IDENTIFICAÇÃO DE SENTIMENTOS EM TEXTOS UTILIZANDO O MODELO *TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY*

SENTIMENT ANALYSIS IN TEXTS USING THE TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY MODEL

Geovana P. P. Landim[1], Guilherme J. Tresso[2], Jefferson A. R. Passerini[3]

[1]Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, geovana.landim@fatec.sp.gov.br
[2]Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, guilherme.tresso@fatec.sp.gov.br
[3]Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, jefferson.passerini@fatec.sp.gov.br

Informação e Comunicação
Subárea: Matemática e Inteligência Computacional

RESUMO

Este trabalho abordou a aplicação de técnicas de Processamento de Linguagem Natural (PLN) na classificação de sentimentos em textos como positivos e negativos. Utilizando a linguagem Python e bibliotecas como Numpy, Pandas, Scikit-Learn, Matplotlib e NLTK, construiu-se um modelo de PLN com ênfase na análise de textos em língua inglesa, utilizando uma base de dados de avaliações de filmes do Rotten Tomatoes. O processo incluiu o pré-processamento dos dados, onde substituímos abreviações e removemos caracteres especiais, com exceção da palavra "*not*", que é fundamental na classificação de textos negativos. Foi investigado o impacto do uso de *stop words*, concluindo que sua manutenção melhora a qualidade da análise, equilibrando o peso das palavras. Comparando os classificadores MultinomialNB e SVM, observou-se que o primeiro teve um desempenho superior de 77,5% de acurácia sem a utilização de *stop words*, devido à sua eficácia na análise de dados textuais. Também houve a utilização de um método de *ensemble*, que combinando os resultados dos classificadores, obteve 76,9% de acurácia sem a utilização de *stop words*, não superando o desempenho do classificador MultinomialNB. Este estudo oferece insights valiosos sobre a importância do pré-processamento de dados e da escolha adequada de classificadores em tarefas de PLN.
Palavras-chave: processamento de linguagem natural; análise de sentimentos ; TF-IDF; SVM; multinomialnb.

ABSTRACT

This work addressed the application of Natural Language Processing (NLP) techniques in sentiment classification of texts as positive and negative. Using the Python language and libraries such as Numpy, Pandas, Scikit-Learn, Matplotlib, and NLTK, a natural language processing model was built with a focus on the analysis of English texts, using a dataset of movie reviews from Rotten Tomatoes. The process included data preprocessing, where abbreviations were replaced, and special characters were removed, except for the word "not," which is crucial in the classification of negative texts. We investigated the impact of using stop words, concluding that their retention improves the quality of the analysis by balancing the weight of words. Comparing the MultinomialNB and SVM classifiers, we observed that the former achieved a superior performance of 77.5% accuracy without the use of stop words due to its effectiveness in textual data analysis. We also explored the use of an ensemble method, which, by combining the results of classifiers, achieved 76.9% accuracy without using stop words, but it did not surpass the performance of the MultinomialNB classifier. This study provides valuable insights into the importance of data preprocessing and the appropriate choice of classifiers in NLP tasks.
Keywords: natural language processing; sentiment analysis; TF-IDF; SVM; multinomialnb.

1 INTRODUÇÃO

Atualmente a informação é um dos bens mais preciosos para todos os processos humanos. A quantidade de informações disponíveis é gigantesca e, além disso, a maioria destas informações estão disponíveis na internet em formato de textos. Este grande volume de dados, denominado *Big Data*, gerou transformações drásticas na sociedade em diferentes aspectos como na cultura, política e na economia. Isso porque, por meio deles, é possível captar opiniões e sentimentos das pessoas, além de possibilitar que instituições financeiras e não financeiras, bem como o governo, tomem melhores decisões e definam suas estratégias.

Porém, coletar as informações sem processá-las adequadamente é apenas um desperdício de recursos. Para isso, o processamento de linguagem natural é uma ferramenta muito importante para realizar esta tarefa. Por meio dela, é possível que as empresas compreendam como as pessoas estão se sentindo em relação a determinados tópicos, produtos e serviços. O processamento de linguagem natural trata de uma grande área da inteligência artificial, que visa a interação entre a linguagem humana com a linguagem do computador. Além disso, permite a identificação de sentimentos em textos, tais como alegria, raiva, tristeza e neutralidade. Esta ferramenta pode ser aplicada para a análise de redes sociais, atendimento ao cliente e pesquisa de mercado. Assim, as empresas podem melhorar a experiência do cliente ao consumir produtos e serviços, bem como agir mais rapidamente para sanar possíveis problemas.

O objetivo geral do trabalho é identificar sentimentos em textos utilizando o processamento de linguagem natural, através de aprendizado de máquina supervisionado. Mais especificamente, o objetivo é compreender quais são os melhores classificadores de texto entre SVM, MultinomialNB e o Método *Ensemble*, usando o vetorizador TF-IDF.

O trabalho justifica-se, pois a análise de sentimentos é uma área de pesquisa em constante evolução e tem sido aplicada em diversas áreas. A identificação de sentimentos em textos pode ser útil para empresas que desejam avaliar a satisfação do cliente em relação a seus produtos ou serviços, bem como monitorar a reputação da marca e identificar problemas antes que eles se tornem críticos.

O uso de modelos de aprendizado de máquina supervisionado, como SVM, MultinomialNB e o Método *Ensemble*, usando o vetorizador TF-IDF, pode ajudar a melhorar a precisão da análise de sentimentos em textos. Esses modelos são capazes de aprender com exemplos rotulados e aplicar esse conhecimento para a classificação de novos textos. A comparação desses modelos pode ajudar a identificar qual modelo é mais adequado para uma determinada tarefa.

2 REFERENCIAL TEÓRICO

Esta seção tem como propósito introduzir os conceitos essenciais para esclarecer a metodologia proposta, abrangendo informações sobre o processamento de linguagem natural, bem como os modelos e algoritmos empregados neste trabalho.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL E APRENDIZADO DE MÁQUINA

A Inteligência Artificial (IA) é um grande campo de estudos que desenvolve tecnologia capaz de tornar as máquinas inteligentes. Mccarthy (2007) descreve Inteligência Artificial como a ciência e engenharia para a criação de máquinas inteligentes, com destaque para softwares inteligentes, capazes de entender a inteligência humana, mas que não se limitam a isso.

Russell e Norving (2013) distinguiram a Inteligência Artificial em quatro objetivos diferentes: pensar como humano, pensar racionalmente, agir como um ser humano e agir racionalmente. O agir como humano pode ser descrito pelo Teste de Turing, proposto por Turing (1950), em que o computador passa por um teste com um interrogador humano e, após várias perguntas, o humano não consegue identificar se as respostas escritas vêm de um computador ou não. Para que o teste fosse viabilizado, o computador teria as capacidades de Processamento de Linguagem Natural, Representação de Conhecimento, Raciocínio Automatizado e Aprendizado de Máquina. Neste trabalho, o foco será o Processamento de Linguagem Natural e o Aprendizado de Máquina.

Iberdrola (2023) explica que Aprendizado de Máquina é uma das disciplinas de Inteligência Artificial que, através de algoritmos, permite ao computador identificar padrões em um grande conjunto de dados e a partir disso fazer previsões. De acordo com Russell e Norvig (2013), existem três principais tipos de aprendizagem: Aprendizagem Não Supervisionada, Aprendizagem por Reforço e Aprendizagem Supervisionada.

Na aprendizagem não supervisionada, são identificados padrões de entrada, mas sem ter um feedback explícito. No caso, os dados não são rotulados. Sendo assim, a tarefa mais comum na aprendizagem não supervisionada é o agrupamento de grupos identificados como úteis. Por exemplo, pode ser desenvolvido gradualmente o conceito de um dia bom ou um dia ruim de tráfego para um taxista, sem essa informação ter sido rotulada por um humano.

Na aprendizagem por reforço, o algoritmo aprende com recompensas e punições. Ainda no exemplo do taxista, a falta de gorjeta após uma corrida dá a indicação de que algo está errado. Bem como a vitória em um jogo de damas indica que o jogador fez a coisa certa. É conferido então ao agente decidir quais das ações anteriores ao reforço são responsáveis por estes resultados.

Na aprendizagem supervisionada, o agente observa exemplos de pares de entrada e saída e adquire um entendimento de como transformar a entrada em saída. No primeiro componente mencionado, as entradas consistem em percepções, e a saída é dada por um instrutor, que indica "Freie!" ou "Vire à esquerda". No segundo componente, as entradas são imagens da câmera, e as saídas são fornecidas por um instrutor que as classifica como "isso é um ônibus". No terceiro componente, a teoria da frenagem é uma função que leva em consideração os estados e as ações relacionadas à frenagem até a distância de parada. Nesse caso, o valor de saída é diretamente derivado da percepção do agente (posteriormente); o ambiente atua como o instrutor.

O Processamento de Linguagem Natural (PLN) é uma tecnologia de Inteligência Artificial, frequentemente associada ao Aprendizado de Máquina, que permite aos computadores interpretar, compreender e manipular a língua humana AWS (2023a). Esta tecnologia está se tornando cada vez mais relevante, uma vez que empresas em todo o mundo lidam com volumes significativos de dados que precisam ser analisados e respondidos, incluindo e-mails, mensagens de texto, comentários em redes sociais, mensagens de voz, vídeos, entre outros. Conforme a AWS (2023b) também destaca, o uso do PLN é extremamente valioso, pois é capaz de interpretar dialetos, gírias e irregularidades gramaticais, além de ter a capacidade de processar documentos extensos, analisar avaliações de clientes e responder aos clientes por meio de *chatbots*. Além disso, o PLN pode classificar e extrair informações de texto.

De acordo com Oracle (2023), a pesquisa sobre Processamento Natural se iniciou após a invenção de computadores digitais por volta da década de 1950, sendo que o processamento de linguagem natural se baseia em Inteligência Artificial, mas os principais avanços dos últimos anos se deram por meio de aprendizado de máquina, juntamente com o aprendizado profundo, que é um tipo de aprendizado de máquina, mas que envolve, além de processar um grande volume de dados, o aprendizado de padrões mais complexos.

Os primeiros modelos de análise de linguagem natural eram simbólicos e estavam baseados em codificar manualmente as regras de linguagem, permitindo a interpretação de tempo, verbos e significado raiz, conforme explica Iberdrola (2023). Entre as décadas de 80 e 90 houve a revolução estatística; neste momento, a inserção de regras e exceções da língua deixou de ser feita manualmente, pois os dados passaram a ser inferidos, ou seja, a análise estatística realiza comparações e identifica padrões, permitindo a compreensão de novas palavras e a detecção de erros.

Iberdrola (2023) demonstra que a maioria dos sistemas utiliza uma combinação de modelos simbólicos e estatísticos. Além disso, no caso dos sistemas de processamento de linguagem natural, são realizados diversos tipos de análises, a saber:

- Morfológica: esta análise distingue os diferentes tipos de palavras e suas variações.
- Sintática: separa uma frase da outra e analisa os seus respectivos componentes.
- Semântica: analisa o significado das palavras individuais e o contexto em que estão inseridas em uma frase.
- Pragmática: identifica a intenção do texto, incluindo a detecção de ambiguidades, ironias ou estados de ânimo.

Os tipos de análise acima permitem a realização de diferentes tipos de aplicação, como classificação de documentos, análise de sentimentos e de opiniões, realizar a comparação de textos e tornar documentos anônimos, removendo menções de dados pessoais em textos. É importante destacar, conforme explicado por Caseli e Nunes (2023), que o Processamento de Linguagem Natural (PLN) é dividido em duas grandes subáreas: Compreensão de Linguagem Natural (*Natural Language Understanding* - NLU) e Geração de Linguagem Natural (*Natural Language Generation* - NLG).

A NLG engloba todos os aspectos relacionados à análise e interpretação da linguagem. A análise envolve a separação e categorização dos elementos linguísticos, como palavras e suas categorias morfológicas e gramaticais, bem como seus atributos semânticos e ontológicos.

A interpretação, por sua vez, refere-se à tentativa de compreender os significados construídos pelos seres humanos. Por exemplo, em uma interação com um *chatbot*, o software processa o texto do usuário e responde de acordo, seja atendendo a uma solicitação ou executando um comando. A precisão da resposta pode variar conforme as expectativas do usuário. O que a máquina interpreta nem sempre corresponde perfeitamente às nossas expectativas humanas.

Por outro lado, em relação à Geração de Linguagem Natural (NLG), conforme explicado por Caseli e Nunes (2023), o objetivo é gerar respostas em linguagem natural. No caso dos *chatbots*, o exemplo de maior êxito é o ChatGPT, que produz respostas ágeis e, frequentemente, é mais fluente do que as respostas humanas.

2.2 RECUPERAÇÃO DE INFORMAÇÃO

Assim como no Processamento de Linguagem Natural, a Recuperação de Informação (RI) é utilizada para lidar com um grande volume de dados, geralmente não estruturados. Este campo interdisciplinar engloba a ciência da computação, linguística e ciência da informação. De acordo com Russell e Norving (2013), a RI possibilita a localização de documentos importantes contendo as informações necessárias para um usuário. Um exemplo notável de aplicação da RI é na *World Wide Web* (WWW), onde o usuário faz uma consulta e recebe uma lista de páginas relevantes com base no que foi inserido.

Conforme Russell e Norvig (2013) explicam, um sistema de recuperação de informação possui as seguintes características:

- Um corpus (um conjunto de textos usado para treinamento e teste de um modelo) de documentos. Cada sistema define o que considera como documento, podendo ser uma sentença, um parágrafo ou várias páginas de texto.

- Consultas em linguagem de consulta, o que significa que quando o usuário realiza uma consulta específica com um termo sobre o que deseja saber, o sistema retorna uma lista de itens que são iguais ou similares. Por exemplo, uma busca por "Livro IA" pode retornar não apenas a frase literal "Livro IA", mas também itens que incluam operadores booleanos, como "Livro e IA", e respostas não booleanas, como "IA próximo a Livro" ou "Livro de IA site: www.aaai.org".

- Um conjunto de resultados que o sistema de RI julga ser relevante para o usuário que fez a consulta.

- Apresentação de um conjunto de dados, que pode variar em complexidade, desde algo simples até algo mais elaborado.

Anteriormente os Sistemas RI utilizavam o modelo booleano. Após a revolução estatística, passou a usar a estatística para contar a frequência de palavras e, atualmente, existe uma função de pontuação, que atribui maior peso para as palavras que são mais importantes no documento.

2.3 *STOP WORDS*

Stop words ou “palavras de parada” em português, conforme explica Silva, Vieira e Osorio (2005), refere-se a uma técnica de remoção de palavras no pré-processamento de textos. Em suma, é uma técnica que remove palavras que não são informativas (pois são amplamente utilizadas na linguagem, mas que raramente contribuem para a identificação de tópicos, sentimentos ou características específicas do texto) ou termos semânticos relacionados ao mesmo termo raiz. As *stop words* são amplamente utilizadas em tarefas como as de Processamento de Linguagem Natural ou Recuperação de Informações.

No uso das *stop words* são corrigidas as abreviações e removidas palavras de conexão como “*the*”, “*is*” e “*are*” que são importantes para dar conexão às frases utilizadas para comunicação entre humanos, mas irrelevantes quanto ao peso para dar sentido à frase. Para mais, conforme explica Moser (2021), a remoção das *stop words* melhora a eficiência do método TF-IDF, uma vez que palavras vazias, como conjunções e artigos, não vão incidir sobre o cálculo.

Resumidamente, *stop words* são as palavras comuns do cotidiano e que costumam ser removidas do texto no pré-processamento dos dados no Processamento de Linguagem Natural. Geralmente, essas palavras não são de grande contribuição para a análise de texto e ainda podem ser um ruído nos dados.

2.4 TF-IDF

TF-IDF é uma sigla em inglês que representa "*Term Frequency - Inverse Document Frequency*", que em português pode ser traduzida como "Frequência do Termo - Frequência Inversa do Documento". De acordo com a explicação fornecida por Moser (2021), o TF-IDF é um mecanismo de Recuperação de Informação associado ao Processamento de Linguagem Natural.

O termo "TF" refere-se à contagem da frequência com que uma palavra ocorre em um conjunto de documentos, atribuindo pesos às palavras com base em sua frequência. Já o "IDF" analisa a importância das palavras, levando em consideração que uma palavra pode aparecer muitas vezes, mas isso não implica necessariamente que ela seja relevante para a análise. Por

exemplo, no inglês, o artigo "*the*", que equivale a "o" ou "a" em português, costuma ocorrer frequentemente, mas não tem grande importância no contexto geral da análise.

Segundo Moser (2021), a fórmula para calcular TF-IDF é $TF\_IDF = TF * IDF$. Sendo que *Term Frequency* (TF) representa a frequência de uma palavra "*p*" em um documento "*d*". É fundamental computar a contagem de todas as palavras; neste caso a fórmula apresentada é:

$$TF(p, d) = \big(count(p, d)\big).$$

É importante destacar que uma palavra pode aparecer com frequência elevada em um documento, mas isso não necessariamente a torna mais importante. Portanto, é comum aplicar uma redução na frequência bruta, utilizando o logaritmo na base 10 da frequência. Além disso, costuma-se adicionar 1 à contagem, uma vez que não é possível calcular o logaritmo de 0. Portanto, a fórmula fica como segue abaixo:

$$TF(p, d) = (count(p, d) + 1).$$

Conforme mencionado anteriormente, o IDF refere-se à Frequência Inversa de Documento e tem como objetivo equilibrar o TF, ou seja, o IDF atribui menor importância às palavras que são frequentes em todos os documentos. O cálculo da equação resulta em um vetor de pesos, um para cada palavra do documento, onde "D" representa o conjunto de documentos e "df" é a função que conta a frequência de um termo "t" em cada documento:

$$IDF(t) = \left(\frac{count(D)}{df(t)}\right)$$

O objetivo desta abordagem é gerar um vetor com os dados preparados para que se possa aplicar um classificador de forma a determinar a predição final do texto.

2.5 CLASSIFICADORES

Russell e Norving (2013) explicam que a classificação em Processamento de Linguagem Natural, também conhecida como categorização de texto, é uma tarefa fundamental que envolve a atribuição de rótulos a documentos de texto, a fim de categorizá-los em grupos predefinidos. Esses grupos podem representar classes, temas, sentimentos ou qualquer outra categoria relevante.

A classificação de texto desempenha um papel crucial em várias aplicações, como identificar o idioma de um texto, categorizar gêneros literários e analisar sentimentos em textos. Um caso comum de classificação de sentimentos é rotular uma análise de produto como "positiva" ou "negativa". Além disso, a classificação é essencial na detecção de spam, onde mensagens de e-mail são categorizadas como "spam" ou "não spam."

Russell e Norving (2013) definem a classificação na aprendizagem de máquina como o processo de identificação de características ou padrões em um conjunto de dados e a atribuição de rótulos ou categorias a esses dados com base em um contexto específico.

Os classificadores são algoritmos usados para prever valores em variáveis categóricas discretas ou contínuas com base em um conjunto de dados de treinamento. Neste trabalho, adotaremos uma abordagem de aprendizado supervisionado para classificar os dados extraídos em duas classes de sentimentos: "bom" e "ruim". Para isso, utilizaremos os classificadores *Multinomial Naive Bayes*, Máquinas de Vetor de Suporte (SVM) e um classificador por votação, que combina as previsões dos dois primeiros.

De acordo com Sriram (2022), o algoritmo *Multinomial Naive Bayes* (MultinomialNB) é um classificador probabilístico amplamente utilizado na análise de texto. Ele calcula a probabilidade de uma etiqueta de texto para uma amostra específica e fornece a etiqueta com a probabilidade mais alta como resultado. Este classificador é particularmente adequado para a

análise de sentimentos com base em texto, uma vez que considera a frequência das palavras nas amostras, o que é essencial para determinar a polaridade do sentimento.

Conforme explicado por Takahashi (2012), o *Support Vector Machine* (SVM) é um algoritmo de aprendizado de máquina que se baseia na criação de uma linha ou superfície de separação entre as classes de dados em um espaço dimensional. O SVM foi escolhido neste contexto devido à sua capacidade de manipular dados de texto, especialmente quando usado em conjunto com o vetorizador TF-IDF (*Term Frequency-Inverse Document Frequency*), que transforma o texto em vetores ponderados. O SVM possui vários hiperparâmetros, como o tipo de kernel, o valor de *gamma* e o parâmetro de regularização (C), que afetam o desempenho do modelo.

Uma das razões fundamentais para a escolha do SVM é o uso do Kernel RBF *(Radial Basis Function*), conforme explicado por Russell e Norving (2013). O Kernel RBF mapeia as informações em um espaço dimensional mais alto e atribui pesos às informações com base em sua proximidade ao centro desse espaço, usando uma função gaussiana. Essa abordagem permite ao modelo lidar com a complexidade inerente aos dados de texto, tornando-o eficaz na classificação.

Como mencionado por Luchese (2023), o SVM possui hiperparâmetros importantes, incluindo o parâmetro de regularização C, que foi configurado com um valor alto (10). Quando C é elevado, o SVM ajusta a fronteira de decisão em relação aos dados de treinamento, minimizando classificações incorretas e aumentando a precisão do modelo. O parâmetro *gamma* controla o nível de generalização do modelo, e a escolha adequada de *gamma* é crucial para evitar generalizações excessivas.

Também foi desenvolvido um classificador usando a técnica de *ensemble*, que combina as previsões dos dois classificadores mencionados anteriormente, SVM e MultinomialNB. A técnica de *ensemble* é eficaz para melhorar a precisão do modelo, pois combina as forças de diferentes algoritmos.

A inclusão ou exclusão de *stop words*, como mencionado por autores como Silva, Vieira e Osorio (2005), será testada como parte do experimento. *Stop words* são palavras comuns, como "e," "o," "de," que podem ser incluídas ou excluídas do processo de classificação. A inclusão de *stop words* geralmente auxilia na distinção e classificação mais precisa das palavras nos textos, melhorando o desempenho dos classificadores, como o MultinomialNB e o SVM, em tarefas de análise de texto.

2.6 *VOTING CLASSIFIER*

O conjunto, ou *ensemble*, é uma técnica amplamente utilizada em modelos de aprendizado de máquina, que envolve o uso de vários classificadores no mesmo modelo com o objetivo de extrair o melhor de cada um deles. Isso pode ser feito de forma independente ou combinando suas previsões.

Um dos métodos de *ensemble* é o *Voting Classifier*, ou classificador por votação. De acordo com Zhou (2012), é uma técnica popular e fundamental quando se trabalha com resultados nominais, ou seja, categorias que não possuem uma ordem intrínseca. O *Voting Classifier* utiliza a análise de vários classificadores para tomar uma decisão final.

O *Voting Classifier* pode ser configurado de duas maneiras: *soft* e *hard*. Segundo Awan-Ur-Rahman (2023), a divisão basicamente apresenta as seguintes diferenças: no modo *soft*, as previsões dos classificadores têm pesos diferentes, formando uma média ponderada. No modo *hard*, todos os classificadores têm o mesmo peso e a decisão final é uma média simples das previsões. Por exemplo, se um *Voting Classifier* tiver três classificadores e dois deles previrem resultados positivos, o resultado será positivo.

Uma das principais vantagens do *Voting Classifier* é a melhoria da generalização dos resultados. Ao combinar os resultados de vários modelos, a precisão e o desempenho geral aumentam. Segundo Simic (2023), isso ocorre porque, ao juntar os classificadores em um *Voting Classifier*, é mais provável que seus erros de precisão sejam diferentes, o que aumenta a probabilidade de obter previsões precisas no conjunto.

Para ilustrar, imagine um *Voting Classifier* com três classificadores treinados na mesma base de dados e com o mesmo pré-processamento. Se o primeiro classificador prevê um resultado negativo, enquanto os outros dois preveem um resultado positivo, é mais provável que a decisão final seja positiva, uma vez que dois modelos chegaram à mesma conclusão.

Em resumo, o *Voting Classifier* é uma técnica de *ensemble* valiosa para melhorar o desempenho de modelos de aprendizado de máquina, combinando as previsões de vários classificadores para obter uma decisão final mais precisa e robusta.

2.7 MÉTRICAS DE DESEMPENHO

Foi utilizado o método *Hold-out*, conforme descrito por Kumar (2023), como uma abordagem para treinar o modelo de aprendizado. Esse método envolve a divisão dos dados em dois grupos: um para treinamento e outro para validação e teste do modelo, sendo o *Hold-out* comumente utilizado para comparar diferentes modelos.

Ao aplicar o método *Hold-out*, é recomendável alocar a maior parte dos dados para o conjunto de treinamento, geralmente em torno de 70%. Observou-se que, tanto na literatura quanto em nossos experimentos, os classificadores apresentaram resultados ligeiramente superiores quando utilizada a proporção de 70%, em comparação a porcentagens mais baixas ou mais altas. Essa escolha se justifica pelo fato de que ambos os classificadores requerem uma quantidade substancial de dados para atingir um desempenho ideal.

Portanto, a opção de alocar 70% dos dados para o conjunto de treinamento não só oferece uma melhor visualização dos resultados, mas também representa uma das abordagens mais apropriadas para esse propósito.

Outra métrica de desempenho utilizada foi a acurácia dos classificadores, segundo Dicio (2023), a “proximidade entre o resultado de um instrumento de medida e o verdadeiro valor do que foi medido”. Ou seja, será feita uma análise do resultado do modelo comparado com o que realmente deveria ser seu resultado, caso tivesse acertado todos os valores.

2.8 BASE DE DADOS

Os dados utilizados foram obtidos do site Rotten Tomatoes (2023), um site em inglês amplamente utilizado para a avaliação de filmes. A base de dados foi dividida em duas partes e classificada com valores, atribuindo o valor 1 a frases positivas e 0 a frases negativas. Frases neutras foram classificadas como positivas, a fim de aprimorar o desempenho do algoritmo.

3 METODOLOGIA

O trabalho propõe a abordagem de processamento de linguagem natural (PLN) para classificação de textos em positivos e negativos, conforme demonstra-se na figura 1. O modelo foi desenvolvido utilizando a linguagem Python (2023) em sua versão 3.7.13, e foram utilizadas as bibliotecas Numpy (2023), Pandas (2023), Scikit-Learn (2023), Team (2023) e a biblioteca NLTK (2023). Para o desenvolvimento e testes do modelo proposto foi utilizada a ferramenta Colaboratory do Google. Esta ferramenta oferece maior facilidade de configuração de ambiente de testes de forma remota com a criação de uma máquina nos servidores da empresa.

A partir da base de dados do Rotten Tomatoes (2023), obteve-se dois arquivos com textos com avaliações de filmes. Tais textos estão na língua inglesa e separados em dois arquivos distintos, onde em um dos arquivos apresentam-se as mensagens de caráter positivo ou neutro, que no processo de desenvolvimento do projeto serão classificadas como “positivas”, ou seja, será atribuído o valor “1”, e um segundo arquivo onde encontram-se as mensagens de caráter negativo que se rotulou com o valor “0”. Os arquivos “negativo” e “positivo” são unificados onde os dados apresentam-se pelas colunas “valor” e “texto”.

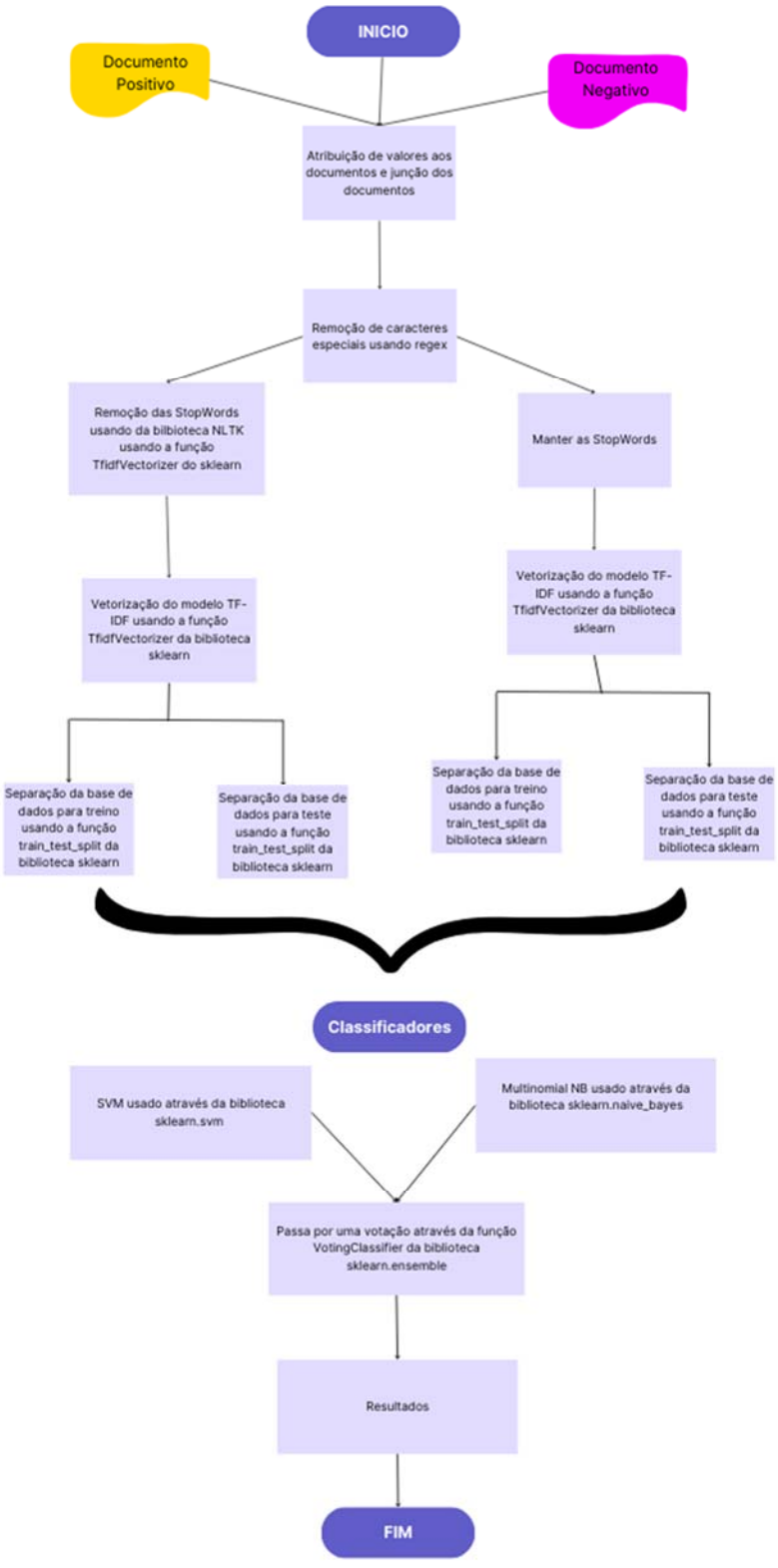
Após a preparação do arquivo a partir da base de dados, realizou-se um pré-processamento a fim de melhorar a qualidade dos dados que o modelo proposto processou. Para isso foi realizada a substituição de abreviações da palavra “*not*” para a própria palavra, o que facilita ao modelo identificar e compreender os textos negativos na base. Em seguida foram removidos todos os caracteres especiais, como "!", "@", "-", "(", ")", entre outros, de forma a deixar o texto mais limpo para a futura classificação. Este trabalho de pré-processamento da base de dados foi realizado através de expressões regulares.

A partir do pré-processamento da base de dados foram trabalhadas duas abordagens para a classificação: uma onde foi realizado um tratamento de relevância das palavras, denominado *stop words*, e outra abordagem onde aplicou-se diretamente o modelo para a classificação. Ferreira e Ferraz (2018) afirmam que classificadores como o MultinomialNB, SVM e outros tendem a apresentar um desempenho aprimorado quando o processo de *stop words* é aplicado. Isso ocorre porque as *stop words* auxiliam o algoritmo a distinguir e classificar melhor as palavras nos textos.

A partir da biblioteca NLTK (2023) foi selecionado um conjunto de *stop words*. Esta lista de palavras-chave será utilizada, com exceção da palavra “*not*”, que foi removida por ser o principal indicador de característica negativa do texto. A sua não remoção poderia afetar negativamente o desempenho do modelo na classificação de frases negativas.

A partir da definição das *stop words*, tem-se as cópias da base de dados nas duas abordagens estudadas: com e sem a aplicação deste algoritmo. Como próximo passo proposto no modelo realizou-se a vetorização dos textos de ambas as bases de dados. Para isso foi utilizado o modelo TD-IDF, como citado anteriormente por Moser (2021), implementada pelo algoritmo *TfidfVectorizer* na biblioteca *sklearn.feature_extraction.text* (Scikit-Learn, 2023). O método *TfidfVectorizer* foi utilizado com seus parâmetros padrão de acordo com a documentação.

Figura 1 – Modelo Proposto para Classificação de Textos



Fonte: Elaborada pelos autores.

Com a aplicação do processo de vetorização dos textos, tem-se as bases preparadas para a classificação. Para expor os dados aos classificadores e aferir o seu desempenho, separou-se

os dados em bases de teste e treinamento. Para isso, utilizou-se o método *Hold-Out* Kumar (2023).

O método *Hold-ou*t foi aplicado através da função *train_test_split* da biblioteca *sklearn.model_selection*, Scikit-Learn (2023), com o parâmetro *test_size* definido em 0.3, o que implica uma divisão de 30% dos textos para teste e 70% para treinamento dos classificadores utilizados no modelo. Utilizou-se também variação no parâmetro *random_state*, de forma a gerar diferentes combinações de dados de teste e treinamento, permitindo a realização de diversos testes a partir da base de dados estudada.

Para a classificação dos dados utilizou-se os classificadores SVM com o algoritmo SVC e MultinomialNB com o algoritmo MultinomialNB, ambos implementados na biblioteca *sklearn* Scikit-Learn (2023). Para o classificador MultinomialNB utilizou-se os parâmetros padrão como proposto em sua documentação. No caso do classificador SVM Scikit-Learn (2023), realizou-se ajustes nos parâmetros, onde o regularizador (C=10), que é considerado um valor adequado, ajudando o modelo a se ajustar melhor à fronteira de decisão e evitar classificações errôneas; o parâmetro Gamma está definido como *scale*, o que implica em adaptar seu valor a cada decisão tomada, escalando o nível de generalização; e para o kernel utilizou-se o RBF (*Radial Basis Function*). Almeida (2017) afirma que o Kernel RBF funciona bem com o TF-IDF, lidando com a complexidade inerente aos dados de maneira eficaz.

Foram realizados testes com ambos os classificadores para ambas as abordagens propostas no pré-processamento das bases de dados, apurando suas acurácias individuais. Visando melhorar o desempenho final do modelo, aplicou-se a função *VotingClassifier* da biblioteca *sklearn* Scikit-Learn (2023). Para a aplicação *VotingClassifier* definiu-se os classificadores SVM (com kernel RBF) e MultinomialNB como *estimators* ou “estimadores”. Definiu-se também o parâmetro *voting* como *hard*, onde buscou-se que a predição final do modelo será determinada pela maioria dos votos dos modelos individuais, aumentando a confiabilidade.

Para avaliar os resultados, tanto dos classificadores individuais quanto da classificação final por votação, utilizou-se como medida de desempenho a acurácia. Para isso utilizou-se a função *accuracy_score* da biblioteca *sklearn.metrics* (Scikit-Learn, 2023).

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Como resultado da aplicação dos modelos propostos na metodologia, obteve-se a acurácia dos algoritmos medidos pela média dos resultados obtidos. No Quadro 1 tem-se listadas as *stop words* consideradas nesta análise. Seu conhecimento é importante para que se entenda como as frases serão afetadas.

Quadro 1 – Exemplos de *Stop Words*

Its
Their
Yours
Ourselves
Himself
you're
Ain
Couldn
Own
Aren

Fonte: Elaborado pelos autores.

Assim pode-se verificar como os classificadores interagem com essa mudança na Tabela 1; visualiza-se o desempenho de ambos os classificadores, onde tende a ser melhor quando não se aplica ao modelo de estudo o processo de *stop words*.

Esse aumento de desempenho se deve ao fato de que ambos os modelos requerem uma grande quantidade de dados para realizar uma classificação precisa, como mencionado por Ferreira e Ferraz (2018). Quando as *stop words* não são utilizadas, o peso das palavras se distribui de forma mais equilibrada entre as palavras importantes do texto. Isso faz com que palavras de maior importância sejam classificadas com valores mais baixos devido à sua frequência elevada. Portanto, nesse caso, é necessário manter as *stop words* para garantir uma melhor qualidade na análise dos resultados.

Tabela 1 – Comparação de desempenho dos classificadores com e sem *Stop Words*

Classificadores	**Com *stop words***	**Sem *stop words***
MultinomialNB	76,2%	77,5%
SVM	75,4%	76,8%

Fonte: Elaborada pelos autores.

Além disso, é possível observar que o classificador MultinomialNB apresentou um desempenho superior. Isso ocorre por conta de o MultinomialNB ser conhecido por sua alta eficácia na análise de dados textuais, especialmente quando se trata de contagem de palavras. Por outro lado, o SVM (*Support Vector Machine*) é um algoritmo de margem máxima que procura encontrar um hiperplano de separação ótimo, mas pode ser mais sensível à escala e à distribuição dos dados. Portanto, a escolha do classificador MultinomialNB parece ser mais apropriada para lidar com esse tipo de análise textual, devido às suas características e desempenho superior observado.

Tabela 2 – Resultado da votação do método *Ensemble*

Método	**Com *stop words***	**Sem *stop words***
Ensemble	76%	76,9%

Fonte: Elaborada pelos autores.

Aplicando-se o *Voting Classifier*, o método de *ensemble* tem como objetivo combinar as análises de ambos os classificadores para determinar se uma frase é negativa ou positiva, seguindo o padrão dos classificadores que o compõem. Isso ajuda a explicar por que a remoção das *stop words* teve um impacto negativo no *ensemble*. No entanto, é evidente que a votação não conseguiu superar o desempenho do classificador MultinomialNB. Isso pode ser atribuído, em grande parte, ao fato de que o modelo SVM não se adaptou bem durante o treinamento, o que pode ter levado a generalizações excessivas.

O aumento de erros também deve ser atribuído ao fato de que foram utilizados apenas dois classificadores, como explicado por Rajawat (2023). No método de *voting ensemble*, quando vários classificadores de naturezas diferentes são treinados de maneira semelhante com a mesma fonte de dados, é comum que apresentem faixas de erro distintas. Isso resulta na seguinte lógica: quanto mais classificadores forem utilizados, maior a qualidade da análise, uma vez que, caso um classificador cometa um erro, outros podem corrigi-lo, resultando em uma decisão mais precisa.

No caso deste trabalho, a acurácia diminuiu devido à utilização de apenas dois classificadores. Quando um deles classifica uma frase como negativa e o outro como positiva,

não há um terceiro classificador para desempatar a decisão. Isso significa que, em uma situação desse tipo, existe uma probabilidade de 50% de erro. Portanto, a falta de concordância entre os classificadores do *ensemble*, influenciada pelo desempenho desfavorável do modelo SVM, parece ser a principal razão pela qual o classificador MultinomialNB continua se destacando como a melhor opção.

5 CONSIDERAÇÕES FINAIS

O estudo teve como objetivo desenvolver um modelo de classificação de textos em positivos e negativos, utilizando processamento de linguagem natural (PLN) com foco na língua inglesa. O método escolhido incluiu o pré-processamento de dados, a utilização de *stop words*, a vetorização de textos com TF-IDF e a aplicação de classificadores, como MultinomialNB e SVM. O pré-processamento dos dados envolveu a substituição de abreviações, a remoção de caracteres especiais e a utilização de expressões regulares. Além disso, a aplicação de *stop words* foi considerada, com a exceção da palavra "*not*" para a classificação de frases negativas.

A análise demonstrou que a utilização de *stop words* teve um impacto negativo no desempenho dos classificadores, uma vez que as *stop words* ajudam a equilibrar o peso das palavras no texto. Portanto, a manutenção das *stop words* é recomendada para melhorar a qualidade da análise. Foi observado que o classificador MultinomialNB teve um desempenho superior em comparação com o SVM. Isso se deve à eficácia do MultinomialNB na análise de dados textuais, especialmente na contagem de palavras. Por outro lado, o SVM pode ser mais sensível à escala e à distribuição dos dados.

A utilização do método de *ensemble*, que combinou os resultados dos classificadores MultinomialNB e SVM, não conseguiu superar o desempenho do MultinomialNB. Isso pode ser atribuído, em parte, ao fato de que o SVM não se adaptou bem durante o treinamento. Além disso, a falta de concordância entre os classificadores do *ensemble* contribuiu para uma redução na acurácia. É importante reconhecer as limitações do estudo, como a utilização de apenas dois classificadores no *ensemble* e a necessidade de explorar métodos de *ensemble* mais complexos. Além disso, pesquisas futuras podem considerar a inclusão de mais classificadores e o desempenho em diferentes idiomas.

Em resumo, o trabalho demonstrou a aplicação eficaz de técnicas de PLN na classificação de textos em positivos e negativos, com ênfase na importância do pré-processamento de dados e na escolha adequada de classificadores. A análise comparativa entre os modelos com e sem *stop words* evidenciou a importância da manutenção dessas palavras-chave. A escolha do classificador MultinomialNB como a melhor opção ressalta a relevância de selecionar algoritmos apropriados para tarefas específicas de PLN. No entanto, é necessário considerar aprimoramentos no método de *ensemble* e investigar mais a fundo as capacidades do modelo em diferentes contextos e línguas. Este estudo contribui para o campo de PLN e oferece insights valiosos para pesquisas futuras nesta área.

REFERÊNCIAS

AWAN-U-R. **Understanding soft voting and hard voting**: a comparative analysis of *ensemble* learning methods. 2023. Disponível em: https://medium.com/@awanurrahman.cse/understanding-soft-voting-and-hard-voting-a-comparative-analysis-of-*ensemble*-learning-methods-db0663d2c008. Acesso em: 31 out. 2023.

ALMEIDA, A. R. O **uso de análise de componentes independentes na extração de características dos sinais transitórios de faltas em linhas de transmissão de energia elétrica**. 2017. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza, 2017.

AWS. **O que é processamento de linguagem natural (pln)?** Disponível em: https://aws.amazon.com/pt/what-is/nlp/. Acesso em: 18 out. 2023a.

AWS. **O que é uma rede neural?** Disponível em: https://aws.amazon.com/pt/what-is/neural-network/#:~:text=Uma%20rede%20neural%20%C3%A9%20um,camadas%2C%20semelhante%20ao%20c%C3%A9rebro%20humano.2023. Acesso em: 8 maio 2023b.

CASELI, H. M.; NUNES, M. G. V. (org.). **Processamento de linguagem natural**: conceitos, técnicas e aplicações em português. São Carlos: Brasileiras em Pln, 2023. Disponível em: https://brasileiraspln.com/livro-pln/1a-edicao/. Acesso em: 7 out. 2023.

FERREIRA, G. A. B. A.; FERRAZ, T. F. **Processamento de linguagem natural para consultas em bancos de dados bibliográficos**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade de São Paulo, São Paulo, 2018. Disponível em: https://repositorio.usp.br/directbitstream/8fce5397-eef2-46d8-9447-fc1a5e96373e/Gabriel%20FERREIRA%20-%20Thiago%20%20FERRAZ%20monografia.pdf. Acesso em: 2 nov. 2023.

RAJAWAT, A. S. **Voting classifiers in machine learning**. 2023. Disponível em: https://medium.com/@imamitsingh/voting-classifiers-in-machine-learning-a532935fe592. Acesso em: 2 nov. 2023.

IBERDROLA. **O que é o processamento de linguagem natural e quais são suas aplicações?** Disponível em: https://www.iberdrola.com/inovacao/pnl-processamento-linguagem-natural. Acesso em: 18 out. 2023.

KUMAR, A. **Hold-out method for training machine learning models**. Disponível em: https://vitalflux.com/hold-out-method-for-training-machine-learning-model/. Acesso em: 6 nov. 2023.

LUCHESE, R. L. **Aplicação de técnicas de machine learning para classificação abc multicritério**. 2021. Tese (Doutorado em Engenharia de Software e Inovação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.

MCCARTHY, J. **What is artificial intelligence?** 2007. Disponível em: https://www-formal.stanford.edu/jmc/whatisai.pdf. Acesso em: 22 out. 2023.

MOSER, G. V. B. **Análise de similaridade entre tf-idf e modelos contextualizados de linguagem baseados em tokens**. 2021. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2021. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/243553/monografia.pdf?sequence=1&isAllowed=y. Acesso em: 10 out. 2023.

NLTK. **Natural language toolkit**. Disponível em: https://www.nltk.org/. Acesso em: 6 nov. 2023.

NUMPY. **Numpy documentation**. Disponível em: https://numpy.org/doc/stable/. Acesso em: 6 nov. 2023.

ORACLE. **O que é processamento de linguagem natural?** Disponível em: https://www.oracle.com/br/artificial-intelligence/what-is-natural-language-processing/. Acesso em: 18 out. 2023.

PANDAS. **User guide**. Disponível em: https://pandas.pydata.org/docs/user_guide/index.html. Acesso em: 6 nov. 2023.

DICIO: Dicionário Online de Português. **Acurácia**. 2023. Disponível em: https://www.dicio.com.br/acuracia/. Acesso em: 31 out. 2023.

PYTHON. **Python 3.12.0 documentation**. Disponível em: https://docs.python.org/3/. Acesso em: 6 nov. 2023.

RUSSELL, S.; NORVING, P. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2013.

SCIKIT-LEARN. **Documentation of scikit-learn 0.21.3**. Disponível em: https://scikit-learn.org/0.21/documentation.html. Acesso em: 6 nov. 2023.

SILVA, C. F.; VIEIRA, R.; OSORIO, F. S. Evaluating the use of linguistic information in the preprocessing phase of text mining. **Revista Iberoamericana de Inteligência Artificial**, v. 9, n. 26, p. 59-66, 2005.

SIMIC, M. **Hard vs. Soft Voting**. 2023. Disponível em: https://www.baeldung.com/cs/hard-vs-soft-voting-classifiers. Acesso em: 31 out. 2023.

SRIRAM. **Multinomial Naive Bayes explicado**: função, vantagens e desvantagens, aplicações em 2023. 2022. Disponível em: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/. Acesso em: 28 set. 2023.

TAKAHASHI, A. **Máquina de vetores-suporte intervalar**. 2012. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2012. Disponível em: https://repositorio.ufrn.br/bitstream/123456789/15225/1/AdrianaT_TESE.pdf. Acesso em: 28 set. 2023.

TEAM. **Plot types**. Disponível em: https://matplotlib.org/stable/plot_types/index.html. Acesso em: 6 nov. 2023.

ROTTEN TOMATOES. **Movies**. Disponível em: https://www.rottentomatoes.com/. Acesso em: 01 nov. 2023.

TURING, A. Computing machinery and intelligence. **Mind**, v. 59, p. 433-460, 1950.

ZHOU, Z. **Ensemble methods**: foundations and algorithms. Londres: A Chapman & Hall Book, 2012.