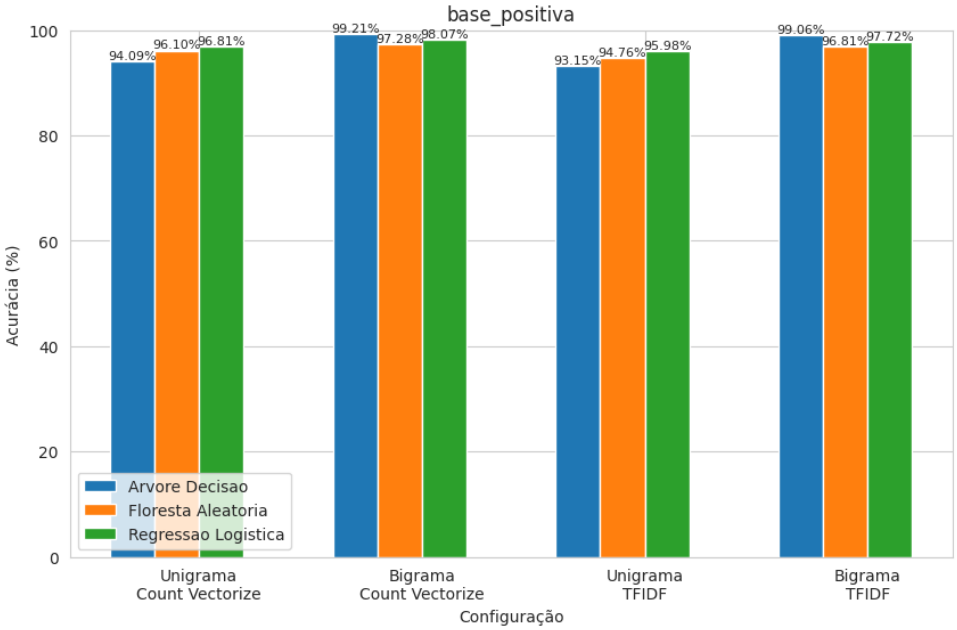
Fonte: autor

Figura 14 – Gráfico acurácia dos modelos - Base balanceada de validação



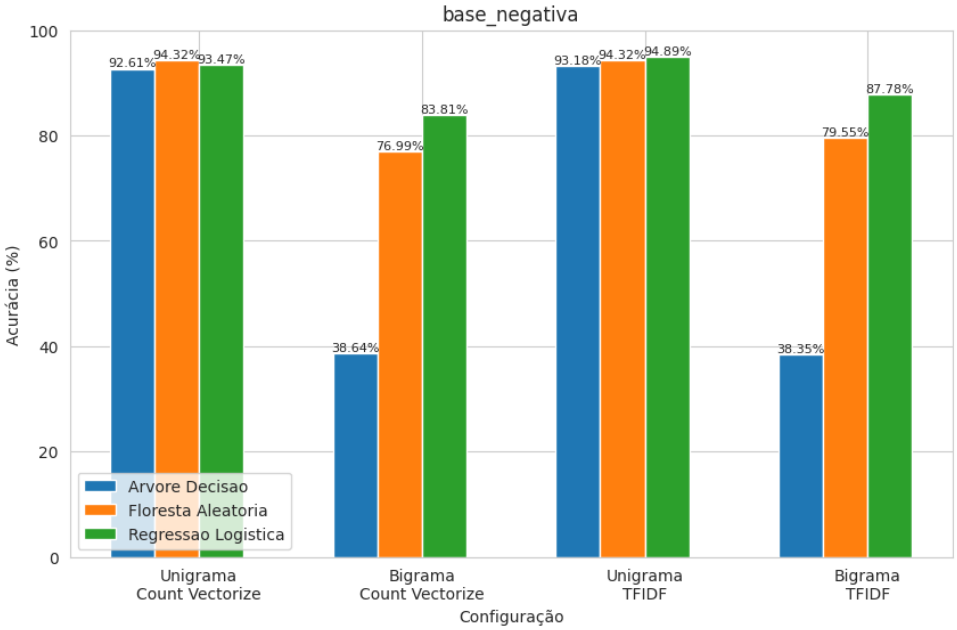
Fonte: autor

Figura 15 – Gráfico acurácia dos modelos - Base positiva de validação



Fonte: autor

Figura 16 – Gráfico acurácia dos modelos - Base negativa de validação



Fonte: autor

6 Considerações Finais

Este trabalho teve como objetivo geral desenvolver e avaliar um modelo de análise de sentimentos aplicado a uma base de dados de avaliações de usuários em uma plataforma de *e-commerce*. Esse objetivo foi alcançado por meio de um processo estruturado, que incluiu a coleta, o tratamento, a vetorização dos dados e a aplicação de diferentes algoritmos de aprendizado de máquina. A análise dos resultados demonstrou a eficácia do modelo proposto, validando a abordagem adotada e fornecendo *insights* relevantes sobre o desempenho das técnicas utilizadas.

Os objetivos específicos também foram alcançados, uma vez que a base de dados e o algoritmo implementados, disponíveis na plataforma Google Colab, demonstram todo o processo de desenvolvimento e análise. A análise realizada não apenas validou a metodologia proposta, mas também revelou aspectos importantes sobre a eficiência de diferentes abordagens para classificação de sentimentos em um contexto de avaliações textuais. Entre os modelos avaliados, destacaram-se a Regressão Logística, a Árvore de Decisão e a Floresta Aleatória, cada um com características e desempenhos distintos, influenciados pelas técnicas de vetorização e pela divisão dos dados em unigramas e bigramas.

Os resultados mostraram que a escolha do modelo e das técnicas de vetorização impacta significativamente na precisão da classificação. De modo geral, a Regressão Logística apresentou o melhor desempenho em termos de acurácia e robustez, alcançando uma taxa de 96,33% de acerto, o que confirma sua capacidade de lidar com bases de dados complexas e balanceadas.

Adicionalmente, foi possível identificar que a coerência entre os comentários textuais e as notas atribuídas pelos usuários é um fator relevante para avaliar a percepção geral dos consumidores. As técnicas de balanceamento aplicadas à base de dados contribuíram para uma avaliação mais precisa dos modelos, especialmente ao permitir que as classes de sentimentos fossem igualmente representadas durante o treinamento e a validação.

Apesar dos resultados positivos, algumas dificuldades foram encontradas ao longo do desenvolvimento deste trabalho. Uma das principais dificuldades foi o encerramento do acesso à API que fornecia os dados utilizados. Isso impediu a coleta de novas avaliações e limitou o volume de dados disponível para a análise. Esse fator ressaltou a importância de explorar múltiplas fontes de dados.

Como continuidade deste trabalho, sugere-se que pesquisas futuras explorem o uso de técnicas mais avançadas de vetorização, como *Word2Vec*, *GloVe* ou *BERT*, que podem aprimorar a representação semântica dos textos e, consequentemente, melhorar os resultados de classificação. Além disso, futuros estudos podem aumentar a diversidade das bases de dados, incluindo avaliações em diferentes idiomas ou oriundas de contextos distintos, para validar a generalização dos modelos desenvolvidos.

Outra possibilidade seria o desenvolvimento de sistemas que integrem a análise de sentimentos em tempo real com ferramentas de monitoramento de *feedback* de consumidores, permitindo que lojistas identifiquem rapidamente problemas ou destaquem produtos bem avaliados. Além disso, futuras pesquisas podem investigar como incorporar métricas adicionais, como análise de sentimentos ao nível de aspecto, para identificar elementos específicos de produtos que impactam na satisfação dos usuários.

Por fim, este trabalho reforça a relevância da análise de sentimentos como ferramenta para compreender as percepções dos usuários em ambientes de *e-commerce*. Os *insights* gerados por meio dessas técnicas podem ser utilizados para aprimorar a experiência do consumidor e apoiar estratégias de negócios mais eficazes. Assim, espera-se que esta pesquisa contribua para o avanço da área e inspire novas investigações nesse campo dinâmico e em constante evolução.

Referências

CARNEIRO, M. S. et al. Aplicação de redes neurais convolucionais para análise de sentimentos para a descoberta de conhecimento sobre o cliente. Universidade Nove de Julho, 2023. Citado na página 11.

CHIAPPE, L. M. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 417–424, Morristown, NJ, USA, 2001. Ass. *Society*, v. 12, n. 3, p. 417–424, 2010. ISSN 0738467X. Disponível em: <http://www.google.com>. Citado na página 13.

ESHAN, S. C.; HASAN, M. S. An application of machine learning to detect abusive bengali text. In: IEEE. *2017 20th International conference of computer and information technology (ICCIT)*. [S.l.], 2017. p. 1–6. Citado na página 18.

FÁTIMA, M. D. et al. *Pesquisa*$_M kt.[S.l. : s.n.], 2011. ISBN 9788578172978. Citado na página 25.$

IGOR, S. Universidade do Estado do Rio de Janeiro Centro de Tecnologia e Ciências Faculdade de Engenharia Igor Pedro Pinto dos Santos Análise de Sentimento Usando Redes Neurais de Convolução Rio de Janeiro*. 2017. 17–26 p. Tese (Doutorado) — Universidade do Estado do Rio de Janeiro, 2017. Citado nas páginas 12 e 13.*

JAVATPOINT, I. Decision Tree Classification Algorithm*. 2024. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm> [Accessed: (29/12/2024)]. Citado na página 16.*

JAVATPOINT, I. Random Forest Algorithm*. 2024. <https://www.javatpoint.com/machine-learning-random-forest-algorithm> [Accessed: (29/12/2024)]. Citado na página 17.*

JÚNIOR, W. J. de A. Métodos de otimização hiperparamétrica: um estudo comparativo utilizando árvores de decisão e florestas aleatórias na classificação binária. Universidade Federal de Minas Gerais, 2018. Citado nas páginas 16 e 18.

KUMAR, V.; SUBBA, B. A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. In: IEEE. 2020 national conference on communications (NCC)*. [S.l.], 2020. p. 1–6. Citado na página 12.*

LETT; OPINIONBOX. OPINIÃO DO CONSUMIDOR : o que é mais importante em uma página de produto no e-commerce? p. 54, 2019. Citado na página 10.

MCC-ENET. Partivipação do E-Commerce no comercio varejista*. Brazil: MCC-ENET, 2022. Citado na página 8.*

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014. Citado na página 12.*

MJVTEAM. Análise de sentimentos: por que você deveria prestar atenção nisso? *2020. Disponível em: <https://www.mjvinnovation.com/pt-br/blog/analise-de-sentimentos/>. Citado na página 11.*

MYLES, A. J. et al. An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society*, Wiley Online Library, v. 18, n. 6, p. 275–285, 2004. Citado na página 14.*

NASCIMENTO, P.; OSIEK, B.; XEXÉO, G. Análise de sentimento de tweets com foco em notícias. Revista Eletrônica de Sistemas de Informação*, v. 14, n. 2, 2015. Citado nas páginas 22 e 23.*

PALMUTI, C. S.; PICCHIAI, D. Mensuração do risco de crédito através de análise estatã stica multivariada. Revista Economia Ensaios*, v. 26, n. 2, 2012. Citado na página 14.*

PEREIRA, J. G. Análise de sentimentos da populaçao brasileira em relaçao a eleiçao presidencial de 2018 através da rede social twitter*. 2019. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2019. Citado na página 11.*

SANTANA, F. B. de. Floresta aleatória para desenvolvimento de modelos multivariados de classificação e regressão em química analítica*. 2020. Tese (Doutorado) — [sn], 2020. Citado na página 15.*

SELLTIZ, W. Métodos de pesquisa nas relações sociais*. 2. ed. São Paulo: [s.n.], 1965. 60 p. Citado na página 25.*

TOMÉ, L. M. COMÉRCIO ELETRONICO. Caderno Setorial ETENE*, v. 205, n. 6, p. 9, dec 2021. Citado na página 8.*

TURKI, T.; ROY, S. S. Novel hate speech detection using word cloud visualization and ensemble learning coupled with count vectorizer. Applied Sciences*, MDPI, v. 12, n. 13, p. 6611, 2022. Citado na página 18.*

VANAJA, S.; BELWAL, M. Aspect-Level Sentiment Analysis on E-Commerce Data. In: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2018. p. 1275–1279. ISBN 978-1-5386-2456-2. Disponível em: <https://ieeexplore.ieee.org/document/8597286/>. Citado nas páginas 11, 14, 23 e 24.*

VIVIAN, R. L. et al. Mineração de Dados Educacionais e Análise de Sentimentos em Ambientes Virtuais de Aprendizagem: um Mapeamento Sistemático. EaD em Foco*, v. 12, n. 2, p. 17, jun 2022. ISSN 2177-8310. Disponível em: <https://eademfoco.cecierj.edu.br/index.php/Revista/article/view/1786>. Citado nas páginas 11 e 20.*