UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO
CAMPUS SALGUEIRO - PE
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Emanuel Flávio dos Santos Silva

Comparação de Ensembles de Classificadores para a Tarefa de Análise de Sentimentos

Salgueiro - PE
2025

Emanuel Flávio dos Santos Silva

Comparação de Ensembles de Classificadores para a Tarefa de Análise de Sentimentos

Trabalho de Conclusão de Curso de Bacharelado em Ciência da Computação apresentado ao Colegiado de Ciência da Computação como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. Me. Débora da Conceição Araújo

Salgueiro - PE
2025

UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO - UNIVASF
Gabinete da Reitoria
Sistema Integrado de Bibliotecas (SIBI)
Av. José de Sá Maniçoba, s/n, Campus Universitário – Centro CEP 56304-917
Caixa Postal 252, Petrolina-PE, Fone: (87) 2101- 6760, biblioteca@univasf.edu.br

Sobrenome do autor, Prenome do autor
* Cutter Título do trabalho / Nome por extenso do autor. - local, ano.
xx (total de folhas antes da introdução em nº romano), 50 f.(total de folhas do trabalho): il. ; (caso tenha ilustrações) 29 cm.(tamanho do papel A4)

Trabalho de Conclusão de Curso (Graduação em nome do curso) - Universidade Federal do Vale do São Francisco, Campus, local, ano

Orientador (a): Prof.(a) titulação e nome do prof(a).

Notas (opcional)

1. Assunto. 2. Assunto. 3. Assunto. I. Título. II. Orientador (Sobrenome, Prenome). III. Universidade Federal do Vale do São Francisco.

* CDD

Ficha catalográfica elaborada pelo Sistema Integrado de Biblioteca SIBI/UNIVASF
Bibliotecário: Nome* e CRB*

* ***Dados inseridos pela biblioteca***

Exemplo:

Souza, José Augusto de
S729c Crianças com dificuldades de aprendizado: estudo nas escolas públicas da cidade de Juazeiro-BA / José Augusto de Souza. – Petrolina - PE, 2009.
xv, 140 f. : il. ; 29 cm.

Trabalho de Conclusão de Curso (Graduação em Psicologia) Universidade Federal do Vale do São Francisco, Campus Petrolina-PE, 2009.

Orientadora: Profª. Drª. Maria de Azevedo.

Inclui referências.

1. Crianças - Ensino. 2. Distúrbios da aprendizagem. 3. Escolas públicas – Juazeiro (BA). I. Título. II. Azevedo, Maria de. III. Universidade Federal do Vale do São Francisco.

370.15

Ficha catalográfica elaborada pelo Sistema Integrado de Biblioteca SIBI/UNIVASF
Bibliotecário: Nome e CRB.

Emanuel Flávio dos Santos Silva

Comparação de Ensembles de Classificadores para a Tarefa de Análise de Sentimentos

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pela banca examinadora.

Salgueiro - PE, 18 de dezembro de 2023.

Prof. Maria Bernadete, Me.
Coordenador do Curso

Banca Examinadora:

Prof. Me. Débora da Conceição Araújo
Presidente da Banca

Prof. X Y Z, Me.
Avaliador
Universidade Federal do Vale do São Francisco

Prof. X Y Z, Dr.
Avaliador
Universidade Federal do Vale do São Francisco

AGRADECIMENTOS

Agradeço primeiramente a Deus e a Nossa Senhora, por sempre estarem ao meu lado, iluminarem meu caminho e fortalecerem minha fé. Cada passo que dei foi abençoado, e por isso consegui chegar até aqui.

A meus pais e à minha família, pelo apoio incondicional, por serem minha base e meu conforto, e por me darem o abrigo de seu abraço mesmo quando errei e tive dias difíceis. Deixo a eles minha mais profunda gratidão.

À Alana, minha namorada, sou grato pelo amor, pelo cuidado, pelas palavras de incentivo e por estar ao meu lado em cada etapa. Seu apoio fez toda a diferença e tornou esta caminhada muito mais leve e significativa.

Aos meus amigos, obrigado por estarem comigo, me encorajando e dando força sempre que precisei.

À minha orientadora, Prof. Me. Débora da Conceição Araújo, meus agradecimentos pela dedicação, profissionalismo, paciência e pela orientação prestada ao desenvolvimento deste trabalho.

A todos que de alguma forma colaboraram para a finalização deste trabalho, deixo aqui meu reconhecimento e gratidão.

RESUMO

O presente estudo teve como objetivo principal comparar diferentes estratégias de construção de *ensembles* heterogêneos aplicados à análise de sentimento em comentários de filmes. Para isso, foram utilizadas duas bases de dados distintas: uma coletada do *Letterboxd*, construída por meio de um *crawler*, contendo mais de 100 mil avaliações rotuladas com base na avaliação dos comentários, onde notas de 0 a 2,5 eram negativas e de 3 a 5 positivas; e a base clássica do *IMDb*. Cada conjunto de dados foi avaliado em dois cenários: com e sem pré-processamento textual. Dez modelos de classificação foram treinados e tiveram seus hiperparâmetros otimizados com o *framework Optuna*, utilizando validação cruzada, formando um *pool* inicial de classificadores. A partir desse conjunto, três estratégias de combinação foram analisadas: Sem Seleção (SS), Seleção por Acurácia (SA) e Seleção por Acurácia e Diversidade (SAD). Os resultados demonstraram diferenças consistentes entre as estratégias, evidenciando que a diversidade entre os modelos é um fator determinante para melhorar o desempenho dos *ensembles*. Um dos métodos de aprendizado de máquina *ensemble*, o Stacking, em particular, mostrou-se mais eficaz na exploração dessa diversidade. Conclui-se que *ensembles* heterogêneos construídos com critérios robustos de seleção (SA e SAD) apresentam desempenho superior e maior estabilidade na tarefa de análise de sentimento. A escolha ideal, no entanto, depende do equilíbrio desejado entre performance e custo computacional.

Palavras-chave: Análise de Sentimentos; Ensemble de Classificadores; Otimização de Hiperparâmetros

ABSTRACT

The main objective of the present study was to compare different heterogeneous *ensemble* construction strategies applied to sentiment analysis in movie reviews. For this purpose, two distinct databases were used: one collected from *Letterboxd*, built using a *crawler*, containing over 100,000 reviews labeled based on the review score, where ratings from 0 to 2.5 were considered negative and 3 to 5 were positive; and the classic *IMDb* database. Each dataset was evaluated in two scenarios: with and without textual preprocessing. Ten classification models were trained and had their hyperparameters optimized with the *Optuna framework*, using cross-validation, forming an initial classifier *pool*. From this set, three combination strategies were analyzed: No Selection (NS), Selection by Accuracy (SA), and Selection by Accuracy and Diversity (SAD). The results demonstrated consistent differences between the strategies, evidencing that the diversity among models is a determining factor for improving *ensemble* performance. One of the *ensemble* machine learning methods, Stacking, in particular, proved more effective in exploring this diversity. It is concluded that heterogeneous *ensembles* constructed with robust selection criteria (SA and SAD) show superior performance and greater stability in the sentiment analysis task. The ideal choice, however, depends on the desired balance between performance and computational cost.

Keywords: Sentiment Analysis; Classifier Ensemble; Hyperparameter Optimization

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nos dias atuais, com o avanço acelerado da internet e da comunicação digital, tornou-se fácil para qualquer pessoa expor suas opiniões online. Esse fenômeno tem tido um impacto significativo nas comunidades virtuais e até mesmo fora delas, influenciando comportamentos, decisões de consumo e debates sociais (Almeida Neto; De Melo, 2023).

Grande parte dessa facilidade e alcance deve-se à popularização das redes sociais, que transformaram a forma como os indivíduos se expressam e compartilham experiências. Essas plataformas não apenas conectam pessoas, mas também se tornaram fontes abundantes de dados textuais gerados pelos próprios usuários, os quais podem ser utilizados em diversas análises (Paes et al., 2022).

Um dos principais usos desses dados textuais é a análise de sentimentos, uma técnica voltada à identificação, extração e classificação de emoções expressas em textos, como opiniões sobre produtos, serviços, marcas ou tópicos variados. Essa tarefa tem se mostrado extremamente útil para empresas interessadas em compreender a percepção dos consumidores, para organizações que desejam medir o impacto de campanhas públicas e para pesquisadores que analisam fenômenos sociais em larga escala (Mohammed; Kora, 2023).

A análise de sentimentos utiliza amplamente técnicas de Processamento de Linguagem Natural (PLN), um subcampo da inteligência artificial que busca permitir que computadores compreendam, interpretem e gerem a linguagem humana de forma significativa (Dang; Moreno-García; De La Prieta, 2020). O PLN abrange uma ampla gama de tarefas computacionais, como tradução automática, sumarização de textos, reconhecimento de entidades nomeadas, classificação de textos e, especialmente, análise de sentimentos. Por meio de modelos estatísticos, redes neurais e algoritmos de aprendizado de máquina, o PLN procura extrair padrões linguísticos que possibilitem o processamento e a interpretação automatizada e inteligente de informações em linguagem natural (Paes et al., 2022).

Outro conceito essencial nesse contexto é o de ensemble learning, uma abordagem de aprendizado de máquina que consiste na combinação de diversos modelos fracos ou de base para formar um modelo mais robusto, preciso e generalizável (Dong et al., 2020). O ensemble é particularmente eficiente na redução de erros de viés e variância e, em muitos casos, supera o desempenho de modelos isolados. Essa técnica pode ser implementada de diversas formas, como bagging, boosting e stacking (Kazmaier; Van Vuuren, 2022). Além disso, os ensembles podem ser divididos em dois grandes grupos: homogêneos, que utilizam modelos da mesma natureza, e heterogêneos, que combinam modelos de diferentes tipos, beneficiando-se da diversidade das decisões (Avelino Júnior, 2022).

A plataforma Letterboxd surge como uma rica fonte de dados para esse tipo de análise. Voltada para entusiastas do cinema, essa rede social permite que seus usuários

registrem os filmes assistidos, atribuam notas, escrevam resenhas, criem listas temáticas e interajam com outras pessoas que compartilham os mesmos interesses (Andrade; Rebs, 2022). Assim, o Letterboxd funciona não apenas como uma comunidade engajada, mas também como um repositório valioso de opiniões e sentimentos sobre filmes. Esse ambiente propício à expressão de sentimentos torna a plataforma uma base ideal para estudos de análise de sentimentos, possibilitando a classificação automatizada das resenhas como positivas, negativas ou neutras (Britto; Pacífico, 2019), oferecendo insights relevantes sobre a recepção crítica de filmes entre o público.

Diante desse cenário, o presente trabalho tem como objetivo analisar a eficácia de diferentes estratégias para a construção de *ensembles* na *análise de sentimentos* aplicada a comentários de filmes, utilizando algoritmos clássicos de classificação supervisionada e técnicas de pré-processamento. Para isso, foi construída uma base de dados a partir do *Letterboxd*, contendo inicialmente cerca de 150 mil comentários acompanhados de suas respectivas classificações em estrelas, variando de 0,5 a 5. A base inclui resenhas em diversos idiomas, com predominância do inglês, o que impõe desafios linguísticos e computacionais ao processo de análise. Este estudo busca comparar as técnicas de *ensemble* aplicadas à tarefa de análise de sentimento e contribuir tanto para o avanço das aplicações em *Processamento de Linguagem Natural* (*PNL*) em contextos reais, quanto para as técnicas de construção e seleção de *ensembles*.

1.1 OBJETIVOS

Os objetivos deste trabalho são subdivididos em objetivos gerais e objetivos específicos. Estes são:

1.1.1 OBJETIVO GERAL

Analisar e comparar o desempenho de técnicas de *ensemble* de classificadores heterogêneos aplicadas à tarefa de análise de sentimentos.

1.1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são:

- Revisar a literatura sobre análise de sentimentos e métodos de ensemble, destacando os conceitos e características dos métodos heterogêneos.
- Selecionar e preparar os conjuntos de dados utilizados no experimento, incluindo a construção de uma nova base a partir de comentários de filmes extraídos do Letterboxd.
- Implementar e treinar modelos de ensemble heterogêneos, com base no desempenho e diversidade dos algoritmos clássicos.

- Avaliar desempenho dos modelos, a partir de métricas de classificação como acurácia, precisão, revocação e F1-score, nos diferentes conjuntos de dados.
- Comparar os resultados obtidos entre os diferentes modelos construídos, discutindo suas vantagens, limitações e aplicabilidade à análise de sentimentos.

1.2 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2, é apresentada a fundamentação com o que é necessário para o entendimento da pesquisa. São introduzidos os conceitos relacionados à Análise de Sentimentos, Pré-processamento de Texto, Representação vetorial com TF-IDF e aos principais algoritmos de aprendizado de máquina utilizados no estudo. Além disso, o capítulo discute a técnica de otimização de hiperparâmetros o framework *Optuna*, bem como métodos de combinação de modelos (*ensembles*), incluindo abordagens tradicionais e estratégias de seleção de modelos base. Ao final, são apresentadas as principais pesquisas relacionadas ao tema.

No Capítulo 3, é descrito o delineamento experimental. Detalha-se o processo de coleta, preparação e pré-processamento dos dados, seguido pela vetorização textual, definição dos conjuntos de treino, validação e teste, além dos procedimentos utilizados para otimização dos hiperparâmetros e construção do *pool* de classificadores. Também são apresentados os critérios de seleção dos modelos para composição dos *ensembles* heterogêneos.

No Capítulo 4, são apresentados os resultados obtidos nos experimentos. Inicialmente, são exibidos os desempenhos individuais dos modelos após a otimização dos hiperparâmetros, e em seguida, são comparados os desempenhos das diferentes configurações de *ensembles*. Este capítulo inclui análises comparativas e tabelas que discutem a influência da diversidade entre modelos, bem como o do desempenho nas diferentes bases utilizadas (Letterboxd e IMDb, com e sem pré-processamento).

Por fim, no Capítulo 5, são sintetizadas as principais contribuições e achados do trabalho. São discutidas as conclusões obtidas a partir do experimento, as limitações encontradas ao longo do processo e sugestões para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Existem diversas pesquisas relevantes nas áreas de Análise de Sentimentos, Modelos de Aprendizado de Máquina, *Ensemble Learning* e Processamento de Linguagem Natural, que contribuem significativamente para o avanço desses campos.

No trabalho de Kazmaier e Van Vuuren (2022), investiga-se como a aprendizagem por conjunto (*ensemble learning*) pode contribuir para a análise de sentimentos. Os autores destacam que, embora haja um crescente interesse por técnicas de ensemble learning na comunidade de aprendizado de máquina, seu uso específico na classificação de sentimentos ainda é limitado. Observa-se, ainda, que grande parte das pesquisas concentra-se em ensembles homogêneos, embora os ensembles heterogêneos possam ser bastante eficazes ao combinar diferentes modelos. O artigo propõe uma nova abordagem para seleção de modelos que compõem o ensemble, evitando o armazenamento das previsões individuais e o retreinamento custoso de todos os modelos candidatos. Utilizando quatro conjuntos de dados de análise de sentimentos, os autores observaram uma melhoria de desempenho de até 5,53% em relação ao melhor modelo individual.

Em Sirqueira e Vidal (2024), é proposta uma avaliação de modelos de ensemble utilizando modelos Transformers para a tarefa de Reconhecimento de Entidades Nomeadas (Named Entity Recognition – NER) em textos públicos brasileiros. O estudo destaca os avanços significativos no Processamento de Linguagem Natural, impulsionados principalmente pelo desenvolvimento de modelos de aprendizado profundo baseados em Transformers. A análise de dados abertos no contexto brasileiro, como documentos publicados no Diário Oficial da União, é considerada crucial para a transparência e o acesso à informação. Os autores testaram um conjunto de modelos baseados em variações do BERT, combinando diferentes estratégias de ensemble, e alcançaram melhorias de até 11% no corpus proposto, em comparação com abordagens clássicas de NER baseadas apenas no BERT.

No trabalho de Santos e Berton (2023), é investigada a classificação de sentimentos em uma amostra de textos publicados no Twitter, em português, sobre as eleições presidenciais brasileiras de 2022. Os autores destacam que o crescimento da internet e das redes sociais facilitou o acesso a informações sobre a opinião pública, mas que a análise manual de grandes volumes de comentários se torna inviável, exigindo o uso de tecnologias. Para isso, foi utilizado o processo de Descoberta de Conhecimento em Banco de Dados, aliado a técnicas de aprendizado de máquina, para analisar e classificar os tweets em opiniões positivas, neutras e negativas. Foram empregadas duas representações textuais clássicas (Bag of Words e TF-IDF) e seis classificadores: Naive Bayes, Árvore de Decisão, Random Forest, K-Nearest Neighbors, MLP e SVM. Os resultados, com base em um conjunto de dados balanceado, indicaram que Jair Bolsonaro apresentou a maior proporção de sentimentos positivos, Luiz Inácio Lula da Silva a maior de sentimentos neutros, e Ciro

Gomes a maior de sentimentos negativos.

O presente trabalho, em relação aos estudos de Kazmaier e Van Vuuren (2022), Sirqueira e Vidal (2024) e Santos e Berton (2023), representa um avanço significativo ao propor uma avaliação abrangente de ensembles heterogêneos aplicados à análise de sentimento. Diferentemente das abordagens anteriores, que se concentraram em apenas um método de seleção, modelos Transformers de alto custo computacional, ou apenas classificadores isolados, este estudo inova ao investigar três estratégias de seleção, utilizando critérios de acurácia e diversidade por meio do Q-statistic.Além disso, foram empregados dez modelos base de menor custo computacional otimizados com o framework Optuna, comparando cenários com e sem pré-processamento. A pesquisa aplica duas técnicas de ensemble e avalia duas bases de dados de comentários de filmes, incluindo uma base inédita do Letterboxd construída especificamente para este estudo.Esses elementos tornam o trabalho mais abrangente, replicável e original em relação ao estado da arte.

2.1 MODELOS DE MACHINE LEARNING E ENSEMBLE LEARNING

Machine Learning é um ramo da Inteligência Artificial cujo principal objetivo é desenvolver métodos que permitam que os computadores aprendam a realizar tarefas a partir de dados, sem que tenham sido programados explicitamente para cada situação específica. Esse aprendizado ocorre por meio da identificação de padrões e regularidades em grandes volumes de dados (Alpaydin, 2021).

2.1.1 NAIVE BAYES(NB)

O algoritmo Naive Bayes (NB) é um classificador estatístico tradicional amplamente utilizado devido à sua estrutura simples, alta eficiência computacional e bom desempenho, mesmo com conjuntos de dados reduzidos (Chen et al., 2021). É empregado em diversas aplicações reais, como sistemas de recomendação de produtos e diagnóstico médico, sendo considerado um dos algoritmos de melhor desempenho em tarefas de mineração de dados (Wickramasinghe; Kalutarage, 2021).

O NB é descrito como um método de aprendizado de máquina que utiliza conceitos de probabilidade e estatística para realizar classificações, sendo eficaz na previsão de eventos futuros com base em experiências passadas (Dwiramadhan; Wahyuddin; Hidayatullah, 2022). Apesar de sua simplicidade, o NB é eficaz e robusto. No entanto, assume a independência entre as características dos dados, o que pode não refletir a realidade em certas aplicações. Ainda assim, o algoritmo pode apresentar bom desempenho mesmo quando há dependência entre atributos (Wickramasinghe; Kalutarage, 2021).

2.1.2 RANDOM FOREST(RF)

O classificador Random Forest (RF) é uma técnica de aprendizado de máquina supervisionado amplamente utilizada para tarefas de classificação e regressão. Sua operação baseia-se no princípio do ensemble learning, combinando diversas árvores de decisão, cada uma treinada com subconjuntos diferentes dos dados, a fim de melhorar a acurácia das previsões e reduzir o risco de sobreajuste (overfitting) associado a árvores individuais (Sun et al., 2024). A previsão final do modelo é obtida por meio da votação da maioria (em problemas de classificação) ou pela média (em problemas de regressão) das saídas das árvores componentes (Gupta et al., 2022).

O desempenho do Random Forest tende a melhorar com o aumento do número de árvores na floresta (Amiri et al., 2024). Entre suas principais vantagens, destacam-se a robustez contra o sobreajuste, a capacidade de lidar com grandes volumes de dados e variáveis, bem como a boa performance preditiva, mesmo em contextos com dados ruidosos ou incompletos (Amini et al., 2022; Purwanto et al., 2022).

2.1.3 SUPPORT VECTOR MACHINE(SVM)

O algoritmo Support Vector Machine (SVM) é amplamente reconhecido por sua eficácia em tarefas de classificação, sendo aplicado com sucesso em áreas como reconhecimento facial, diagnóstico de doenças, reconhecimento de texto e, especialmente, análise de sentimentos (Abdullah; Abdulazeez, 2021). Sua popularidade decorre da capacidade de lidar bem com conjuntos de dados pequenos, problemas não lineares e de alta dimensionalidade.

Uma das principais vantagens do SVM é sua busca por uma solução ótima global, o que contribui para altos níveis de acurácia em tarefas preditivas. No entanto, seu desempenho depende fortemente da escolha adequada da função kernel, responsável por projetar os dados em espaços de maior dimensionalidade, onde a separação entre as classes torna-se viável. Apesar de competitivo, especialmente em bases com poucas amostras, o SVM pode enfrentar limitações quando aplicado a grandes volumes de dados, devido ao aumento da complexidade computacional (Abdullah; Abdulazeez, 2021).

2.1.4 REGRESSÃO LOGÍSTICA(RL)

A Regressão Logística é amplamente utilizada como um método estatístico simples e eficaz para resolver problemas de classificação binária, como o acontecimento ou não de um evento (Zhou, 2021). Ela permite estimar como variáveis independentes influenciam a probabilidade de um desfecho, podendo ser aplicada em áreas médicas, industriais e em modelos preditivos. A Regressão Logística transforma o desfecho binário em uma variável contínua que é chamada de *Logit* (logaritmo de chances), possibilitando modelar a relação entre preditores e o resultado. A partir dos coeficientes do modelo, é possível calcular

probabilidades e interpretar o efeito das variáveis por meio dos *Odds Ratio* (OR), que indica como a chance de um evento muda conforme cada preditor varia. Odds Ratios maiores que 1 aumentam a chance do evento; menores que 1, reduzem. Algumas condições precisam ser atendidas para que o modelo seja válido: independência entre as observações, relação linear entre variáveis contínuas e o Logit, ausência de colinearidade entre preditores e inexistência de outliers influentes.(Zabor et al., 2022)

2.1.5 K-NEAREST NEIGHBORS(KNN)

O *K-Nearest Neighbors* (KNN) é um algoritmo de aprendizado supervisionado usado principalmente em tarefas de classificação. A ideia desse modelo é que novos dados podem ser classificados por seus K vizinhos mais próximos do conjunto de treinamento. Ele calcula as distâncias entre os pontos de consulta e identifica os K mais próximos. Após isso, realiza uma votação majoritária, atribuindo ao novo ponto a classe mais frequente entre seus vizinhos.(Uddin; Al., 2022) Ele possui diversas aplicações que vão desde *IoT* (*Internet das Coisas*) até sistemas de recomendação, e tem um papel relevante em sistemas modernos relacionados à Indústria 4.0. Ele é um método não-paramétrico e de aprendizado baseado em instâncias, ou seja, não é construído um modelo explícito durante o treinamento; ele apenas realiza cálculos quando recebe uma nova entrada. Seu princípio fundamental é que instâncias semelhantes tendem a estar próximas no espaço, permitindo prever a classe de uma nova amostra com base na similaridade dos exemplos existentes.(Halder; Al., 2024)

2.1.6 DECISION TREE(DT)

O modelo Decision Tree (DT) é uma técnica bastante utilizada em *machine learning*, processamento de imagens e reconhecimento de padrões. Esse modelo tem uma estrutura hierárquica de testes simples, em que os atributos numéricos são comparados a valores de corte para gerar regras fáceis de interpretar, diferente de modelos mais complexos. Por sua simplicidade, os DT possuem diversas aplicações práticas. Existem alguns tipos de DT, como os mais antigos ID3, C4.5 e CART, que têm diferenças nos critérios de divisão, variáveis aceitas e estratégias de poda. Esse modelo funciona por meio de alguns conceitos, como a entropia e o ganho de informação. A entropia mede a impureza dos dados, sendo que, quanto mais próximo de zero, melhor, e o ganho de informação indica o quanto a divisão dos dados reduz essa impureza nesse caso, quanto mais alto, melhor. Os modelos DT apresentam vantagens como facilidade de interpretação, velocidade e capacidade de trabalhar tanto com dados categóricos como numéricos, porém há limitações, como sensibilidade ao aumento do número de amostras e suscetibilidade a decisões subótimas quando mal configurados.(Charbuty; Abdulazeez, 2021)

2.1.7 MULTI-LAYER PERCEPTRON(MLP)

O Multi-Layer Perceptron (MLP) é uma rede neural do tipo *feed-forward* que é muito utilizada em tarefas de classificação, regressão e predição em vários domínios, como detecção de intrusões e na saúde. Ele é composto de três tipos de camada: camada de entrada, camada oculta (que pode ser mais de uma) e camada de saída. Essas camadas são responsáveis por coletar as características, extrair padrões e gerar previsões. É um modelo que é capaz de aprender tanto funções lineares quanto não lineares.(Al Bataineh; Manacek, 2022)

O seu funcionamento é baseado principalmente por meio do algoritmo de retropropagação, que ajusta os pesos e vieses para que o erro possa ser minimizado entre as saídas. Para que funcione, as funções de ativação, como *sigmoid* e *reLU*, devem ser diferenciáveis, para que possa se introduzir a não linearidade e possibilitar o cálculo do gradiente. As escolhas do número de neurônios, camadas ocultas e funções de ativação são os hiperparâmetros do modelo. É um modelo em que pode ser difícil otimizar esses hiperparâmetros, já que podem haver cenários onde os dados apresentam alta variabilidade.(Al Bataineh; Manacek, 2022; Naskath; Sivakamasundari; Begum, 2023)

2.1.8 ADABOOST

O AdaBoost é um método de ensemble baseado na reponderação dos exemplos de treinamento, permitindo que o algoritmo apresente bom desempenho mesmo quando há poucos dados disponíveis. Inicialmente, todos os exemplos recebem pesos iguais. A cada iteração, um classificador fraco é treinado considerando a distribuição de pesos atual. Em seguida, calcula-se o erro ponderado desse classificador e determina-se sua importância no modelo final. Exemplos classificados incorretamente têm seus pesos aumentados, enquanto os corretamente classificados têm seus pesos reduzidos, fazendo com que o algoritmo concentre a atenção nos casos mais difíceis. Após várias iterações, o AdaBoost combina todos os classificadores fracos, ponderando cada um pelo seu desempenho, e produz um classificador final obtido pelo sinal da soma ponderada das predições.(Ding et al., 2022; Ramakrishna et al., 2023)

2.1.9 STOCHASTIC GRADIENT DESCENT(SGD)

O *Stochastic Gradient Descent* (SGD) é uma técnica bastante reconhecida por ser simples e eficiente, sendo muito indicada para treinar classificadores lineares e regressores, como *SVM* e Regressão Logística. Uma das principais vantagens é a facilidade de implementação e a eficiência, que o fazem ser apropriado para problemas de grande escala e para cenários com dados esparsos, o que acontece em classificação de textos e tarefas de *Processamento de Linguagem Natural*. (Pinho et al., 2024)

No caso do classificador, é implementado um modelo linear regularizado treinado por meio do SGD, em que o gradiente da função de perda é estimado a cada amostra individual, permitindo sempre atualizações contínuas. O SGD é um modelo que, do ponto de vista prático, é bastante eficiente e ajustável, oferecendo muitas opções para o processo de aprendizagem, como a taxa de aprendizado, o que favorece um refinamento de um treinamento muito mais eficaz.(Scikit-learn developers, 2025)

2.1.10 LIGHTGBM

O LightGBM é um algoritmo que foi desenvolvido pela Microsoft e é baseado no Gradient Boosting Decision Tree (GBDT). Visto que o GBDT usava muita memória e tinha um tempo de treinamento bastante elevado, o LightGBM foi criado com o foco de acelerar o treinamento e performar de forma eficiente em conjuntos de dados massivos. O GBDT funciona percorrendo todo o conjunto de dados diversas vezes a cada iteração, o que pode causar travamentos e gargalos. Isso porque os dados podem não caber na memória, e o processo se torna mais lento devido aos acessos repetidos no armazenamento. (Li et al., 2024)

O LightGBM funciona por meio de um método baseado em Histogramas, onde são armazenadas características contínuas em bins discretos. Isso reduz o uso da memória e diminui o custo dos cálculos dos ganhos em cada split. Esse algoritmo também utiliza uma estratégia chamada leaf-wise, onde, em vez de expandir a árvore camada por camada, é selecionada a folha que produz maior ganho.(Hajihosseinlou; Maghsoudi; Ghezelbash, 2023)

2.1.11 ENSEMBLE LEARNING

O ensemble learning consiste na combinação de múltiplos modelos de aprendizado de máquina com o objetivo de melhorar a capacidade de generalização e reduzir erros que modelos individuais, isoladamente, não conseguem evitar (Zhou, 2021). Segundo Kazmaier e Van Vuuren (2022), essa abordagem é especialmente promissora na análise de sentimentos, uma vez que diferentes modelos apresentam vieses indutivos distintos, cujas previsões combinadas podem compensar fraquezas individuais.

Apesar de seu uso já consolidado em diversas áreas da aprendizagem de máquina, a aplicação de ensembles na análise de sentimentos ainda é limitada. Os autores destacam que ensembles heterogêneos aqueles que combinam algoritmos distintos, como SVM, Regressão Logística e redes neurais tendem a superar ensembles homogêneos, justamente por aproveitarem melhor a diversidade dos modelos base.

Nos ensembles homogêneos, temos as tecnicas de bagging e boosting. O bagging é uma técnica de ensemble, com o objetivo aumentar a precisão de modelos preditivos, especialmente aqueles considerados instáveis, como árvores de decisão e redes neurais. O principal mecanismo do bagging consiste em gerar diversas versões de um mesmo modelo

a partir de subconjuntos de dados obtidos por amostragem com reposição (bootstrap), e, em seguida, combinar suas previsões para formar uma decisão final mais robusta (Breiman, 1996). Já no boosting temos que é um método de aprendizado de máquina que constrói um modelo preditivo forte por meio da combinação sequencial de diversos modelos fracos (ou base learners), de forma que cada novo modelo corrige os erros cometidos pelos anteriores (Friedman, 2001).

Quando falamos de ensembles heterogêneos, temos uma tecnica chamada stacking, ou stacked generalization, é uma técnica de combinação de modelos preditivos que busca melhorar a acurácia da predição ao integrar as saídas de múltiplos algoritmos de aprendizado de máquina. Diferentemente de métodos como bagging ou boosting, o stacking utiliza um modelo de segunda camada, chamado meta-modelo, que é treinado para aprender a melhor forma de combinar as previsões dos modelos de base (Wolpert, 1992). Outra técnica para *ensembles* heterogêneos é a *Voting*, um modelo menos complexo, mas muito eficaz. Possui duas variações: *Soft Voting* e *Hard Voting*. No *Soft Voting*, a previsão é feita com base nas probabilidades que cada algoritmo calcula para cada classe. Essas probabilidades são combinadas, e a predição final é calculada por meio da média ponderada das probabilidades previstas nos modelos. No *Hard Voting*, são computadas as predições de cada modelo para a instância do conjunto. A classe predita é definida pela maioria dos votos dos modelos, sendo, assim, estabelecida por uma votação majoritária entre os modelos que o compõem.(Tauil, 2024)

2.1.12 POOL DE CLASSIFICADORES

Pool de classificadores refere-se a um conjunto de modelos preditivos treinados (ou previamente preparados) com o objetivo de serem utilizados em técnicas de combinação ou seleção dinâmica de classificadores. Esses pools são projetados para explorar a diversidade e complementaridade entre os modelos, de forma que, mesmo que um classificador individual tenha desempenho limitado em determinadas regiões do espaço de atributos, o conjunto como um todo possa alcançar resultados superiores por meio da colaboração (Avelino Júnior, 2022; Sousa, 2020; Manastarla, 2024).

2.2 ANÁLISE DE SENTIMENTO

A Análise de Sentimento é um subcampo fundamental da classificação de textos no Processamento de Linguagem Natural (PLN), cujo objetivo principal é classificar automaticamente documentos textuais com base nos sentimentos, emoções e opiniões expressos (Abdar et al., 2021). Essa técnica busca identificar e extrair informações subjetivas de textos, como avaliações, comentários, postagens em redes sociais e resenhas de produtos, possibilitando uma compreensão mais aprofundada das percepções, intenções e emoções dos usuários em diferentes contextos.

Com o crescimento exponencial da produção de conteúdo textual nas plataformas digitais, a análise de sentimento tornou-se uma ferramenta indispensável para a tomada de decisões. Empresas utilizam essa técnica para entender a percepção dos clientes sobre produtos e serviços, enquanto governos e organizações públicas a empregam para avaliar a opinião popular sobre políticas públicas, eventos sociais e questões econômicas (Onan, 2022). Nesse sentido, a análise de sentimento estabelece uma ponte entre a linguagem humana e a interpretação computacional, fornecendo insights valiosos em tempo real.

A análise de sentimento pode ser dividida em duas vertentes principais: mineração de opinião e mineração de emoções. A mineração de opinião refere-se à detecção da polaridade textual, ou seja, identificar se uma determinada mensagem expressa um sentimento positivo, negativo ou neutro, além de quantificar a intensidade dessa polaridade (Pereira, 2021). Por exemplo, em resenhas de produtos, é possível determinar se um cliente está satisfeito ou insatisfeito, bem como o grau de intensidade desse sentimento.

2.3 OPTUNA

O *Optuna* é um otimizador de hiperparâmetros cujo funcionamento é baseado em Otimização Bayesiana. Ele serve para encontrar o melhor conjunto de parâmetros para um modelo de *machine learning* e se utiliza de modelos como *Tree-structured Parzen Estimator* (TPE), *Covariance Matrix Adaptation* (CMA), *Gaussian Processes* (GPs) e *Asynchronous Successive Halving* (ASHA), oferecendo uma série de benefícios, como maior flexibilidade, eficiência e capacidade de lidar com hiperparâmetros contínuos e discretos.(Imani; Arabnia, 2023a)

Para um modelo *performar* bem, não depende apenas dos algoritmos selecionados; os hiperparâmetros são de suma importância. Os modelos possuem uma série de hiperparâmetros que são definidos antes do treinamento, e otimizá-los é uma etapa que é muito importante, porém trabalhosa e custosa computacionalmente. Existem diversos métodos, como os tradicionais *Grid Search*, *Random Search* e *Algoritmos Genéticos*. Porém, esses modelos de otimização possuem limitações, pois são custosos e pouco eficientes, e Algoritmos Genéticos tendem a convergir para ótimos locais. Por isso, o *Optuna* vem como a solução para essas limitações, pois ele aprende continuamente com otimizações anteriores, direcionando a busca para regiões mais promissoras do espaço de parâmetros.(Imani; Arabnia, 2023b; Srinivas; Katarya, 2022)

2.4 SELEÇÃO

A eficácia de um modelo de *ensemble* depende das escolhas dos classificadores que o irão compor. A simples adição de vários modelos não garante a melhoria; o ganho está fortemente ligado à seleção dos modelos, para que eles se complementem. Em outras palavras, *ensembles* funcionam melhor quando seus membros cometem erros *não correlacionados*.

Portanto, para a seleção, é crucial considerar duas características: o *desempenho individual* e a *diversidade* entre os modelos, de modo a explorar variações no padrão de erro.(Jurek et al., 2014)

Dessa forma, a seleção baseada apenas no desempenho pode ser insuficiente em alguns casos, visto que muitos modelos funcionam de forma semelhante, o que diminui o ganho coletivo.

Visto que a seleção é de suma importância para a construção de *ensembles* eficazes, em Yang (2011), é demonstrado que *ensembles* bem-sucedidos dependem do equilíbrio entre a acurácia individual e a diversidade entre modelos, e que classificadores redundantes podem prejudicar o conjunto. A seleção é tratada como um problema de *otimização*, no qual um subconjunto, retirado do *pool* de classificadores, é escolhido de forma criteriosa, combinando métricas de desempenho e diversidade. Ainda em Yang (2011) também é demonstrado que *ensembles* menores, mas bem selecionados, performavam igual ou melhor que *ensembles* completos. Isso é benéfico, pois além de manter métricas elevadas, o *custo computacional* é reduzido.

3 DELINEAMENTO METODOLÓGICO

Nesta seção, são discutidos os procedimentos utilizados para o desenvolvimento do estudo, incluindo a descrição das bases de dados utilizadas e como foram construídas, o processo de rotulação, as técnicas de pré-processamento aplicadas e os protocolos de experimentação adotados. O objetivo principal deste trabalho é analisar a eficácia de diferentes estratégias de pré-processamento textual na análise de sentimentos aplicada a comentários de filmes, utilizando e comparando algoritmos clássicos de classificação supervisionada e técnicas de ensemble learning. A pesquisa tem natureza experimental, com foco na aplicação prática de algoritmos, bem como na avaliação de desempenho das estratégias de pré-processamento e dos modelos utilizados.

3.1 BASE DE DADOS

Duas bases de dados compostas por comentários de filmes foram utilizadas neste estudo: uma proveniente do Letterboxd e outra do IMDb.

3.1.1 LETTERBOXD

A primeira base de dados foi composta por comentários e notas de usuários da plataforma Letterboxd. Inicialmente, foram coletados 150.000 comentários por meio de um crawler desenvolvido em Python, utilizando a biblioteca Selenium. A extração foi realizada a partir das avaliações de 50 filmes diversos disponíveis na plataforma.

Os comentários estavam em vários idiomas, sendo a maioria em inglês. Para garantir a uniformidade linguística da base, foram utilizados apenas os comentários em inglês. Para isso, os dados foram separados em duas bases uma com comentários em inglês e outra com os demais idiomas utilizando a biblioteca langdetect, também em Python. Cada comentário vinha acompanhado de uma nota atribuída pelo usuário, variando de 0.5 a 5 estrelas. Comentários sem nota receberam, inicialmente, o valor 0 durante a extração, mas foram posteriormente removidos.

Durante a definição das classes, observou-se que a nota 3 não se comportava adequadamente como classe neutra, ainda que, inicialmente, este estudo tivesse sido planejado para trabalhar com três classes, e não duas, como acabou sendo conduzido. Embora na literatura a nota intermediária seja frequentemente interpretada como indicativa de neutralidade(Aftab et al., 2023; Ojeda; Zalewski; Maletzke, 2024; Silva et al., 2023). As análises exploratórias realizadas neste trabalho demonstraram que essa suposição introduzia inconsistências e prejudicava o desempenho dos classificadores.

Diante desse cenário, foram conduzidos experimentos adicionais com técnicas de agrupamento e inspeção estrutural dos dados. Os resultados revelaram que a nota 3 não constituía uma classe intermediária coerente, apresentando grande heterogeneidade

interna. Verificou-se, ainda, que a maior parte dos comentários originalmente rotulados como neutros apresentava maior afinidade linguística e semântica com a classe positiva.

Com base nessas evidências, optou-se por agrupar a nota 3 junto à classe positiva. Essa decisão reduziu a ambiguidade da rotulagem, melhorou a separação entre as classes e contribuiu para classificadores mais estáveis e com melhor desempenho.

Após os tratamentos descritos, a base final contou com 114.817 comentários em inglês. A rotulação dos sentimentos foi feita com base nas notas atribuídas pelos usuários, utilizando o seguinte critério:

- Positivo: Comentários com nota igual ou superior a 3
- Negativo: Comentários com nota inferior a 3.

Essa estratégia de rotulação reflete uma suposição comum em análise de sentimentos, segundo a qual notas mais altas indicam sentimentos positivos e notas baixas indicam sentimentos negativos. Essa abordagem permite a construção de um conjunto de dados rotulado de forma automática, facilitando o treinamento dos modelos de classificação.

3.1.2 IMDB

A segunda base de dados foi obtida do IMDb (Internet Movie Database) e é amplamente utilizada na literatura sobre análise de sentimentos. Diferentemente da base do Letterboxd, essa base já se encontra pré-processada e rotulada, o que elimina a necessidade de rotulação manual. Ela contém 50.000 avaliações de filmes, com classificações binárias (positivo ou negativo), sendo comumente utilizada como benchmark em tarefas de classificação de sentimentos.

A base do IMDb complementa a do Letterboxd ao oferecer uma maior diversidade de estilos de escrita, formatos de comentários e temas de filmes, o que é essencial para testar a capacidade de generalização dos modelos. A combinação dessas duas fontes de dados permite uma análise mais robusta e abrangente.

3.2 TÉCNICAS DE PRÉ-PROCESSAMENTO

A construção de modelos de classificação baseados em texto requer a aplicação de técnicas de pré-processamento, que têm um impacto direto na performance dos algoritmos. Estudos demonstram que a aplicação adequada dessas técnicas pode melhorar significativamente os resultados obtidos (Almeida Neto; De Melo, 2023). As técnicas utilizadas neste trabalho foram: Remoção de caracteres especiais e emojis, remoção das stopwords, Lemmatization e Stemming. Durante essa etapa é importante apontar que também foi utilizada uma etapa de pré-processamento chamada lowercase, que foi a de deixar todas em minúsculas, que foi aplicada em todos.

Caracteres especiais são símbolos não alfanuméricos, como pontuações (!.,?), símbolos (@"(/), entre outros (Brandão et al., 2023). Já os emojis são representações gráficas de emoções, objetos, lugares, etc., podendo ser compostos por códigos Unicode ou combinações de caracteres (Paula, 2019). A remoção desses elementos visa gerar um texto mais limpo, excluindo itens que não agregam valor semântico relevante à tarefa de análise de sentimentos. Essa etapa foi implementada com o uso de expressões regulares (regex), utilizando a biblioteca re do Python.

Stopwords são palavras muito comuns em um idioma como artigos, preposições e pronomes que geralmente não contribuem significativamente para a compreensão do conteúdo semântico do texto. A remoção dessas palavras permite concentrar a análise nas informações mais relevantes (Kaur; Buttar, 2018). Para isso, utilizou-se a biblioteca NLTK, que disponibiliza listas de stopwords para diversos idiomas, tendo sido utilizada, neste caso, a lista em inglês.

A lematização (*lemmatization*) consiste em reduzir uma palavra à sua forma base, considerando seu contexto gramatical e semântico. Por exemplo, as palavras "correu" e "corrida" são ambas reduzidas à forma base "correr" (Brandão et al., 2023). Essa técnica reduz a variabilidade linguística e facilita a identificação de padrões pelos modelos. A lematização foi realizada com o auxílio da biblioteca NLTK.

O *stemming* é uma técnica que reduz palavras aos seus radicais, por meio da remoção de sufixos e prefixos, sem considerar o contexto semântico (Souza et al., 2021). Por exemplo, as palavras "correu" e "corrida" seriam reduzidas ao radical "corr". Assim como a lematização, essa técnica busca reduzir a variabilidade linguística, mas de forma mais simplificada. Também foi utilizada a biblioteca NLTK para a implementação dessa etapa.

3.3 PROTOCOLO DE EXPERIMENTOS

Nesta seção, são descritas as ferramentas, técnicas e procedimentos utilizados para treinar, ajustar e avaliar os modelos propostos.

3.3.1 DIVISÃO DOS DADOS

Em cada uma das bases (Letterboxd e IMDb) foram divididas em três subconjuntos distintos:

- 80% para o conjunto de treino, utilizado no treinamento dos modelos;
- 10% para validação, utilizado para ajuste de hiperparâmetros e seleção de modelos;
- 10% para teste, utilizado para avaliar o desempenho final dos modelos.

Essa divisão foi feita de forma estratificada, garantindo que a proporção entre classes fosse mantida em cada subconjunto. Foi utilizada a função `train_test_split` da biblioteca `Scikit-learn` da linguagem `Python`.

3.3.2 FERRAMENTAS E BIBLIOTECAS

Todos os experimentos foram realizados utilizando a linguagem de programação `Python` com o ambiente de desenvolvimento sendo o `Jupyter Notebook`. As bibliotecas e *frameworks* adotados foram: `Scikit-learn` para a implementação dos algoritmos de *machine learning* e validação cruzada; `NLTK` que é uma biblioteca de pré-processamento textual; *Optuna* que é uma biblioteca do Python para otimização de hiperparâmetros; `Selenium` que foi a biblioteca utilizada para o *crawler* pegar os dados da plataforma `Letterboxd`; e `Pandas` para manipulação de dados.

3.3.3 REPRESENTAÇÃO DOS DADOS

Após o pré-processamento textual, os comentários foram convertidos em representações numéricas por meio da técnica `TF-IDF` (*Term Frequency–Inverse Document Frequency*). Essa técnica calcula a frequência relativa de uma palavra em um documento em comparação à sua frequência em todo o *corpus*, atribuindo pesos maiores às palavras mais relevantes e penalizando termos muito frequentes.

Essa representação permite que os algoritmos de classificação se concentrem em termos informativos e discriminativos para a tarefa de análise de sentimentos. A implementação foi realizada por meio da função `TfidfVectorizer`, da biblioteca `Scikit-learn`.

3.3.4 VALIDAÇÃO CRUZADA E AJUSTE DE HIPERPARÂMETROS

Com o objetivo de garantir robustez na otimização dos hiperparâmetros, foi adotado o método de validação cruzada estratificada com 5 *folds* (k=5), utilizando a classe `StratifiedKFold` da biblioteca `Scikit-learn` para . A estratificação assegura que a proporção entre as classes seja preservada em cada subdivisão dos dados, evitando viés.

Para o ajuste dos hiperparâmetros, utilizou-se a técnica de otimização automática fornecida pelo framework *Optuna*. A escolha por esse framework deve-se ao fato de que métodos tradicionais, como *grid search* e *random search*, tendem a ser menos eficientes e significativamente mais lentos quando comparados ao processo de otimização baseado em busca bayesiana e seleção adaptativa utilizado pelo *Optuna*.

A otimização foi realizada de forma independente para cada conjunto de dados (*Letterboxd* com pré-processamento, *Letterboxd* sem pré-processamento, *IMDb* com pré-processamento e *IMDb* sem pré-processamento). Em cada cenário, o *Optuna* executou 30 tentativas (*trials*) por modelo, buscando maximizar as métricas de desempenho definidas.

Esse processo resultou em um conjunto de hiperparâmetros ótimo para cada algoritmo avaliado, os quais são apresentados nas tabelas a seguir:

Tabela 1 – Hiperparâmetros otimizados — Letterboxd com pré-processamento

Modelo	**Hiperparâmetros**
Naive Bayes	alpha = 0.6758407653651551
Logistic Regression	C = 0.6246792786048405, penalty = l2
SVM Linear	C = 0.06013530856715907, loss = squared_hinge
Random Forest	n_estimators = 115, max_depth = 30, min_samples_split = 2, min_samples_leaf = 1
KNN	n_neighbors = 5, weights = distance, p = 2
MLP	hidden_layer_sizes = (100, 50), learning_rate_init = 0.00014744878823096062, alpha = 0.00025996826132761928, max_iter = 370
Decision Tree	max_depth = 20, min_samples_split = 4, criterion = gini
LightGBM	num_leaves = 36, learning_rate = 0.09677801420002084, n_estimators = 238, max_depth = 9, subsample = 0.6292147199320182, colsample_bytree = 0.8785528356077164
AdaBoost	n_estimators = 300, learning_rate = 0.7947974759368932
SGD	loss = modified_huber, alpha = 0.0002752762905896779

Tabela 2 – Hiperparâmetros otimizados — Letterboxd sem pré-processamento

Modelo	**Hiperparâmetros**
Naive Bayes	alpha = 0.9737981758173101
Logistic Regression	C = 0.5838191361607037, penalty = l2
SVM Linear	C = 0.4765256659616241, loss = hinge
Random Forest	n_estimators = 198, max_depth = 30, min_samples_split = 4, min_samples_leaf = 1
KNN	n_neighbors = 5, weights = distance, p = 2
MLP	hidden_layer_sizes = (50, 50), learning_rate_init = 0.00010400993294201546, alpha = 0.008482443671690949, max_iter = 482
Decision Tree	max_depth = 19, min_samples_split = 6, criterion = gini
LightGBM	num_leaves = 45, learning_rate = 0.09929154892388725, n_estimators = 276, max_depth = 10, subsample = 0.726920757114998, colsample_bytree = 0.8424583748814402
AdaBoost	n_estimators = 299, learning_rate = 0.6456811835697023
SGD	loss = log_loss, alpha = 0.00001743455841195683 5

Tabela 3 – Hiperparâmetros otimizados — IMDb com pré-processamento

Modelo	**Hiperparâmetros**
Naive Bayes	alpha = 0.02927511356754202
Logistic Regression	C = 1.661735240578435, penalty = l2
SVM Linear	C = 0.5097630076639009, loss = hinge
Random Forest	n_estimators = 299, max_depth = 30, min_samples_split = 3, min_samples_leaf = 1
KNN	n_neighbors = 15, weights = uniform, p = 2
MLP	hidden_layer_sizes = (100,), learning_rate_init = 0.0007955551045346731, alpha = 0.0025435382319358075, max_iter = 471
Decision Tree	max_depth = 18, min_samples_split = 6, criterion = gini
LightGBM	num_leaves = 27, learning_rate = 0.08289881794330034, n_estimators = 266, max_depth = 8, subsample = 0.8134698312657079, colsample_bytree = 0.6418085519195297
AdaBoost	n_estimators = 159, learning_rate = 0.8314950879497945
SGD	loss = modified_huber, alpha = 0.00003544253928216622

Tabela 4 – Hiperparâmetros otimizados — IMDb sem pré-processamento

Modelo	**Hiperparâmetros**
Naive Bayes	alpha = 0.9153390147081293
Logistic Regression	C = 1.2968000704228235, penalty = l2
SVM Linear	C = 0.11110818042262983, loss = squared_hinge
Random Forest	n_estimators = 397, max_depth = 30, min_samples_split = 6, min_samples_leaf = 2
KNN	n_neighbors = 14, weights = distance, p = 2
MLP	hidden_layer_sizes = (50, 50), learning_rate_init = 0.0004978496912038466, alpha = 0.00012550127377080177, max_iter = 486
Decision Tree	max_depth = 20, min_samples_split = 4, criterion = gini
LightGBM	num_leaves = 25, learning_rate = 0.05767611522807929, n_estimators = 300, max_depth = 8, subsample = 0.8790404915770229, colsample_bytree = 0.6497297150122884
AdaBoost	n_estimators = 296, learning_rate = 0.7201004649039641
SGD	loss = modified_huber, alpha = 0.00018257582631215738

3.3.5 FLUXOGRAMA DOS EXPERIMENTOS

A imagem abaixo mostra o fluxo que o trabalho irá seguir:

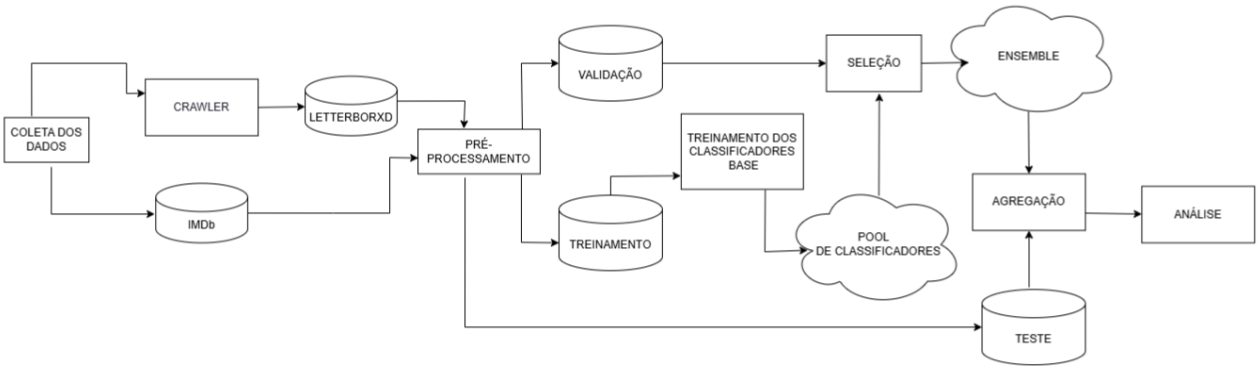

Figura 1 – Fluxograma do seguimento do trabalho. Fonte: Elaborado pelo autor

O processo experimental inicia-se com a seleção e preparação dos conjuntos de dados. Foram utilizadas duas fontes principais: a base `IMDb`, previamente estruturada e rotulada, e a base construída a partir de comentários extraídos do `Letterboxd` por meio de um *crawler* em `Python`, com o auxílio da biblioteca `Selenium`.

Após a coleta, os dados passam por um processo de pré-processamento, que inclui a remoção de *stopwords*, *emojis* e caracteres especiais, *stemming* e conversão para letras minúsculas (*lowercase*). Em seguida, os textos são transformados em vetores numéricos utilizando o *TF-IDF* (*Term Frequency-Inverse Document Frequency*), por meio do `TfidfVectorizer` da biblioteca `Scikit-learn`.

Com os dados vetorizados, é feita a otimização dos hiperparâmetros com o *Optuna*, onde serão selecionados os hiperparâmetros ótimos para o treinamento dos modelos. Para essa otimização, foi utilizada a validação cruzada com ($k = 5$).

Posteriormente, os dados são divididos em três subconjuntos: 80% para treinamento, 10% para validação e 10% para teste, com estratificação para preservar o equilíbrio entre as classes.

Na etapa seguinte, são implementados e treinados algoritmos clássicos de classificação supervisionada, como *Naive Bayes*, *SVM*, *Regressão Logística*, *MLP (Multi-Layer Perceptron)*, *Decision Tree*, *AdaBoost*, *LightGBM*, *SGD (Stochastic Gradient Descent)*, *KNN* e *Random Forest*. Esses modelos resultantes compõem um *pool de classificadores*.

A partir do *pool de classificadores*, são selecionados os modelos mais promissores para a composição dos *ensembles* heterogêneos, por meio da *Seleção por Acurácia* (*SA*) e *Seleção por Acurácia e Diversidade* (*SAD*). No *SA*, foram escolhidos os modelos que tiveram o melhor desempenho individual. Já para o *SAD*, foi considerada também a *diversidade*, utilizando-se o método estatístico *Q-statistic* para mensurá-la. O *Q-statistic* mede a quantidade de erros não correlacionados entre os modelos, priorizando aqueles

que erram em exemplos diferentes. Foram gerados *ensembles* com quantidades de modelos variando entre [2, 3, 4, 5], conforme a literatura (Dietterich (2000) e Kuncheva (2004)) indica que *ensembles* menores podem performar muito próximo ou igual a conjuntos com muitos modelos. Além desses, foi incluído o *ensemble* com todos os modelos, denominado *Sem Seleção* (*SS*). Esses *ensembles* combinam as previsões dos modelos base, gerando decisões mais robustas.

Por fim, os *ensembles* são avaliados com base no conjunto de teste final, utilizando métricas de avaliação como acurácia, precisão, revocação (*recall*) e *F1-score*.

4 RESULTADOS E DISCURSÕES

Nesta seção, são apresentados todos os dados e resultados obtidos ao longo dos experimentos. Primeiro, descrevem-se as métricas de avaliação dos modelos de classificação individuais utilizados no estudo. Em seguida, são exibidas as métricas dos ensembles construídos por meio das diferentes estratégias de seleção. Por fim, são realizadas comparações entre os modelos e discutidos os resultados observados, destacando o impacto das técnicas aplicadas no desempenho final.

4.1 DESEMPENHOS DOS MODELOS INDIVIDUAIS

Foram avaliados dez modelos de classificação em quatro cenários distintos: Letterboxd com pré-processamento, Letterboxd sem pré-processamento, IMDb sem pré-processamento e IMDb com pré-processamento.

Nos resultados obtidos, já podemos perceber as diferenças de desempenho entre os modelos e o impacto do pré-processamento sobre a eficácia dos algoritmos.

As tabelas a seguir detalham os resultados obtidos para cada cenário, apresentando as métricas de avaliação: Precisão, Recall e F1-Score para as classes Negativa e Positiva.

Tabela 5 – Métricas dos Modelos Individuais(Letterborxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
KNN	0.54	0.75	0.63	0.67	0.43	0.52
NB	0.79	0.78	0.78	0.81	0.82	0.81
RL	0.80	0.79	0.80	0.82	0.83	0.82
SGD	0.80	0.80	0.80	0.82	0.82	0.82
SVM	0.80	0.79	0.80	0.82	0.83	0.82
RF	0.78	0.69	0.73	0.75	0.83	0.79
LightGBM	0.77	0.80	0.78	0.81	0.79	0.80
AdaBoost	0.74	0.81	0.77	0.81	0.75	0.78
MLP	0.77	0.81	0.79	0.82	0.79	0.80
DT	0.60	0.84	0.70	0.78	0.49	0.61

No cenário da base Letterboxd com pré-processamento, pode-se observar que os *modelos lineares* apresentaram métricas melhores que os de outro tipo. Os modelos *SVM Linear*, *Regressão Logística* e *SGD* apresentaram F1-Scores entre *0.80* e *0.82*, tanto para a classe positiva quanto para a negativa. Quanto aos modelos que não tiveram uma boa performance, destacam-se *KNN* e a *Decision Tree*, com F1-Scores entre *0.52* e *0.70*.

Tabela 6 – Métricas dos Modelos Individuais(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
NB	0.79	0.77	0.78	0.80	0.82	0.81
RL	0.80	0.79	0.79	0.82	0.82	0.82
SVM	0.80	0.80	0.80	0.82	0.82	0.82
RF	0.79	0.66	0.72	0.74	0.84	0.79
KNN	0.56	0.71	0.62	0.66	0.50	0.57
MLP	0.77	0.78	0.78	0.81	0.79	0.80
DT	0.71	0.42	0.53	0.63	0.85	0.72
LightGBM	0.77	0.80	0.79	0.82	0.79	0.81
AdaBoost	0.74	0.79	0.77	0.80	0.76	0.78
SGD	0.79	0.80	0.80	0.82	0.81	0.82

No contexto da base Letterboxd sem pré-processamento, pode-se observar melhorias nos resultados. Os modelos lineares ainda foram superiores, com F1-Scores próximos a *0.82*. Isso sugere que o pré-processamento aplicado pode ter removido informações relevantes do texto, ou seja, a remoção de palavras neste caso prejudicou o desempenho do modelo. Os modelos *KNN* e *Decision Tree* ainda tiveram um desempenho abaixo, e os demais foram intermediários.

Tabela 7 – Métricas dos Modelos Individuais(IMDB - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
NB	0.87	0.83	0.85	0.84	0.88	0.86
RL	0.90	0.88	0.89	0.88	0.90	0.89
SVM	0.91	0.88	0.89	0.88	0.91	0.89
RF	0.87	0.82	0.84	0.83	0.87	0.85
KNN	0.78	0.63	0.70	0.69	0.82	0.75
MLP	0.88	0.88	0.88	0.88	0.88	0.88
DT	0.76	0.66	0.71	0.70	0.79	0.74
LightGBM	0.88	0.85	0.86	0.85	0.89	0.87
AdaBoost	0.86	0.83	0.85	0.84	0.87	0.85
SGD	0.88	0.88	0.88	0.88	0.88	0.88

Na base do IMDb sem pré-processamento, que é uma base mais estruturada, os modelos apresentaram desempenhos significativamente melhores. No entanto, os modelos lineares ainda foram superiores: *SVM Linear*, *Regressão Logística* e *SGD* obtiveram um F1-Score de aproximadamente *0.89*. Isso pode ser explicado pela qualidade da base de

dados, que tem menos ruído. Os modelos *KNN* e *Decision Tree* continuaram sendo os piores, com F1-Scores na faixa dos *0.70*.

Tabela 8 – Métricas dos Modelos Individuais(IMDB - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
NB	0.87	0.84	0.86	0.84	0.88	0.86
RL	0.90	0.88	0.89	0.88	0.91	0.89
SVM	0.90	0.88	0.89	0.88	0.91	0.89
RF	0.87	0.81	0.84	0.82	0.88	0.85
KNN	0.74	0.69	0.71	0.71	0.75	0.73
MLP	0.88	0.86	0.87	0.86	0.88	0.87
DT	0.79	0.64	0.71	0.70	0.83	0.76
LightGBM	0.88	0.84	0.86	0.85	0.89	0.87
AdaBoost	0.87	0.84	0.86	0.85	0.88	0.86
SGD	0.91	0.86	0.89	0.87	0.92	0.89

Por fim, na base do IMDb com pré-processamento, pode-se observar que as métricas dos melhores modelos foram praticamente idênticas às do cenário sem pré-processamento, ou seja, pode ser melhor não realizar o pré-processamento nesse dataset, tendo em vista que é muito custoso computacionalmente, principalmente para bases tão grandes. Isso indica também que a base do IMDb já é bem estruturada e suficientemente limpa para que o pré-processamento adicional não traga benefícios e nem prejudique. Os mesmos modelos foram superiores (*SVM Linear*, *Regressão Logística* e *SGD*), com valores entre *0.88* e *0.89*, enquanto o *KNN* e a *Decision Tree* ainda foram os menos eficazes.

Em resumo, os resultados demonstram que *classificadores lineares* (*SVM Linear*, *Regressão Logística* e *SGD*) são os mais adequados para a tarefa de análise de sentimento utilizando a representação *TF-IDF* mostrando eficácia superior tanto em bases menores (Letterboxd) quanto em bases maiores e mais estruturadas (IMDb). Isso ocorreu porque seus datasets são binários. Ou seja, o que podemos concluir é que esse classificadores tiveram os melhores resultados, no contexto estudado. Não podemos concluir que serão os melhores sempre e em qualquer tarefa de análise de sentimentos.

Os modelos *KNN* e *Decision Tree* apresentaram um desempenho consistentemente inferior em todos os cenários. Contudo, seu uso foi relevante ao contribuir com a diversidade de métodos nos esquemas de *ensemble*.

Já os modelos baseados em árvores (*Random Forest*, *LightGBM* e *AdaBoost*) exibiram um desempenho intermediário, porém satisfatório, posicionando-se como opções robustas.

Tais achados reforçam o que é estabelecido na literatura: *modelos lineares são frequentemente superiores* quando se trata de classificação textual com TF-IDF. (Hassan;

Ahamed; Ahmad, 2022; Galke et al., 2022; Kucuk et al., 2024) Além disso, a análise evidenciou o impacto variável do pré-processamento em diferentes bases, podendo prejudicar, melhorar ou não ter efeito, dependendo da base.

4.2 DESEMPENHO DOS ENSEMBLES SEM SELEÇÃO

Uma das etapas dos experimentos foi a avaliação do *ensemble* sem seleção, com o objetivo de observar o comportamento e como a diversidade de modelos impacta a performance do *ensemble* tanto no *Voting* quanto no *Stacking*.

Tabela 9 – Métricas para Ensembles com 10 Modelos Base(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.77	0.83	0.80	0.84	0.78	0.81
Stacking	0.80	0.81	0.81	0.83	0.82	0.83

Na base *Letterboxd* com pré-processamento, os resultados mostraram que o *Stacking* obteve um desempenho superior, conseguindo atingir um F1-Score de *0.81* e *0.83* nas duas classes, enquanto o *Voting* obteve valores de *0.80* e *0.81*. O Stacking foi capaz de aproveitar melhor essa diferença entre os modelos.

Tabela 10 – Métricas para Ensembles com 10 Modelos Base(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.81	0.80	0.83	0.81	0.82
Stacking	0.80	0.80	0.80	0.82	0.82	0.82

Na versão sem pré-processamento da Letterboxd, ambos os ensembles obtiveram desempenho semelhante, com F1-Scores entre *0.80* e *0.82*. O Stacking ainda apresentou um leve ganho na classe positiva, mas a diferença geral foi menor do que a observada no cenário com pré-processamento.

Na base *IMDb* sem pré-processamento, os ensembles atingiram ótimos resultados. Tanto o *Voting* quanto o *Stacking* alcançaram F1-Scores próximos de *0.90*. O Stacking ainda apresentou um leve ganho, mas, na prática, ambos os métodos tiveram desempenho muito similar.

Tabela 11 – Métricas para Ensembles com 10 Modelos Base(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.89	0.90	0.90	0.90	0.89	0.90
Stacking	0.90	0.89	0.90	0.90	0.90	0.90

Tabela 12 – Métricas dos Ensembles com 10 Modelos Base(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.90	0.90	0.90	0.90	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Para finalizar, na base do *IMDb* com pré-processamento, os resultados foram bem parecidos com os do cenário sem pré-processamento, porém houve um leve ganho tanto no Voting quanto no Stacking. Com isso, confirma-se que o desempenho na base IMDb *não é fortemente influenciado pelo pré-processamento*, e ensembles de grande porte podem ser robustos mesmo com variações no tratamento do texto.

Os resultados mostraram que *ensembles* com 10 modelos apresentam um desempenho bastante sólido em diversos cenários. Quando há ruído, o *stacking* tende a ser superior ao *voting*, enquanto o *voting* já tem um bom desempenho quando os modelos base são mais consistentes.

4.3 SELEÇÃO POR ACURÁCIA

Nesta seção, serão mostrados os resultados dos *ensembles* construídos por meio da seleção por acurácia (*SA*), onde somente os modelos individuais com o maior desempenho são escolhidos para a composição do *ensemble*. Aqui, os *ensembles* serão formados por subconjuntos contendo 2, 3, 4 ou 5 modelos. Essa estratégia busca avaliar *ensembles* de diferentes tamanhos.

As tabelas a seguir mostram os valores das acuracias de cada modelo que foram usadas como base para a formação dos ensembles:

Tabela 13 – Acurácia dos Modelos Individuais (Letterboxd - Sem Pré-Processamento)

Modelo	**Acurácia**
NB	0.7950
RL	0.8076
SVM	0.8101
RF	0.7577
KNN	0.5971
MLP	0.7882
DT	0.6497
LightGBM	0.7967
AdaBoost	0.7744
SGD	0.8078

Tabela 14 – Acurácia dos Modelos Individuais (Letterboxd - Com Pré-Processamento)

Modelo	**Acurácia**
NB	0.7981
RL	0.8118
SVM	0.8107
RF	0.7645
KNN	0.5833
MLP	0.7972
DT	0.6583
LightGBM	0.7916
AdaBoost	0.7769
SGD	0.8099

Tabela 15 – Acurácia dos Modelos Individuais (IMDb - Sem Pré-Processamento)

Modelo	**Acurácia**
NB	0.8584
RL	0.8924
SVM	0.8930
RF	0.8426
KNN	0.7226
MLP	0.8694
DT	0.7330
LightGBM	0.8634
AdaBoost	0.8616
SGD	0.8894

Tabela 16 – Acurácia dos Modelos Individuais (IMDb - Com Pré-Processamento)

Modelo	**Acurácia**
NB	0.8538
RL	0.8902
SVM	0.8930
RF	0.8466
KNN	0.7286
MLP	0.8784
DT	0.7274
LightGBM	0.8658
AdaBoost	0.8510
SGD	0.8816

Os valores de acurácia apresentados nesta seção são fundamentais para identificar os modelos com melhor desempenho em cada cenário e, assim, orientar a composição dos *ensembles*, tanto pela estratégia de Seleção por Acurácia (SA) quanto pela Seleção por Acurácia e Diversidade (SAD). Foi observado que os modelos que tiveram um desempenho superior foram: Regressão Logística, SVM e SGD.

4.3.1 DESEMPENHO DOS ENSEMBLES COM 2 MODELOS

Nesta subseção, analisam-se os *ensembles* compostos por dois modelos, selecionados pela estratégia de Seleção por Acurácia (SA), destacando o impacto da escolha dos modelos sobre o desempenho geral do ensemble.

Tabela 17 – Métricas para Ensembles SA 2 modelos: Regressão Logística, SVM Linear (Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.80	0.80	0.83	0.82	0.82
Stacking	0.80	0.80	0.80	0.82	0.82	0.82

Tabela 18 – Métricas para Ensembles SA 2 modelos: SVM Linear e SGD (Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.78	0.81	0.80	0.83	0.80	0.82
Stacking	0.79	0.80	0.79	0.82	0.81	0.82

Tabela 19 – Métricas para Ensembles SA 2 modelos: SVM Linear e Regressão Logística (IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.89	0.90	0.90	0.90	0.89	0.89
Stacking	0.89	0.90	0.90	0.90	0.89	0.89

Tabela 20 – Métricas para Ensembles SA (2 modelos): SVM Linear e Regressão Logística (IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.91	0.90	0.91	0.89	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Nos *ensembles* com 2 modelos, pode-se observar que a combinação foi feita pelos modelos lineares mais fortes, sendo *Regressão Logística*, *SVM Linear* e *SGD*, que apresentaram desempenhos sólidos, tanto no *voting* quanto no *stacking*. No *dataset* do *Letterboxd*, os F1-Scores ficaram entre *0.79* e *0.82*, e na base do *IMDb*, foram alcançados *0.90*. Com este resultado, demonstra-se que, para tarefas de análise de sentimento, um *ensemble* formado por apenas dois modelos fortes pode produzir um desempenho favorável.

4.3.2 DESEMPENHO DOS ENSEMBLES COM 3 MODELOS

Nesta subseção, analisam-se os *ensembles* compostos por três modelos, selecionados pela estratégia de Seleção por Acurácia (SA), demonstrando o impacto da adição do terceiro modelo na performance geral do *ensemble*.

Tabela 21 – Métricas para Ensembles SA 3 modelos: Regressão Logística, SVM Linear, SGD(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.80	0.80	0.80	0.82	0.82	0.82
Stacking	0.80	0.80	0.80	0.82	0.82	0.82

Tabela 22 – Métricas para Ensembles SA 3 modelos: SVM Linear, SGD, Regressão Logística(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.80	0.79	0.82	0.82	0.82
Stacking	0.79	0.80	0.79	0.82	0.82	0.82

Tabela 23 – Métricas para Ensembles SA 3 modelos: SVM Linear, Regressão Logística, SGD (IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.90	0.90	0.90	0.90	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Tabela 24 – Métricas para Ensembles SA 3 modelos: SVM Linear, Regressão Logística, SGD (IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.90	0.90	0.90	0.90	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Nos *ensembles* formados por 3 modelos, com a adição de mais um modelo, que ainda foi um modelo linear, tanto o *voting* quanto o *stacking* mantiveram o desempenho do *ensemble* com dois modelos. Isso mostra que, quando os classificadores são muito parecidos em comportamento e estrutura, os ganhos são limitados.

4.3.3 DESEMPENHO DOS ENSEMBLES COM 4 MODELOS

Já nesta subseção, analisam-se os *ensembles* compostos por quatro modelos, selecionados pela estratégia de Seleção por Acurácia (SA), demonstrando como a inclusão do quarto modelo influencia os resultados e o desempenho geral do *ensemble*.

Tabela 25 – Métricas para Ensembles SA 4 modelos: Regressão Logística, SVM Linear, SGD, Naive Bayes(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.81	0.80	0.83	0.81	0.82
Stacking	0.80	0.80	0.80	0.82	0.83	0.82

Tabela 26 – Métricas para Ensembles SA 4 modelos: SVM Linear, SGD, Regressão Logística, LightGBM(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.81	0.80	0.83	0.81	0.82
Stacking	0.79	0.80	0.80	0.83	0.81	0.82

Tabela 27 – Métricas para Ensembles SA 4 modelos: SVM Linear, Regressão Logística, SGD, MLP(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.89	0.90	0.90	0.90	0.89	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Tabela 28 – Métricas para Ensembles SA (4 modelos): SVM Linear, Regressão Logística, SGD, MLP(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.88	0.91	0.90	0.91	0.88	0.89
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Com a inclusão de um quarto modelo, puderam ser notados pequenos ganhos de diversidade, principalmente nas bases do *Letterboxd*. A presença de um método diferente (*Naive Bayes* ou *LightGBM*) contribuiu para uma leve melhoria no equilíbrio entre precisão e *recall* em algumas classes; contudo, esses ganhos foram discretos e não superaram de maneira expressiva os *ensembles* com 3 modelos.

4.3.4 DESEMPENHO DOS ENSEMBLES COM 5 MODELOS

Por fim, nesta subseção, analisam-se os *ensembles* compostos por cinco modelos, selecionados pela estratégia de Seleção por Acurácia (SA), demonstrando como a ampliação do *ensemble* afeta sua capacidade de generalização e desempenho sobre o conjunto de dados.

Tabela 29 – Métricas para Ensembles SA 5 modelos: Regressão Logística, SVM Linear, SGD, Naive Bayes, MLP(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.80	0.80	0.80	0.83	0.82	0.82
Stacking	0.80	0.81	0.80	0.83	0.82	0.83

Tabela 30 – Métricas para Ensembles SA 5 modelos: SVM Linear, SGD, Regressão Logística, LightGBM, Naive Bayes(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.80	0.79	0.82	0.82	0.82
Stacking	0.79	0.81	0.80	0.83	0.81	0.82

Tabela 31 – Métricas para Ensembles SA 5 modelos: Linear SVM, Regressão Logística, SGD, MLP, LightGBM(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.89	0.90	0.89	0.90	0.90
Stacking	0.90	0.89	0.90	0.89	0.90	0.90

Tabela 32 – Métricas para Ensembles SA 5 modelos: SVM Linear, Regressão Logística, SGD, MLP, LightGBM(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.90	0.90	0.90	0.90	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Por fim, nos *ensembles* com 5 modelos, com a adição de outro modelo de natureza diferente (*MLP* e *Naive Bayes*), foi observado um comportamento mais consistente na base do *Letterboxd* e estabilidade completa nas bases do *IMDb*. O *voting* e o *stacking* produziram resultados praticamente equivalentes, com F1-Scores variando entre *0.80* e *0.83* no *Letterboxd* e atingindo *0.90* no *IMDb*. Esses resultados demonstram que a adição de diversidade ao conjunto, mesmo que de maneira moderada, pode trazer equilíbrio às métricas de avaliação, mas não resultará em um grande ganho, visto que *ensembles* menores já continham modelos muito consistentes e dominantes.

No geral, a técnica de *SA* confirma que *ensembles* pequenos, formados por modelos com alta acurácia, já são capazes de apresentar um desempenho favorável e comparável aos *ensembles* maiores. No entanto, o ganho principal está no tempo de treinamento, pois *ensembles* com menos modelos treinam mais rapidamente e mantêm métricas competitivas. Os resultados reforçam a força dos modelos lineares na análise de sentimento.

4.4 SELEÇÃO POR ACURÁCIA E DIVERSIDADE

Na abordagem *SAD* (Seleção por Acurácia e Diversidade), temos a combinação de dois critérios: o desempenho individual dos modelos e o grau de diversidade entre as suas previsões. Essa técnica de seleção tem como objetivo escolher subconjuntos de classificadores que sejam tanto *competentes quanto diversos*. Modelos com maior diversidade podem reduzir a variância e os vieses por oferecerem perspectivas diferentes sobre os dados, já que seu funcionamento é distinto. Logo, nesta seção, serão discutidos e apresentados os resultados dos *ensembles* formados por 2, 3, 4 e 5 modelos selecionados pelo método *SAD*.

4.4.1 DESEMPENHO DOS ENSEMBLES COM 2 MODELOS

Nesta subseção, apresentam-se os resultados obtidos pelos *ensembles* compostos por dois modelos, construídos a partir da técnica de Seleção por Acurácia e Diversidade (SAD), nos quais são mostradas as métricas de desempenho por classe, permitindo identificar quais pares apresentam melhor desempenho no *ensemble*.

Tabela 33 – Métricas para Ensembles SAD 2 modelos: RandomForest, KNN (Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.56	0.91	0.69	0.82	0.37	0.51
Stacking	0.75	0.79	0.77	0.80	0.77	0.79

Tabela 34 – Métricas para Ensembles SAD 2 modelos: Random Forest e KNN (Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.57	0.88	0.69	0.79	0.42	0.55
Stacking	0.75	0.77	0.76	0.79	0.77	0.78

Tabela 35 – Métricas para Ensembles SAD 2 modelos: KNN e MLP(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.74	0.93	0.82	0.91	0.67	0.77
Stacking	0.87	0.88	0.87	0.87	0.87	0.87

Tabela 36 – Métricas para Ensembles SAD (2 modelos): MLP e LightGBM(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.84	0.93	0.89	0.93	0.83	0.87
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Nos *ensembles* com 2 modelos, pode-se observar como o método *SAD* já performa, selecionando modelos estruturalmente diferentes, como o *Random Forest* e *KNN* na base *Letterboxd* ou *MLP* e *LightGBM* na base *IMDb*. Essa seleção demonstra o papel da *diversidade*: mesmo quando o desempenho individual de alguns desses modelos não está entre os melhores, a combinação pode gerar *ensembles* mais equilibrados. No entanto, os ganhos não ocorrem em todos os casos. Na base *Letterboxd*, especialmente na versão com pré-processamento, o *Voting* apresentou um forte desequilíbrio entre as classes, enquanto o *Stacking*, por ser mais robusto funcionando com um meta-aprendiz, exibiu resultados superiores e muito mais estáveis, alcançando F1-Scores próximos de *0.79* e *0.77* para as duas versões da base. Na base *IMDb*, o impacto da diversidade foi mais consistente: pares como *MLP–KNN* ou *MLP–LightGBM* produziram *ensembles* com F1-Scores entre *0.87* e *0.90*, destacando o potencial da técnica em bases mais robustas.

4.4.2 DESEMPENHO DOS ENSEMBLES COM 3 MODELOS

Aqui, apresentam-se os resultados obtidos pelos *ensembles* compostos por três modelos, construídos a partir da técnica de Seleção por Acurácia e Diversidade (SAD), mostrando o efeito da adição de um terceiro modelo sobre a performance geral do *ensemble*.

Tabela 37 – Métricas para Ensembles SAD 3 modelos: MLP, RandomForest, KNN(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.76	0.81	0.79	0.82	0.78	0.80
Stacking	0.78	0.82	0.80	0.83	0.80	0.82

Tabela 38 – Métricas para Ensembles SAD 3 modelos: KNN, Random Forest, AdaBoost(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.76	0.76	0.76	0.79	0.79	0.79
Stacking	0.75	0.77	0.76	0.79	0.77	0.78

Tabela 39 – Métricas para Ensembles SAD 3 modelos: MLP, Random Forest, KNN(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.88	0.85	0.86	0.86	0.88	0.87
Stacking	0.89	0.88	0.89	0.88	0.89	0.89

Tabela 40 – Métricas para Ensembles SAD 3 modelos: LightGBM, RandomForest, MLP(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.87	0.89	0.88	0.90	0.89
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Adicionando mais um modelo, pôde-se perceber um comportamento mais sólido nos *ensembles*, tanto no *Voting* quanto no *Stacking*. A inclusão de um terceiro classificador aumentou a capacidade de equilíbrio entre a precisão e o *recall*, principalmente na base do *Letterboxd*. No cenário pré-processado, o *ensemble* formado por *MLP*, *Random Forest* e *KNN* apresentou F1-Scores entre *0.79* e *0.82*, enquanto na versão sem pré-processamento os resultados permaneceram próximos, mas com menos estabilidade. Na base *IMDb*, os modelos escolhidos pelo *SAD* (*MLP*, *Random Forest* e *KNN*, ou *LightGBM*, *MLP* e *Random Forest*) alcançaram um desempenho consistentemente alto, chegando a F1-Scores de *0.89* para *voting* e *0.90* para *stacking*. Isso mostra que, ao contrário da seleção apenas por acurácia, a seleção por acurácia e diversidade tende a produzir benefícios quando combina modelos de naturezas distintas.

4.4.3 DESEMPENHO DOS ENSEMBLES COM 4 MODELOS

Já nesta parte, apresentam-se os resultados obtidos pelos *ensembles* compostos por quatro modelos, construídos a partir da técnica de Seleção por Acurácia e Diversidade (SAD), mostrando como a inserção de um quarto modelo impacta a eficácia.

Tabela 41 – Métricas para Ensembles SAD 4 modelos: Regressão Logística, RandomForest, LightGBM, MLP(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.77	0.84	0.80	0.85	0.78	0.81
Stacking	0.80	0.81	0.80	0.83	0.82	0.82

Tabela 42 – Métricas para Ensembles SAD 4 modelos: KNN, Naive Bayes, AdaBoost, Decision Tree(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.73	0.82	0.77	0.82	0.73	0.78
Stacking	0.79	0.77	0.78	0.80	0.82	0.81

Tabela 43 – Métricas para Ensembles SAD 4 modelos: MLP, Linear SVM, Naive Bayes, Regressão Logística(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.89	0.90	0.90	0.90	0.88	0.89
Stacking	0.89	0.90	0.90	0.90	0.89	0.90

Tabela 44 – Métricas para Ensembles SAD 4 modelos: Regressão Logística, KNN, AdaBoost, RandomForest(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.88	0.90	0.89	0.90	0.87	0.89
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Nos *ensembles* compostos por 4 modelos, a técnica *SAD* demonstrou maior capacidade de explorar diversidade, montando um classificador que combinava diversos tipos de classificadores, como modelos lineares, modelos baseados em árvores, redes neurais e algoritmos *instance-based*. Os resultados da base de dados *Letterboxd* indicaram um aumento na estabilidade, com F1-Scores entre *0.80* e *0.82* para *voting* e *stacking*. Na versão sem pré-processamento, o conjunto mais diverso, incluindo *KNN*, *Naive Bayes*, *AdaBoost* e *Decision Tree*, mostrou que essa heterogeneidade pode compensar fraquezas individuais. Já no *IMDb*, tanto com como sem pré-processamento, os resultados permaneceram elevados, com F1-Scores entre *0.89* e *0.90*, com o *Stacking* atingindo *0.90* em todos os cenários.

Isso reforça que *ensembles* moderadamente grandes e diversificados conseguem capturar diferentes aspectos dos dados sem reduzir a consistência geral.

4.4.4 DESEMPENHO DOS ENSEMBLES COM 5 MODELOS

Por fim, nesta subseção, apresentam-se os resultados obtidos pelos *ensembles* compostos por cinco modelos, construídos a partir da técnica de Seleção por Acurácia e Diversidade (SAD), analisando como a ampliação do ensemble com um quinto modelo impacta as métricas de desempenho.

Tabela 45 – Métricas para Ensembles SAD 5 modelos: Regressão Logística, RandomForest, LightGBM, MLP, KNN(Letterboxd - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.79	0.81	0.80	0.83	0.81	0.82
Stacking	0.80	0.81	0.81	0.83	0.82	0.83

Tabela 46 – Métricas Consolidadas para Ensembles SAD 5 modelos: KNN, Naive Bayes, AdaBoost, Decision Tree, LightGBM(Letterboxd - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.78	0.77	0.78	0.80	0.81	0.80
Stacking	0.80	0.79	0.79	0.81	0.82	0.82

Tabela 47 – Métricas para Ensembles SAD 5 modelos: MLP, Linear SVM, Naive Bayes, Regressão Logística, KNN(IMDb - Sem Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.90	0.89	0.90	0.89	0.90	0.90
Stacking	0.89	0.90	0.90	0.90	0.89	0.90

Tabela 48 – Métricas para Ensembles SAD 5 modelos: Regressão Logística, KNN, AdaBoost, RandomForest, MLP(IMDb - Com Pré-Processamento)

Modelo	**Classe Negativa**			**Classe Positiva**		
	Precisão	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
Voting	0.91	0.88	0.90	0.89	0.91	0.90
Stacking	0.90	0.90	0.90	0.90	0.90	0.90

Com 5 modelos formando o *ensemble*, estes foram os que mais demonstraram estabilidade dentro da abordagem *SAD*. Na base do *Letterboxd*, tanto com pré-processamento quanto sem, o *voting* e o *stacking* alcançaram F1-Scores entre *0.79* e *0.83*, mostrando ganhos de robustez conforme mais modelos diversos eram incluídos. Na base do *IMDb*, os *ensembles* apresentaram excelentes desempenhos, com F1-Scores de aproximadamente *0.90* em todos os cenários, tanto no *voting* quanto no *stacking*. Com isso, pode-se notar que *ensembles* compostos por combinações de modelos lineares, redes neurais e modelos de árvores, como o conjunto *Regressão Logística*, *KNN*, *AdaBoost*, *Random Forest* e *MLP*, demonstraram ser equilibrados entre os valores de precisão e *recall* nas duas classes.

Os resultados do *SAD* mostram que a combinação entre acurácia e diversidade permite selecionar subconjuntos de modelos que são capazes de alcançar desempenho elevado mesmo com a inclusão de classificadores com desempenho individual inferior. A técnica se mostrou eficaz quando os *ensembles* eram formados por 3 ou mais modelos e teve um melhor desempenho quando aplicada à base de dados do *IMDb*. Com isso, podemos concluir que a diversidade desempenha um papel importante e fundamental para melhorar a estabilidade, compensar vieses e aumentar a robustez no contexto da análise de sentimentos.

A comparação entre os três tipos de seleção e as técnicas de *ensemble* mostram diferenças importantes na forma como cada estratégia reage às características das bases de dados e das técnicas de pré-processamento. Os *ensembles* sem seleção, formados por todos os 10 modelos, apresentaram um *desempenho sólido* em todos os cenários. A grande *diversidade estrutural* tornou esse *ensemble* bastante robusto, o que resultou em métricas estáveis, principalmente na base do *IMDb*, onde os F1-Scores ficaram em torno de *0.90*. Entretanto, para o *Voting*, a inclusão de tantos modelos fracos não foi benéfica, resultando em um aumento do custo computacional e do tempo de treinamento.

A seleção por acurácia (*SA*) mostrou um comportamento que já era esperado e eficiente, pois priorizou os modelos de melhor desempenho individual, principalmente os *modelos lineares*, como *SVM Linear*, *Regressão Logística* e *SGD*. Nesse cenário, os *ensembles* tornaram-se eficazes com poucos modelos em sua composição (a partir de 3), o que os torna mais rápidos no treinamento que os *ensembles* maiores. Os *ensembles* com 3 modelos já produziam resultados idênticos aos modelos com 4 e 5. Esses *ensembles* pequenos, dominados por modelos lineares, especialmente na base de dados *IMDb*, dominaram o problema de classificação.

Já na seleção por acurácia e diversidade (*SAD*), os resultados tiveram padrões distintos. Modelos com desempenho mais modestos passaram a compor o *ensemble*, aumentando a diversidade e complementando as previsões. Contudo, isso resultou em um *Voting* menos estável, especialmente na base de dados do *Letterboxd*, onde diferenças muito grandes entre os classificadores geraram decisões conflitantes. No entanto, com o *Stacking*, essa diversidade foi uma vantagem que superou o *Voting* em praticamente todos os cená-

rios e aproximou-se dos melhores resultados obtidos pela *SA*. Em bases com mais ruído, como a do *Letterboxd*, a diversidade contribuiu para o equilíbrio entre precisão e *recall*.

5 CONCLUSÕES

Este trabalho investigou o uso de técnicas de Processamento de Linguagem Natural aplicadas à *análise de sentimentos* em comentários de filmes, com ênfase na avaliação dos modelos individuais e dos *ensembles* construídos por meio da *Seleção por Acurácia* (*SA*) e *Seleção por Acurácia e Diversidade* (*SAD*). A pesquisa concentrou-se na comparação entre diferentes abordagens de *ensembles*, por meio de técnicas como *Voting* e *Stacking*, com o intuito de compreender como a junção de vários modelos pode melhorar o desempenho dos classificadores.

Os resultados deste projeto contribuem com a área ao demostrar a eficácia dos *ensembles* heterogêneos em tarefas de análise de sentimentos e fortalecendo a importância da aplicação da *diversidade* entre *ensembles*, já que os desempenhos obtidos foram satisfatórios. Tanto o *Voting* quanto o *Stacking* tiveram ganhos relevantes, com o segundo sendo capaz de obter um desempenho superior. Isso demonstra que a diversidade contribui positivamente para a robustez e a generalização dos *ensembles*.

Ainda sobre os resultados, foi mostrado que cada abordagem se destaca por um aspecto diferente: o *ensemble* com todos os modelos se revelou o mais *robusto*; os *ensembles SA* foram os mais *eficientes* e frequentemente os mais precisos, graças ao subconjunto de modelos de alto desempenho; e os *SAD*, principalmente o *Stacking*, foram os que melhor aproveitaram a diversidade para recuperar erros e lidar com conjuntos mais heterogêneos. Na base *IMDb*, pôde-se perceber que o tipo de *ensemble* ou o pré-processamento tinham pouco impacto, enquanto na *Letterboxd* a escolha da estratégia era mais perceptível. Portanto, foi definido que *ensembles* pequenos e fortes, como o caso do *SA* com 3 modelos, já eram bastante estáveis, enquanto os *ensembles SAD* ganham mais relevância quando a base é mais ruidosa.

Os objetivos que foram propostos no início do trabalho foram alcançados: foi realizada a revisão da literatura sobre análise de sentimentos e métodos de *ensemble*; foram preparados os conjuntos de dados, com a inclusão de uma nova base; foram construídos, analisados e comparados tipos de *ensembles* com base nos desempenhos dos algoritmos clássicos; foi verificado o impacto do pré-processamento; foram comparados os resultados dos diferentes modelos construídos; e foi demonstrado de forma quantitativa e fundamentada que a *Seleção por Acurácia* e a *Seleção por Acurácia e Diversidade* são estratégias eficientes para melhorar o desempenho em tarefas de análise de sentimentos.

Para finalizar, como trabalhos futuros, pode-se explorar: Outras bases de dados, visando ampliar a variedade de fontes e aumentar o número de instâncias por classe para melhorar a representatividade. Outros modelos de seleção para *ensembles*, visto que neste trabalho apenas duas técnicas foram analisadas. A realização do experimento com mais modelos, incluindo modelos de *Deep Learning* e *Large Language Models(LLM)*, para que haja uma seleção mais ampla e maior diversidade. O aumento do número de modelos no

ensemble, visando testar os métodos *SA* e *SAD* com mais de 5 componentes. A análise de outras técnicas de pré-processamento, visando um melhor entendimento e otimização para o modelo. O uso de outros métodos de *ensembles*, além do *Voting* e *Stacking* utilizados neste estudo.

REFERÊNCIAS

ABDAR, Moloud et al. Uma revisão da quantificação de incertezas em aprendizado profundo: técnicas, aplicações e desafios. **Information Fusion**, v. 76, p. 243–297, 2021.

ABDULLAH, Dakhaz Mustafa; ABDULAZEEZ, Adnan Mohsin. Machine learning applications based on SVM classification a review. **Qubahan Academic Journal**, v. 1, n. 2, p. 81–90, 2021.

AFTAB, Farhan et al. A comprehensive survey on sentiment analysis techniques. **International Journal of Technology**, IJTech, v. 14, n. 6, p. 1288–1298, 2023.

AL BATAINEH, Ali; MANACEK, Sarah. MLP-PSO hybrid algorithm for heart disease prediction. **Journal of Personalized Medicine**, MDPI, v. 12, n. 8, p. 1208, 2022.

ALMEIDA NETO, José A. de; DE MELO, Tiago. Exploring Supervised Learning Models for Multi-Label Text Classification in Brazilian Restaurant Reviews. In: ANAIS do XX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC). Belo Horizonte/MG: Sociedade Brasileira de Computação, 2023. p. 126–140. DOI: 10.5753/eniac.2023.233843.

ALPAYDIN, Ethem. **Machine learning**. [S.l.]: MIT press, 2021.

AMINI, Saeid et al. Urban land use and land cover change analysis using random forest classification of landsat time series. **Remote Sensing**, v. 14, n. 11, p. 2654, 2022.

AMIRI, Ahmed Faris et al. Faults detection and diagnosis of PV systems based on machine learning approach using random forest classifier. **Energy Conversion and Management**, v. 301, p. 118076, 2024.

ANDRADE, Juliana de; REBS, Rebeca Recuero. **A estética do consumo de filmes na plataforma Letterboxd**. [S.l.*:* s.n.], 2022. Trabalho apresentado em evento ou artigo sem informação completa.

AVELINO JÚNIOR, Juscelino Sebastião. **Uma abordagem de seleção dinâmica de classificadores para predição de defeitos de software**. 2022. Dissertação (Mestrado em Ciência da Computação).

BRANDÃO, Michele A. et al. Impacto do pré-processamento e representação textual na classificação de documentos de licitações. In: SIMPÓSIO Brasileiro de Banco de Dados (SBBD). [S.l.]: SBC, 2023. p. 102–114.

BREIMAN, Leo. Bagging predictors. **Machine learning**, v. 24, p. 123–140, 1996.

BRITTO, Larissa; PACÍFICO, Luciano. Análise de sentimentos para revisoes de aplicativos mobile em português brasileiro. In: ANAIS do XVI Encontro Nacional de Inteligência Artificial e Computacional. [S.l.]: SBC, 2019.

CHARBUTY, Bahzad; ABDULAZEEZ, Adnan. Classification based on decision tree algorithm for machine learning. **Journal of applied science and technology trends**, Journal of Applied Science e Technology Trends, v. 2, n. 01, p. 20–28, 2021.

CHEN, Hong et al. Improved naive Bayes classification algorithm for traffic risk management. **EURASIP Journal on Advances in Signal Processing**, v. 2021, n. 1, p. 30, 2021.

DANG, Nhan Cach; MORENO-GARCÍA, Maria N.; DE LA PRIETA, Fernando. Análise de sentimento baseada em aprendizagem profunda: um estudo comparativo. **Eletrônica**, v. 3, p. 483, 2020.

DIETTERICH, Thomas G. Ensemble methods in machine learning. In: SPRINGER. INTERNATIONAL workshop on multiple classifier systems. [S.l.*:* s.n.], 2000. p. 1–15.

DING, Yi et al. An efficient AdaBoost algorithm with the multiple thresholds classification. **Applied Sciences**, v. 12, n. 12, p. 5872, 2022.

DONG, Xibin et al. A survey on ensemble learning. **Frontiers of Computer Science**, v. 14, p. 241–258, 2020.

DWIRAMADHAN, Farhan; WAHYUDDIN, Mohammad Iwan; HIDAYATULLAH, Deny. Sistem Pakar Diagnosa Penyakit Kulit Kucing Menggunakan Metode Naive Bayes Berbasis Web. **J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)**, v. 6, n. 3, p. 429–437, 2022. Apesar do formato de artigo, incluído como @ARTICLE para manter consistência.

FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, p. 1189–1232, 2001.

GALKE, Lukas et al. Are we really making much progress in text classification? A comparative review. **arXiv preprint arXiv:2204.03954**, 2022.

GUPTA, Ishu et al. **PCA-RF: an efficient Parkinson's disease prediction model based on random forest classification**. [S.l.], 2022.

HAJIHOSSEINLOU, Mahsa; MAGHSOUDI, Abbas; GHEZELBASH, Reza. A novel scheme for mapping of MVT-type Pb–Zn prospectivity: LightGBM, a highly efficient gradient boosting decision tree machine learning algorithm. **Natural Resources Research**, v. 32, n. 6, p. 2417–2438, 2023.

HALDER, Rajib Kumar; AL., et. Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. **Journal of Big Data**, Springer, v. 11, n. 1, p. 113, 2024.

HASSAN, Sayar Ul; AHAMED, Jameel; AHMAD, Khaleel. Analytics of machine learning-based algorithms for text classification. **Sustainable operations and computers**, Elsevier, v. 3, p. 238–248, 2022.

IMANI, Mehdi; ARABNIA, Hamid Reza. Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis. **Technologies**, v. 11, n. 6, 2023. ISSN 2227-7080. DOI: 10.3390/technologies11060167. Disponível em: https://www.mdpi.com/2227-7080/11/6/167.

IMANI, Mehdi; ARABNIA, Hamid Reza. Hyperparameter optimization and combined data sampling techniques in machine learning for customer churn prediction: a comparative analysis. **Technologies**, MDPI, v. 11, n. 6, p. 167, 2023.

JUREK, Anna et al. A survey of commonly used ensemble-based classification techniques. **The Knowledge Engineering Review**, Cambridge University Press, v. 29, n. 5, p. 551–581, 2014.

KAUR, Jashanjot; BUTTAR, P. Kaur. A systematic review on stopword removal algorithms. **International Journal on Future Revolution in Computer Science & Communication Engineering**, v. 4, n. 4, p. 207–210, 2018.

KAZMAIER, Jacqueline; VAN VUUREN, Jan H. The power of ensemble learning in sentiment analysis. **Expert Systems with Applications**, v. 187, p. 115819, 2022.

KUCUK, Ekrem et al. Comparative analysis of machine learning algorithms for biomedical text document classification: A case study on cancer-related publications. **Medicine Science**, v. 13, n. 1, 2024.

KUNCHEVA, Ludmila I. Classifier ensembles for changing environments. In: SPRINGER. INTERNATIONAL workshop on multiple classifier systems. [S.l.*:* s.n.], 2004. p. 1–15.

LI, Shaojie et al. Utilizing the LightGBM algorithm for operator user credit assessment research. **arXiv preprint arXiv:2403.14483**, 2024.

MANASTARLA, Alberto. **Otimização da seleção dinâmica de ensemble em classificação: integrando seleção de protótipos e metaclassificadores**. 2024. Tese (Doutorado).

MOHAMMED, Ammar; KORA, Rania. A comprehensive review on ensemble deep learning: Opportunities and challenges. **Journal of King Saud University-Computer and Information Sciences**, v. 35, n. 2, p. 757–774, 2023.

NASKATH, J.; SIVAKAMASUNDARI, G.; BEGUM, A. Alif Siddiqua. A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN. **Wireless Personal Communications**, Springer, v. 128, n. 4, p. 2913–2936, 2023.

OJEDA, Daniel Zonta; ZALEWSKI, Willian; MALETZKE, André Gustavo. Utilizando a quantificação na análise de sentimentos em reviews de produtos. In: SBC. ESCOLA Regional de Banco de Dados (ERBD). [S.l.*:* s.n.], 2024. p. 71–80.

ONAN, Aytuğ. Arquitetura de rede neural recorrente convolucional bidirecional com mecanismo de aprimoramento por grupo para classificação de sentimentos em texto. **Journal of King Saud University-Computer and Information Sciences**, v. 34, n. 5, p. 2098–2117, 2022.

PAES, Vinícius J. et al. Análise de sentimento em tweets relacionados ao desmatamento da floresta amazônica. In: BRAZILIAN Workshop on Social Network Analysis and Mining (BraSNAM). [S.l.]: SBC, 2022. p. 61–72.

PAULA, Hildon Eduardo Lima de. Quantificando a importância de emojis e emoticons para a identificação de polaridade. In: Trabalho/Artigo sem informação completa.

PEREIRA, Denilson Alves. A survey of sentiment analysis in the Portuguese language. **Artificial Intelligence Review**, v. 54, n. 2, p. 1087–1115, 2021.

PINHO, Cintia et al. Aplicação de técnicas de inteligência artificial para classificação de fuga ao tema em redações. **Educação em Revista**, v. 40, 2024.

PURWANTO, Anang Dwi et al. Decision tree and random forest classification algorithms for mangrove forest mapping in Sembilang National Park, Indonesia. **Remote Sensing**, v. 15, n. 1, p. 16, 2022.

RAMAKRISHNA, Mahesh Thyluru et al. Homogeneous adaboost ensemble machine learning algorithms with reduced entropy on balanced data. **Entropy**, v. 25, n. 2, p. 245, 2023.

SANTOS, Daiana Kathrin Santana; BERTON, Lilian. Analysis of Twitter users' sentiments about the first round 2022 presidential election in Brazil. In: ENCONTRO Nacional de Inteligência Artificial e Computacional (ENIAC). [S.l.]: SBC, 2023. p. 880–893.

SCIKIT-LEARN DEVELOPERS. **sklearn.linear_model.SGDClassifier — scikit-learn documentation**. [S.l.: s.n.], 2025. `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html`. Acesso em: 20 nov. 2025.

SILVA, Lincoln Wallace Valentim da Costa et al. Análise de Sentimentos: Impacto da Tradução Neural na Avaliação de Desempenho. Universidade Federal da Paraíba, 2023.

SIRQUEIRA, Eutino Júnior Vieira; VIDAL, Flávio de Barros. Evaluation of Named Entity Recognition using Ensemble in Transformers Models for Brazilian Public Texts. In: ANAIS do XXI Encontro Nacional de Inteligência Artificial e Computacional (ENIAC). Belém/PA: Sociedade Brasileira de Computação, 2024. p. 966–977. DOI: `10.5753/eniac.2024.245227`.

SOUSA, Thiago Fernandes de. **Combinação de classificadores para sistema de automated fact checking**. 2020. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife.

SOUZA, Ellen et al. Assessing the impact of stemming algorithms applied to brazilian legislative documents retrieval. In: ANAIS do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. [S.l.]: SBC, 2021.

SRINIVAS, Polipireddy; KATARYA, Rahul. hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost. **Biomedical Signal Processing and Control**, v. 73, p. 103456, 2022. ISSN 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.103456. Disponível em: https://www.sciencedirect.com/science/article/pii/S1746809421010533.

SUN, Zhigang et al. An improved random forest based on the classification accuracy and correlation measurement of decision trees. **Expert Systems with Applications**, v. 237, p. 121549, 2024.

TAUIL, Yasser Bulaty. **Avaliação de algoritmos de aprendizado de máquina e modelo explicativo para a análise da satisfação quanto à economia e seus principais fatores de contribuição**. 2024. B.S. thesis – Universidade Tecnológica Federal do Paraná.

UDDIN, Shahadat; AL., et. Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. **Scientific Reports**, Nature Publishing Group, v. 12, n. 1, p. 6256, 2022.

WICKRAMASINGHE, Indika; KALUTARAGE, Harsha. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. **Soft Computing**, v. 25, n. 3, p. 2277–2293, 2021.

WOLPERT, David H. Stacked generalization. **Neural networks**, v. 5, n. 2, p. 241–259, 1992.

YANG, Liying. Classifiers selection for ensemble learning based on accuracy and diversity. **Procedia Engineering**, Elsevier, v. 15, p. 4266–4270, 2011.

ZABOR, Emily C. et al. Logistic Regression in Clinical Studies. **International Journal of Radiation Oncology*Biology*Physics**, v. 112, n. 2, p. 271–277, 2022. ISSN 0360-3016. DOI: https://doi.org/10.1016/j.ijrobp.2021.08.007. Disponível em: https://www.sciencedirect.com/science/article/pii/S0360301621026468.

ZHOU, Zhi-Hua. **Machine learning**. [S.l.]: Springer nature, 2021.