PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

CLASSIFICAÇÃO DE SENTIMENTOS USANDO MÁQUINA DE VETOR DE SUPORTE (SVM)

IGOR FELIPE JARDIM DA SILVA

GOIÂNIA
2025

IGOR FELIPE JARDIM DA SILVA

CLASSIFICAÇÃO DE SENTIMENTOS USANDO MÁQUINA DE VETOR DE SUPORTE (SVM)

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador:

Prof. Dr. Clarimar José Coelho

GOIÂNIA

2025

IGOR FELIPE JARDIM DA SILVA

CLASSIFICAÇÃO DE SENTIMENTOS UTILIZANDO MÁQUINA DE VETOR DE SUPORTE (SVM)

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia da Computação.

Aprovado em 03 de Dezembro de 2025

BANCA EXAMINADORA

_____

Prof. Dr. Clarimar José Coelho
Orientador/PUC Goiás

_____

Professor
Prof. Me. Daniel Corrêa da Silva/PUC Goiás

_____

Professor
Prof. Dr. Ernesto Fonseca Veiga/PUC Goiás

A Deus, amigos e às mulheres da minha vida,

Jane e Cintya.

AGRADECIMENTOS

Agradeço a toda a minha família, especialmente minhas mães e minha avó Terezinha por todo o suporte e paciência durante todos esses anos de graduação. Não foi fácil e nem rápido, mas chegamos até aqui.

Aos colegas de curso, a quem pude me apoiar e ser apoio durante tantos momentos. Passamos por muitas dificuldades, mas vencemos.

Aos professores da escola politécnica da PUC-Goiás. Sou grato por todo ensinamento compartilhado ao longo desses anos.

Agradeço ao professor Clarimar J Coelho, que aceitou me orientar ao longo de todo o projeto e me ajudou com seus conselhos. Obrigado pelo apoio.

À Hemilyne, que em tão pouco tempo demonstrou ser um pilar de apoio muito importante. Obrigado por confiar em mim e me dar forças quando necessário durante a reta final.

Às comunidades Canto de Maria e Cristo Alegria, que me acompanharam ao longo desses anos e sempre me mostraram o valor de ser um missionário e como agir nessa terra. Obrigado por me mostrarem sempre a melhor parte de mim.

Por fim, obrigado a você que está lendo este trabalho. Agradeço o interesse e seu tempo.

RESUMO

A análise de sentimentos é uma área em expansão no campo do Processamento de Linguagem Natural (PLN), permitindo compreender emoções e opiniões expressas em textos. Este trabalho tem como objetivo desenvolver um modelo de classificação de sentimentos capaz de identificar automaticamente o tom positivo, negativo ou neutro em textos curtos. Para isso, utiliza-se a técnica de Máquina de Vetor de Suporte (SVM), aliada à vetorização TF-IDF, aplicada sobre uma base de dados pública de mensagens do Twitter. O modelo foi treinado, comparados com outros modelos de classificação como Naive Bayes e Regressão Logística e avaliado rigorosamente com métricas de desempenho como acurácia, precisão, *recall* e *F1-score*, a fim de validar sua eficácia. Os resultados obtidos demonstraram que o método apresenta excelente desempenho na análise de textos curtos, atingindo uma acurácia de 92%, evidenciando o potencial do SVM como ferramenta de apoio em aplicações de mineração de opinião e monitoramento de redes sociais

Palavras-chave: Análise de sentimentos. Processamento de linguagem natural. Máquina de vetor de suporte. TF-IDF. Classificação de texto.

ABSTRACT

Sentiment analysis is an expanding field within Natural Language Processing (NLP), enabling the understanding of emotions and opinions expressed in text. This work aims to develop a sentiment classification model capable of automatically identifying positive, negative, or neutral tones in short texts. To achieve this, the Support Vector Machine (SVM) technique is employed together with TF-IDF vectorization, applied to a public dataset of Twitter messages. The model was trained, compared with other classification models such as Naive Bayes and Logistic Regression, and rigorously evaluated using performance metrics such as accuracy, precision, recall, and F1-score in order to validate its effectiveness. The results obtained demonstrate that the method shows excellent performance in analyzing short texts, achieving an accuracy of 92%, highlighting the potential of SVM as a supporting tool for opinion mining and social media monitoring applications.

Keywords: Sentiment analysis. Natural language processing. Support vector machine. TF-IDF. Text classification.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

AUC – *Area Under Curve*

FN – *False Negative* (Falso Negativo)

FP – *False Positive* (Falso Positivo)

NLP – *Natural Language Processing*

RBF – *Radial Basis Function*

RegEx – *Regular Expressions*

ROC – *Receiver Operating Characteristic*

SVM – *Support Vector Machine*

TF-IDF – *Term Frequency – Inverse Document Frequency*

TN – *True Negative* (Verdadeiro Negativo)

TP – *True Positive* (Verdadeiro Positivo)

URL – *Uniform Resource Locator*

SUMÁRIO

1. INTRODUÇÃO

O avanço contínuo e a popularização da Análise de Sentimentos (também conhecida como Mineração de Opiniões) representam um dos campos de pesquisa mais relevantes no Processamento de Linguagem Natural (PLN) atualmente. Esta linha de pesquisa visa extrair de textos uma atitude, emoção ou sentimento em relação a um tópico, extração essa realizada por meio de técnicas e modelos que classificam o texto em classes como "positivo", "negativo" e "neutro" (Ricci, 2020). Este trabalho, focado na Classificação de Sentimentos usando Máquina de Vetor de Suporte (SVM), explora a capacidade de sistemas automáticos identificarem e categorizarem atitudes, emoções e opiniões expressas em textos.

A motivação para o desenvolvimento deste estudo decorre da crescente quantidade de informações produzidas diariamente na internet. Redes sociais como Twitter, Facebook e Instagram reúnem bilhões de textos que refletem percepções individuais e coletivas sobre diversos acontecimentos. Diante desse grande volume de dados, torna-se inviável a análise manual, surgindo a necessidade de ferramentas automáticas que auxiliem na identificação rápida e precisa dos sentimentos presentes nessas mensagens. Técnicas de classificação automática são, portanto, essenciais para a sociedade contemporânea, permitindo desde o monitoramento de políticas públicas até a análise de satisfação de clientes e prevenção de crises institucionais, áreas em que modelos robustos como o SVM apresentam significativa utilidade. Além disso, pesquisas nesse domínio contribuem para ampliar a compreensão científica sobre linguística computacional, aprendizado de máquina e processamento massivo de dados textuais.

A antiga rede social Twitter (atualmente X) é o ambiente escolhido para a coleta e análise dos dados. Sua seleção é justificada pelo alto volume de informações atualizadas em tempo real e pela facilidade de acesso aos dados por meio de APIs ou bases disponibilizadas em pesquisas anteriores (Camelo, 2017). Essa característica de grande fonte de dados curtos e atuais é destacada por Silva (2016), que identifica a rede como uma das fontes mais ricas para estudos de mineração de opinião.

Cortes e Vapnik (1995) introduziram as Máquinas de Vetor de Suporte, que se destacaram por sua forma robusta de lidar com dados de alta dimensionalidade, muito presentes em representações textuais. Historicamente, os primeiros estudos

estruturados em análise de sentimentos remontam ao início dos anos 2000, impulsionados pela popularização da internet e pela necessidade de interpretar opiniões de usuários em ambientes digitais. Trabalhos como os de Pang e Lee (2002) são considerados marcos na área, apresentando metodologias que inspiraram grande parte das pesquisas posteriores. No contexto brasileiro, pesquisas como as de Silva (2016), Corrêa (2017), Araújo et al. (2013) e Bordin Junior (2018) investigaram diferentes abordagens para classificação de sentimentos, utilizando desde métodos probabilísticos, como Naive Bayes, até modelos lineares mais robustos, como Regressão Logística e SVM. Em paralelo, contribuições teóricas importantes sobre técnicas de aprendizagem, como as apresentadas por Bishop (2006) e Shalev-Shwartz e Ben-David (2014), consolidaram o SVM como uma das ferramentas mais eficientes para problemas de alta dimensionalidade. Uma das principais dificuldades observadas nesses estudos envolve o pré-processamento de textos curtos, a grande variedade linguística presente em redes sociais e o desbalanceamento entre classes de sentimentos, desafios que este trabalho também enfrentou e abordou em sua metodologia. No meio de NLP, combinações entre SVM e representações vetoriais como TF-IDF (*Term Frequency - Inverse Document Frequency*) têm apresentado resultados expressivos na classificação de sentimentos e detecção de spam (SCIKIT-LEARN, 2024).

O presente trabalho propõe o desenvolvimento de um modelo de classificação de sentimentos utilizando o algoritmo de Máquinas de Vetor de Suporte (SVM), comparando-o com outras abordagens tradicionais, como Naive Bayes (nas versões Multinomial e Complement) e Regressão Logística. Para isso, realiza-se um fluxo completo de NLP, envolvendo coleta, organização, pré-processamento, vetorização via TF-IDF, treinamento supervisionado e avaliação por meio de métricas amplamente reconhecidas, como acurácia, precisão, recall, F1-score, matriz de confusão e curva ROC (*Receiver Operating Characteristic*). O uso do SVM apresenta vantagens importantes em relação a métodos probabilísticos, principalmente ao lidar com dados esparsos gerados pela vetorização TF-IDF, característica típica de tarefas textuais. Em testes conduzidos neste trabalho, o SVM demonstrou desempenho superior na maioria das métricas, confirmando resultados já observados na literatura e evidenciando sua adequação para problemas de classificação de sentimentos.

Com base nessas etapas, este trabalho busca demonstrar a aplicabilidade e o desempenho do modelo SVM na classificação de sentimentos em textos de redes

sociais, contribuindo para o avanço de pesquisas em Processamento de Linguagem Natural e mineração de opinião.

2. MATERIAL E MÉTODOS

O objetivo do trabalho é desenvolver um modelo de classificação de emoções utilizando o algoritmo SVM e avaliar sua superioridade em relação a outros modelos de classificação como Naive Bayes e Regressão Logística, aplicando conceitos de NLP para que os modelos consigam lidar da melhor forma possível com os dados propostos.

2.1 APRENDIZADO DE MÁQUINA

A Aprendizagem de Máquina (*Machine Learning*) é uma área da Inteligência Artificial focada no desenvolvimento de algoritmos que podem identificar padrões e realizar previsões com base em dados. De acordo com Bishop (2006), o processo de aprendizado ocorre por meio da otimização de funções matemáticas que ajustam o comportamento do modelo a partir de exemplos rotulados.

Shalev-Shwartz e Ben-david (2014) dizem que o aprendizado de máquina pode ser compreendido como a construção de funções de aproximação que representam relações entre entradas e saídas, sendo essa relação refinada por meio da generalização sobre novos conjuntos de dados.

Existem aprendizagens supervisionadas, não supervisionadas e por reforço. A aprendizagem pode ser considerada supervisionada quando os dados possuem rótulos de saída conhecidos; não supervisionada quando busca padrões sem rótulos; e por reforço quando o aprendizado se dá pela interação com um ambiente. Neste trabalho, a aprendizagem pode ser classificada como supervisionada, pois, cada texto contido na base de dados contém uma classe associada de sentimento (positivo, negativo ou neutro).

Para classificações de texto, como análise de sentimentos de redes sociais, é uma abordagem amplamente utilizada (Silva, 2016).

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL (NLP)

O Processamento de Linguagem Natural (do inglês, *Natural Language Processing* - NLP) é uma área da inteligência artificial que estuda como fazer com que os computadores compreendam, interpretem e produzam linguagem humana de

forma automática. O NLP combina princípios de linguística e ciência da computação para transformar textos não estruturados em representações que possam ser processadas por algoritmos matemáticos (Bird, Klein e Looper, 2009).

Segundo Silva (2016), o NLP tem papel fundamental em atividades que envolvam análise de sentimentos, traduções automáticas, reconhecimento de entidades nomeadas e classificação de texto, devido à atuação entre a linguagem humana e o processamento computacional. Aplicações desses tipos envolvem processos de tokenização, remoção de *stopwords*, lematização, análise sintática e vetorização. Técnicas essas que foram aplicadas neste trabalho.

Os maiores desafios do NLP são associados a ambiguidades como ironias, sarcasmos, abreviações e variações regionais; probabilísticas, raciocínios sobre o mundo. Em textos curtos, como os do Twitter, esses desafios são ainda mais evidentes (Araújo, Gonçalves e Benevenuto, 2013). Para lidar com essas dificuldades, técnicas de representação textual, como TF-IDF e *Word Embeddings*, podem ser utilizadas de forma que permitam o modelo capturar informações semânticas relevantes.

Neste projeto, o NLP foi aplicado para pré-processar, limpar e representar os textos antes do treinamento do modelo de classificação, garantindo que as mensagens fossem convertidas em dados numéricos compreensíveis pelo SVM.

2.3 ANÁLISE DE SENTIMENTOS

A Análise de Sentimentos é uma subárea do Processamento de Linguagem Natural (NLP) que tem como objetivo identificar emoções, opiniões ou julgamentos expressos em textos. Segundo Araújo, Gonçalves e Benevenuto (2013), esse campo busca quantificar e compreender reações humanas diante de produtos, eventos e marcas, utilizando grandes volumes de dados provenientes da internet.

A rede social Twitter (X) é uma das fontes mais populares para esse tipo de pesquisa devido ao grande número de mensagens publicadas diariamente e à natureza pública dos tweets (Aslam, 2020). Como afirmam Corrêa (2017) e Silva (2016), a concisão dos textos do Twitter facilita a extração de sentimentos, apresentando desafios como o uso de gírias, abreviações e ironias.

A análise de sentimentos é, portanto, uma ferramenta estratégica para empresas, instituições e pesquisadores que buscam compreender tendências sociais

e a percepção coletiva de determinados temas (Gonçalves, Benevenuto e Almeida, 2013).

Como exemplo de funcionamento dessa teoria, serão usadas três frases, a primeira; "Eu amo estudar de manhã", a segunda; "Eu estudo de manhã" e a terceira "Eu odeio estudar de manhã". A primeira frase recebe uma classificação "positiva", a segunda "neutra" e a terceira "negativa". Isso acontece devido a interpretação da frase e o impacto das palavras nelas. A palavra "adoro" faz com que o texto se aproxime do cunho "positivo", já "odeio" tem o efeito inverso fazendo com que a frase se aproxime mais da classificação "negativa". Com os extremos de classificação definidos, pode-se analisar que a segunda frase se aproxima da classificação "neutra" pois não contém palavras que possam direcionar o sentido do texto para "positivo" ou "negativo".

2.4 BASE DE DADOS

A base de dados é um componente essencial para qualquer projeto de aprendizado de máquina. Bordin Junior (2018) diz que a qualidade dos dados influencia diretamente o desempenho dos modelos, sendo fundamental garantir sua representatividade e consistência.

Neste trabalho, foi utilizado um conjunto de dados públicos extraídos do Twitter denominado *Twitter Sentiment Analysis*[1]. Esses dados contém rótulos definidos de acordo com o sentido predominante do texto (positivo, negativo, neutro e irrelevante).

A base contém dois arquivos .csv (*twitter_training.csv* e *twitter_validation.csv*), um para treino e outro para teste que, juntos, somam mais de 70 mil registros com identificador, texto e rótulo. Esses arquivos foram unificados com a finalidade de formar uma base de dados mais robusta, semelhante ao tratamento descrito por Araújo et al. (2013) em estudos sobre sentimentos em redes sociais.

Além disso, realizou-se uma padronização da coluna *sentiment*, convertendo a classe “*Irrelevant*” para “*Neutral*”, conforme prática comum em trabalhos como Silva (2016) e Corrêa (2017), onde sentimentos irrelevantes tendem a ser agrupados na categoria neutra.

[1] Base de dados disponível em: https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis/data. Acesso em: 20 de março de 2025.

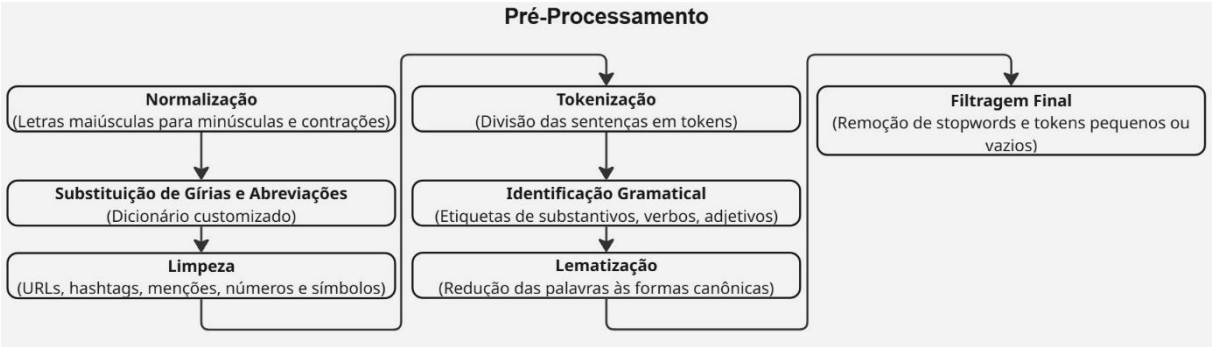
2.5 PRÉ-PROCESSAMENTO

O pré-processamento textual é uma etapa de extrema importância em análises de NLP, tendo a função de remover ruídos e padronizar os textos para o processo de vetorização que ocorrerá posteriormente. Bird, Klein e Loper (2009) descrevem o pré-processamento como o processo de transformar linguagem natural em uma forma estruturada e compreensível para algoritmos computacionais. Após todas essas etapas, os tokens processados são unidos novamente em uma *string*, formando uma coluna final *cleaned_text*, que será utilizada na etapa de vetorização.

Neste trabalho, o pré-processamento foi implementado em múltiplas etapas: normalização de texto (conversão para letras minúsculas e expansão de contrações), substituição de gírias e abreviações, remoção de URLs, hashtags, menções (@usuário), números, símbolos, palavras vazias (*stopwords)* e palavras curtas, tokenização e lematização. A Figura 1 mostra o fluxo de toda a etapa de pré-processamento.

Essas operações seguem boas práticas documentadas por Silva (2016) e Corrêa (2017), permitindo que o texto seja representado numericamente sem perda significativa de informação semântica.

Figura 1 – Fluxograma do funcionamento do pré-processamento



Fonte: Própria autoria

Ao fim da etapa de pré-processamento, toda a base de dados deve estar formatada e otimizada para iniciar o processo de vetorização.

2.5.1 Normalização inicial do texto

A primeira etapa consiste na padronização de todos os caracteres para minúsculas. Essa etapa é fundamental para evitar duplicidade representacional de tokens como “*Good*”, “*good*” e “*GOOD*”.

2.5.2 Expansão de contrações

Um desafio recorrente em textos informais em inglês é o uso de contrações (ex.: *don't*, *isn't*, *I'm*). A biblioteca *contractions* foi utilizada para expandir automaticamente esses casos, transformando, por exemplo "*don't*" em "*do not*" ou "*I'm*" em "*I am*". Essa expansão melhora a lematização feita posteriormente e permite que palavras gramaticalmente importantes sejam preservadas.

2.5.3 Substituição de gírias e abreviações

Por se tratar de textos de redes sociais, é comum encontrar gírias do próprio ambiente digital como "*u*", que significa "*you*"; "*idk*", que significa "*I don't know*"; "*btw*", que significa "*by the way*". Por isso, foi criado manualmente um dicionário de gírias para fazer a substituição e padronização dos termos. A Tabela 1 mostra os termos utilizados para esse processo. Estudos como Silva (2016) e Araújo et al. (2013) destacam o impacto positivo dessa estratégia em tarefas de classificação de tweets.

Tabela 1 – Dicionário de Gírias

Dicionário de Gírias	
Gíria/Abreviação	**Termo Padronizado**
u	you
ur	your
pls	please
plz	please
thx	thanks
imo	in my opinion
idk	i don’t know

btw	by the way
omg	oh my god
lol	laughing
lmao	laughing
rofl	laughing
tbh	to be honest
bc	because
bcz	because
brb	be right back
gr8	great
smh	shaking my head

Fonte: Própria autoria

2.5.4 Remoção de ruídos

Utilizando expressões regulares (RegEx), foram removidos URLs (http://..., www...); menções como "@usuário"; hashtags (#topic); números isolados; caracteres especiais como emojis e símbolos não ASCII; sequências repetidas de caracteres não alfabéticos.

O uso de regex permite filtrar padrões complexos de forma eficiente, e é amplamente usado em pré-processamento textual (BIRD; KLEIN; LOPER, 2009).

2.5.5 Tokenização

A tokenização é o processo de dividir um texto contínuo em unidades menores denominadas tokens, que geralmente correspondem a palavras, pontuações ou símbolos significativos. Entretanto, a tokenização em tweets é significativamente mais complexa do que em textos formais, como artigos ou livros. Isso ocorre porque tweets apresentam características peculiares, como uso de abreviações e siglas frequentemente; onomatopeias e repetições de caracteres; emojis e emoticons; URLs encurtadas; hashtags e menções; ausência de pontuação correta; uso de gírias da internet.

Devido a esses desafios, tokenizadores tradicionais, como *word_tokenize*, da biblioteca *NLKT* do *Python*, não foram projetados para lidar adequadamente com

essas características, podendo quebrar tokens incorretamente ou ignorar elementos importantes para a análise de sentimento (BIRD; KLEIN; LOPER, 2009). Para contornar esses problemas, este trabalho utiliza o *TweetTokenizer*, desenvolvido especificamente para a tokenização de conteúdos provenientes do Twitter.

Figura 2 – Exemplo de sentença separada em tokens

Fonte: Guilherme de Oliveira (2024).[2]

2.5.6 Identificação de classes gramaticais e lematização

Em seguida, aplicou-se uma etiqueta gramatical a cada token: substantivo, verbo, adjetivo, advérbio etc. Esse processo é fundamental para a etapa de lematização, pois o significado de uma palavra só pode ser reduzido corretamente à sua forma canônica quando a classe gramatical é conhecida. Por exemplo: "*running*" como *VERB* "*run*" ou "*better*" como *ADJ* "*good*".

A lematização é responsável por fazer essas reduções das palavras às suas formas base (*lemma*). Diferente da *stemmização*, a lematização preserva significado semântico, sendo mais adequada para análise de sentimentos (SILVA, 2016; FREIRE, 2023). Por exemplo: a sentença “Os gatos estão caçando ratos no jardim” após o processo de lematização é escrita como “o gato estar caçar rato em o jardim”.

[2] NLP para iniciantes: Introdução à Tokenização Básica. Disponível em: https://medium.com/@guilherme.davedovicz/nlp-para-iniciantes-introdução-à-tokenização-básica-fbba9de852e8 Acesso em: 19 de novembro de 2025.

2.5.7 Remoção de *stopwords* e filtragem final dos tokens

Utilizando o corpus de *stopwords* da *NLTK*, foram removidas palavras extremamente comuns da língua inglesa, como: "*the*", "is*",* "*at*", "*from*", "*with*". Essas palavras geralmente não contribuem para distinguir sentimentos e podem aumentar ruídos na vetorização TF-IDF.

Também foram removidos tokens com menos de três caracteres, tokens compostos apenas por símbolos e tokens vazios gerados pelo processo de limpeza. Essa filtragem reduz termos irrelevantes, tornando a matriz TF-IDF mais compacta e eficiente.

2.6 TF-IDF

O TF-IDF (do inglês, *Term Frequency - Inverse Document Frequency*) é a representação numérica de textos. Os algoritmos de aprendizado de máquina não operam diretamente sobre as palavras e sim sobre valores numéricos. Uma das técnicas mais utilizadas para essa finalidade é o TF-IDF, introduzida originalmente por Salton e McGill (1983) e bastante utilizada em estudos posteriores, como os de Bird, Klein e Loper (2009) e Silva (2016).

Essa representação combina duas medidas:

- TF (*Term Frequency*): mede a frequência com que um termo é repetido dentro de um documento. Quanto mais vezes aparece, maior seu peso relativo naquele texto.
- IDF (*Inverse Document Frequency*): avalia a importância global do termo dentro de todo o *corpus*, atribuindo menor peso a palavras muito comuns (como "*the*", "*and*", "de") e maior peso a termos mais raros e informativos.

O valor do TF-IDF é computado pela equação:

$$TFIDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right) \quad (1)$$

onde N é o número total de frases no documento e DF(t) é o número de frases que contêm o termo t.

Dessa forma, palavras raras no conjunto total recebam pesos mais altos, o que permite diferenciar documentos com base em vocabulários específicos.

Bishop (2006) diz que a utilização de representações ponderadas como o TF-IDF reduz a influência de ruídos e melhora o desempenho de classificadores lineares como o SVM.

2.7 NAIVE BAYES

Os classificadores do tipo Naive Bayes estão entre os métodos mais tradicionais e amplamente utilizados em tarefas de classificação de textos, especialmente devido à sua simplicidade, rapidez computacional e desempenho satisfatório em cenários com grandes volumes de dados. Seu princípio é baseado no Teorema de Bayes, assumindo que os atributos são condicionalmente independentes entre si dado a classe, uma hipótese “ingênua”, mas que funciona bem em muitos problemas de NLP (BIRD; KLEIN; LOPER, 2009; GONÇALVES, 2020). No contexto da classificação de sentimentos, dois modelos se destacam: Multinomial Naive Bayes e Complement Naive Bayes.

O MultinomialNB é um dos algoritmos mais utilizados para dados textuais, pois considera a frequência das palavras em cada documento. A partir da vetorização TF-IDF ou Bag-of-Words, o modelo estima a probabilidade de um documento pertencer a uma classe com base na distribuição das palavras associadas a cada rótulo. Trabalhos como os de Araújo et al. (2013) e Silva (2016) demonstram que, apesar de simples, o MultinomialNB apresenta resultados competitivos na detecção de sentimentos, principalmente quando o texto contém padrões bem distribuídos entre as classes.

Já o ComplementNB foi proposto como uma alternativa mais estável em cenários com fortes desbalanceamentos de classes. Em vez de estimar a probabilidade diretamente sobre os exemplos da própria classe, ele utiliza as palavras presentes **no conjunto complementar** de cada classe, o que reduz a sensibilidade a atribuições equivocadas de probabilidade (RENDELL et al., 1989; BORDIN JUNIOR, 2018). Essa característica o torna particularmente útil em bases nas quais uma ou mais classes são mais numerosas, como frequentemente ocorre em bases de tweets.

Ambos os modelos foram utilizados neste trabalho para fins comparativos, permitindo avaliar como algoritmos probabilísticos se comportam em relação a métodos lineares e marginais.

2.8 REGRESSÃO LOGÍSTICA

A Regressão Logística é um método estatístico amplamente utilizado para problemas de classificação binária e multiclasse, sendo um dos modelos lineares mais consolidados tanto academicamente quanto na prática. Embora seu nome sugira uma técnica de regressão, o modelo utiliza uma função logística para estimar a probabilidade de um dado pertencer a cada categoria, produzindo uma fronteira de decisão linear no espaço das características (FIGUEIRA, 2006).

Em NLP, especialmente em tarefas como análise de sentimentos, a Regressão Logística apresenta desempenho consistente, principalmente quando combinada com vetores TF-IDF de alta dimensionalidade. Estudos como os de Pang, Lee e Vaithyanathan (2002), além de trabalhos recentes como Freire (2023), demonstram que o método é capaz de capturar padrões relevantes de polaridade textual, mesmo sem modelar explicitamente dependências entre palavras.

Sua principal vantagem está na interpretabilidade e na capacidade de lidar bem com dados esparsos, uma característica típica de modelos textuais. Além disso, por ser um modelo discriminativo, a Regressão Logística otimiza diretamente a separação entre classes, o que pode resultar em melhor desempenho em certos contextos.

Entretanto, como toda técnica linear, o modelo encontra limitações quando as classes não são linearmente separáveis no espaço de atributos, ou quando há sobreposição semântica significativa, como acontece frequentemente com textos neutros e polaridades intermediárias.

Nesse sentido, a inclusão da Regressão Logística no conjunto de modelos testados permite analisar seu comportamento frente a algoritmos mais robustos e probabilísticos, contribuindo para uma avaliação comparativa mais abrangente no contexto de classificação de sentimentos.

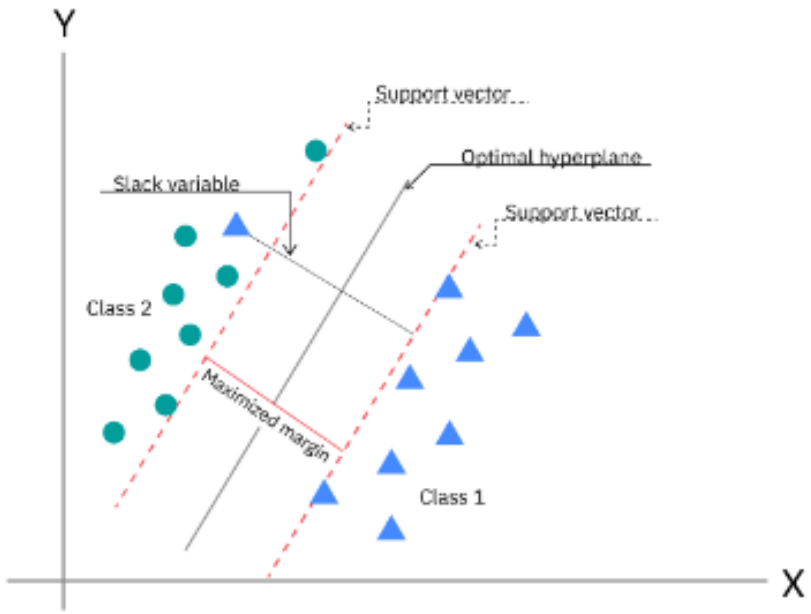
2.9 MÁQUINAS DE VETOR DE SUPORTE (SVM)

As Máquinas de Vetor de Suporte (*Support Vector Machines*) constituem uma das técnicas de aprendizado supervisionado mais utilizadas em tarefas de classificação, especialmente em contextos que envolvem dados de alta dimensionalidade, como ocorre em textos representados por vetores TF-IDF. Esses modelos foram introduzidos por Cortes e Vapnik (1995) e se destacam por sua capacidade de maximizar a margem entre classes, gerando decisões robustas mesmo em ambientes com ruído ou com fronteiras complexas (BISHOP, 2006).

2.9.1 Conceito Fundamental

O princípio básico da SVM consiste em encontrar um hiperplano ótimo que separe duas classes no espaço de atributos. Para isso, o algoritmo identifica as amostras mais próximas da fronteira de decisão (também chamados de vetores de suporte) e maximiza a distância (margem) entre essas amostras e o hiperplano. Esse processo é ilustrado pela Figura 3:

Figura 3 - Margem máxima, hiperplano ótimo e vetores de suporte



Fonte: IBM[3]

[3] IBM. O que são máquinas de vetores de suporte (SVMs)? Disponível em: https://www.ibm.com/br-pt/think/topics/support-vector-machine . Acesso em: 25 de maio de 2025.

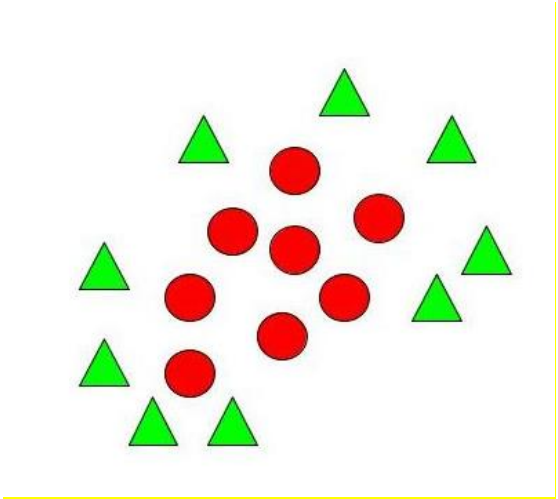
Na figura, as linhas pontilhadas representam as margens da SVM, e os pontos que tocam essas margens são os devores de suporte, que definem a fronteira da classificação. Quanto maior a margem, maior tende a ser a generalização do modelo em novos dados.

2.9.2 Dados Linearmente Separáveis e Não Separáveis

Em cenários ideais, as classes são linearmente separáveis, ou seja, existe uma linha (em 2D) ou hiperplano que separa perfeitamente as classes. Porém esse cenário é incomum em situações reais, como o deste trabalho, pois os textos apresentam grande variabilidade e sobreposição semântica. Normalmente, problemas reais são não lineares.

A Figura 4 mostra um exemplo de dados não linearmente separáveis, onde um hiperplano simples não consegue separar as classes:

Figura 4 – Dados Separáveis não Lineares



Fonte: Sandeep Kumar (2020).[4]

Para lidar com esse cenário, a SVM utiliza duas estratégias principais:

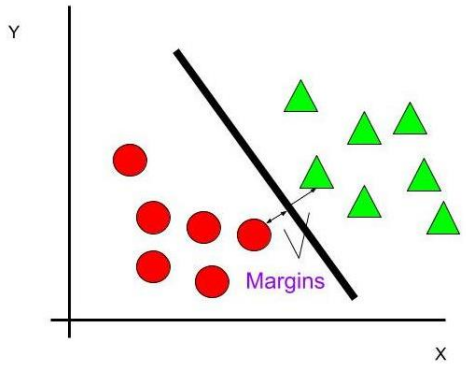
- Variáveis de folga (*slack variables*), que permitem erros controlados na classificação;

- Funções de kernel, que projetam os dados para um espaço de maior dimensionalidade, onde eles se tornam separáveis.

[4] SVM: Difference between linear and non-linear models. Disponível em: https://www.aitude.com/svm-difference-between-linear-and-non-linear-models/. Acesso em: 25 de maio de 2025.

A Figura 5 exemplifica um caso em que o hiperplano consegue separar as classes após esse processo:

Figura 5 – Dados Lineares Separáveis



Fonte: Sandeep Kumar (2020).[5]

2.9.3 Truque do Kernel

Também chamado de kernel trick, permite a SVM resolver problemas não lineares sem realizar explicitamente a transformação para espaços de alta dimensão.

Entre os kernels mais utilizados estão:

- Linear: apropriado quando os dados já são bem separáveis no espaço original (comumente utilizado em TF-IDF)

- RBF (*Radial Basis Function*): usado quando já relações complexas, comum em redes sociais.

- Polynomial: cria fronteiras curvas ajustáveis.

No presente trabalho, foram avaliados os kernels linear e RBF, otimizados por meio do *GridSearchCV*, conforme recomendado por Bishop (2006) para comparações sistemáticas de hiperparâmetros.

2.10 VALIDAÇÃO CRUZADA

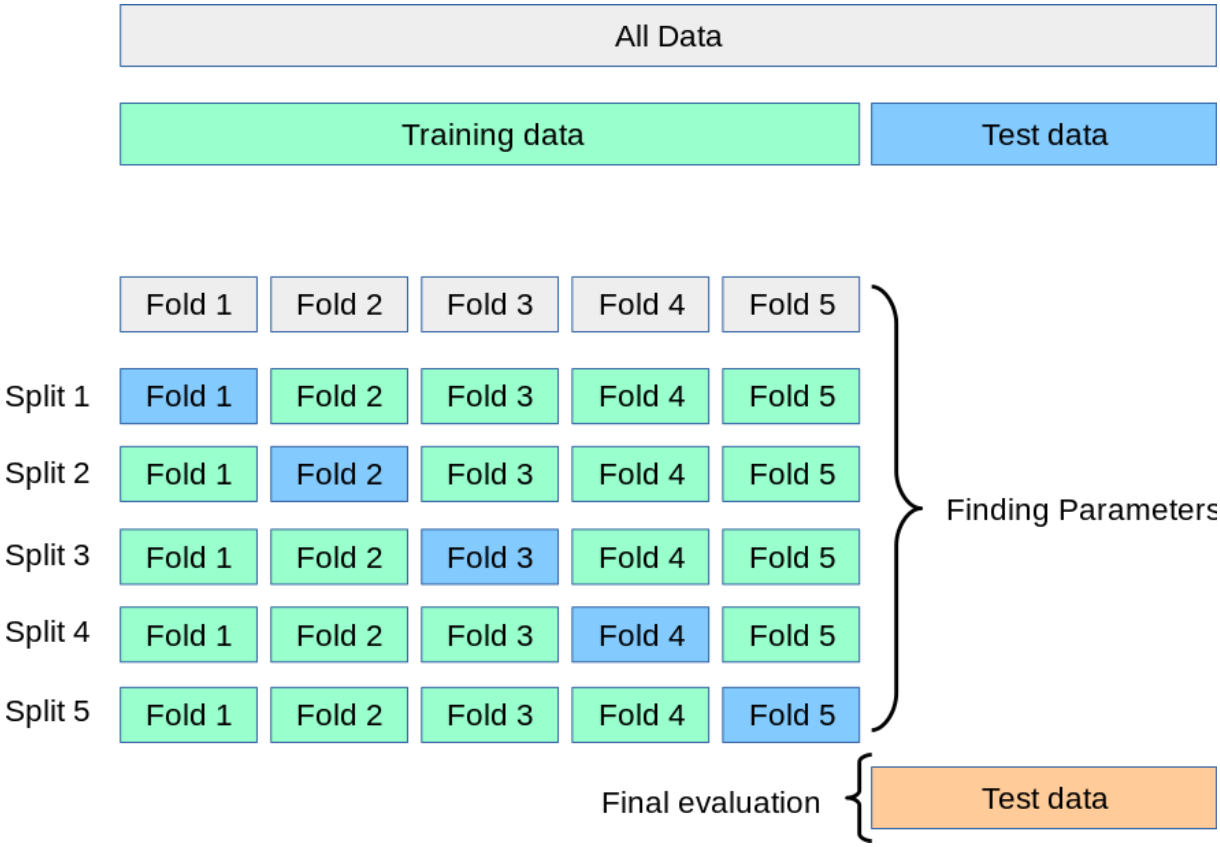
A validação cruzada (*cross-validation*) é um método estatístico utilizado para avaliar o desempenho de modelos de aprendizado supervisionado. Segundo Shalev-Shwartz e Ben-David (2014), essa técnica permite estimar a capacidade de generalização do modelo, evitando o sobreajuste (*overfitting*).

[5] SVM: Difference between linear and non-linear models. Disponível em: https://www.aitude.com/svm-difference-between-linear-and-non-linear-models/. Acesso em: 25 de maio de 2025.

Neste trabalho, foi empregada a *k-fold cross-validation*, com a função *GridSearchCV* da biblioteca *Scikit-learn*, configurada com três partições de validação (cv=3). Foram testadas diferentes combinações dos parâmetros C, kernel e gamma, sendo selecionado o modelo que apresentou melhor desempenho médio.

A Figura 6 ilustra o funcionamento da validação cruzada k-fold. Nessa abordagem, o conjunto de treinamento é dividido em k subconjuntos (*folds*) aproximadamente do mesmo tamanho, e o modelo é treinado k vezes, utilizando um *fold* diferente para validação a cada iteração, enquanto os demais são usados para treino. Ao final, calcula-se a média das métricas de desempenho obtidas em todas as iterações. Essa média representa uma estimativa mais confiável do desempenho real do modelo.

Figura 6 – Funcionamento da Validação Cruzada K-Fold



Fonte: scikit-learn[6]

[6] Cross-validation: evaluating estimator performance. Disponível em: https://scikit-learn.org/stable/modules/cross_validation.html. Acesso em: 28 de outubro de 2025.

2.11 MÉTRICAS DE AVALIAÇÃO

Para avaliar o desempenho do modelo, foram aplicadas métricas clássicas de classificação supervisionada: acurácia, precisão, recall e F1-score. Segundo Figueira (2006), essas métricas permitem mensurar o equilíbrio entre falsos positivos e falsos negativos, fornecendo uma visão mais completa da performance do classificador.

Foram calculadas as seguintes métricas para avaliação do modelo, onde:

- TP – *true positives* (verdadeiros positivos)
- TN – *true negatives* (verdadeiros negativos)
- FP – *false positives* (falsos positivos)
- FN – *false negatives* (falsos negativos)

- **Acurácia:** Representa a proporção total de classificações corretas realizadas pelo modelo em relação ao total de instâncias avaliadas.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

- **Precisão:** Mede a proporção de predições positivas que realmente pertencem à classe positiva. Indica o quão confiável é o modelo quando afirma que um tweet pertence à determinada classe.

$$Precisão = \frac{TP}{TP + FP} \quad (3)$$

- **Revocação (*Recall*):** Mede a capacidade do modelo de identificar corretamente todos os exemplos de uma classe. Isso ajuda a identificar se o modelo está deixando de detectar textos positivos ou negativos.

$$Revocação = \frac{TP}{TP + FN} \quad (4)$$

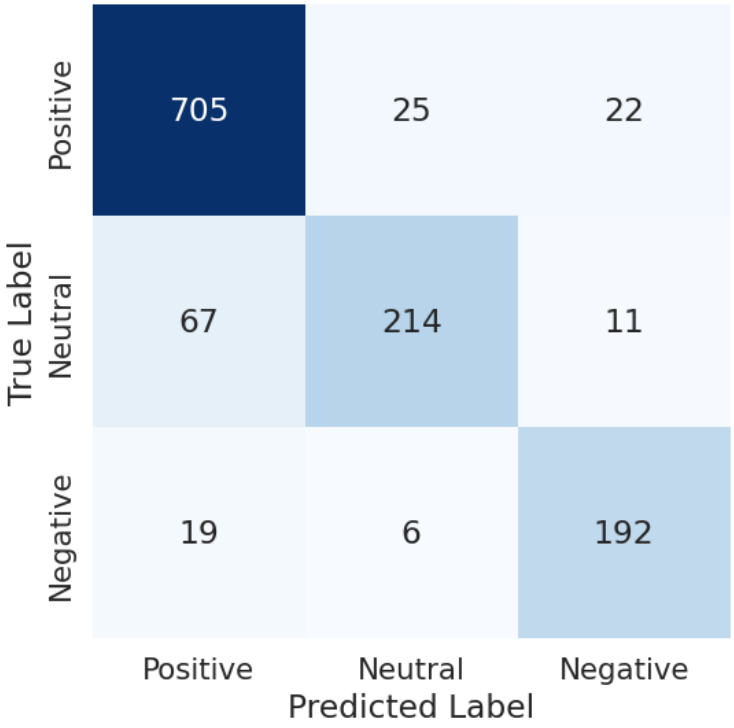
- **F1-Score:** média harmônica entre precisão e revocação. Equilibra os erros do tipo FP e FN.

$$F1 = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação} \quad (5)$$

- **Matriz de Confusão**

A matriz de confusão foi utilizada para visualizar a distribuição de acertos e erros em cada classe, como adotado por Silva (2016) e Corrêa (2017). Essa ferramenta gráfica facilita a identificação dos casos em que o modelo tende a confundir sentimentos semelhantes, como “neutro” e “positivo”. Para problemas com múltiplas classes (positivo, negativo e neutro), a matriz apresenta uma estrutura quadrada onde linhas representam as classes verdadeiras e as colunas representam as classes previstas pelo modelo. A Figura 7 demonstra uma matriz de confusão semelhante à utilizada neste trabalho. A diagonal principal destacada diz respeito aos valores verdadeiros e os demais espaços dizem respeito aos erros de classificação do modelo.

Figura 7 – Exemplo de Matriz de Confusão



Fonte: researchgate[7]

[7] Sentiment Analysis of Solar Energy in U.S. Cities: A 10-Year Analysis Using Transformer-Based Deep Learning - Scientific Figure on ResearchGate. Disponível em: https://www.researchgate.net/figure/Confusion-Matrix-for-Sentiment-Classification-Using-RoBERTa-The-matrix-shows-the_fig2_383914663. Acesso em: 26 de novembro de 2025.

2.12 FERRAMENTAS E BIBLIOTECAS UTILIZADAS

A implementação[8] do sistema foi realizada em Python, linguagem amplamente utilizada em pesquisas científicas devido à sua sintaxe simples e vasto ecossistema de bibliotecas. As principais ferramentas utilizadas foram:

- Pandas: manipulação e integração dos dados;
- NLTK: pré-processamento linguístico e tokenização;
- *Scikit-learn*: vetorização TF-IDF, modelagem com SVM, validação cruzada e métricas de avaliação;
- Matplotlib: geração da matriz de confusão e visualizações gráficas.

Esses recursos garantem reprodutibilidade, precisão e flexibilidade na experimentação, conforme as boas práticas descritas por Wazlawick (2009) em metodologias para pesquisa aplicada em Computação.

[8] O código fonte do projeto pode ser encontrado em: https://github.com/Igorfjs/tcc-classificacao-sentimentos-svm.git

3. RESULTADOS E DISCUSSÃO

Este capítulo apresenta e discute os resultados obtidos a partir da aplicação dos três algoritmos de classificação: Máquina de Vetor de Suporte (SVM), Naive Bayes e Regressão Logística, utilizando o mesmo conjunto de dados pré-processados. Esses dados foram gerados no *script* principal do modelo SVM, que salva um arquivo *csv* contendo os textos já limpos e normalizados previamente para garantir consistência experimental entre os classificadores.

Todos os experimentos foram conduzidos sob as mesmas condições, variando apenas o algoritmo de classificação e os parâmetros de vetorização, configurados buscando os melhores resultados para cada modelo de classificação. Para cada modelo, foram calculadas as métricas de avaliação: acurácia, precisão, revocação, F1-score, além de matrizes de confusão e curvas ROC (*Receiver Operating Characteristic*). O objetivo é oferecer uma comparação justa e detalhada do desempenho de cada modelo.

3.1 Modelo *Support Vector Machine* (SVM)

Para o modelo SVM, a melhor configuração de hiperparâmetros obtida após a validação cruzada foi: C = 10; *gamma* = *scale* e kernel = rbf. O modelo alcançou uma acurácia de 92,44% em 14.653 amostras de teste.

3.2 Modelo Regressão Logística

O modelo alcançou uma acurácia de 81,53% em 14.653 amostras de teste.

3.3 Modelo Naive Bayes

Os algoritmos Naive Bayes demonstraram, após a validação cruzada, melhores resultados com o valor de *alpha* = 0,0001. O modelo ComplementNB alcançou uma acurácia de 82,77%, já o MultinomialNB alcançou uma acurácia de 84,67% ambos em 14.653 amostras de teste.

3.4 Relatórios de Classificação

Para permitir uma análise comparativa clara e objetiva entre os diferentes algoritmos de classificação aplicados neste trabalho, todos os **relatórios de classificação** gerados pelos modelos **Máquina de Vetor de Suporte (SVM)**, **Naive Bayes** e **Regressão Logística** são apresentados em conjunto neste tópico. Os relatórios apresentam as métricas de precisão, revocação, f1-score e suporte (número de instâncias por classe). Esses valores são descritos nas tabelas 2, 3, 4 e 5 a seguir:

Tabela 2 – Relatório de Classificação do modelo SVM (%)

	Precisão	Recall	F1-Score	Suporte
Negativo	93	93	93	4430
Neutro	94	92	93	6139
Positivo	90	92	91	4084

Fonte: Própria autoria

Tabela 3 – Relatório de Classificação do modelo ComplementNB (%)

	Precisão	Recall	F1-Score	Suporte
Negativo	80	87	83	4430
Neutro	89	78	83	6139
Positivo	78	85	82	4084

Fonte: Própria autoria

Tabela 4 – Relatório de Classificação do modelo MultinomialNB (%)

	Precisão	Recall	F1-Score	Suporte
Negativo	86	86	86	4430
Neutro	85	85	85	6139
Positivo	83	82	83	4084

Fonte: Própria autoria

Tabela 5 – Relatório de Classificação do modelo Regressão Logística (%)

	Precisão	Recall	F1-Score	Suporte
Negativo	81	85	83	4430
Neutro	85	78	82	6139
Positivo	77	83	80	4084

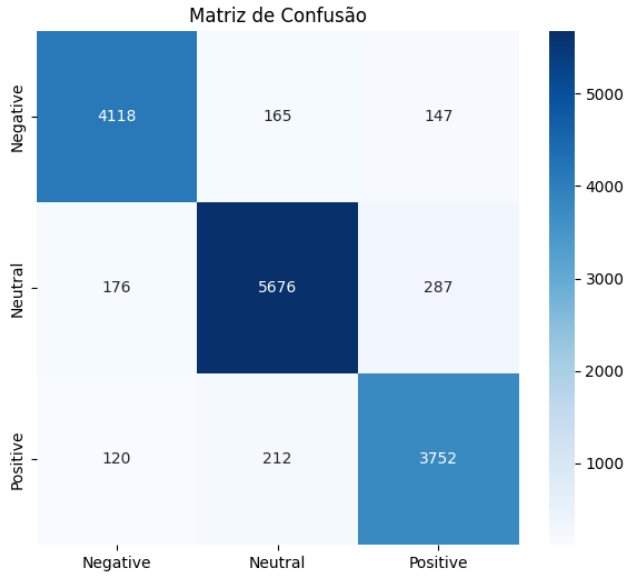
Fonte: Própria autoria

Ao analisar os quatro relatórios de classificação, observa-se que o modelo SVM obteve desempenho superior em todas as classes, apresentando os maiores valores de precisão, recall e f1-score. O modelo de Regressão Logística apresentou os menores valores, embora dentro de um intervalo aceitável. Os modelos Naive Bayes obtiveram desempenho intermediário e relativamente equilibrado sendo eficiente em relação à simplicidade operacional.

3.5 Matrizes de Confusão

Neste tópico, todas as matrizes de confusão são apresentadas visando comparar de forma mais direta os tipos de erro cometidos por cada modelo. As matrizes são apresentadas a seguir nas figuras 8, 9, 10 e 11:

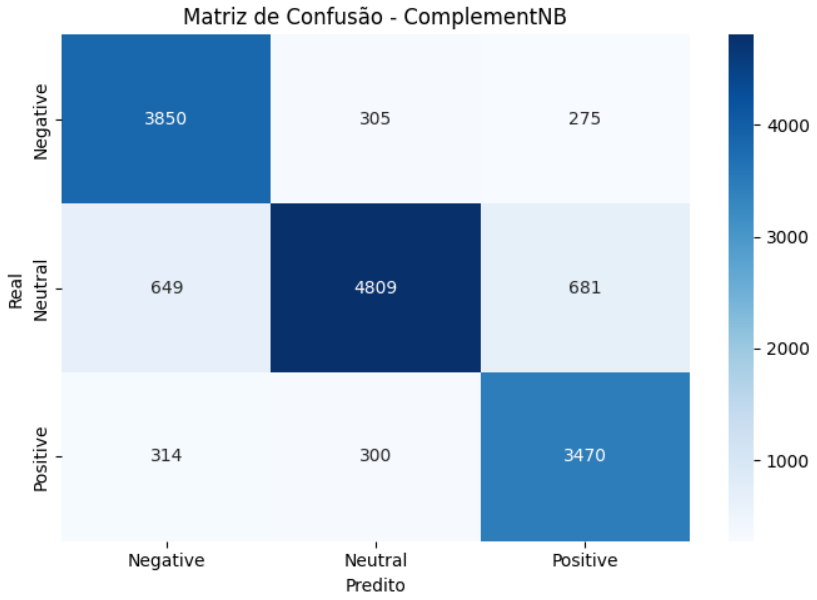
Figura 8 – Matriz de Confusão do modelo SVM



Fonte: Própria Autoria

O modelo SVM apresenta a matriz de confusão mais equilibrada entre todas as classes. Pode-se observar um número mais alto de acertos (diagonal principal) em relação aos outros modelos, o que indica alta capacidade de distinção entre sentimentos negativos, positivos e neutros. A classe negativa obteve poucos falsos positivos e falsos negativos, a classe neutra obteve maior precisão geral e a classe positiva apresenta a maior taxa de erros com 434 classificações falso positivas.

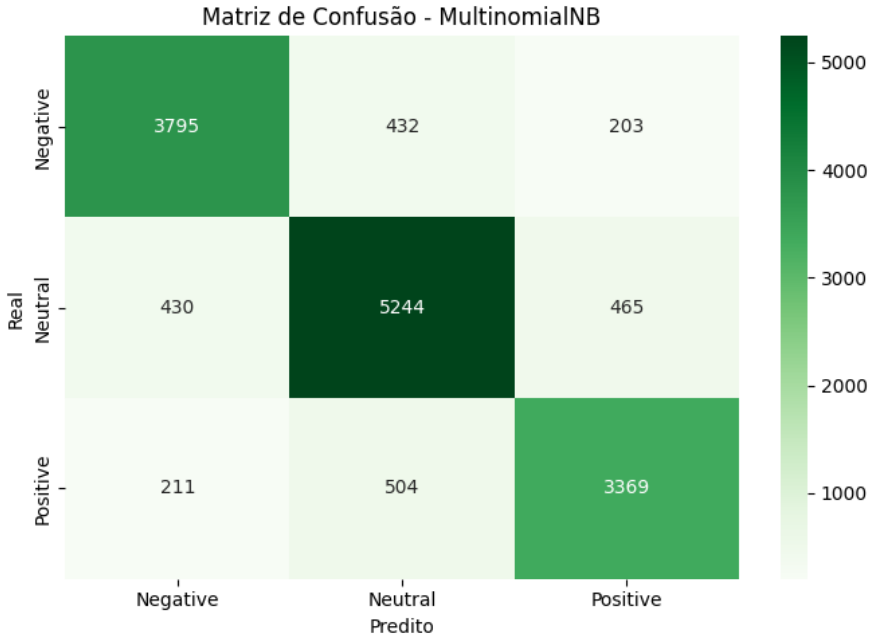
Figura 9 – Matriz de Confusão do modelo ComplementNB



Fonte: Própria Autoria

O modelo ComplementNB apresentou maior número de acertos nas classes negativa e positiva em relação ao modelo MultinomialNB porém, menos acertos em todas as classes em relação ao modelo SVM. Apresentou 2524 erros de classificação sendo a maior concentração em falso negativos, com 963 classificações erradas.

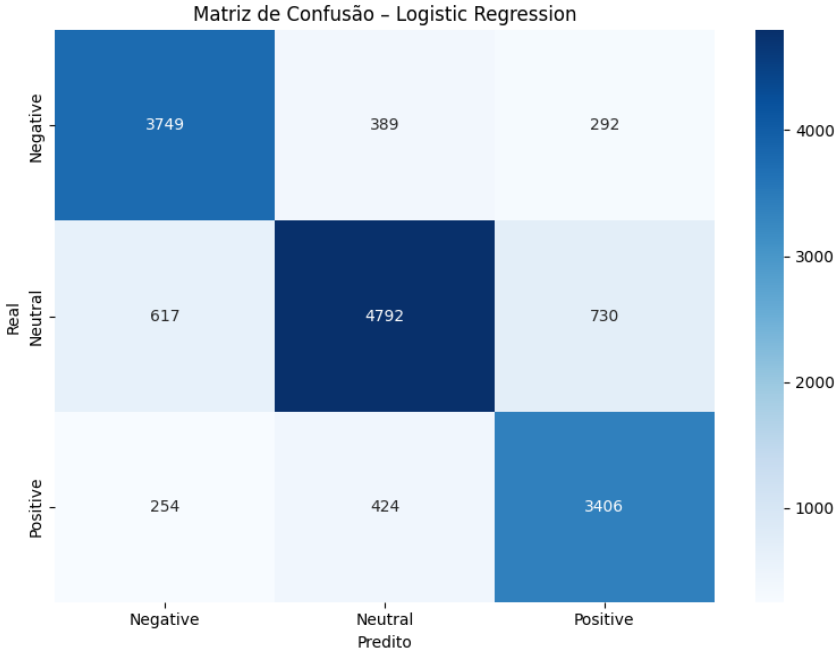
Figura 10 – Matriz de Confusão do modelo MultinomialNB



Fonte: Própria Autoria

O modelo MultinomialNB apresentou maior número de acertos em relação ao modelo ComplementNB. A classe com menos erros foi a neutra, com 936 classificações falso neutras. Também foi o modelo com segunda maior taxa de acertos totais de classificação, com 12408 acertos.

Figura 11 – Matriz de Confusão do modelo Regressão Logística



Fonte: Própria Autoria

O modelo Regressão Logística apresenta maiores índices de confusão em relação ao modelo SVM. Foi o modelo que apresentou maior número de erros de classificação (2706 erros), sendo a classe positiva com maior taxa de erros com 1022 classificações falso positivas.

A análise dessas matrizes permite concluir que, em concordância com os relatórios de classificação, o modelo SVM apresenta melhor desempenho geral contendo o menor número de confusões entre as classes. A classe *Neutral* demonstra ter sido a mais difícil para classificação nos modelos, reforçando a sua natureza ambígua de significados. A Tabela 6 foi criada para melhor visualização dos erros e acertos entre os modelos.

Tabela 6 – Dados das Matrizes de Confusão dos modelos

	SVM	ComplementNB	MultinomialNB	Regressão Logística
Verdadeiro Negativo	4118	3850	3795	3749
Verdadeiro Neutro	5676	4809	5244	4792
Verdadeiro Positivo	3752	3470	3369	3406
Falso Negativo	296	963	641	871
Falso Neutro	377	605	936	813
Falso Positivo	434	956	668	1022
Total de Acertos	13546	12129	12408	11947
Total de Erros	1107	2524	2245	2706

Fonte: Própria Autoria

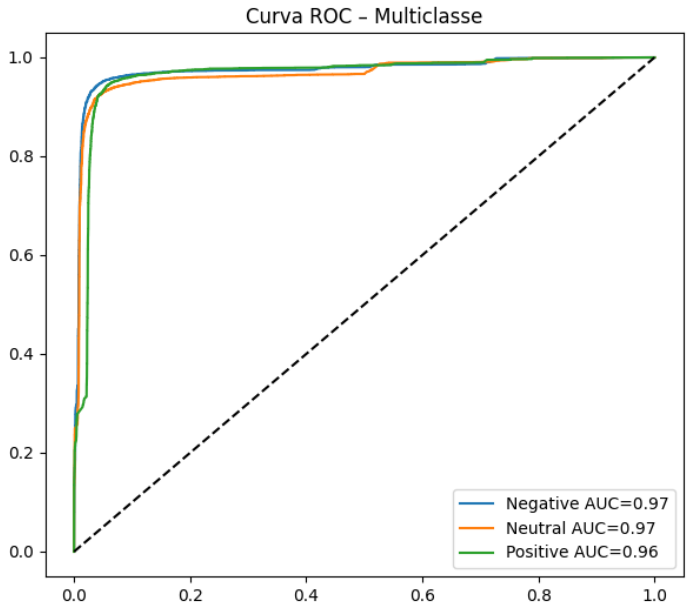
A partir da Tabela 6, foi possível visualizar de maneira mais simples que o modelo com maior quantidade de acertos foi o SVM (13546 acertos) e com a menor foi o de Regressão Logística (11947 acertos). Também pode-se analisar que o modelo com mais classificações falso negativas foi o ComplementNB, falso neutras o MultinomialNB e falso positivas o de Regressão Logística.

3.5 Curvas ROC e valores AUC

As curvas ROC de todos os modelos estão apresentadas a seguir, nas figuras 12, 13, 14 e 15, visando avaliar a capacidade dos modelos de distinguir entre as classes. Além disso, os valores AUC quantificam a capacidade discriminativa dos modelos.

A Figura 12 mostra as curvas ROC do modelo SVM muito próximas do canto superior esquerdo para todas as classes, o que indica alta capacidade discriminativa. Os altos valores de AUC demonstram que o modelo foi capaz de capturar padrões relevantes no espaço vetorial TF-IDF e separar de forma eficiente os sentimentos analisados. Esses resultados enfatizam a superioridade do modelo SVM em relação aos outros em comparação no proposto cenário.

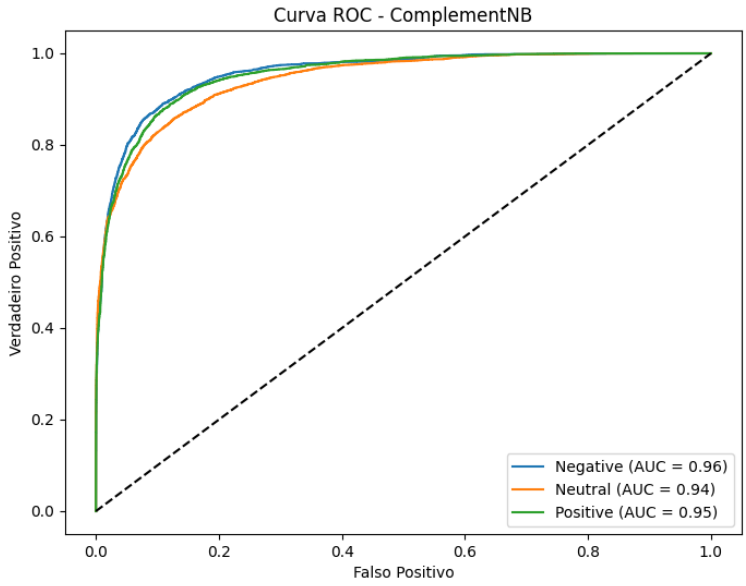
Figura 12 – Curva ROC e valores AUC do modelo SVM



Fonte: Própria Autoria

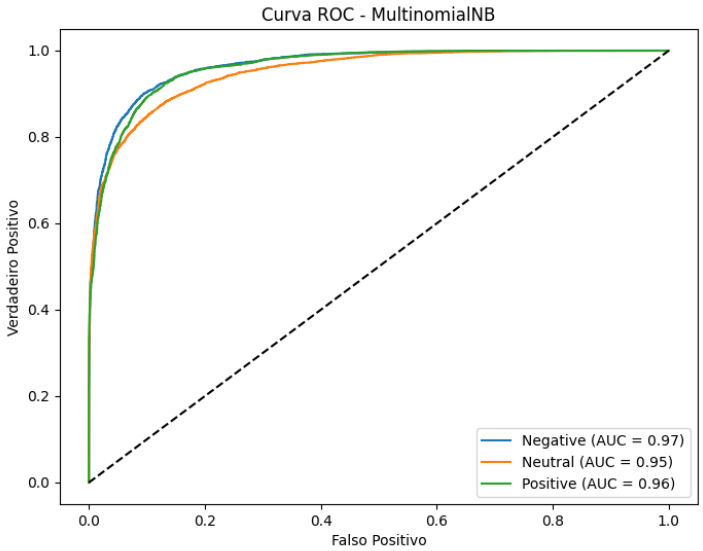
O ComplementNB demonstrou valores AUC menores que o modelo SVM, mas boa capacidade de classificação, com as curvas acima da diagonal aleatória. O que indica utilidade prática no contexto de análise de sentimentos. Essa análise pode ser concluída a partir da Figura 13.

Figura 13 – Curva ROC e valores AUC do modelo ComplementNB



Fonte: Própria Autoria

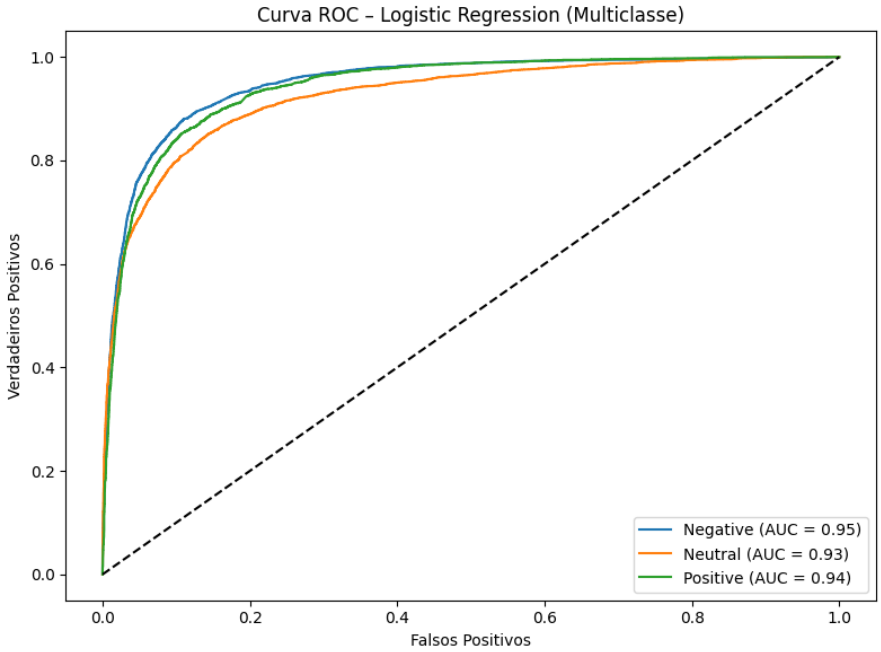
Figura 14 – Curva ROC e valores AUC do modelo MultinomialNB



Fonte: Própria Autoria

O MultinomialNB apresentou maiores valores AUC em todas as classes em relação ao modelo ComplementNB. Em relação ao SVM, o modelo apresentou desempenho inferior principalmente na classe neutra. O que pode ser visto na Figura 14.

Figura 15 – Curva ROC e valores AUC do modelo Regressão Logística



Fonte: Própria Autoria

A Figura 15 apresenta os resultados do modelo Regressão Logística, que apresenta uma área ligeiramente inferior em comparação aos outros modelos,

principalmente o SVM, evidenciando menos robustez em cenários mais complexos do espaço vetorial.

Ao comparar os quatro modelos, observa-se o seguinte ranking em termos de desempenho (pela métrica AUC):

1. **SVM** – melhor desempenho em todas as classes, curvas mais próximas do ideal;
2. **MultinomialNB** – bom desempenho, especialmente para classes mais frequentes;
3. **ComplementNB** – desempenho sólido, porém ligeiramente inferior ao MultinomialNB;
4. **Regressão Logística** – apesar de consistente, apresenta as menores AUCs.

4. CONCLUSÃO E TRABALHOS FUTUROS

A partir dos experimentos realizados, observou-se que o classificador Máquina de Vetor de Suporte (SVM) apresentou desempenho superior em comparação aos modelos Naive Bayes (ComplementNB e MultinomialNB) e Regressão Logística, alcançando acurácia de **92,44%** e melhores resultados em precisão, revocação e F1-score para todas as classes avaliadas. Esse desempenho está de acordo com a literatura, que reconhece o SVM como um dos métodos mais eficazes para dados de alta dimensionalidade e esparsos, características inerentes a representações TF-IDF de textos. Ademais, as matrizes de confusão e os valores AUC evidenciaram a capacidade superior do SVM em diferenciar corretamente sentimentos positivos, negativos e neutros, com menor ocorrência de classificações equivocadas.

Outro ponto relevante observado foi a influência do pré-processamento na qualidade final da classificação. Técnicas como expansão de contrações, substituição de gírias, remoção de ruídos, tokenização especializada e lematização mostraram-se essenciais para reduzir ambiguidades textuais e melhorar a representatividade dos dados vetorizados. Esse resultado reforça o que é discutido em trabalhos como os de Silva (2016), Corrêa (2017) e Freire (2023), nos quais a etapa de preparação dos textos exerce impacto direto no desempenho dos modelos.

A comparação entre os classificadores permitiu concluir que, embora modelos de Naive Bayes apresentem simplicidade computacional e resultados consistentes, não alcançam a mesma capacidade discriminativa do SVM. Já a Regressão Logística, apesar de amplamente usada em tarefas textuais, obteve o menor desempenho entre os métodos, especialmente devido às limitações em lidar com sobreposições semânticas comuns em textos informais.

Os resultados finais demonstram que a utilização de SVM combinada ao TF-IDF constitui uma abordagem robusta, eficiente e altamente aplicável a sistemas reais de análise de sentimentos, especialmente em bases de grande escala. O modelo desenvolvido neste trabalho apresenta potencial para ser integrado a sistemas de monitoramento de redes sociais, ferramentas de suporte à decisão corporativa e plataformas de atendimento digital.

Como perspectivas futuras, sugere-se a investigação de modelos baseados em representações distribuídas, como Word2Vec, GloVe ou *embeddings*

contextualizados derivados de arquiteturas Transformers, como BERT. Além disso, explorar técnicas de balanceamento de classes, expansão da base de dados e análise de sentimentos multilíngue pode contribuir para ampliar a robustez e generalização do sistema. Métodos de aprendizagem profunda também podem ser avaliados para comparação com os modelos tradicionais testados neste estudo.

O poder computacional foi um ponto de destaque durante a execução do projeto, demonstrando alta utilização da CPU principalmente no processo de validação cruzada no modelo SVM. Um ponto de melhoria no presente trabalho seria utilizar o auxílio da GPU para o treinamento dos modelos utilizando frameworks modernos como PyTorch, Keras ou cuML que fariam com que os modelos fossem treinados mais rapidamente.

Em síntese, o trabalho cumpriu seus objetivos ao demonstrar a eficácia do SVM na classificação de sentimentos e ao evidenciar a importância das etapas de pré-processamento e escolha adequada de hiperparâmetros na construção de modelos de NLP. Os resultados obtidos contribuem para a área acadêmica e aplicada, reforçando o potencial do processamento automático de textos como ferramenta estratégica na análise de grandes volumes de dados contemporâneos.

REFERÊNCIAS

APOLINÁRIO, F. *Metodologia da ciência: filosofia e prática da pesquisa.* 2. ed. São Paulo: Penso, 2011.

ARAÚJO, M.; GONÇALVES, P.; BENEVENUTO, F. Measuring sentiments in online social networks. In: *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web.* New York: ACM, 2013. p. 97–104.

ASLAM, S. Twitter by the Numbers: Stats, Demographics & Fun Facts. 2020. Omnicore. Disponível em: https://www.omnicoreagency.com/twitter-statistics/. Acesso em: 12 nov. 2025.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* Sebastopol, CA: O'Reilly Media, 2009.

BISHOP, C. M. *Pattern Recognition and Machine Learning.* New York: Springer, 2006.

BORDIN JUNIOR, A. *Aplicação de programação genética na análise de sentimentos.* 142 f. Dissertação (Mestrado) – Universidade Federal de Goiás, Goiânia, 2018.

CAMELO, J. H. GetOldTweets-python. 2017. Disponível em: https://github.com/Jefferson-Henrique/GetOldTweets-python. Acesso em: 9 de novembro de 2025.

CORRÊA, I. T. *Análise dos sentimentos expressos na rede social Twitter em relação aos filmes indicados ao Oscar 2017.* 72 f. Monografia (Graduação) – Universidade Federal de Uberlândia, 2017.

FIGUEIRA, C. V. *Modelos de regressão logística.* 138 f. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

FREIRE, F. E. S. *Aprendizado de máquina para classificação de sentimentos: uma análise comparativa.* Campinas: Unicamp, 2023. Disponível em: 20.500.12733/17674. Acesso em: 18 nov. 2025.

GIL, A. C. *Como elaborar projetos de pesquisa.* 5. ed. São Paulo: Atlas, 2010.

GONÇALVES, P.; BENEVENUTO, F.; ALMEIDA, V. O que tweets contendo emoticons podem revelar sobre sentimentos coletivos? In: *Anais do II Brazilian Workshop on Social Network Analysis and Mining.* Porto Alegre: SBC, 2013. p. 128–139.

GONÇALVES, T. Teorema de Bayes: o que é e qual sua aplicação? 2020. Disponível em: https://www.voitto.com.br/blog/artigo/teorema-de-bayes. Acesso em: 5 nov. 2025.

JUNG, F. C. *Metodologia para pesquisa e desenvolvimento aplicada a novas tecnologias, produtos e processos.* Rio de Janeiro: Axcel Books, 2004.

KOCHE, J. C. *Fundamentos de metodologia científica: teoria da ciência e prática da pesquisa.* 23. ed. Petrópolis: Vozes, 2006.

LAKATOS, E. M.; MARCONI, M. A. *Fundamentos de metodologia científica.* 7. ed. São Paulo: Atlas, 2010.

MARSHALL, I.; GOMES, T. *Fundamentos e aplicações de máquinas de vetor de suporte.* Porto Alegre: UFRGS, 2018.

PLATT, J. Probabilistic outputs for SVM and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers.* Cambridge: MIT Press, 1999.

SEVERINO, A. J. *Metodologia do trabalho científico.* 23. ed. São Paulo: Cortez, 2007.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge: Cambridge University Press, 2014.

SILVA, N. F. F. da. *Análise de sentimentos em textos curtos provenientes de redes sociais.* 112 f. Tese (Doutorado) – Universidade de São Paulo, São Carlos, 2016.

STARMER, J. StatQuest: Random Forests – Parte 1. YouTube, 2018. Disponível em: https://youtu.be/J4Wdy0Wc_xQ. Acesso em: 8 de outubro de 2025.

WAZLAWICK, R. S. *Metodologia de pesquisa para Ciência da Computação.* Rio de Janeiro: Elsevier, 2009.

YOUTUBE. How Support Vector Machines Work – StatQuest. Disponível em: https://www.youtube.com/watch?v=ba7tMJZbGyA. Acesso em: 19 de maio de 2025.

YOUTUBE. Curso de Machine Learning – Playlist. Disponível em: https://www.youtube.com/watch?v=ePZswmBSLvc&list=PL5TJqBvpXQv5CBxLkdqmou_86syFK7U3Q. Acesso em: 20 de maio de 2025.