



UNIVERSIDADE FEDERAL DO MARANHÃO
Curso de Ciência da Computação

Michele Soraya de Almeida Carvalho

**Análise de Sentimentos em Conversas de
Suporte Técnico usando o classificador Naive
Bayes**

São Luís
2023

Michele Soraya de Almeida Carvalho

Análise de Sentimentos em Conversas de Suporte Técnico usando o classificador Naive Bayes

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Ivo José da Cunha Serra

São Luís

2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Almeida Carvalho, Michele Soraya de.

Análise de Sentimentos em Conversas de Suporte Técnico usando o classificador Naive Bayes / Michele Soraya de Almeida Carvalho. - 2023.

57 p.

Orientador(a): Ivo José da Cunha Serra.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, SAO LUIS - MA, 2023.

1. Análise de Sentimentos. 2. Naive Bayes. 3. Processamento de Linguagem Natural. I. Cunha Serra, Ivo José da. II. Título.

Michele Soraya de Almeida Carvalho

Análise de Sentimentos em Conversas de Suporte Técnico usando o classificador Naive Bayes

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado em São Luís, 21 de dezembro de 2023:

Prof. Dr. Ivo José da Cunha Serra
Orientador

**Prof. Me. Carlos Eduardo Portela
Serra de Castro**
Examinador

Profa. Dra. Simara Vieira da Rocha
Examinador

São Luís
2023

Agradecimentos

Agradeço àquele que guiou meu caminho e proporcionou a oportunidade de concluir mais um objetivo: meu melhor amigo, Deus. Não posso deixar de expressar minha profunda gratidão à minha família, com destaque especial para minha mãe, que sempre prioriza minha educação e é um pilar fundamental em minha jornada acadêmica, além de ser um exemplo inspirador de mulher.

Agradeço ao meu professor e orientador Ivo, sempre receptivo em ajudar, por sua orientação sábia e apoio constante ao longo desta jornada.

E por fim, expresso minha gratidão aos ex-colegas de trabalho que disponibilizaram tempo para fornecer a base de dados necessária. Agradeço também aos amigos que me deram apoio e orientações em vários momentos, em especial a Benjamin e Lucas, meu sincero agradecimento.

"Lancem sobre Ele toda a sua ansiedade, porque ele tem cuidado de vocês."

1 Pedro 5:7

Resumo

A análise de sentimentos explora a atribuição de valores emocionais para cada sentença textual, possibilitando uma interpretação mais profunda dos dados de entrada. Detectar e compreender as emoções presentes em diálogos entre profissionais da informática e usuários é um aspecto importante na busca do aprimoramento contínuo na qualidade do serviço oferecido. A avaliação de conversas e a extração de informações sobre o desempenho do atendimento contribuem para obter uma perspectiva dos remetentes, possibilitando ajustes e melhorias necessárias. Este trabalho propõe estudar expressões compostas em chats de suporte de Tecnologia da Informação (TI) e categorizar os sentimentos em negativos, positivos e neutros, transmitidos nas interações dos usuários. Para tanto, foi desenvolvida uma solução computacional baseada em técnicas de Processamento de Linguagem Natural (PLN) em conjunto do algoritmo Naive Bayes, um classificador probabilístico, famoso pela facilidade e precisão na manipulação de dados textuais. Os resultados alcançados revelam que o classificador Naive Bayes, apesar da relativa simplicidade, se mostra uma opção a ser considerada, pela sua eficácia, em aplicações de análise de sentimentos.

Palavras-chave: Análise de Sentimentos; Naive Bayes; Chat de Suporte Técnico; Multinomial; Processamento de Linguagem Natural.

Abstract

Sentiment analysis explores the attribution of emotional values to each textual sentence, enabling a deeper interpretation of the input data. Detecting and understanding the emotions present in dialogues between IT professionals and users is an important aspect in the search for continuous improvement in the quality of the service offered. Evaluating conversations and extracting information about service performance helps to obtain a perspective from senders, enabling necessary adjustments and improvements. This work proposes to study compound expressions in Information Technology (IT) support chats and categorize feelings into negative, positive and neutral, transmitted in user interactions. To this end, a computational solution was developed based on Natural Language Processing (NLP) techniques together with the Naive Bayes algorithm, a probabilistic classifier, famous for its ease and precision in manipulating textual data. The results achieved reveal that the Naive Bayes classifier, despite its relative simplicity, is an option to be considered, due to its effectiveness, in sentiment analysis applications.

Keywords: Sentiment Analysis; Naïve Bayes; Technical Support Chat; Multinomial; Natural Language Processing.

Lista de ilustrações

Figura 1 – Tela de conversas do Spark com 2 profissionais de TI e usuários de outros polos	24
Figura 2 – Visão parcial da base de dados do Whatsapp	26
Figura 3 – Visão parcial da base de dados do Spark	26
Figura 4 – Gráfico com estatísticas da quantidade de sentimentos neutros, positivos e negativos	28
Figura 5 – Dedução de sentimentos	29
Figura 6 – Processo da distribuição das mensagens para etapa de treinamento e teste	31
Figura 7 – Pré-processamento dos dados do Whatsapp	35
Figura 8 – Modelo Multinomial Naive Bayes	45
Figura 9 – Matriz de confusão multiclasse	47
Figura 10 – Relatório de classificação do MNB	50
Figura 11 – Modelo MNB testado em conversa em grupo	51

Lista de tabelas

Tabela 1 – Descrição de 3 técnicas de Processamento de Linguagem Natural	20
Tabela 2 – Descrição das Fontes das mensagens	24
Tabela 3 – Exemplos de mensagens do chat do Suporte de TI	26
Tabela 4 – Exemplos de mensagens classificadas do suporte de TI	27
Tabela 5 – Mensagens e suas polaridades	42
Tabela 6 – Matriz de confusão 3x3 para Classificação de sentimentos	46
Tabela 7 – Matriz da classe Neutra	47

Sumário

1	INTRODUÇÃO	12
1.1	Trabalhos Relacionados	13
1.2	Objetivos	14
1.2.1	Objetivos Específicos	15
1.3	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Análise de sentimentos e o Processamento de Linguagem Natural	16
2.2	Pré-processamento de textos	19
2.3	Modelo de classificação Naive Bayes	20
3	ESTUDO DE CASO	23
3.1	Coleta de dados	23
3.2	Classificação dos Sentimentos	26
3.3	Naive Bayes na análise de sentimentos e suas ferramentas	30
3.3.1	Jupyter Notebook	31
3.3.2	NLTK	32
3.3.3	SpaCy	32
3.4	Pré-processamento de Dados: Técnicas e Métodos	33
3.4.1	Tokenização	35
3.4.2	Stopwords	36
3.4.3	Lematização	37
3.5	Análise de Dados	38
3.5.1	Scikit-Learn	38
3.5.2	TfidfVectorizer	38
3.6	Treinamento e Teste	39
3.7	Modelos Naive Bayes	41
3.7.1	Classificador Multinomial Naive Bayes	41
4	RESULTADOS	46
4.1	Matriz de Confusão	46
4.2	Resultados do experimento	48
5	CONCLUSÃO	53

REFERÊNCIAS 55

1 Introdução

O campo da Tecnologia da Informação (TI) abrange diversos aspectos relacionados ao processamento e análise de informações. Na esfera da comunicação, a TI facilitou a conexão instantânea entre indivíduos, ultrapassando barreiras geográficas e permitindo a troca de opiniões em tempo real. Essa modernização impactou dentro do setor tecnológico empresarial, levando a perceber que as responsabilidades de uma equipe de gestão, redes, operações, e outras, não se restringem apenas a compreensão de hardware e software, mas também a interação com os usuários, sejam eles clientes ou funcionários. Entender a mensagem transmitida, extrair os dados relevantes e encontrar soluções são habilidades fundamentais no cenário de prestação de serviços ao usuário. Esse suporte se dedica a solucionar uma diversidade de problemas técnicos, desde defeitos em dispositivos até a prestação de orientações e assistência no uso de aplicativos e sistemas. Sabendo da crescente dependência da tecnologia na sociedade, a necessidade de oferecer ajuda às pessoas na resolução de defeitos tornou-se evidente. Assegurar que as operações continuem sem interrupções e que os clientes possam realizar suas atividades é não apenas comum, mas essencial. Para isso, são necessários profissionais e medidas que facilitem a comunicação e alcancem resultados satisfatórios.

Durante o pós-pandemia, uma pesquisa conduzida pela Sercom revelou que 64% das pessoas têm preferido os canais digitais das empresas (como WhatsApp/Telegram), chat (28%) e redes sociais (8%) em comparação com os métodos tradicionais (como e-mail e telefone). Isso revela uma das necessidades das pessoas em relatar e resolver problemas de maneira rápida e imediata, impulsionando o uso crescente de interações digitais e em tempo real como ponto de contato com o usuário. O contexto atual demanda que os funcionários adotem uma série de estratégias de comunicação para lidar com os clientes. Isso inclui interpretar as informações fornecidas, compreender os sentimentos expressos e identificar soluções viáveis para resolver as demandas. Tendo a abordagem digital como realidade, surge a questão da automatização e aprimoramento contínuo dos meios utilizados.

No processo de interpretação de dados a tecnologia disponibiliza métodos que facilitem a manipulação de textos como conversas de chats. Para descobrir as informações úteis dos dados presentes, são realizadas diversas ações como inspeção, limpeza, transformação e modelagem (RANI; TANDON, 2018). A análise de sentimentos nesse contexto é uma ferramenta essencial, permitindo uma compreensão mais profunda das necessidades, preferências e experiências dos usuários. Com o Processamento de Linguagem Natural, é possível extrair informações e características sobre o sentimento implícito nas conversas, através de suas técnicas computacionais. Isso não apenas auxilia na identificação de problemas e preocupações dos usuários, mas também permite a identificação de pontos

críticos no atendimento e a otimização contínua da experiência dos funcionários. Diferente de um email que contém uma linguagem com tom mais formal e não dê de descrever toda a problemática, os chats proporcionam um ambiente de resposta rápida e expressão de sentimentos bons ou ruins imediatos, potencializando a análise de dados para aprimorar futuros atendimentos. Este tipo de interação foi utilizada neste trabalho, com chats reais de profissionais de TI com usuários de duas empresas.

Neste trabalho, optou-se por aplicar o processo de análise de sentimentos em interações de suporte técnico de TI, nas quais os funcionários dialogam por meio de aplicativos como o Whatsapp e Spark, no período entre 2021 e 2023. Essas conversas abordam desde problemas em equipamentos até informativos e dúvidas, proporcionando um panorama detalhado das necessidades de suporte, ajudando a identificar demandas recorrentes e questões relevantes. Isso significa que as expressões de satisfação, insatisfação ou frustração são frequentemente transmitidas de maneira imediata, tornando-as um alvo ideal para avaliação de sentimentos. O propósito central é justamente obter informações significativas dessas interações para identificar padrões emocionais e tendências, que podem ser usados para aprimorar a qualidade do suporte técnico.

Para realizar a análise de sentimentos é necessário escolher o algoritmo de classificação. Neste contexto, o Naive Bayes foi a opção escolhida. Naive Bayes é um dos algoritmos mais utilizados em problemas de classificação devido à sua simplicidade e eficácia (ARAR; AYAN, 2017). A versão Multinomial desse algoritmo, em especial, demonstra um desempenho superior em comparação ao modelo Bernoulli e, em alguns casos, até mesmo em relação a outros classificadores (MCCALLUM; NIGAM, 1998); (ZIVOLO; PRADO, 2021). Considerando esses elementos, este estudo optou por empregar o algoritmo Multinomial Naive Bayes na análise das interações nas conversas de suporte técnico de TI, visando obter uma compreensão mais profunda e abrangente dos sentimentos expressos nesses diálogos.

1.1 Trabalhos Relacionados

Com a crescente diversidade de redes sociais, torna-se cada vez mais crucial adaptar-se e analisar as opiniões expressas pelas pessoas. Isso vai desde uma publicação à todo um comportamento. A compreensão dos sentimentos tornam-se essenciais para interpretar o cenário social e responder de maneira adequada às demandas dos usuários. Nesta seção há trabalhos que serviram como base e referência no decorrer deste estudo de caso na análise de sentimentos.

O WhatsApp é uma plataforma amplamente utilizada tanto para fins comerciais quanto para interações sociais. O artigo de Renukadevi et al. (2023) se dedica a examinar a frequência e a participação dos membros de um grupo do WhatsApp. A base de

dados utilizada no estudo é composta por 262 registros, dos quais 195 são destinados ao treinamento do modelo, enquanto 67 são reservados para testes. O foco é identificar as pessoas que são muito ativas no grupo, por meio de dois algoritmos: Naive Bayes e Árvore de Decisão. Os resultados obtidos com esses algoritmos foram notáveis, atingindo taxas de acurácia de 95,45% e 98,48%, respectivamente.

A análise de sentimentos em posts educacionais (COUTINHO et al., 2016) foi analisada por Coutinho et al. (2020), por via do ambiente virtual de aprendizagem SOLAR, que facilita a comunicação entre alunos e tutores de cursos. O estudo empregou a classificação de mensagens por meio do algoritmo Multinomial Naive Bayes, utilizando como referência um dicionário prévio composto por tweets e palavras avulsas para o treinamento. A aplicação alcançou um desempenho satisfatório de 79%. No entanto, vale ressaltar que a dificuldade em encontrar bases textuais de qualidade classificadas em português brasileiro representou um desafio significativo para o desenvolvimento do estudo.

Dificuldade em relação ao texto ocorreu semelhantemente com Hassan et al. (2022), devido a complexidade gramatical do idioma. É abordado em seu artigo a análise de sentimentos em Bangla, com 1.141 dados coletados a partir de vários filmes e roteiros de curtas-metragens. No estudo utilizou-se de vários algoritmos de aprendizado de máquina, como: máquina de vetores de suporte, Multinomial Naive Bayes, k-vizinhos mais próximos, regressão logística, árvore de decisão e floresta aleatória. Entre os testes o Multinomial Naive Bayes conseguiu uma acurácia de 85.15%, tendo o 2º melhor desempenho.

A dissertação de Cássio Castaldi (2017) aborda a análise de sentimentos em diversos tíquetes de TI, totalizando 34.895, de cinco organizações. Esses tíquetes abrangem expressões relacionadas a erros, requisições e soluções. O autor optou por utilizar três métodos de classificação: um léxico, um estrutural e outro híbrido, para realizar a análise das polaridades positiva, negativa e neutra. Na implementação foi utilizado o SentiWordNet como léxico de sentimentos e o WordNet como thesaurus de sinônimos. Durante a aplicação dos dicionários foi necessário adequar as palavras para o cenário de TI, tendo em vista jargões comuns da área, como "salvar". O estudo obteve como melhores resultados uma precisão e revocação de 82,83% e 88,42%, superando outras soluções semelhantes apresentadas em seu trabalho.

1.2 Objetivos

Este estudo tem como objetivo desenvolver uma solução computacional para análise de sentimentos de mensagens do serviço de suporte técnico. Tendo o propósito de compreender as emoções expressas nas conversas, classificando-as em categorias de polaridade positiva, negativa e neutra. Além de explorar as potencialidades do classificador

Naive Bayes, avaliando como esse modelo se comporta na tarefa específica de classificar sentimentos em conversas técnicas de TI.

1.2.1 Objetivos Específicos

Destacam-se como objetivos específicos deste trabalho:

- Desenvolver uma solução computacional baseada em técnicas de Processamento de Linguagem Natural em conjunto com o algoritmo de classificação Naive Bayes para análise de sentimentos em conversas de suporte técnico de TI.
- Conduzir uma avaliação da solução proposta para análise de sentimentos de forma a preliminarmente demonstrar sua eficácia.

1.3 Organização do Trabalho

Este trabalho segue uma estrutura organizada em cinco capítulos, visando proporcionar uma compreensão clara e abrangente do conteúdo apresentado.

O Capítulo 2 explora a fundamentação teórica dos conceitos relacionados à análise de sentimentos, destacando as técnicas de Processamento de Linguagem Natural e a relevância do classificador Naive Bayes. O Capítulo 3 descreve a construção do estudo de caso, destacando as ferramentas e métodos utilizados no desenvolvimento do classificador de sentimentos, proporcionando uma visão abrangente da implementação. O Capítulo 4 apresenta os resultados obtidos, analisando o desempenho do algoritmo Multinomial Naive Bayes, assim como a aplicação de métricas para sua avaliação. Por fim, o Capítulo 5 traça as conclusões e propõe perspectivas para futuras pesquisas.

2 Fundamentação Teórica

Neste capítulo, será feita uma fundamentação teórica relacionada à análise de sentimentos, destacando as técnicas empregadas no Processamento de Linguagem Natural e a aplicabilidade do classificador Naive Bayes.

2.1 Análise de sentimentos e o Processamento de Linguagem Natural

As opiniões são fundamentais para quase todas as atividades humanas e são os principais influenciadores dos nossos comportamentos (LIU, 2012). A análise de sentimento, também conhecida como mineração de opinião, faz parte de uma das áreas provenientes do Processamento de Linguagem Natural (do inglês, Natural Language Processing - NLP), busca identificar em um conjunto de dados textuais os valores e emoções de um indivíduo, deste modo classificando-o em um sentimento positivo, negativo e neutro. Whitelaw (2005) traz uma forma diferente de polarizar textos, ele concentra seu trabalho na extração e análise de grupos de avaliação adjetivais encabeçados por um adjetivo de avaliação (como “bonito” ou “chato”) e opcionalmente modificados por uma sequência de modificadores (como “muito”, “mais ou menos” ou “não”). Com isso ele pôde obter informações valiosas sobre a linguagem do sentimento, tendo reflexo na avaliação dos mesmos. Embora seja atribuído o nome mineração de opinião no presente trabalho, existe o termo mineração de emoção (YADOLLAHI; SHAHRAKI; ZAIANE, 2017), baseado na identificação e compreensão das emoções expressas em textos não estruturados. Ela vai além da análise de sentimentos tradicional, que geralmente se concentra em determinar a polaridade do texto, para identificar emoções específicas, como alegria, raiva, tristeza, etc. Diversas técnicas podem ser implementadas com informações disponibilizadas nas redes sociais, assim como em ambientes que visem a necessidade de discernir o comportamento e expressão dos usuários, útil por exemplo, na avaliação de um atendimento em uma determinada empresa. E vai além de extrair informações de textos, podendo capturar sentimentos e emoções através de expressões faciais, movimentos, fala, música, entre outros.

O Processamento de Linguagem Natural (PLN) se concentra na comunicação entre a linguagem humana e a tecnologia, realizando o tratamento dos códigos de máquina através de etapas de processamento, para gerar informações entendíveis. Para Liu (2012), a análise de sentimento oferece uma ótima plataforma para pesquisadores de PLN fazerem progressos tangíveis em todas as frentes do PLN com potencial de causar um enorme impacto prático (LIU, 2012). Segundo Shanahann, Qu e Wiebe (2006) analisar a atitude e

o afeto no texto é um problema completo de “PLN”. A interpretação da atitude e do afeto depende do público, contexto e conhecimento de mundo. Além disso, ainda há muito a aprender sobre os aspectos psicológicos e relações biológicas entre emoção e linguagem. A importância em extrair um parecer correto de sentenças hipotéticas, especulativas ou reais reside na capacidade de compreender e interpretar com precisão a intenção e o contexto do autor, especialmente quando há ambiguidades no texto. A habilidade de reconhecer diferentes nuances de sentimento e significado contribui para resultados mais acurados em campos como análise de sentimentos.

Ao analisar cada palavra-chave de um conjunto de dados, como “bom” e “mau”, há aqueles que consideram a busca pela classificação de uma polaridade de sentimento descomplicada, por se tratar de termos simples e compreensíveis. Entretanto, termos como “bom” e “mau” podem variar em significado dependendo da colocação e das palavras circundantes. Ao verificar textos, uma das dificuldades seria a linguagem subjetiva das sentenças, além do uso da ironia e sarcasmo. Wiebe(2004) aborda em seu trabalho, os conceitos e experimentos ao lidar com artigos de notícias, ele verifica cada conjunto de dados e seus elementos específicos. Informa que a subjetividade na linguagem natural refere-se a aspectos da linguagem usados para expressar opiniões, avaliações e especulações. Ele comenta a análise de fatos, já que por ser objetivo não significa ser verdadeiro, o mesmo vale para algo subjetivo. Então além de consultar um conjunto de léxicos, deve-se avaliar o escopo em que os mesmos estão empregados.

O uso de técnicas para evitar resultados incorretos nas aplicações que utilizam do PLN é fundamental para lidar com a complexidade associada à compreensão da linguagem humana. Essas técnicas envolvem desde o pré-processamento avançado dos dados disponibilizados até a utilização de modelos de aprendizado de máquina sofisticados, como redes neurais e algoritmos de classificação, que consideram não apenas palavras isoladas, mas também a estrutura, a semântica e até mesmo a sutileza das frases. A incorporação de recursos como dicionários de sentimento e análise de padrões, combinando métodos supervisionados e não supervisionados, auxilia na melhoria da qualidade e da precisão dos resultados nas tarefas de análise de sentimentos.

Assim, ao realizar a tarefa de classificação das sentenças contidas em um amplo conjunto de dados, é necessário o uso de ferramentas de PLN para organizar o texto de forma sistemática. Dentro da análise morfosintática, encontra-se o processo de tokenização, o qual divide um fluxo de caracteres em unidades menores, chamados de *tokens*, pode ser entendido como qualquer sequência de caracteres que estejam divididos por espaços em branco, pontuações ou quebra de linha. Ademais, a marcação de partes do discurso (POS) e o reconhecimento de entidades nomeadas, funciona respectivamente, definindo se uma determinada palavra é um substantivo, verbo ou adjetivo, e identifica se são nomes de pessoas, cidades, etc. Além disso, a remoção de stopwords, a normalização de palavras e a

aplicação de técnicas de lematização ajudam a reduzir a complexidade do texto. Mas nem todos os sistemas exigem todas essas ferramentas.

No livro *Machine Learning*, Mitchell (1997), conceitua aprendizado de máquina como a construção de programas de computadores que melhorem o desempenho de alguma tarefa por meio da experiência. Assim, ele aborda alguns pontos pertinentes ao assunto na aplicação de projetos, incluindo a escolha do tipo de experiência de treinamento, a função alvo a ser aprendida, uma representação para esta função alvo e um algoritmo para aprender a função alvo a partir de exemplos de treinamento. Esses componentes formam a base do processo de aprendizado de máquina, em que a seleção adequada de cada um deles influencia diretamente na capacidade do modelo de realizar previsões e tomar decisões corretas em situações reais.

O sucesso da análise de sentimentos depende em grande parte da qualidade dos dados utilizados durante o treinamento, é importante a manipulação e aplicação correta no processo de preparo do conjunto. Liu (2012) demonstra como um classificador treinado usando documentos de opinião de um domínio geralmente tem um desempenho ruim em dados de teste de outro domínio. A razão é que as palavras e mesmo as construções linguísticas utilizadas em diferentes domínios para expressar opiniões podem ser bastante diferentes. Para piorar a situação, a mesma palavra num domínio pode significar positivo, mas noutro domínio pode significar negativo.

Uma opção utilizada para análise de opinião é o uso da abordagem baseada em léxicos, onde ocorre a utilização de dicionários na detecção do sentimento referente a cada item analisado. Ele funciona atribuindo um valor emocional, ou seja, uma polaridade positiva, negativa ou neutra, sendo a pontuação resultante o sentimento predominante da palavra verificada. Embora seja vantajoso para lidar com grandes volumes de dados e destaque pela eficiência sem a necessidade de um treinamento prévio, apenas usar na categorização de cada palavra não é o suficiente, a análise de sentimentos é algo bem mais complexo. Braz (2017), descreve em sua dissertação a análise de sentimentos na aplicação de tíquetes de suporte de TI, utilizando a abordagem não supervisionada por meio de dicionários e uma abordagem híbrida. O estudo se baseou em cinco organizações reais, contendo fontes como: Telecomunicação, Desenvolvimento de Software, Seguros e Previdência, e Pesquisa. Os textos continham: postagem de erro (infraestrutura, sistema), requisição (extração de dados, manutenção de sistema, infraestrutura, administrativa) e feedback (área de TI, sistema, infraestrutura). Como o autor Braz utilizou léxicos, pode-se verificar em seu trabalho os detalhes ao manusear mensagens técnicas, devido alguns termos terem significado diferente no mundo tecnológico. Assim, é necessário a adaptação do léxico utilizado para aprimoramento em relação aos termos específicos do mundo da TI, a fim de garantir que a classificação de sentimentos tenha um desempenho satisfatório. Por outro lado, se fosse preciso empregar técnicas de aprendizagem de máquina, seria

possível treinar modelos que compreendessem as nuances emocionais desses termos técnicos, contribuindo para análises mais precisas e contextualizadas.

Uns dos algoritmos de aprendizagem de máquina mais conhecidos atualmente e com suas próprias vantagens são: Redes Neurais, SVM (máquina de vetores de suporte)(KUMAR; SUBBA, 2020), Naive Bayes (PAK; PAROUBEK, 2010) e Árvore de Decisão. Cada um deles atua de forma eficaz em determinadas aplicações dependendo do cenário em que o autor deseja empregá-los. Pak e Paroubek (2010), utilizando a API do Twitter na coleta de dados, retrataram o uso do Multinomial Naive Bayes e obtiveram o melhor resultado em relação a SVM e Conditional Random Fields (LAFFERTY; MCCALLUM; PEREIRA, 2001). Através de técnicas de Processamento de Linguagem Natural construíram classificadores para a análise da presença de n-gramas e para informações de distribuição de classes gramaticais (POS). Isso permitiu a apresentação detalhada dos métodos empregados e o desempenho de cada procedimento no contexto de sua pesquisa.

2.2 Pré-processamento de textos

Antes da aplicação de qualquer técnica de Processamento de Linguagem Natural, é crucial o conhecimento da sua base de dados e suas determinadas características. Pak e Paroubek (2010) utilizaram como conjunto de textos (corpora), tweets para analisar a polaridade sentimental. Em seu processo de filtragem removeram links de URL, nomes de usuários do Twitter, palavras especiais do Twitter (como “RT”⁶) e emoticons. Isso evidencia a peculiaridade de cada base textual, destacando a importância de identificar a abordagem ideal de categorização e determinar quais ferramentas utilizar para extrair de maneira eficaz os elementos relevantes, a fim de simplificar a análise de sentimentos.

Como abordado na seção 2.1, um dos processos de tratamento da base de dados é a tokenização. Na documentação da biblioteca NLTK (Natural Language Toolkit), os tokenizadores são definidos como responsáveis por dividir strings em listas de substrings. Tenha a frase como exemplo:

“Não estou conseguindo imprimir. Poderia me ajudar?”

Frase após o processo de tokenização:

“Não - estou - conseguindo - imprimir - . - Poderia - me - ajudar - ?”

A fragmentação do texto nessa etapa acarreta em elementos mais manejáveis e passíveis de análise. Outro passo importante é a normalização do texto, que envolve a aplicação de técnicas como a remoção de caracteres especiais, a conversão para letras minúsculas e a eliminação de stopwords. Stopwords podem ser exemplificados no idioma português-br como: ‘a’, ‘o’, ‘de’, etc. Vale ressaltar que cada língua possui suas peculiaridades e abordagens específicas para a utilização das ferramentas de Processamento

de Linguagem Natural. Portanto, ao realizar essas etapas, o objetivo é padronizar o texto, minimizando a influência de elementos sem relevância que poderiam comprometer a análise de sentimentos e outras atividades relacionadas ao procedimento.

Os lematizadores são analisadores morfológicos que associam variantes do mesmo termo a uma forma de raiz. A raiz pode ser a forma que normalmente seria encontrada como uma entrada em um dicionário (JACKSON; MOULINIER, 2002). Palavras como “brincou”, “brincando”, resultam em “brincar”. Stemming refere-se a uma versão mais simples de lematização, na qual apenas retiramos os sufixos do final da palavra (JURAFSKY; MARTIN, 2000). Logo, aplicando algumas das técnicas anteriormente citadas a frase acima resultaria em:

“não estar conseguir imprimir poder eu ajudar”

Além disso, no processamento de textos, a identificação de partes do discurso (POS) e entidades nomeadas (EN) são pistas úteis para a estrutura e o significado das frases. Saber se uma palavra é um substantivo ou um verbo nos informa sobre prováveis palavras vizinhas (JURAFSKY; MARTIN, 2000). A adequada extração e reconhecimento de Entidades Nomeadas e a atribuição de Partes do Discurso podem aprimorar consideravelmente a análise de sentimentos e outras atividades no âmbito do Processamento de Linguagem Natural. Esses procedimentos promovem uma compreensão mais profunda da relação entre as entidades identificadas nos textos e as emoções que são expressas. Isso resulta em uma interpretação mais completa das emoções presentes nas conversas e discursos analisados. A Tabela 1 demonstra de forma simplificada algumas das principais técnicas de PLN citadas nesta seção.

Tabela 1 – Descrição de 3 técnicas de Processamento de Linguagem Natural

	Tokenização	Lematização	Partes do discurso
Definição	Divide um texto em palavras individuais	Reduz palavras à forma base (lemas)	Atribui etiquetas de classe gramatical a cada palavra
Objetivo	Dividir o texto em tokens, preparar para análise	Eliminar flexões e sufixos, normalizar palavras	Identificar relações das palavras em frases
Exemplo	[“cachorros”, “correm”]	[“cachorro correr”]	“cachorros” - substantivo “correm” - verbo

2.3 Modelo de classificação Naive Bayes

Após a definição e estruturação do texto de entrada, a próxima etapa na resolução de classificação é a escolha dos métodos de aprendizado de máquina a serem empregados para prever os sentimentos. Essa escolha pode ser híbrida, combinando abordagens supervisionadas e não supervisionadas, ou pode se concentrar na seleção de um único

método a ser aplicado como modelo. Um desses modelos amplamente conhecidos é o algoritmo Naive Bayes (NB), popular por sua simplicidade e eficiência na análise de sentimentos (WICKRAMASINGHE; KALUTARAGE, 2021); (AKRE et al., 2023), devido à sua base probabilística sólida. Como exemplo de sistema, o Piegas, bem sucedido na análise de sentimentos em tuítes, sendo uma opção competitiva em relação à complexidade de outros algoritmos (GRANDIN; ADAN, 2016). O Naive Bayes é um algoritmo de classificação que se baseia no Teorema de Bayes, estudo formulado por Thomas Bayes, para calcular a probabilidade de um documento pertencer a uma classe específica, neste caso, um sentimento particular. Ele assume ingenuamente que as características são independentes umas das outras, ou seja, a presença de uma característica não afeta a presença de outras, daí o termo “naive” (ingênuo). A principal premissa é que cada palavra contribui de forma independente para a probabilidade de pertencer a uma classe de sentimento. Arar e Ayan(2017) discutem sobre a relação de independência do algoritmo, propondo o método, denominado Feature Dependent Naive Bayes (FDNB), que leva em consideração recursos pares no cálculo de probabilidades, o qual obteve um desempenho satisfatório em relação ao NB padrão. Jurafsky e Martin (2000) descrevem em seu livro a aplicação probabilística na prática, conforme especificado pelas fórmulas.

(Equação 2.1) Para um documento \mathbf{d} , de todas as classes $c \in C$ o classificador retorna a classe \hat{c} que tem a probabilidade posterior máxima dada o documento

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|\mathbf{d}) \quad (2.1)$$

$P(\mathbf{x}|\mathbf{y})$ é a probabilidade condicional da classe x dado um documento y

$P(\mathbf{y}|\mathbf{x})$ é a probabilidade condicional de um documento y pertencer à classe x , considerando as características do documento.

$P(\mathbf{x})$ é a probabilidade prévia da classe x

$P(\mathbf{y})$ é a probabilidade de um documento y

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \quad (2.2)$$

Substituindo e simplificando a equação da Regra de Bayes:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|\mathbf{d}) = \underset{c \in C}{\operatorname{argmax}} \frac{P(\mathbf{d}|c)P(c)}{P(\mathbf{d})} = \underset{c \in C}{\operatorname{argmax}} P(\mathbf{d}|c)P(c) \quad (2.3)$$

Veja a aprimoração da equação de probabilidade do NB e sua adequação para lidar com as posições das palavras ao realizar a classificação de textos, percorrendo um índice através de cada posição de palavra no documento. Na seção 3.7 será abordado com mais detalhes sua utilização na análise de sentimentos.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j) \quad (2.4)$$

A eficiência do algoritmo Naive Bayes é notável em várias tarefas, incluindo a classificação de sentimentos, onde se destaca pelo seu desempenho competitivo. Embora algoritmos mais complexos possam superá-lo em algumas situações, ele continua como uma ótima escolha, sendo semelhante em desempenho a outras técnicas de aprendizado de máquina (PANG; LEE; VAITHYANATHAN, 2002). Essa eficiência torna o Naive Bayes uma opção valiosa em muitos cenários de análise de sentimentos e classificação de texto. Yu e Hatzivassilglou (2003) obtiveram um f-measure de 97% em um dos modelos apresentados em seu trabalho, e ao analisar a polaridade das sentenças conseguiram 90% de precisão. Diante disso, a opção de utilizar o NB neste trabalho partiu de sua confiabilidade ao analisar textos, assim como sua facilidade de implementação e flexibilidade devido suas variações (MCCALLUM; NIGAM, 1998); (METSIS; ANDROUTSOPOULOS; PALIOURAS, 2006) sendo interessante em diferentes tipos de domínios (LANGLEY; IBA; THOMPSON, 1992). Contudo, é amplamente reconhecido que ao adotar uma abordagem supervisionada como o Naive Bayes (JUNQUEIRA; FERNANDES, 2018), há uma necessidade crítica de contar com um conjunto de dados de treinamento que esteja cuidadosamente etiquetado com informações sobre os sentimentos presentes. Essa tarefa implica em rotular um grande volume de dados com diversas polaridades de sentimentos, neste caso, mensagens existentes em chats de suporte de TI. Devido à presença de ambiguidades, ironia e à combinação de estilos de linguagem informal e palavras de aspecto técnico, considerou-se valioso avaliar o desempenho do classificador Naive Bayes como parte deste estudo.

3 Estudo de Caso

Este capítulo apresenta um estudo de caso que consistiu no desenvolvimento de uma solução computacional para análise de sentimentos em conversas de suporte técnico de TI. Esta solução utiliza técnicas de Processamento de Linguagem Natural em conjunto com o algoritmo de classificação Naive Bayes.

A seção 3.1 inicia com uma descrição do corpus de dados extraído para este trabalho, detalhando como as mensagens foram coletadas, manipuladas e tratadas para análise. A seção 3.2 fornece informações sobre as características do processo de classificação de sentimentos. Na seção 3.3, o foco recai sobre o classificador Naive Bayes e as ferramentas utilizadas na análise de sentimentos. A seção 3.4 é dedicada às técnicas empregadas no pré-processamento dos dados textuais. No que se refere à seção 3.5, ela trata do processamento dos dados na etapa de análise. A seção 3.6 aborda os parâmetros das etapas de treinamento e teste na análise de dados. Por fim, a seção 3.7 descreve o algoritmo Multinomial Naive Bayes e seu funcionamento na classificação dos sentimentos das mensagens.

3.1 Coleta de dados

Os trechos de texto selecionados como o conjunto de dados foram adquiridos através da coleta de chats de suporte técnico de T.I. de duas organizações reais, que neste estudo foram identificadas como Empresa A e Empresa B. As conversas obtidas contém um bate papo entre usuários e profissionais da área, relatando dúvidas, problemas em equipamentos e sistemas, além de comunicados. Cada empresa dispõe dos seus padrões, isto é, o conteúdo das mensagens não têm necessariamente a mesma nomenclatura de dispositivos e sistemas. O corpus foi composto por 11 conversas, sendo 3 conversas no privado e 1 em grupo pelo aplicativo Whatsapp, e 7 conversas pelo programa Spark. O restante das mensagens, sendo 85 no privado e 1 em grupo, foram usadas para testes com o modelo. Dessas fontes foram distribuídas a porcentagem referente aos dados de treinamento e de teste, que serão comentados na seção 3.6. A Tabela 2 descreve alguns detalhes do volume de chats utilizado.

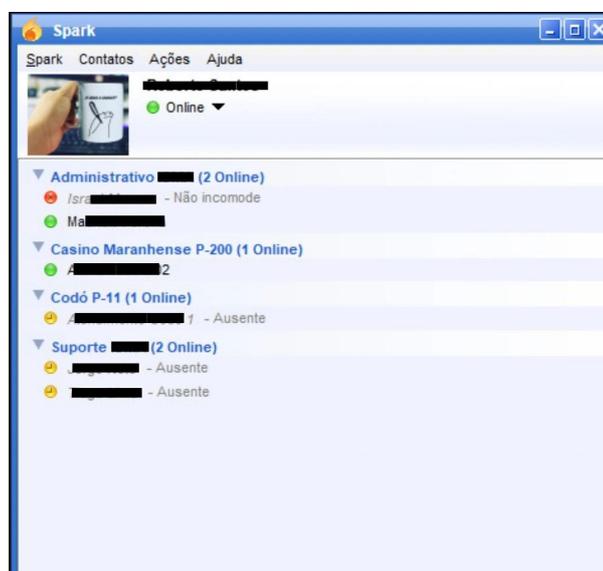
Inicialmente foi feita uma análise preliminar das conversas para checar os elementos textuais de atendimento técnico ou uma mensagem informal fora do aspecto de problemas e erros. Foi verificado que os tipos de chat privado tinham mais probabilidade de conter gírias e subjetividade, e no chat em grupo haviam textos mais formais e repetitivos. Como foi realizado o processamento de textos de duas empresas distintas, as mesmas contêm diferenças nos métodos para solucionar problemas, assim como o nível de expressões contidas em cada conversa. No chat de suporte técnico da Empresa B, é comum notar o uso frequente da palavra “núcleo” devido ao formato dessas interações, que geralmente

Tabela 2 – Descrição das Fontes das mensagens

Empresa	Aplicativo	Tipo de Suporte de TI	Conversas	Formato
Empresa A	Spark (programa principal)	Comunicados e alertas, Resolução de problemas individuais (privado)	77 KB	.xml
	Whatsapp	Resolução de problemas individuais (privado)	38 KB	.txt
Empresa B	Whatsapp	Comunicados e alertas, Resolução de problemas em grupo	85 KB	.txt

ocorrem em grupos envolvendo participantes de diferentes municípios. Nesse contexto, a identificação da localização do usuário é fundamental para proporcionar suporte técnico personalizado, uma vez que as necessidades podem variar de acordo com a região. Por outro lado, na Empresa A, a presença da identificação do local nas conversas é menos comum, já que o programa principal utilizado é o Spark e nele cada usuário é nomeado previamente de acordo com o posto em que está trabalhando, ilustrado na Figura 1. Esses são um dos detalhes que influenciam diretamente na frequência das palavras e conseqüentemente na resolução da classificação.

Figura 1 – Tela de conversas do Spark com 2 profissionais de TI e usuários de outros polos



As conversas em grupo da Empresa B são de julho de 2021. Devido à atividade de vários usuários, notou-se uma grande quantidade de mensagens repetitivas, arquivos multimídia e frases desnecessárias da própria estrutura do whatsapp. Como medida de obter as mensagens foi utilizado a opção de “exportar conversa” no aplicativo em questão, assim sendo incluso frases como “<Arquivo de mídia oculto>” e “Lucas entrou usando o link de convite deste grupo”, acarretando em alguns dados irrelevantes. Além disso, é importante mencionar que existem informações presentes nas conversas que não são de

grande utilidade para o propósito deste trabalho. Isso inclui dados como data, hora e remetente das mensagens. A presença desses elementos e o formato padronizado em que são apresentados nos arquivos de conversa exigem uma abordagem cuidadosa no tratamento textual, uma vez que não contribuem diretamente para a análise de sentimentos e podem aumentar a complexidade do manuseio dos dados.

Em vista das conversas serem repletas de termos técnicos, é notório o desafio que surge quando juntam várias emoções em uma única mensagem textual. Isso pode dificultar a análise da polaridade, uma vez que a presença de informações técnicas e a expressão de sentimentos coexistem no mesmo texto. Além disso, é intrigante a identificação de palavras e expressões frequentemente empregadas pelos técnicos de suporte para demonstrar sua atenção e comprometimento em solucionar os problemas dos usuários. Essas palavras e frases específicas podem funcionar como indicadores valiosos de interações positivas e construtivas, independentemente do teor técnico das mensagens. Por exemplo, a palavra “verificando”. Em muitos casos, essa palavra representa a resposta do profissional a uma situação relatada pelo usuário, indicando uma abordagem positiva na qual o problema está sendo minuciosamente analisado e investigado para oferecer a melhor solução possível.

Ao lidar com as estruturas distintas do Spark e do Whatsapp, e da quantidade de dados escolhidos para serem processados, foi necessário automatizar a leitura do arquivo, sendo desenvolvido um código em Python que desempenhou um papel essencial. Assim, foi possível extrair os dados dos chats de suporte técnico de forma eficiente, processá-los e prepará-los para o treinamento do modelo de análise de sentimentos. Essa abordagem automatizada permitiu agilizar o processo de anotação, economizando tempo e recursos que seriam necessários caso a escrita fosse realizada diretamente no documento. Entretanto, do mesmo modo durou um grande período devido ter que analisar, compreender e escolher o sentimento adequado para a mensagem. Considerando que nem todas as frases apresentam uma carga emocional significativa, é importante observar que em algumas situações é comum receber reclamações sobre problemas que, na verdade, não são erros, e outras situações em que não dá para detectar o que significa o problema. Um exemplo disso é quando os usuários relatam a falta de acesso a um determinado site, quando, na realidade, essa é a maneira como o sistema deve funcionar. Casos como esse devem ser cuidadosamente identificados para que ao realizar a etiquetagem dos sentimentos, seja detectado a polaridade dominante que o remetente esteja passando. A Tabela 3 mostra alguns exemplos de mensagens recebidas pelo suporte de TI.

Para manipular os dados do arquivo .xml, foi necessário utilizar a biblioteca *xml.etree.ElementTree*. Essa biblioteca é extremamente empregada na análise e processamento de documentos XML em linguagem Python. Ela permite a leitura e a manipulação de elementos e atributos XML, tornando mais fácil a remoção de informações específicas de arquivos.

Tabela 3 – Exemplos de mensagens do chat do Suporte de TI

<i>“Fui abrir a Netflix não deu certo”</i>	há bloqueio em determinados sites
<i>“A estagiária que chegou hoje n tá conseguindo acessar o SAGAP”</i>	permissão para acesso de estagiários a sistemas são determinados após dados encaminhados pela Sup. de Estágio
<i>“não consigo logar, minha senha bloqueou eu troquei e esqueci a nova e coloquei a antiga várias vezes”</i>	há bloqueio do sistema após sucessivas tentativas de acesso sem sucesso
<i>“meu token é nas nuvens, mas o meu pc do núcleo não está configurado”</i>	há uma dificuldade de interpretação

As Figuras 2 e 3 ilustram as estruturas pertencentes a cada base de dados utilizada por via dos programas de chat distintos, Whatsapp e Spark. Compreender esses padrões é essencial para garantir a correta extração e interpretação dos dados das conversas em ambos os programas. É notável a existência de informações irrelevantes na análise de sentimentos do texto. Ter uma familiaridade com esses padrões de estruturação das conversas é valiosa para o processo de pré-processamento dos dados.

Figura 2 – Visão parcial da base de dados do Whatsapp

```
07/01/2022 10:37 - Gabriel: @██████████
07/01/2022 10:37 - Gabriel: @██████████
07/01/2022 10:37 - Polly: *10/01/2022 à 14/01/2022*
Zé Doca, Viana, Bom Jardim, Santa Inês, Arari, Penalva Tuntum

17/01/2022 à 21/01/2022
Pinheiro, Parnarama, Matões, Pedreiras, Rosário, Colinas e São João dos Patos

24/01/2022 à 28/01/2022
Alcantara, Chapadinha, Esperantinópolis, Coroatá, Codó e São Mateus

31/01/2022 à 04/02/2022
Vitorino Freire / lago da pedra / Santa Helena / bacabal
```

Figura 3 – Visão parcial da base de dados do Spark

```
<transcript>
<messages>
<message>
<to>██████████@gerions</to>
<from>impribamar@gerions/Spark 2.6.3</from>
<body>oi</body>
<date>2022-10-14 13:51:48.604 GMT-03:00</date>
</message>
<message>
<to>██████████@gerions</to>
<from>impribamar@gerions/Spark 2.6.3</from>
<body>boa tarde</body>
<date>2022-10-14 13:51:55.834 GMT-03:00</date>
</message>
</messages>
```

3.2 Classificação dos Sentimentos

Ao examinar os textos, torna-se evidente a necessidade de empregar três categorias de polaridade: positivo, negativo e neutro. Esta abordagem se justifica pelo fato de que

a base de dados se concentra predominantemente na resolução de problemas técnicos, o que resulta em mensagens com um teor informativo e explicativo considerável. Muitas dessas mensagens não expressam sentimentos fortemente positivos ou negativos, mas sim uma neutralidade que reflete a natureza técnica e objetiva das interações. Ao considerar o espectro de polaridades, a inclusão da categoria “neutro” é essencial para uma análise de sentimentos abrangente, permitindo a identificação precisa das mensagens que não carregam emoções expressas de forma intensa, mas ainda desempenham um papel crucial no contexto da assistência técnica e suporte ao cliente. A Tabela 4 mostra alguns exemplos de mensagens positivas, negativas e neutras.

Tabela 4 – Exemplos de mensagens classificadas do suporte de TI

Polaridade	Mensagem
Positiva	<i>“Acredito que agora normalizou o retorno de dados”</i>
Negativa	<i>“deu certo não”</i>
Neutra	<i>“vou imprimir uma página de teste”</i>

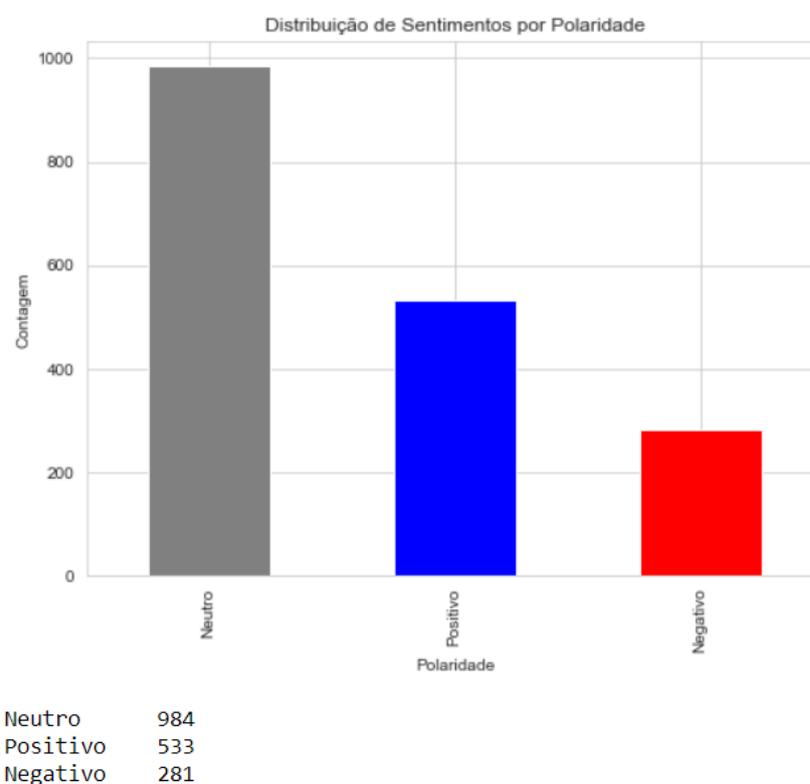
As mensagens retiradas da base de dados do suporte de TI, ilustram como as interações entre usuários e os profissionais de informática frequentemente carecem de detalhes e informações explicativas. Muitas vezes, os técnicos de suporte já conseguem deduzir qual problema está sendo relatado apenas pelo costume e sem a necessidade de uma descrição minuciosa da situação.

Quando se trata do uso de uma abordagem supervisionada, é fundamental lembrar que os dados precisam passar por uma fase de treinamento. No contexto deste trabalho, que envolve chats de suporte técnico de TI, a preparação dos dados de treinamento é uma etapa crítica. Isso ocorre porque o algoritmo Naive Bayes requer um conjunto de dados rotulados com informações sobre os sentimentos presentes nas mensagens. No caso específico dessas conversas de suporte, a rotulação dos dados envolve atribuir polaridades de sentimentos às mensagens, como positivo, negativo ou neutro, para que o algoritmo possa aprender com esses exemplos e posteriormente classificar novas mensagens com base nesse aprendizado. Deste modo, foram atribuídos aos sentimentos positivo, negativo e neutro, respectivamente, a identificação de 1, 2 e 0. As conversas classificadas e já passadas por uma pré-formatação inicial, mesclando os chats do Spark e Whatsapp totalizaram 162KB.

Dentre alguns desafios ao desenvolver utilizando uma abordagem supervisionada, é notável a dificuldade no manuseio dos dados, especialmente na etapa de rotulação. Isso é, os custos da rotulagem manual, as interpretações subjetivas, a personalidade e perspectiva do autor, entre outras situações. É importante mencionar que a tarefa de classificação para treinamento foi realizada por uma única pessoa, o que levou a um intervalo de tempo considerável para concluir essa etapa do projeto. Assim, tendo uma base de dados de treinamento pequena, é importante escolher o melhor modelo probabilístico para o volume

de dados, além de atentar-se para a escolha das mensagens que passarão pelo processo de análise de sentimentos, já que o algoritmo funciona por estimativa de probabilidades. Por exemplo, o WhatsApp, frequentemente apresenta em suas conversas de texto exportadas mensagens que possuem termos padrões, como situações onde pessoas entram, saem ou são excluídas de determinado grupo, aparecendo uma notificação na conversa e assim sendo salva no histórico da mesma. Esses tipos de mensagens são comuns, além de casos quando os usuários excluem ou enviam fotos, e apagam mensagens em uma conversa. Ao realizar a análise desse cenário, é importante considerar como essas mensagens são tratadas, contadas e registradas, pois isso pode ter um impacto significativo na precisão de qualquer modelo de PLN ou algoritmo de análise de texto. A presença frequente dessas mensagens pode afetar a acurácia do modelo de várias maneiras. Primeiro, essas mensagens geralmente não carregam informações úteis em termos de sentimento, o que significa que atribuir uma polaridade emocional a elas pode ser inadequado. Segundo, a frequência dessas mensagens “ruído” pode distorcer as estatísticas gerais do conjunto de dados, levando a conclusões imprecisas. Portanto, é importante realizar uma filtragem do que passará no processo de treinamento com suas polaridades emocionais corretas.

Figura 4 – Gráfico com estatísticas da quantidade de sentimentos neutros, positivos e negativos

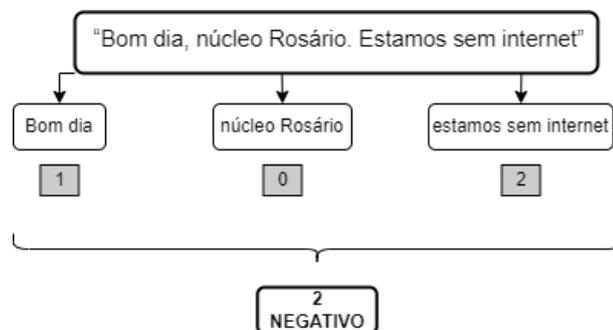


A Figura 4 apresenta uma representação visual dos sentimentos neutros, positivos e negativos presentes nos dados analisados. Nota-se que a categoria de sentimentos neutros é predominante em comparação com as categorias positivas e negativas. Isso é uma observação significativa, considerando que a base de dados se concentra em conversas

relacionadas a assuntos técnicos, que geralmente não envolvem uma carga emocional forte. No entanto, a presença de sentimentos positivos e negativos também é evidente, indicando que, mesmo em contextos técnicos, pode haver situações que geram respostas emocionais variadas.

Na etapa de rotulação é analisada cada mensagem e a emoção predominante feita pelo próprio autor. A partir disso será calculada a probabilidade considerando os pesos, frequência e relevância de cada termo presente nos dados. Este processo de atribuir rótulos às mensagens de treinamento é fundamental para que o modelo Naive Bayes possa aprender a distinguir diferentes polaridades de sentimentos e fazer previsões com base nos padrões identificados durante o treinamento. Diante disso, ao analisar a Figura 5, pode-se entender inicialmente que a frase em questão deveria resultar em neutro, verificando cada elemento unicamente, entretanto a carga emocional de uma frase não é simplesmente a soma das associações emocionais das palavras que a compõem. Além disso, as emoções muitas vezes não são explicitamente declaradas (MOHAMMAD, 2016). Com isso, nesse exemplo em questão, a frase é algo negativo, significando que a localidade está sem acesso à internet. Esta entre outras questões tornam a abordagem supervisionada, através do treinamento, diferente das tarefas que seriam realizadas em uma abordagem com léxicos, onde há o uso de dicionários.

Figura 5 – Dedução de sentimentos



Nesse processo de polarização das conversas nos chats de suporte técnico, é comum encontrar palavras e frases isoladas que, por si só, podem parecer desprovidas de significado. No entanto, ao contextualizá-las e analisá-las em relação às mensagens anteriores da conversa, muitas vezes essas palavras e frases isoladas adquirem um significado mais claro e relevante. Neste estudo não foram examinadas as mensagens atuais em conjunto com as anteriores, resultando em algumas mensagens classificadas como neutras, dependendo da situação. Por exemplo, ao analisar o envio de algumas mensagens, é comum aparecer a palavra “tá” sem complementos textuais, e esta palavra pode ter diferentes significados, como confirmação (“tá bom”), presença de algo (“está”) ou a resposta para uma pergunta anterior importante. Assim, para não comprometer o desempenho do algoritmo optou-se por, nesse caso, atribuir um valor neutro, já que a palavra isoladamente dificulta o entendimento da emoção transmitida. No entanto, esta abordagem não comprometeu

o desempenho do classificador, uma vez que o objetivo principal era destacar o uso do Naive Bayes em chats de suporte de TI e explorar diferentes variáveis e métodos. Esta escolha metodológica permitiu demonstrar a aplicabilidade do algoritmo Naive Bayes e sua capacidade de lidar com a complexidade das conversas técnicas de suporte, mesmo sem a análise contextual mais elaborada. Dessa forma, o estudo fornece insights valiosos sobre a eficácia do Naive Bayes em cenários específicos de atendimento ao usuário, demonstrando que, mesmo com uma abordagem simplificada, ele pode oferecer resultados relevantes e úteis na análise de sentimentos em chats de suporte técnico.

3.3 Naive Bayes na análise de sentimentos e suas ferramentas

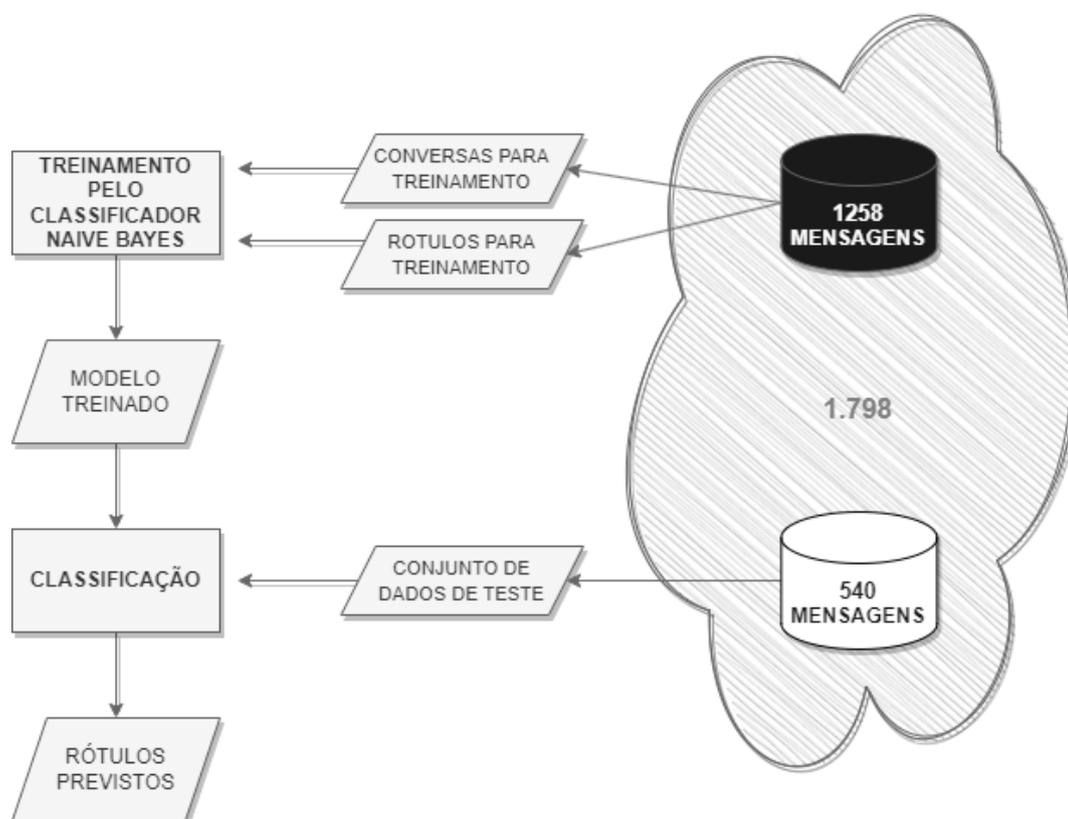
As empresas modernas estão cada vez mais preocupadas em compreender a satisfação do cliente e avaliar a qualidade de seus produtos e serviços com base nas avaliações dos consumidores. Nesse sentido, a análise de sentimentos desempenha um papel crucial, pois permite que as empresas avaliem como os clientes se sentem em relação aos serviços e experiências. Ao lidar com chats de suporte técnico entre a empresa e o cliente, a análise de sentimentos torna-se igualmente relevante, pois está intrinsecamente ligada à resolução de problemas. Os usuários que buscam assistência técnica frequentemente expressam preocupações, dúvidas e frustrações, e compreender essas emoções é essencial para fornecer um suporte eficaz ao usuário.

É comum em empresas, ao final de um suporte técnico, solicitar uma avaliação da qualidade do atendimento prestado à pessoa. No entanto, é frequente que muitos usuários optem por não avaliar ou comentar sobre a experiência de serviço, o que pode dificultar a coleta de feedbacks e informações importantes para aprimorar a qualidade do suporte. Essa ausência de avaliações pode ser atribuída a diversos fatores, incluindo a falta de incentivo por parte da empresa, a pressa do solicitante ou até mesmo a sua satisfação ou insatisfação com o suporte, que pode não ser expressa por meio de avaliações formais. Diante disso, avaliar as emoções do usuário por meio de um bate papo com o profissional de TI facilita a percepção do atendimento prestado. Quando os usuários interagem diretamente com os agentes de suporte técnico em um chat, suas emoções e sentimentos podem ser expressos de forma mais imediata e natural, proporcionando aos funcionários a oportunidade de identificar prontamente qualquer sinal de satisfação, insatisfação ou frustração.

Para analisar o cenário da TI e compreender suas nuances por meio das emoções, a escolha foi utilizar a base de dados, já abordada na seção 3.1, sendo processada com o algoritmo Naive Bayes. Para alcançar resultados precisos, ferramentas de Processamento de Linguagem Natural foram empregadas no pré-processamento do texto, visando a limpeza, normalização e etiquetagem das mensagens para uma análise mais eficaz. A combinação da base de dados, do algoritmo Naive Bayes e das técnicas de PLN possibilitou uma

análise aprofundada das interações entre o suporte e usuários. A Figura 6 ilustra o método utilizado, expondo alguns processos principais realizados: a etapa de treinamento, a etapa de teste e os rótulos finais usados na análise de sentimentos. O processo começa com a etapa de treinamento, na qual o modelo NB é alimentado com um conjunto de dados de treinamento contendo mensagens previamente rotuladas com sentimentos. Durante essa fase, o algoritmo aprende a associar palavras e padrões específicos a sentimentos, permitindo que ele faça previsões em novos textos.

Figura 6 – Processo da distribuição das mensagens para etapa de treinamento e teste



Embora existam programas que facilitam o uso deste algoritmo, como Weka, Orange e R, a escolha foi pelo Jupyter Notebook devido à sua versatilidade e eficácia na implementação personalizada das etapas de pré-processamento e análise de sentimentos. O Jupyter Notebook em conjunto com a linguagem Python permitiram uma abordagem mais adaptável e controlada, adequada às necessidades específicas deste estudo, proporcionando maior transparência no processo de análise e garantindo resultados precisos na avaliação das emoções presentes nas conversas de suporte técnico em TI.

3.3.1 Jupyter Notebook

O Jupyter Notebook é uma ferramenta de código aberto baseada em navegador que funciona como um laboratório virtual altamente flexível e interativo. Projetado para acomodar fluxos de trabalho de análise de dados, programação, documentação

e visualização, o Jupyter Notebook é uma escolha popular entre cientistas de dados, pesquisadores e programadores. Ele permite a criação de documentos que combinam células de código, onde os usuários podem escrever e executar código em linguagens como Python, com células de texto para documentação explicativa. Isso se revela extremamente útil na análise de sentimentos, pois permite que os analistas documentem seu processo de análise, expliquem as decisões tomadas e compartilhem resultados de maneira transparente e eficaz. Além disso, o Jupyter suporta bibliotecas populares de Processamento de Linguagem Natural, como NLTK e spaCy, que simplificam a extração e o processamento de texto, tornando-o uma escolha ideal para a análise de sentimentos.

Dentro do ambiente do Jupyter, o algoritmo Naive Bayes encontra um terreno fértil para a implementação na análise de sentimentos. O Naive Bayes é conhecido por sua simplicidade e eficiência, e o Jupyter oferece uma plataforma amigável para construir e treinar modelos NB. Usando bibliotecas como scikit-learn, os analistas podem importar, preparar e treinar conjuntos de dados de análise de sentimentos, avaliar o desempenho dos modelos e até mesmo criar visualizações interativas para demonstrar os resultados. A combinação do Jupyter Notebook, da linguagem Python e do Naive Bayes proporciona um ambiente robusto e adaptável para abordar tarefas de análise de sentimentos com facilidade e precisão.

3.3.2 NLTK

Natural Language Toolkit (Kit de ferramentas de linguagem natural), conhecido como NLTK, é uma plataforma de código aberto conhecida por proporcionar uma variedade de recursos em Python, ideal para trabalhar com análise de texto em PLN. Ele oferece uma vasta gama de recursos e ferramentas para tarefas relacionadas à linguagem, como tokenização, análise de sentimentos, classificação de texto, entre outros. Além disso, ele fornece acesso a diversos recursos linguísticos, como corpora (conjuntos de textos de exemplo) e léxicos (dicionários de palavras e suas informações linguísticas), tornando-o uma ferramenta valiosa para a análise e processamento de texto em uma variedade de aplicações. Neste trabalho, ele será utilizado no processo de tokenização, stopwords e em outros momentos oportunos. Entretanto, optou-se pelo uso do Spacy na etapa de lematização.

3.3.3 SpaCy

Assim como o NLTK, o Spacy é uma biblioteca usada para trabalhar com dados em Python. Possui suporte para uso em tokenização, POS, lematização, entre outros. Ele tem uma capacidade de lidar com texto em vários idiomas e seu suporte a modelos pré-treinados para esses idiomas, o que o torna uma escolha popular para projetos multilíngues. Por mais que tenha semelhanças com o NLTK, os mesmos têm abordagens diferentes. O Spacy

se mantém melhor quando o assunto é atualização e simplicidade ao utilizar sua API. Ele funciona com abordagem de orientação a objetos, enquanto o NLTK trabalha no processamento de strings. É claro, que a escolha de um ou outro depende das necessidades do projeto e dos recursos disponíveis. Logo, optou-se pelo uso do NLTK, mas ao longo do desenvolvimento viu-se a necessidade do uso da biblioteca Spacy. Seu propósito foi devido a aplicação da lematização usando o NLTK não ter sido satisfatória, devido sua construção, método linguístico e funcionalidade, e por escolha própria decidiu-se usar a do Spacy, que foi mais interessante junto ao conjunto de técnicas empregadas.

3.4 Pré-processamento de Dados: Técnicas e Métodos

A análise textual realizada neste trabalho envolve diálogos no idioma português brasileiro, que, embora predominantemente consista em português, pode conter a presença de gírias ou termos de outros idiomas. Este aspecto linguístico requer uma consideração cuidadosa ao escolher as bibliotecas e métodos que serão empregados ao longo da pesquisa. É essencial que a interpretação, etiquetagem e o tratamento dos textos sejam conduzidos de maneira precisa e em conformidade com a gramática e o contexto linguístico do Brasil. No entanto, ao lidar com idiomas menos comuns ou com nuances linguísticas específicas, pode ser necessário recorrer a traduções para realizar uma análise de sentimentos precisa. A tradução de texto para um idioma mais amplamente suportado pode simplificar a tarefa, mas é importante lembrar que a tradução pode introduzir perdas de sutilezas linguísticas e nuances culturais que podem afetar os resultados da análise.

Nem todas as bibliotecas utilizadas para o processamento de textos oferecem as mesmas funcionalidades em relação à normalização de gírias, extração do radical das palavras ou detecção da classe gramatical. Portanto, ao abordar tarefas específicas como essas, é imperativo examinar as bibliotecas com base em suas capacidades e funcionalidades individuais. Se o objetivo for normalizar gírias ou palavras coloquiais, é necessário escolher uma biblioteca que possua recursos adequados para essa finalidade, e analisar se a mesma trará um resultado satisfatório, já que nem todas as bibliotecas contam com um conjunto de gírias amplo em português. Por exemplo, a gíria “top” pode ser utilizada para indicar algo positivo, e seu significado pode ser perdido se não for normalizada adequadamente. Da mesma forma, ao lidar com a análise morfológica ou gramatical, a seleção de uma biblioteca com capacidade para identificar as classes gramaticais das palavras é crucial para a precisão do processo.

O processo de análise de sentimentos com o algoritmo Naive Bayes é conduzido através de três etapas essenciais neste trabalho, cada uma desempenhando um papel na classificação das emoções presentes nos chats de suporte da TI. Sendo elas conhecidas como o pré-processamento, a classificação de expressões e a probabilidade total do sentimento do

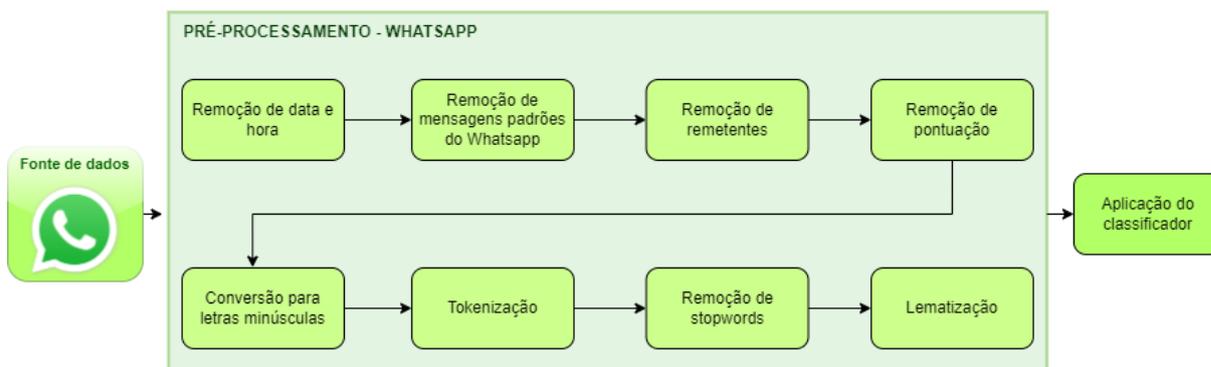
texto. Na etapa inicial do processo de treinamento dos dados, optou-se por mesclar uma parte das conversas privadas e das conversas em grupo. Essa abordagem foi adotada com o objetivo de evitar que o classificador ficasse excessivamente condicionado a um cenário específico. Ao combinar esses dois tipos de conversas, tornou-se possível criar um modelo de classificação de sentimentos mais abrangente, capaz de lidar com uma variedade de contextos de interação.

Inicialmente, é necessário realizar uma formatação dos dados que serão lidos e examinados, verificar caracteres e pontuações, a fim de facilitar o processamento e desempenho do algoritmo. Essa preparação dos dados visa tornar o texto mais legível e compatível com as operações subsequentes, além de realizar a limpeza e simplificação desses dados. Algumas das etapas comuns de formatação de textos incluem a conversão deles para letras minúsculas, e remoção de caracteres especiais, como símbolos, emojis, números, e a eliminação de pontuações desnecessárias, o que ajuda a normalizar o texto e reduzir a complexidade.

Deste modo, no processo de preparação dos dados textuais para análise de sentimentos, foi realizada a etapa de remoção de pontuação e caracteres não alfabéticos que não contribuem diretamente para a análise. Para realizar essa tarefa, foi aplicada a função *re.sub()* em Python. Essa função permite identificar e substituir padrões de caracteres indesejados por uma string vazia, removendo eficazmente esses elementos do texto. Diante disso optou-se pela escolha consciente de não tratar dos emojis na análise de emoção por vários fatores. Uma delas foi devido a complexidade e diversidade de emoticons presentes nas conversas do WhatsApp, bem como às diferenças em seu design em comparação com os emoticons usados no Spark. Essa decisão de não incluir emojis na análise de emoções foi tomada para simplificar o processo de análise e focar nas palavras e expressões textuais presentes nas conversas de TI, assim verificando o desempenho do Naive Bayes sem esse acréscimo importante dos emojis. Entretanto, a escolha em verificar sua polaridade seria muito útil e relevante em trabalhos futuros, já que esses elementos gráficos carregam informações valiosas sobre sentimentos. Além do uso da biblioteca *re.sub()* nessa tarefa, também foi usada suas funcionalidades *re.search()* e *re.match()*, empregadas na formatação do texto utilizado por meio do Whatsapp. A Figura 7 apresenta o fluxo de trabalho realizado na base de dados do Whatsapp, empregando o tratamento do texto até a aplicação do Naive Bayes para análise de sentimentos.

Outra etapa importante no pré-processamento do texto foi a conversão de todas as palavras para letras minúsculas. Nessa tarefa contou-se com a função *lower()*. A decisão de usar essa função durante o pré-processamento foi tomada com base nas características gerais do volume de dados e na necessidade de uniformizar o formato das palavras, para torná-las facilmente comparáveis. Entretanto, em outras fontes de dados, especialmente em chats online, as pessoas podem usar letras maiúsculas para enfatizar ou transmitir

Figura 7 – Pré-processamento dos dados do Whatsapp



emoções, como raiva ou entusiasmo. Quando palavras ou frases são digitadas em letras maiúsculas, isso pode ser interpretado como se a pessoa estivesse “gritando” ou enfatizando fortemente um ponto. No caso da Empresa A, que utiliza um programa diariamente, onde é necessário escrever com letras maiúsculas, optou-se pela formatação das palavras em minúsculas durante o pré-processamento. Isso ocorreu devido à peculiaridade desse ambiente de trabalho, em que o Caps Lock frequentemente permanece ativo como uma conveniência para os funcionários. Essa decisão visa evitar interpretações errôneas das conversas ao serem analisadas. Diante disso, é importante levar em conta a natureza específica do contexto de comunicação e as práticas de escrita predominantes. Portanto, ao lidar com dados textuais é essencial garantir que a análise de sentimentos reflita com precisão a intenção e as emoções expressas pelos usuários, levando em consideração as particularidades linguísticas e de formatação de cada conjunto de dados.

Na seção 2.2, foi citada algumas técnicas de PLN que costumam ser empregadas em análise de dados. Neste contexto, serão exploradas abordagens baseadas nas conversas que serão examinadas, aplicando técnicas de pré-processamento de texto. Este pré-processamento envolve etapas como tokenização, remoção de stopwords e lematização. Vale ressaltar que esta abordagem busca explorar a capacidade do Naive Bayes em lidar com tais dados em seu estado básico. Isso permitirá determinar até que ponto a simplicidade do modelo e o pré-processamento mínimo podem ser eficazes na análise de sentimentos nesse contexto específico.

3.4.1 Tokenização

Ao lidar com qualquer tipo de volume de dados que envolva o processamento de texto, seja ele gerado por meio da fala, da escrita ou de informações visuais, torna-se fundamental a aplicação de técnicas de Processamento de Linguagem Natural. Estas técnicas são essenciais para extrair informações significativas, realizar análises e tomar decisões informadas a partir desse tipo de conteúdo textual. Diante disso, foram utilizadas algumas técnicas para proporcionar uma melhor análise do texto. Entre elas a Tokenização,

que consiste na divisão do texto em unidades menores, chamadas de "*tokens*", que podem ser palavras, frases ou até mesmo caracteres, dependendo do nível de granularidade desejado. Para realizar esse procedimento, foi necessária a utilização da biblioteca NLTK e, em particular, o seu módulo de tokenização (*tokenize*). O NLTK é uma biblioteca amplamente reconhecida e utilizada em PLN que oferece uma variedade de recursos para a análise textual. A biblioteca oferece diversos tipos de formatação e manuseio ao gerar tokens, dependendo da necessidade. Essa flexibilidade é importante ao lidar com diferentes tipos de textos e idiomas. Para utilizar a função *word_tokenize*, disponibilizada no NLTK, foi necessário instalar o pacote de recursos *punkt*, ele contém modelos de tokenização pré-treinados que podem segmentar um texto em palavras, sentenças e outros elementos linguísticos. Esses modelos são essenciais para segmentar um texto de maneira adequada, levando em consideração as regras gramaticais específicas do idioma em questão. Para realizar a tokenização em um contexto da língua portuguesa, você pode utilizar o seguinte trecho de código como exemplo:

```
tokens = word_tokenize(conteudo, language='portuguese')
```

Após este processo, tem-se como resultado uma lista, onde cada elemento da lista representa um token único.

3.4.2 Stopwords

Ao lidar com um grande volume de dados de texto, é crucial simplificar o processamento e economizar tempo removendo informações que não contribuem significativamente para a análise desejada. Uma dessas técnicas eficazes é a remoção de stopwords, que são palavras muito comuns e que geralmente não carregam um peso informativo significativo, como artigos (“o”, “a”), preposições (“de”, “para”) e conjunções (“e”, “ou”). Ao eliminar essas stopwords do texto, você consegue se concentrar nas palavras-chave e nos termos mais relevantes para a análise de sentimentos ou qualquer outra tarefa de PLN. Essa etapa de pré-processamento ajuda a reduzir a dimensionalidade dos dados, economizar recursos computacionais e melhorar a eficácia dos modelos de PLN. É sempre bom lembrar que elas são listadas e checadas com base no idioma a ser utilizado, neste caso o português brasileiro. Sabendo assim que cada idioma possui suas próprias stopwords específicas, pois as palavras que são consideradas irrelevantes ou de pouco significado podem variar entre idiomas.

Ao utilizar a biblioteca NLTK, é possível definir exceções às stopwords de acordo com as necessidades específicas de análise de texto. Já que em certos casos, pode ser útil manter algumas dessas palavras se elas tiverem importância para o contexto da análise. Para definir exceções às stopwords no NLTK, você pode criar sua própria lista personalizada de stopwords, incluindo as palavras que deseja manter, ou no caso presente adicionar uma

única palavra. No código abaixo, pode ser visto um exemplo de implementação incluindo a exceção da palavra “não”.

```
stopwords_br = set(stopwords.words('portuguese'))
stopwords_br.remove('não')
```

Lidar com a negação pode ser uma preocupação importante na análise relacionada à opinião e ao sentimento (PANG; LEE, 2008). A presença e ausência da palavra “não” em contextos de sarcasmo, ironia e sentimentos de empolgação podem representar emoções diferentes das aparentemente expressas. Manter ela entre as mensagens de Suporte de TI se baseou em analisar o desempenho do algoritmo na interpretação das frases. Já que em alguns casos, a palavra "não" pode inverter completamente o significado de uma declaração, e sua exclusão poderia resultar em interpretações incorretas.

Diante disso, você pode adaptar a lista de stopwords de acordo com o contexto da análise de texto, mantendo as palavras que são relevantes para sua tarefa específica. Isso permite uma análise mais precisa e relevante dos dados textuais.

3.4.3 Lematização

Um procedimento importante a ser realizado na Análise de Sentimentos é usar de técnicas que ajudem a reduzir a dimensionalidade de um texto, tornando-o mais fácil de processar e analisar. Devido a isso utilizou-se da lematização, que vem com o propósito de simplificar as palavras em um texto, transformando-as em sua forma base ou lemas, removendo flexões verbais e nominais, e pluralização. Por exemplo, as palavras "trabalhando", "trabalhavam" e "trabalham" seriam todas reduzidas ao mesmo lema "trabalhar".

Para aplicação da lematização no desenvolvimento e análise desta base de dados contou-se com o suporte oferecido pela biblioteca Spacy, sabendo que a sua simplicidade e funcionalidade ao idioma português brasileiro é muito eficiente. Como a Spacy tem abordagem orientada a objetos, você pode obter os métodos e funções a partir da chamada do seu objeto criado. Sendo assim, é possível carregar através da função `load()` um modelo pré-treinado para uma língua específica, no caso português, e a partir do momento que for incluído o texto a ser analisado utilizar o atributo `lemma_()` para realizar a transformação em forma base da palavra. Na linha de código abaixo pode ser verificada a descrição do idioma que deve ser usado.

```
lem = spacy.load('pt_core_news_sm')
```

3.5 Análise de Dados

Após o pré-processamento são iniciadas as etapas para o processamento dos dados. Nesta fase, a biblioteca Pandas desempenha um papel importante, facilitando o manuseio e organização das mensagens e dos sentimentos. Pandas oferece uma estrutura de dados, conhecida como DataFrame, que permite armazenar e manipular eficazmente os dados. Com a combinação do Pandas e da Scikit-learn, é possível realizar tarefas essenciais, como dividir o conjunto de dados em conjuntos de treinamento e teste, criar características relevantes para o treinamento e avaliar o desempenho dos modelos.

3.5.1 Scikit-Learn

Uma biblioteca bastante utilizada em Machine Learning (Aprendizado de máquina) e mineração de dados em Python é a Scikit-Learn, também conhecida como sklearn. Com ela é possível, através de uma API simples, trabalhar com o pré-processamento de dados até a construção e avaliação dos modelos de Naive Bayes utilizados. Ela oferece uma ampla variedade de algoritmos de aprendizado supervisionado e não supervisionado, incluindo classificação, regressão, agrupamento e outros. Sua facilidade de uso, documentação abrangente e compatibilidade com outras bibliotecas a tornam uma escolha interessante para desenvolvedores. Sendo compatível com o Pandas, proporciona uma melhor aplicação dela nos dados que serão manuseados neste trabalho.

3.5.2 TfidfVectorizer

O TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer, em português, "frequência do termo – inverso da frequência nos dados", é uma técnica aplicada em PLN que funciona transformando um texto em representações numéricas significativas usadas para compreender e ajustar um algoritmo de aprendizado de máquina. O TF-IDF atribui esses valores numéricos a cada termo de um documento com base em sua relevância e frequência. A relevância é calculada considerando quantas vezes a palavra aparece no documento (frequência de termo - TF) e quantos documentos no conjunto de dados contêm a palavra (frequência inversa de documento - IDF). Assim é convertido esse conjunto de dados num espaço vetorial, isso resultará em uma matriz com a primeira linha representando um documento e cada coluna representa uma palavra única, e os valores nas células dessa matriz representam o peso TF-IDF de cada palavra em relação ao documento em questão ($TF_{ij} \times IDF_i$).

Em termos simples, a frequência do termo(TF) calcula o quão frequentemente uma palavra ocorre em um texto específico, e esse valor aumenta à medida que a palavra aparece mais vezes dentro do documento em questão. Podendo ser representada por:

$$TF_{ij} = \frac{f_{ij}}{n_j} \quad (3.1)$$

f_{ij} é a frequência do termo i em um documento j

n é o número total de palavras no documento j

Já a frequência inversa do documento (IDF) identifica quais palavras são únicas e pouco comuns em comparação com o conjunto geral de documentos. Quando uma palavra é rara, seu IDF será alto, destacando sua importância relativa em relação ao conjunto de dados completo. Isso ajuda a dar mais peso às palavras que são distintas e potencialmente mais informativas, permitindo que o modelo compreenda quais termos são menos comuns e, portanto, mais relevantes em um contexto específico. Sendo representada pela equação abaixo:

$$IDF_i = 1 + \log\left(\frac{N}{c_i}\right) \quad (3.2)$$

Ao utilizar a TF-IDF no Python é necessária a ajuda das funções `fit_transform`, que são definidas pela sua capacidade de calcular e atribuir as pontuações TF-IDF aos termos presentes nos textos do corpus. O processo começa com o ajuste (fit) do vetorizador TF-IDF das mensagens de treinamento, o que permite que ele aprenda a frequência dos termos e calcule o IDF para cada palavra. Em seguida, a função `transform` é aplicada às mensagens de teste, que converte o texto em uma matriz numérica. Essas funções permitem que os pesos TF-IDF sejam calculados com base nos dados de treinamento e, em seguida, aplicados a outros conjuntos de dados para análise de texto. Nestes procedimentos, é importante notar os ajustes que serão feitos para a aplicação dessas funções. Nas abordagens a serem realizadas, o foco estará na avaliação do desempenho dos modelos em dados de treinamento e teste. Portanto, é crucial realizar uma divisão adequada dos dados em conjuntos de treinamento e teste para avaliar como o algoritmo Naive Bayes generaliza em relação a novos dados.

3.6 Treinamento e Teste

Para ter um bom resultado utilizando o Naive Bayes, foi necessário empregar um conjunto de dados diversificado, composto por mensagens relacionadas ao suporte de TI, tanto de conversas em grupo quanto de mensagens privadas. Isso permitiu que o modelo fosse treinado com uma ampla variedade de formas de comunicação e expressões, todas abordando o mesmo assunto. Deste modo, para manipular e processar os dados textuais dentro do contexto desse domínio específico, recorreu-se ao uso do TDIF Vectorizer

juntamente com a função *fit_transform*. No entanto, é importante ressaltar que essa aplicação ocorreu após a divisão dos dados em conjuntos de treinamento e teste, o que normalmente é realizado utilizando a função *train_test_split()*. Essa função trará a possibilidade de ajustar algumas variáveis para aprimoramento do modelo de treinamento e assim colaborar na validação e aferição da precisão na análise de sentimentos.

Após atribuir as mensagens e serem calculados seus pesos, inicia a etapa de definir a base de treinamento e de testes, um subconjunto será usado para treinar um modelo de aprendizado de máquina e outro subconjunto será usado para testar a performance deste modelo. A função *train_test_split* da Scikit-learn foi ajustada da seguinte forma neste trabalho “*X, df[‘Sentiment’], test_size=0.3, random_state=22*”. Definindo os argumentos atribuídos a função, tem-se:

- **X**: As mensagens que serão divididas
- **df[‘Sentiment’]**: Os rótulos associados às mensagens
- **test_size**: A porcentagem do conjunto de dados que serão alocados para o conjunto de teste (30%)
- **random_state**: Controlador de aleatoriedade, divide os dados e garante a reprodutibilidade deles cada vez que o código é executado

E em código Python:

```
_train, X_test, y_train, y_test = train_test_split (X, df[‘Sentiment’], test_size=0.3,  
                                                random_state=22)
```

É crucial analisar os ajustes dos parâmetros, eles moldam o funcionamento do treinamento desses dados. Por exemplo, definir a quantidade de dados para teste pode depender do tamanho da sua base de dados. Ter um valor específico para a divisão dos dados em cada execução, pode dar uma reprodutividade exata e controle da execução. Em resumo, essas combinações dependem do conjunto de dados e do problema em questão. Devido o modelo ter ajustado 30% dos dados para teste, logo terão 1258 mensagens para a etapa de treinamento e 540 mensagens para a etapa de teste. Na divisão das amostras de treino, com o auxílio do parâmetro *random_state*, obteve-se 687 mensagens para a classe neutra, 361 para a classe positiva e 210 para a classe negativa. Essa desigualdade das taxas das categorias é suscetível de influenciar o resultado dos testes que envolvem fatores como peso e contagem das classes (RENNIE et al., 2003).

Este particionamento, com suas definições e parâmetros serão definidos com base na avaliação do desempenho do modelo, isso é, a maneira como se divide os dados em

treinamento e teste, bem como a proporção desses conjuntos, deve ser determinada considerando como ele irá generalizar esses dados não vistos.

3.7 Modelos Naive Bayes

Os algoritmos Naive Bayes são os algoritmos de classificação estatística que ajudam a encontrar a probabilidade condicional de ocorrência de dois eventos com base nas probabilidades de ocorrência de cada evento individual (SINGH et al., 2019). Ele é amplamente utilizado em tarefas de classificação, como análise de sentimentos. Existem diferentes versões populares do algoritmo Naive Bayes, cada uma com suas próprias suposições e características, dentre elas o NB Gaussiano, NB Multinomial e NB Bernoulli.

Este trabalho utilizou o modelo Multinomial para analisar e classificar os sentimentos presentes nas mensagens de texto relacionadas à área de TI. Embora o modelo Bernoulli também seja uma opção comum em tarefas de classificação de texto, ele é mais adequado para lidar com dados binários, como documentos que contenham palavras-chave ou não. No trabalho de Nigam e McCallum(1998), são apresentadas algumas observações interessantes sobre o desempenho dos modelos Naive Bayes, especificamente em relação ao Bernoulli Multivariado e ao Multinomial, em diferentes cenários de tamanho de vocabulário. O que chama atenção é o bom desempenho do Bernoulli em vocabulários menores, diferente do Multinomial. Entretanto, a frequência das palavras parece ser uma métrica mais informativa do que apenas a presença. Portanto, optou-se por utilizar o modelo Multinomial devido ao seu desempenho superior na consideração da frequência das palavras, o que pode ser mais preciso e eficaz na análise de sentimentos dessas mensagens. Além de ser considerado mais apropriado para lidar com as características das mensagens que geralmente são curtas, diretas e contêm uma combinação de palavras técnicas e informações específicas sobre problemas técnicos.

3.7.1 Classificador Multinomial Naive Bayes

O modelo Multinomial Naive Bayes (MNB) é um dos vários classificadores de aprendizagem de máquina utilizados para análise de texto. Tem destaque por possuir um método simples e veloz no processo de classificação. Ele é uma variação do algoritmo NB e funciona calculando a probabilidade de cada categoria possível para uma determinada amostra textual e, em seguida, atribui a categoria com a maior probabilidade como a previsão final. O MNB assume que o documento é um pacote de palavras e leva em consideração a frequência das palavras e as informações (ABBAS et al., 2019).

Ao utilizar o modelo Multinomial para classificação de texto, como análise de sentimentos. Primeiro considerasse a estimativa de máxima verossimilhança nos dados de

treinamento, onde simplesmente usa-se a frequência dos dados. Por exemplo, analise a Tabela 5 com as frases e polaridades, e a aplicação dos textos nas estimativas (3.4)(3.5)(3.6).

Tabela 5 – Mensagens e suas polaridades

				Total
Positivo	Bom dia	Está funcionando		2
Negativo	Não está imprimindo	Deu erro	Internet caiu	3
Neutro	Conecte o cabo de rede			1

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}} \quad (3.3)$$

$$Positivos - P(pos) = \frac{2}{6} \quad (3.4)$$

$$Negativos - P(neg) = \frac{3}{6} \quad (3.5)$$

$$Neutros - P(neu) = \frac{1}{6} \quad (3.6)$$

Também há a estimativa dos parâmetros, onde é ajustado os hiperparâmetros do modelo, é analisada uma fração de vezes em que uma palavra(w) aparece entre todas as outras palavras do documento da classe(c_j). Usa-se a frequência de w no documento em questão. Além de ocorrer a contagem das palavras normalizadas e assim a conclusão da probabilidade. Analise a equação (3.8) a seguir:

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)} \quad (3.7)$$

$$P("boa" | positivo) = \frac{\text{count}("boa", positivo)}{\sum_{w \in V} \text{count}(w, positivo)} \quad (3.8)$$

Há situações onde a palavra encontrada aparece pela primeira vez no documento e não se tem uma estimativa da sua frequência, isso pode ocasionar uma etiquetagem de polaridade incorreta. Isso pode ser problemático, pois a frequência de uma palavra é um dos fatores-chave para calcular as probabilidades no modelo Naive Bayes Multinomial. Para lidar com esse desafio, geralmente é aplicada uma técnica chamada suavização Laplaciana (também conhecida como suavização de Laplace). Essa técnica consiste em adicionar uma contagem fictícia de 1 a todas as palavras em todas as categorias, garantindo assim que nenhuma palavra tenha uma probabilidade condicional igual a zero. Essas etapas são críticas no treinamento do modelo, pois afeta diretamente a precisão e o desempenho do classificador.

O MNB é aplicado no algoritmo após o procedimento de particionamento dos dados para treinamento e teste. A Scikit-learn conta com a função *MultinomialNB()*, a qual foi utilizada neste trabalho para a criação e treinamento do classificador Multinomial Naive Bayes. A função vem com parâmetros opcionais a serem ajustados, dentre eles um valor para suavização Laplace/Lidstone (*alfa*), além de atributos e métodos. Entretanto, pode-se utilizar do classificador com os valores padrões, como neste caso, e ajustar quando necessário dependendo do código.

Após a etapa da escolha do classificador, deve-se utilizar o método *fit()* na instância criada do MNB. Ele será usado para treinar o modelo com base em seu conjunto de dados de treinamento, para que o modelo possa fazer previsões ou classificações em dados futuros. O método recebe *X_train* e *y_train*, respectivamente, são as mensagens de treinamento e a representação dos rótulos dessas mensagens, já abordado na seção 3.6. *X_train* é representado como um *DataFrame*, onde cada linha representa uma mensagem e cada coluna uma característica dentro dessas mensagens. Devido a técnica TF-IDF utilizada, as mensagens são reproduzidas como vetores numéricos, que formam a matriz *X_train*. O *y_train* é um vetor que contém as classes associadas às mensagens de treinamento. Sendo simbolizadas por 0 (neutro), 1 (positivo) e 2 (negativo), indicando o sentimento associado a cada mensagem. Durante o treinamento, o MNB aprende as probabilidades associadas a diferentes palavras e suas frequências em relação às classes (rótulos) fornecidas em *y_train*. É importante ressaltar que o MNB considera a representação numérica das palavras e seus rótulos durante o treinamento para criar um modelo que possa generalizar o suficiente para fazer previsões precisas em novos dados não vistos. O código ficaria semelhante a este:

```
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train, y_train)
```

São apresentadas no Quadro 3.1, as sete palavras mais frequentes ao analisar os dados e suas polaridades. No Quadro 3.2 são expostas as dez mensagens mais frequentes e o número de vezes que há a ocorrência dessas mensagens. Compreender quais palavras são mais influentes ajuda a interpretar como o modelo está tomando suas decisões de classificação.

<p>Palavras mais importantes para classe positiva: ['verificar', 'bom', 'ok', 'dia', 'obrigado', 'tarde', 'pronto']</p> <p>Palavras mais importantes para classe negativa: ['de', 'sem', 'não', 'estar', 'bom', 'internet', 'acesso']</p> <p>Palavras mais importantes para classe neutra: ['sim', 'de', 'em', 'ser', 'não', 'isso', 'ta']</p>

Quadro 3.1 – Palavras mais frequentes dos dados analisados

Mensagens mais frequentes:	
verificar:	84
bom dia:	66
:	59
ok:	42
sim:	37
obrigado:	22
bom tarde:	22
ta:	18
pronto:	17
certo:	11

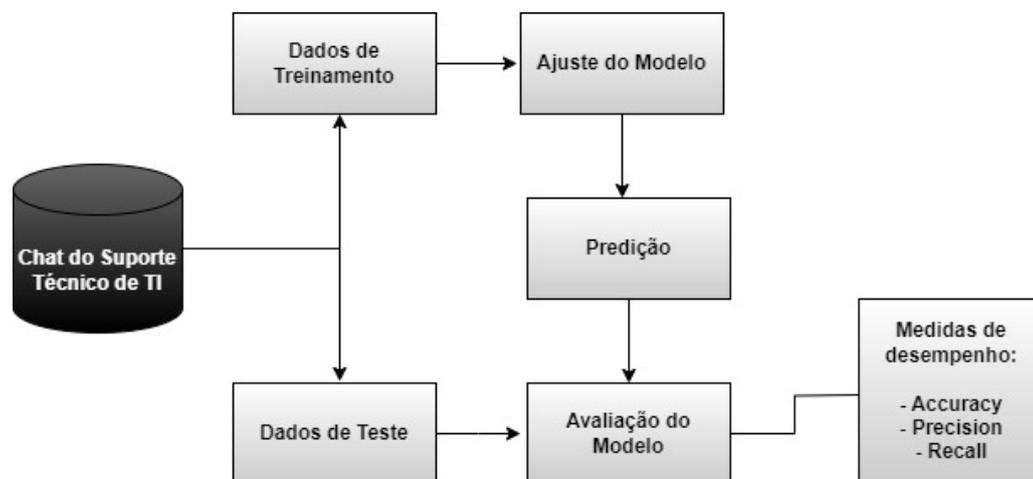
Quadro 3.2 – As 10 mensagens mais frequentes dos dados analisados

É evidente ao analisar as mensagens do chat de suporte que as palavras mais frequentes no conjunto de treinamento não seguem o padrão tradicional de aspecto positivo e negativo. Palavras como “bom”, “ruim”, “excelente” e “mau” são exemplos de características que normalmente indicam polaridade clara em tarefas de análise de sentimentos. No entanto, neste contexto específico de suporte técnico de TI, a ênfase recai em termos relacionados a desempenho, funcionalidade e solução de problemas, em vez de expressões emocionais diretas. Isso demonstra como a escolha do domínio de dados influencia diretamente as palavras-chave e os padrões de linguagem que se tornam mais relevantes.

A próxima etapa da análise tem o objetivo de aplicar os dados da fase de treinamento nos futuros volumes na busca de obter previsões ou classificações com base nos padrões aprendidos durante o treinamento, onde será utilizada a função *predict()*. Ela é aplicável a uma ampla variedade de algoritmos de classificação e regressão e é especialmente importante em tarefas como análise de sentimentos. Seu funcionamento é relativamente simples: após o treinamento do modelo com as mensagens de treinamento, é preparado um conjunto de dados de teste da mesma maneira, realiza-se a chamada da função *predict()* no MNB treinado com esses dados de teste e obtém as previsões para as amostras. No processo de avaliação do desempenho do modelo, usa-se *predict(X_test)* para fazer previsões com base nas mensagens de teste. *X_test* representa os recursos (vetores TF-IDF) dos dados de teste. As previsões resultantes são armazenadas em *y_pred_test*.

A Figura 8 mostra as etapas dos processos realizados no MNB. A base de dados tendo o conjunto de mensagens de suporte técnico, e sua divisão em treinamento e teste, devido a utilização de uma abordagem supervisionada. Logo em seguida, há os ajustes e predição do modelo, onde é ensinado o modelo a reconhecer os sentimentos e a realização de previsões com os dados de teste. E por fim, a avaliação do MNB, empregando técnicas de análise e verificação do seu desempenho.

Figura 8 – Modelo Multinomial Naive Bayes



4 Resultados

Para avaliar o desempenho do modelo MNB, é essencial empregar técnicas de verificação da qualidade e precisão do modelo. Essa etapa é importante para entender como o modelo está se saindo e se atende aos requisitos desejados. Existem várias abordagens e métricas para avaliar um modelo de classificação, e devem ser escolhidas aquelas que se alinham com os objetivos específicos do projeto. A Scikit-learn oferece funções que facilitam a análise do desempenho do classificador NB, através da importação da biblioteca *sklearn.metrics*. Dentre essas métricas será aplicada a matriz de confusão, técnica crucial para averiguar o contexto e seus alarmes falsos. É fundamental considerar a situação específica do problema, os custos associados aos diferentes tipos de erros e escolher as medidas apropriadas que melhor atendam às necessidades da aplicação.

4.1 Matriz de Confusão

É uma métrica para a avaliação de modelos de classificação, que fornece uma visão detalhada do desempenho do modelo em cada classe. Ela é representada em uma tabela de erros e acertos do modelo escolhido, neste caso o MNB, comparando com o resultado previsto. No caso das mensagens do suporte técnico de TI, por exemplo, verdadeiros positivos são os textos previstos corretamente como sentimentos positivos. É comum encontrar a tabela com esses dados com duas classes (matriz binária), isto é, uma matriz 2x2. Entretanto neste trabalho é usada a matriz multiclases para análise de sentimentos, sendo dividida em classes neutras, positivas e negativas, resultando em uma matriz 3x3. Ela é composta pelos elementos representados na Tabela 6, exibidos com a nomenclatura das polaridades para fácil entendimento.

Tabela 6 – Matriz de confusão 3x3 para Classificação de sentimentos

		<i>previsto</i>		
		Neutro	Positivo	Negativo
<i>r e a l</i>	Neutro	Verdadeiro Positivo 0	Falso Positivo 1 Falso Negativo 0	Falso Positivo 2 Falso Negativo 0
	Positivo	Falso Positivo 0 Falso Negativo 1	Verdadeiro Positivo 1	Falso Positivo 2 Falso Negativo 1
	Negativo	Falso Positivo 0 Falso Negativo 2	Falso Positivo 1 Falso Negativo 2	Verdadeiro Positivo 2

De acordo com o exemplo da Tabela 7, pode-se ver a classe neutra em destaque. Analisando sua coluna é possível visualizar seus falsos positivos neutro (0), ou seja, não era para serem neutros e foram classificados como neutros. O valor real nesse caso é positivo(1)

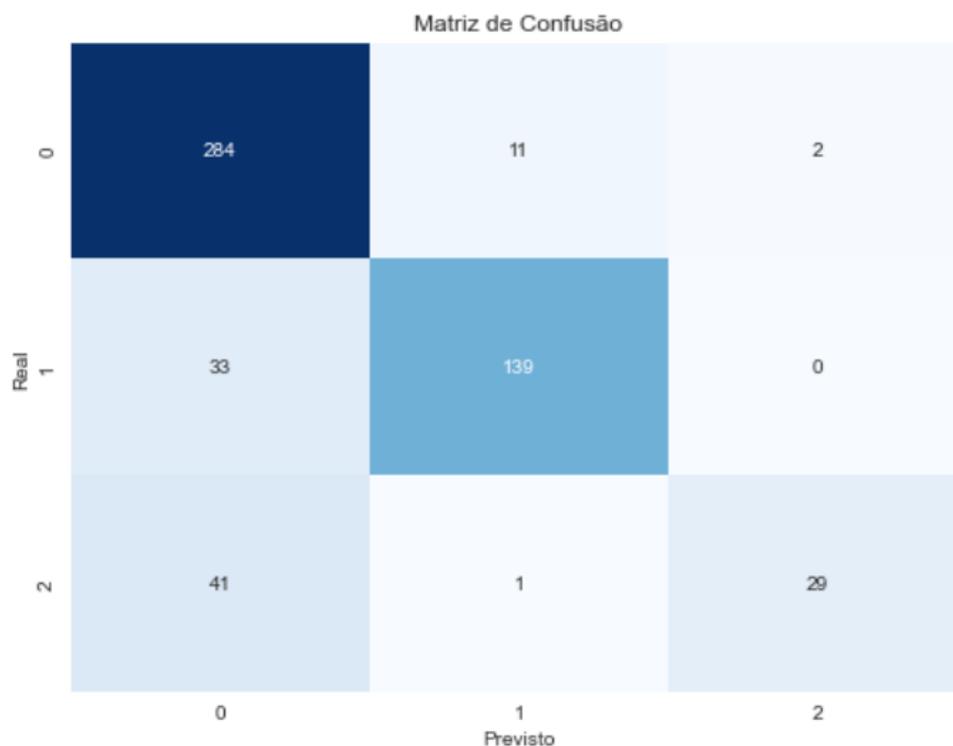
e negativo(2), mas foram classificados de forma incorreta. Agora analisando a linha aparece seus falsos positivos positivo(1) e negativo(2). Ou seja, tem-se as previsões classificadas como positivo e negativo, porém o valor real é neutro. Portanto, é importante verificar o que são os dados previstos e os dados reais.

Tabela 7 – Matriz da classe Neutra

	Neutro	Positivo	Negativo
Neutro	284	11	2
Positivo	33		
Negativo	41		

Utilizando como exemplo ainda a Tabela 7, conclui-se que 13 mensagens foram classificadas como outra polaridade, quando eram para serem neutras. Sendo que 11 foram classificadas como positivas e 2 como negativas. Outra situação analisando a matriz de confusão, é que 74 mensagens positivas e negativas foram classificadas incorretamente como neutras. Sendo dentre elas, 33 positivas e 41 negativas. Com base nessa matriz, pode-se calcular várias métricas de desempenho, como precisão, recall, F1-score e acurácia. A Figura 9 mostra a matriz de confusão do Multinomial Naive Bayes, aplicada e exibida com o auxílio da função `confusion_matrix()`, contendo as classes reais dos dados de teste e as previsões do modelo.

Figura 9 – Matriz de confusão multiclasse



Por meio da matriz de confusão é fornecido dados úteis para obtenção de outras métricas. As quais estão presentes no relatório de classificação disponível por meio da Scikit-

learn, que inclui precisão, recall, F1-score e suporte para cada classe, é uma ferramenta essencial para avaliar o desempenho de um modelo, como o MNB, e pode ser obtido através da função `classification_report()`. Ele fornece uma visão mais detalhada do quão bem o modelo está realizando a classificação, levando em consideração métricas adicionais além da acurácia geral. Isso é particularmente importante quando o conjunto de dados não é balanceado, ou seja, quando há uma distribuição desigual entre as classes. A precisão, embora seja uma métrica crucial, não é suficiente por si só, pois pode ser enganosa em casos em que o desequilíbrio de classes afeta a interpretação do desempenho do modelo.

A precisão e o recall separadamente não são métricas úteis. Considere um exemplo com mensagens de chat de suporte em uma empresa de tecnologia. Se tiver uma precisão de 1,0 para a classe “Problema Resolvido”, isso significa que todas as mensagens classificadas como “Problema Resolvido” são, de fato, casos em que o problema foi resolvido. No entanto, isso não diz quantas mensagens podem ter sido incorretamente rotuladas como “Problema Resolvido” quando, na verdade, o problema não foi solucionado. Por outro lado, um recall de 1,0 para a classe “Problema Resolvido” indicaria que todas as mensagens em que o problema foi resolvido foram identificadas corretamente como “Problema Resolvido”. No entanto, isso não informa quantas mensagens que não deveriam ter sido rotuladas como “Problema Resolvido” também foram incluídas. Em outras palavras, o recall diz o quão bom o modelo é em encontrar todos os casos reais de “Problema Resolvido”, mas não diz quantos falsos positivos (mensagens incorretamente rotuladas como “Problema Resolvido”) ocorreram.

O F1 Score mede o equilíbrio entre a precisão e o recall. Seu valor máximo é de 1,0, indicando um equilíbrio de precisão e recall perfeitos, e a pontuação mínima é 0, se a precisão ou o recall forem zero.

Além dessas métricas é necessário avaliar o desempenho do MNB, verificando a proporção de predições corretas em comparação ao total de predições realizadas pelo modelo. Sendo assim, a acurácia consegue analisar o quão preciso o classificador foi, concedendo uma visão geral da performance. Entretanto, pode haver classes desequilibradas que poderiam ocasionar uma acurácia enganosa. Por exemplo, tendo uma base de dados com 1000 mensagens de TI, onde 900 são positivas e 100 são negativas, ao chegar na etapa de treinamento e teste onde a classe positiva é majoritária, pode acarretar em um modelo simples e limitado, detectando a maioria das outras mensagens como positivas e tendo uma alta acurácia.

4.2 Resultados do experimento

O desenvolvimento deste modelo de aprendizado de máquina envolveu a análise do comportamento dos métodos diante de diversas fontes de variação. Isso incluiu a

investigação dos efeitos do tamanho da base de dados, ou seja, o número de conversas, e sua complexidade, que se refere à quantidade de informações por mensagem. Posteriormente, foi realizada uma avaliação da capacidade de generalização em relação às duas empresas. Este processo visou compreender como o MNB se comporta com conversas reais, repletas de gírias e termos técnicos.

A qualidade e a eficácia do treinamento têm um reflexo direto nas etapas de teste e, por fim, no desempenho geral do modelo. Se o treinamento foi bem-sucedido e o modelo foi capaz de generalizar os padrões do conjunto de treinamento, espera-se que ele também tenha um bom desempenho nos dados de teste. Sendo assim, foram seguidas algumas regras ao avaliar as polaridades das mensagens, dentre elas:

1 - A análise de polaridade deve ser feita com base no cenário de TI, seguindo o contexto e padrão da empresa.

2 - As mensagens devem ser avaliadas com foco no que está sendo relatado. Se o usuário está insatisfeito mesmo que sem fundamento, isso demonstra uma emoção negativa na visão dele.

No início deste experimento, ao fornecer ao algoritmo novos dados já etiquetados e submetê-los ao processo de treinamento, foi notável uma acurácia alta, apesar da base de dados ainda ser relativamente pequena. Este resultado pode ser atribuído à limitada quantidade de mensagens disponíveis para a fase de treinamento, além das suas classes desbalanceadas, acarretando em uma diversidade vocabular restrita e semelhante, assim não havendo diversos tipos de palavras para serem comparadas e testadas. Outro fator que contribuiu para esse desempenho foi o fato do tratamento do texto não ter passado por uma filtragem de dados, havendo repetições de frases como “*arquivo de mídia oculto*” e “*bom dia*”. Isso teve um impacto significativo na classificação realizada pelo modelo, levando a uma alta precisão na identificação de palavras como “*bom dia*”, enquanto a maior parte do texto foi classificada como tendo polaridade neutra. Essas repetições frequentes de frases específicas podem ter criado uma espécie de viés no modelo, levando-o a associar certas palavras a sentimentos específicos. Isso ressalta a importância de considerar a qualidade e a diversidade dos dados de treinamento, pois a presença excessiva de frases repetidas pode impactar a capacidade do modelo de discernir nuances. Assim, ao expandir o conjunto de dados com uma variedade maior de mensagens, equilibrando as polaridades e reduzindo a presença dessas repetições através do pré-processamento do texto, o modelo aprimorou mais sua capacidade de classificação de sentimentos.

Ocorreram algumas nuances interessantes nas conversas da Empresa A. Notou-se que a palavra “brother” era frequentemente usada de maneira informal como uma forma positiva de cumprimento durante o atendimento ao usuário. No entanto, essa mesma palavra causou equívocos em outra mensagem, onde um remetente se referiu a “Brother” como a marca de uma impressora, e o classificador interpretou erroneamente a polaridade como positiva.

Além disso, houve situações em que perguntas foram erroneamente classificadas como negativas, devido à interpretação do classificador como uma afirmação negativa. Contudo, o modelo obteve um resultado de desempenho satisfatório com vista na simplicidade e técnicas na aplicação do Naive Bayes.

A Figura 10 apresenta o relatório do experimento executado pela função `classification_report()`. Pode ser vista as seguintes colunas: “taxa de precisão”, “taxa de recuperação”, “F1-score” e “número de amostras”. Nas linhas têm as polaridades do experimento, “acurácia”, “média macro” e “média ponderada”.

Figura 10 – Relatório de classificação do MNB

Relatório de Classificação:				
	precision	recall	f1-score	support
0	0.79	0.96	0.87	297
1	0.92	0.81	0.86	172
2	0.94	0.41	0.57	71
accuracy			0.84	540
macro avg	0.88	0.72	0.77	540
weighted avg	0.85	0.84	0.83	540

Analisando o relatório é possível ver que o MNB não previu corretamente todas as mensagens das classes 0, 1 e 2. Para constatar a eficiência do relatório, será demonstrado o cálculo da precisão da polaridade 0 (neutro), levando em consideração a matriz de confusão da Figura 9.

$$Precisão = \frac{VP}{(VP + FP)} \quad (4.1)$$

VP - verdadeiro positivo, FP - falso positivo

$$Precisão = \frac{284}{(284 + 33 + 41)} = 0,79 \quad (4.2)$$

É possível notar no cálculo que o resultado foi preciso como esperado. Ainda analisando o relatório da classificação e a matriz de confusão, nota-se o recall da polaridade neutra(0) superior à precisão, significando que há menos Falsos Negativos 0 do que Falsos Positivos 0. Em contrapartida, na polaridade negativa(2) ve-se uma alta precisão em comparação ao recall, significando que há mais Falsos Negativos 2 do que Falso Positivo 2.

Continuando a avaliar o desempenho do classificador, tem-se outras métricas gerais úteis: a média macro e a média ponderada. A média macro do recall é calculada pela média das taxas de recall de todas as classes, sem levar em consideração o tamanho das classes. Por outro lado, a média ponderada do recall é calculada pela média das taxas de

recall, com um peso proporcional ao tamanho das classes. Diante disso, as mesmas são calculadas da seguinte forma:

$$M.Macro(recall) = \frac{0,96 + 0,81 + 0,41}{3} = 0,72 \quad (4.3)$$

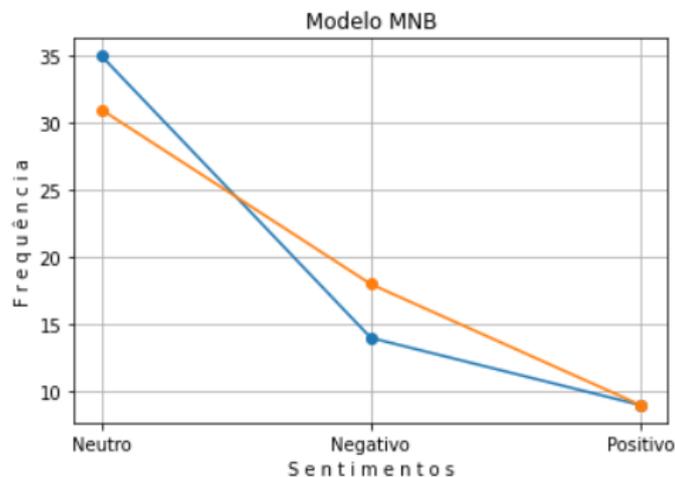
$$M.Ponderada(recall) = \frac{(0,96 * 297) + (0,81 * 172) + (0,41 * 71)}{297 + 172 + 71} = 0,84 \quad (4.4)$$

No experimento final, após alterações e testes, obteve-se uma acurácia nos dados de treinamento de 88% e nos dados de teste 84% com o modelo Multinomial Naive Bayes, como mostra o cálculo abaixo.

$$Acurácia = \frac{\text{Numero de previsões corretas}}{\text{Numero total de previsões}} = \frac{452}{540} = 0,837 = 84\% \quad (4.5)$$

É importante analisar a precisão no treinamento e no teste, para verificar se os dados estão super ajustados e se as classes estão desequilibradas. Ao realizar um teste com uma mensagem privada entre um usuário e um técnico de suporte da Empresa B, foi verificado que dentre 64 mensagens o modelo classificou 51 com sentimentos neutros, 1 com sentimentos positivos e nenhum sentimento negativo. Entretanto, foram encontradas 5 mensagens que originalmente deveriam ser classificadas como negativas e 3 que eram, na verdade, positivas, mas o modelo MNB as rotulou erroneamente como neutras. Esse desvio pode ser atribuído à presença frequente da palavra “travar” nessa conversa específica, a qual raramente aparecia nos dados de treinamento do modelo, levando a classificações imprecisas. Isso demonstra o baixo recall exibido no relatório de classificação, onde o modelo está classificando mensagens originalmente negativas com outra polaridade. Esses casos ressaltam a importância de ajustar e aprimorar o modelo continuamente para melhor lidar com uma ampla variedade de palavras e não ocorrer um desequilíbrio entre as classes.

Figura 11 – Modelo MNB testado em conversa em grupo



Valor previsto ■ Valor real ■

Sentimentos Neutros - 35, 31

Sentimentos Negativos - 14, 18

Sentimentos Positivos - 9, 9

Outro teste realizado fez uso de uma parte de uma conversa em grupo, a base de dados continha 35 mensagens etiquetadas com sentimentos neutros, 14 com sentimentos negativos e 9 com sentimentos positivos, como demonstra a Figura 11. Ao realizar uma análise mais detalhada de cada mensagem, foram identificadas algumas discrepâncias na classificação dos sentimentos. Constatou-se que 3 mensagens originalmente negativas e 1 originariamente positiva foram erroneamente classificadas como neutras. Além disso, uma mensagem que deveria ser rotulada como negativa foi classificada como positiva. Um exemplo dessa situação é a mensagem: *“net horrível aq bom jardim”*. A presença da palavra *“bom”* nessa frase pode ter levado o modelo a interpretá-la como algo positivo, embora o contexto geral da mensagem seja negativo, já que descreve uma conexão de internet ruim. Esses casos ressaltam desafios na análise de sentimentos, especialmente quando palavras ambíguas ou contextos específicos podem afetar a classificação.

Neste contexto, uma conversa com 58 mensagens obteve 5 classificadas incorretamente. Em vista que o MNB foi aplicado com poucas técnicas de PLN, num volume de dados relativamente pequeno e difícil de interpretar, o modelo conseguiu realizar a tarefa de análise de sentimentos em chats de TI de forma bastante precisa, errando em apenas um pequeno número de casos.

5 Conclusão

Neste trabalho, desenvolveu-se uma solução tecnológica focada no aprofundamento de emoções presentes em conversações online entre indivíduos. Foram utilizadas técnicas de PLN para filtrar e extrair as informações relevantes, e o classificador Naive Bayes para realizar a análise e classificação dos sentimentos das mensagens textuais. Tendo como base de dados chats de usuários comuns com profissionais de TI, nota-se diálogos e termos técnicos próprios de cada empresa verificada. Além das características referentes a padronização das estruturas pertencentes aos aplicativos Whatsapp e Spark. Essas particularidades acrescentaram mais realismo durante a avaliação do algoritmo, empregando mais nuances interpretativas e enriquecendo as etapas de teste.

Analisando a aplicação do Multinomial Naive Bayes na base de dados é perceptível o bom desempenho nas mensagens, tendo uma acurácia de 84%. É notável a detecção do algoritmo na classificação de comunicados gerais e pedidos de ajuda para solucionar defeitos em dispositivos. O fato das frases serem curtas, diretas e com gírias, são uns dos desafios de obter a polaridade correta após a aplicação da limpeza dos dados, já que os chats por serem informais carregam poucas palavras chaves. Em determinados casos, eventos que poderiam ser percebidos como negativos podem não ser corretamente identificados, devido à baixa frequência de ocorrência da palavra no texto. No entanto, mesmo diante desses desafios, o Naive Bayes destacou-se na categorização de sentimentos nas mensagens específicas de chats na área de informática. Quando comparado a outros estudos na literatura, mesmo que não compartilhem a mesma linha de dados, observa-se bons resultados obtidos. Coutinho et. al (2020) e Zulfikar et. al (2023) obtiveram uma acurácia, respectivamente, de 79% e 90,25% de acertos, utilizando do mesmo classificador.

A análise resultante das métricas de avaliação mostram valores esperados, levando em conta a divisão de classes, a quantidade de dados e a complexidade da linguagem dos chats. Este resultado não apenas valida a eficácia do método utilizado, mas também contribui para futuras análises em conversas do segmento técnico, ampliando o entendimento sobre a aplicação de técnicas de Processamento de Linguagem Natural em contextos específicos. A aplicação do classificador Naive Bayes em conversas de grupo proporciona uma compreensão mais clara do modo como todos os usuários se comunicam e interagem. Paralelamente, as conversas privadas apresentam um nível adicional de detalhamento sobre os problemas relatados, fornecendo percepções diretas para contê-los.

Diante disso, este estudo pôde mostrar ser aplicável em cenários de contexto técnico. Para trabalhos futuros seria interessante considerar um conjunto de dados de treinamento mais extenso, e a aplicação de mais técnicas de PLN como, por exemplo, bigramas e

reconhecimento de entidades nomeadas. A implementação dessas abordagens propostas poderia resultar em uma análise mais minuciosa e refinada das mensagens. Ademais, estratégias adicionais podem contribuir significativamente para análises em tempo real nos chats, considerando variáveis como o momento de abertura do atendimento, sua duração e a finalização correspondente, permitindo avaliar se o atendimento ocorreu de maneira bem-sucedida. Outro ponto a se analisar seria a aplicação de outros algoritmos de classificação, analisando sua performance e desempenho em chats de Suporte Técnico de TI.

Referências

- ABBAS, M.; MEMON, K. A.; JAMALI, A. A.; MEMON, S.; AHMED, A. Multinomial naive bayes classification model for sentiment analysis. *International Journal of Computer Science and Network Security*, 2019. Citado na página 41.
- AKRE, P.; MALU, R.; JHA, A.; TEKADE, Y.; BISEN, W. Sentiment analysis using opinion mining on customer review. *International Journal of Engineering and Management Research*, 2023. Citado na página 21.
- ARAR Ömer F.; AYAN, K. A feature dependent naive bayes approach and its application to the software defect prediction problem. *Applied Soft Computing Journal*, 2017. ISSN 15684946. Citado 2 vezes nas páginas 13 e 21.
- BLAZ, C. C. A. Análise de sentimentos em tíquetes para o suporte de ti. 2017. Disponível em: <<http://hdl.handle.net/10183/172455>>. Citado 2 vezes nas páginas 14 e 18.
- COUTINHO, E. F.; MOREIRA, L. O.; PAILLARD, G. A. L.; NETO, E. T. L. Análise do sentimento de mensagens de chats em uma turma de graduação de um curso de educação à distância. 2016. Citado na página 14.
- GRANDIN, P. H.; ADAN, J. M. Piegas: A systems for sentiment analysis of tweets in portuguese. *IEEE Latin America Transactions*, v. 14, p. 3467–3473, 2016. Disponível em: <<https://api.semanticscholar.org/CorpusID:21882021>>. Citado na página 21.
- HASSAN, M.; SHAKIL, S.; MOON, N. N.; ISLAM, M. M.; HOSSAIN, R. A.; MARIAM, A.; NUR, F. N. Sentiment analysis on bangla conversation using machine learning approach. *International Journal of Electrical and Computer Engineering (IJECE)*, 2022. Citado na página 14.
- JACKSON, P.; MOULINIER, I. *Natural Language Processing for Online Applications: Textretrieval, Extraction, and Categorization*. [S.l.]: John Benjamins Publishing Company, 2002. Citado na página 20.
- JUNQUEIRA, K. T. C.; FERNANDES, A. M. da R. Análise de sentimento em redes sociais no idioma português com base em mensagens do twitter. IX Computer on the Beach 2018, 2018. Citado na página 22.
- JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*. [S.l.]: Pearson, 2000. ISBN 0130950697. Citado 2 vezes nas páginas 20 e 21.
- KUMAR, V.; SUBBA, B. A tfidfvectorizer and svm based sentiment analysis framework for text data corpus. National Conference on Communications (NCC), 2020. Citado na página 19.
- LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, 2001. Citado na página 19.
- LANGLEY, P.; IBA, W.; THOMPSON, K. *An Analysis of Bayesian classifiers*. [S.l.]: AAAI Press, 1992. 223–228 p. ISBN 0262510634. Citado na página 22.

- LIU, B. *Sentiment Analysis and Opinion Mining*. Morgan Claypool Publishers, 2012. Disponível em: <<https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>>. Citado 2 vezes nas páginas 16 e 18.
- MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. AAAI Conference on Artificial Intelligence, 1998. ISSN 15684946. Disponível em: <<https://api.semanticscholar.org/CorpusID:7311285>>. Citado 3 vezes nas páginas 13, 22 e 41.
- METSIS, V.; ANDROUTSOPOULOS, I.; PALIOURAS, G. Spam filtering with naive bayes - which naive bayes? In CEAS, Vol. 17, 2006. Citado na página 22.
- MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. ISBN 0070428077. Citado na página 18.
- MOHAMMAD, S. M. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion Measurement*, abs/2005.11882, p. 201–237, 2016. Disponível em: <<https://api.semanticscholar.org/CorpusID:3058582>>. Citado na página 29.
- PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of the Seventh International Conference on Language Resources and Evaluation, 2010. Citado na página 19.
- PANG, B.; LEE, L. Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2008. Citado na página 37.
- PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up? sentiment classification using machine learning techniques. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, 2002. Disponível em: <<https://aclanthology.org/W02-1011>>. Citado na página 22.
- RANI, R.; TANDON, S. Chat summarization and sentiment analysis techniques in data mining. 2018 4th International conference on Computing Scienses (ICCS), Jalandhar, India, 2018. Citado na página 12.
- RENNIE, J. D. M.; SHIH, L.; TEEVAN, J.; KARGER, D. R. Tackling the poor assumptions of naive bayes text. AAAI Press, 2003. Citado na página 40.
- RENUKADEVI, N.; NANTHITHA, S.; SARASWATHI, K.; SHOBICA, S.; KARTHIKA, R. Whatsapp group chat analysis by using machine learning. International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023. Citado na página 13.
- RIBEIRO, M. S.; COUTINHO, E. F. Aplicação de técnicas de análise de sentimento para classificação de postagens em ambientes virtuais de aprendizagem. 2020. Citado 2 vezes nas páginas 14 e 53.
- SHANAHAN, J. G.; QU, Y.; WIEBE, J. Springer Dordrecht, 2006. Citado na página 16.
- SINGH, G.; KUMAR, B.; GAUR, L.; TYAGI, A. Comparison between multinomial and bernoulli naïve bayes for text classification. International Conference on Automation, Computational and Technology Management (ICACTM), 2019. Citado na página 41.

- WHITELAW, C.; GARG, N.; ARGAMON, S. Using appraisal groups for sentiment analysis. Proceedings of the 14th ACM international conference on Information and knowledge management, 2005. Citado na página 16.
- WICKRAMASINGHE, I.; KALUTARAGE, H. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft computing, 2021. Citado na página 21.
- WIEBE, J.; WILSON, T.; BRUCE, R.; BELL, M.; MARTIN, M. Learning subjective language. Computational Linguistics, v. 30, 2004. Citado na página 17.
- YADOLLAHI, A.; SHAHRAKI, A. G.; ZAIANE, O. R. Current state of text sentiment analysis from opinion to emotion mining. ACM Computing Surveys, Volume 50, Issue 2, 2017. Citado na página 16.
- YU, H.; HATZIVASSILOGLU, V. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003. Disponível em: <<https://aclanthology.org/W03-1017>>. Citado na página 22.
- ZIVOLO, M. V.; PRADO, S. C. C. Estudo comparativo dos algoritmos de aprendizado de maquina naive bayes, arvore de decisao e redes neurais artificiais para analise de sentimento – implementação com pandas e scikit-learn. 2021. Disponível em: <<https://congresso.fatecmococa.edu.br/index.php/congresso/article/view/323>>. Citado na página 13.
- ZULFIKAR1, W. B.; ATMADJA, A. R.; PRATAMA, S. F. Sentiment analysis on social media against public policy using multinomial naive bayes. *Scientific Journal of Informatics*, 2023. Disponível em: <<https://doi.org/10.15294/sji.v10i1.39952>>. Citado na página 53.