

EDUARDO PERES DE LIMA

**APLICAÇÃO DE PLN PARA CLASSIFICAÇÃO DE COMENTÁRIOS DE
JOGOS EM LÍNGUA PORTUGUESA**

Trabalho de Conclusão de Curso apresentado
à banca avaliadora do Curso de Engenharia
de Computação, da Escola Superior de
Tecnologia, da Universidade do Estado do
Amazonas, como pré-requisito para obtenção
do título de Engenheiro de Computação.

Orientador(a): Prof. Dr. Luís Cuevas Rodríguez

Manaus – Junho – 2026

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

L732a

Lima, Eduardo Peres de

Aplicação de PLN para classificação de comentários de jogos em
língua portuguesa / Eduardo Peres de Lima. Manaus : [s.n], 2026.
65 f.: color.; 21.0 cm.

TCC - Graduação em Engenharia de Computação- Universidade do
Estado do Amazonas, Manaus, 2026.

Inclui Bibliografia.

Orientador: Rodriguez, Luis Cuevas.

1. Processamento de linguagem natural. 2. Classificação
multirrótulo. 3. Jogos digitais. 4. Aprendizado de máquina. 5.
Rotulação automática. I. Rodriguez, Luis Cuevas (Orient.) II.
Universidade do Estado do Amazonas. III. Título

CDU(1997)62:004



FOLHA DE APROVAÇÃO

APLICAÇÃO DE PLN PARA CLASSIFICAÇÃO DE COMENTÁRIOS DE JOGOS EM LÍNGUA PORTUGUESA

EDUARDO PERES DE LIMA

Trabalho de Conclusão de Curso II defendido e aprovado pela banca
avaliadora constituída pelos professores:

Prof. Dr. Luis Cuevas Rodriguez
Presidente

Prof. Dr. Sergio Cleger Tamayo
Avaliador

Prof. MSc. Eduardo Jorge Lira
Avaliador

Manaus, 22 de junho de 2026.

Resumo

Este trabalho investiga a aplicação de técnicas de Processamento de Linguagem Natural (PLN) na classificação multirrótulo de comentários de jogos digitais em língua portuguesa, no contexto de análise automática de opiniões de usuários. A proposta consiste na construção de um pipeline híbrido que combina modelagem de tópicos (Latent Dirichlet Allocation), rotulagem automática com Modelos de Linguagem (LLMs) e estratégias de consenso entre modelos para aumentar a consistência dos rótulos gerados. A partir desse conjunto de dados, são avaliados modelos de aprendizado supervisionado com diferentes configurações de pré-processamento textual e representações vetoriais.

Inicialmente, comentários reais foram coletados e processados por técnicas de normalização e limpeza textual, sendo posteriormente enriquecidos com rótulos automáticos, reduzindo a dependência de anotação manual. Em seguida, foram aplicados algoritmos como Support Vector Machines (SVM), Random Forest e Regressão Logística em cenários multirrótulo. Os resultados indicam variações significativas de desempenho entre as configurações avaliadas, evidenciando a influência do pré-processamento e da qualidade dos rótulos gerados automaticamente. De forma geral, observa-se que a integração entre aprendizado supervisionado e métodos modernos de construção de dados apresenta potencial para melhoria de desempenho, embora ainda existam desafios relacionados à consistência dos rótulos e à generalização dos modelos. Como trabalhos futuros, propõe-se a ampliação do conjunto de dados, a adoção de estratégias mais robustas de consenso entre LLMs e a inclusão de novos modelos de aprendizado supervisionado.

Palavras-chave: Processamento de Linguagem Natural, Classificação Multirrótulo, Jogos Digitais, Aprendizado de Máquina, Modelos de Linguagem, Rotulagem Automática.

Abstract

This work investigates the application of Natural Language Processing (NLP) techniques to the multi-label classification of Portuguese-language digital game reviews, in the context of automatic user opinion analysis. The proposed approach consists of a hybrid pipeline that combines topic modeling (Latent Dirichlet Allocation), automatic labeling using Large Language Models (LLMs), and consensus strategies among multiple models to improve label consistency. Based on this dataset, supervised learning models are evaluated under different configurations of text preprocessing and vector representations.

Initially, real user comments were collected and processed using text normalization and cleaning techniques, and subsequently enriched with automatically generated labels, reducing the dependence on manual annotation. Then, classification algorithms such as Support Vector Machines (SVM), Random Forest, and Logistic Regression were applied in multi-label settings. The results indicate significant performance variations across the evaluated configurations, highlighting the impact of preprocessing strategies and the quality of automatically generated labels. Overall, the integration of supervised learning with modern data construction methods shows potential for performance improvement, although challenges remain regarding label consistency and model generalization. As future work, we propose expanding the dataset, adopting more robust consensus strategies among LLMs, and incorporating additional supervised learning models.

Keywords: Natural Language Processing, Multi-label Classification, Digital Games, Machine Learning, Language Models, Automatic Labeling.

Sumário

Lista de Figuras	ix
1 Introdução	1
1.1 Objetivos	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	3
1.2 Justificativa	3
1.3 Metodologia	5
1.3.1 Seleção dos Jogos	6
1.3.2 Coleta dos Comentários	6
1.3.3 Definição dos Rótulos	6
1.3.4 Rotulagem Automática dos Comentários	7
1.3.5 Pré-processamento Textual	7
1.3.6 Treinamento dos Modelos	7
1.3.7 Avaliação Experimental	7
1.3.8 Ferramentas Utilizadas	8
2 Fundamentação Teórica	9
2.1 Inteligência Artificial (IA)	9
2.2 Processamento de Linguagem Natural	10
2.3 Modelagem de Tópicos	11
2.3.1 Latent Dirichlet Allocation (LDA)	11

2.3.2	Coerência de Tópicos	11
2.3.3	Seleção do Número de Tópicos (Elbow Method)	12
2.4	Rotulagem Automática de Comentários com LLMs	12
2.4.1	Prompt Engineering (Zero-shot)	12
2.4.2	Consenso entre Modelos de Linguagem	13
2.5	Representação Vetorial de Texto	13
2.5.1	Bag of Words	14
2.5.2	TF-IDF	14
2.6	Aprendizado de Máquina	15
2.6.1	Aprendizado Supervisionado	15
2.6.2	<i>One vs All</i>	16
2.6.3	<i>One-vs-One</i>	17
2.7	Modelos de Inteligência Artificial	18
2.7.1	<i>Support Vector Machine (SVM)</i>	18
2.7.2	<i>Random Forest</i>	19
2.7.3	<i>Logistic Regression (LR)</i>	20
2.8	Técnicas de Pré-Processamento	22
2.9	Validação de Modelos	23
2.9.1	Validação Cruzada	23
2.9.2	Validação Cruzada Estratificada Multirrótulo	24
2.10	Otimização de Hiperparâmetros	24
2.10.1	Grid Search	25
2.11	Métricas de Avaliação	25
2.11.1	Precisão (<i>Precision</i>)	25
2.11.2	Revocação (<i>Recall</i>)	26
2.11.3	F1-Score	26
2.11.4	Hamming Loss	27

3	Trabalhos Relacionados	29
3.1	Classificação Multirrótulo em Reviews de Restaurantes Brasileiros	29
3.2	Modelos Supervisionados para Classificação de Texto	30
3.3	Construção e Rotulagem Automática de Dados	31
3.4	Síntese e Lacunas da Literatura	31
4	Desenvolvimento	33
4.1	Obtenção do Dataset	33
4.1.1	Seleção dos Jogos	33
4.1.2	Coleta dos Comentários	35
4.1.3	Análise Exploratória do Dataset	37
4.2	Definição dos Rótulos	38
4.2.1	Modelagem de Tópicos com LDA	39
4.2.2	Definição da Quantidade de Tópicos	41
4.2.3	Interpretação dos Tópicos e Definição dos Rótulos	43
4.3	Rotulagem Automática Utilizando LLMs	45
4.3.1	Seleção dos Modelos	45
4.3.2	Engenharia de Prompt	46
4.3.3	Processamento das Respostas	47
4.3.4	Construção dos Datasets por Consenso	48
4.4	Treinamento dos Modelos de Classificação	49
4.4.1	Representação Vetorial dos Comentários	50
4.4.2	Estratégias de Pré-Processamento Textual	51
4.4.3	Estratégia de Classificação Multirrótulo	52
4.4.4	Algoritmos Avaliados	52
4.4.5	Validação Cruzada	53
4.4.6	Treinamento da Rede Neural MLP	54
4.5	Análise dos Resultados	55
4.6	Conclusão	59

Lista de Figuras

1.1	Fonte: PGB (Pesquisa Game Brasil, 2026).	4
4.1	Distribuição dos comentários por intervalo de tamanho.	37
4.2	Maior coerência obtida para cada quantidade de tópicos.	41
4.3	Resultados de coerência para os diferentes pré-processamentos avaliados.	42
4.4	Distribuição dos rótulos no 3º dataset.	52

Capítulo 1

Introdução

A indústria de jogos digitais tem se consolidado como um dos setores de maior crescimento no entretenimento global, e o Brasil acompanha essa tendência com uma comunidade ativa e crescente de jogadores (ABRAGAMES, 2023). Com a popularização das plataformas digitais, tornou-se comum a publicação de comentários por usuários que expressam opiniões, críticas e sugestões sobre diferentes aspectos dos jogos, como qualidade, progressão, conteúdo de jogo e problemas técnicos.

Esse volume crescente de dados textuais representa uma fonte relevante para análise da experiência do usuário e compreensão de percepções dos jogadores. Entretanto, sua análise manual é inviável devido à escala, à diversidade e à natureza não estruturada desses comentários, que frequentemente são curtos, informais e abordam múltiplos aspectos simultaneamente.

Nesse cenário, técnicas de Processamento de Linguagem Natural (PLN) são fundamentais para a extração automática de conhecimento a partir de textos em linguagem natural. Apesar dos avanços recentes na área, grande parte das soluções ainda é desenvolvida para a língua inglesa, enquanto aplicações em português brasileiro permanecem menos exploradas e apresentam desafios adicionais relacionados à variação linguística e à escassez de dados anotados.

Diante desse contexto, a tarefa de **classificação multirrótulo** (*multi-label classification*) é particularmente adequada, pois permite que múltiplos aspectos sejam atribuídos a um único comentário, refletindo melhor a natureza real dos dados. Esse tipo de problema é mais complexo do que a classificação tradicional, exigindo modelos capazes de capturar múltiplas dimensões

semânticas simultaneamente.

Além disso, embora métodos clássicos de aprendizado supervisionado, como *Support Vector Machines*, *Random Forests* e *Logistic Regression*, ainda sejam amplamente utilizados em tarefas de classificação textual devido à sua eficiência e competitividade, sua aplicação isolada limita-se à dependência de conjuntos de dados rotulados de alta qualidade.

Neste trabalho, essa limitação é mitigada por meio de uma abordagem híbrida de construção de dados, que combina modelagem de tópicos (LDA) para estruturação inicial dos comentários, rotulagem automática baseada em Modelos de Linguagem (LLMs) e mecanismos de consenso entre diferentes modelos para refinamento dos rótulos. Essa estratégia reduz a dependência de anotação manual e busca aumentar a consistência do conjunto de dados utilizado no treinamento.

A partir desse conjunto, são avaliados modelos supervisionados tradicionais com diferentes representações vetoriais e configurações de pré-processamento, como lematização e remoção de *stopwords*, com o objetivo de analisar seu desempenho na tarefa de classificação multirrótulo de comentários em português.

Dessa forma, este trabalho contribui para o avanço das aplicações de PLN no domínio de jogos digitais e para a literatura em língua portuguesa, ao investigar uma abordagem integrada que combina técnicas clássicas de aprendizado de máquina com estratégias modernas de construção e refinamento automático de dados.

1.1 Objetivos

A seguir, são apresentados o Objetivo Geral e os Objetivos Específicos deste trabalho.

1.1.1 Objetivo Geral

Investigar e comparar o desempenho de diferentes abordagens de aprendizado de máquina na tarefa de classificação multirrótulo de comentários de jogos em português, combinando técnicas de Processamento de Linguagem Natural (PLN), modelos de aprendizado supervisionado

e estratégias de rotulagem automática baseadas em Modelos de Linguagem (LLMs), com o objetivo de identificar tópicos relevantes nos comentários de usuários e gerar insights úteis para desenvolvedores de jogos digitais.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Aplicar e avaliar técnicas de pré-processamento textual em PLN, incluindo *stemming*, *lemmatization* e normalização textual, visando adequar os dados para as etapas de modelagem.
- Construir um conjunto de dados rotulado a partir de estratégias híbridas de modelagem de tópicos (LDA) e rotulagem automática com LLMs, incluindo mecanismos de consenso entre modelos.
- Treinar e avaliar diferentes algoritmos de aprendizado supervisionado para classificação multirrótulo, incluindo SVM, Random Forest e Regressão Logística, além de uma rede neural MLP.
- Comparar o desempenho dos modelos utilizando métricas apropriadas para problemas multirrótulo, como F1-score (micro e macro) e Hamming Loss.
- Analisar os resultados obtidos e identificar as abordagens mais eficazes para a tarefa proposta, considerando qualidade preditiva e robustez dos modelos.

1.2 Justificativa

A indústria de jogos digitais tem se consolidado como um dos setores de maior crescimento no mercado global e nacional, acompanhando o aumento contínuo da base de jogadores no Brasil. Dados recentes indicam que uma parcela significativa da população brasileira consome jogos digitais de forma recorrente, evidenciando a relevância econômica e social desse segmento (ABRAGAMES, 2023; PGB, 2026).

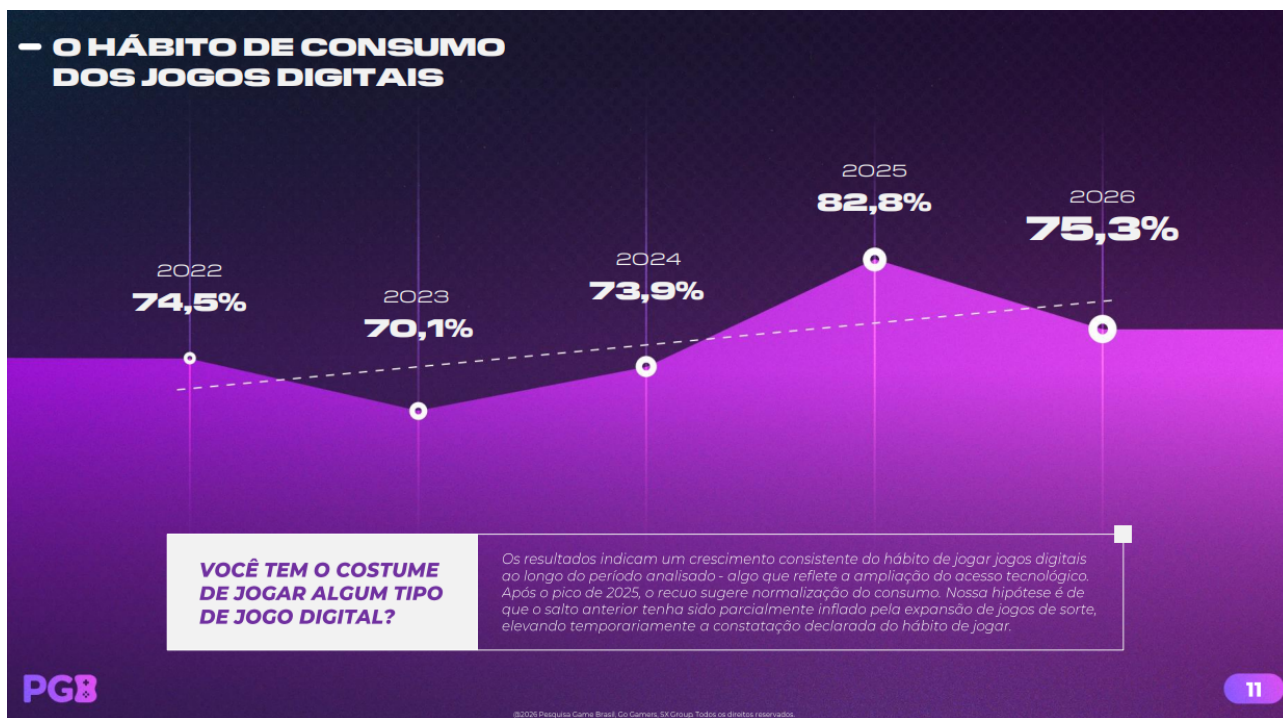


Figura 1.1: Fonte: PGB (Pesquisa Game Brasil, 2026).

Nesse cenário, observa-se um crescimento expressivo na produção de comentários de usuários em plataformas digitais, como lojas de aplicativos e redes sociais. Esses comentários constituem uma fonte relevante de informação sobre a experiência do jogador, permitindo a análise de diferentes aspectos dos jogos, como jogabilidade, gráficos, narrativa e desempenho técnico.

Entretanto, devido ao grande volume e à natureza não estruturada desses dados, sua análise manual torna-se inviável na prática. Dessa forma, técnicas de *Processamento de Linguagem Natural* (PLN) são essenciais para automatizar a extração de informações a partir de textos em linguagem natural, especialmente em contextos de língua portuguesa, ainda menos explorados em comparação a idiomas como o inglês.

Dentro desse contexto, a tarefa de **classificação multirrótulo** (*multi-label classification*) se mostra adequada, uma vez que um único comentário pode abordar simultaneamente múltiplos aspectos de um jogo, exigindo modelos capazes de capturar diferentes dimensões semânticas de forma simultânea.

Além disso, abordagens clássicas de aprendizado supervisionado continuam sendo amplamente utilizadas em tarefas de classificação textual, especialmente quando associadas a repre-

representações vetoriais esparsas como Bag of Words e TF-IDF, devido à sua eficiência computacional e desempenho competitivo em diversos cenários (MINAEE et al., 2021; PETERS et al., 2023). Modelos como *Support Vector Machines*, *Random Forests* e *Logistic Regression* ainda são frequentemente empregados como fortes baselines em problemas de classificação de texto.

No entanto, a integração dessas abordagens com estratégias modernas de construção e rotulagem de dados, como o uso de Modelos de Linguagem (LLMs) para anotação automática e técnicas de consenso para refinamento de rótulos, ainda é pouco explorada no domínio de comentários de jogos em língua portuguesa, evidenciando uma lacuna na literatura recente de PLN aplicado.

Neste trabalho, propõe-se uma abordagem híbrida para construção do conjunto de dados, combinando modelagem de tópicos (LDA) com rotulagem automática baseada em LLMs e mecanismos de consenso entre diferentes modelos, visando reduzir o custo de anotação manual e aumentar a consistência dos rótulos.

A partir desse conjunto de dados, são avaliadas diferentes representações vetoriais e modelos supervisionados, com o objetivo de analisar o impacto de cada configuração no desempenho da classificação multirrótulo.

Dessa forma, este estudo contribui para o avanço das aplicações de PLN no domínio de jogos digitais e para a literatura em língua portuguesa, ao investigar a integração entre técnicas clássicas de aprendizado supervisionado e estratégias modernas de construção de dados assistidas por LLMs.

1.3 Metodologia

A metodologia adotada neste trabalho foi dividida em etapas sequenciais, abrangendo desde a construção do conjunto de dados até o treinamento e avaliação dos modelos de classificação multirrótulo. Inicialmente foi realizada a coleta de comentários de jogos eletrônicos na Google Play Store, seguida pela definição dos rótulos por meio de modelagem de tópicos. Posteriormente, técnicas de processamento de linguagem natural e modelos de linguagem de grande porte foram utilizadas para rotular automaticamente os comentários. Por fim, diferentes algo-

ritmos de aprendizado de máquina foram treinados e avaliados utilizando validação cruzada e métricas específicas para classificação multirrótulo.

1.3.1 Seleção dos Jogos

Inicialmente foi realizada uma seleção de jogos populares disponíveis na Google Play Store utilizando a plataforma AppBrain (AppBrain, 2026). Foram considerados jogos com mais de cinco milhões de instalações e pertencentes a diferentes gêneros, buscando garantir diversidade temática e volume suficiente de avaliações. Após a obtenção de informações adicionais por meio da biblioteca `google_play_scraper`, os jogos foram ordenados pela quantidade de avaliações disponíveis e os dez com maior volume foram selecionados para a coleta dos comentários.

1.3.2 Coleta dos Comentários

Os comentários foram coletados utilizando a biblioteca `google_play_scraper`, aplicando filtros de idioma e país para obter apenas avaliações em português brasileiro. Para cada jogo selecionado, buscou-se obter 200 mil comentários, e, posteriormente, selecionar os comentários mais extensos. Ao final, foram mantidos 20 mil comentários de cada jogo, resultando em um conjunto balanceado com 200 mil comentários.

1.3.3 Definição dos Rótulos

Como não existia um conjunto de categorias previamente definido para os comentários analisados, foi utilizada modelagem de tópicos baseada em LDA (*Latent Dirichlet Allocation*) para identificar temas recorrentes presentes no dataset. Diferentes estratégias de pré-processamento textual e combinações de parâmetros foram avaliadas utilizando a métrica de coerência semântica. Os tópicos gerados pelos modelos selecionados foram analisados manualmente e consolidados em um conjunto final de rótulos utilizado na tarefa de classificação.

1.3.4 Rotulagem Automática dos Comentários

A rotulagem dos comentários foi realizada utilizando cinco modelos de linguagem executados localmente por meio da plataforma Ollama. Cada comentário foi submetido às cinco LLMs utilizando uma abordagem *zero-shot*. As respostas obtidas foram processadas, normalizadas e combinadas por meio de uma estratégia de consenso entre modelos. Diferentes critérios de consenso foram avaliados, resultando na geração de múltiplas versões do dataset rotulado.

1.3.5 Pré-processamento Textual

Antes do treinamento dos classificadores, os comentários passaram por diferentes estratégias de pré-processamento textual. As abordagens avaliadas incluíram lematização, filtragem por classe gramatical, remoção de termos específicos do domínio e normalização textual. O objetivo foi analisar o impacto dessas transformações no desempenho dos modelos de classificação.

1.3.6 Treinamento dos Modelos

Foram avaliados três algoritmos clássicos de aprendizado de máquina, SVM, Regressão Logística e Random Forest, além de uma rede neural MLP. Os comentários foram representados utilizando TF-IDF e os rótulos foram transformados para representação binária por meio do `MultiLabelBinarizer`. Como o problema é multirrótulo, foi utilizada a estratégia One-vs-Rest para treinamento dos classificadores.

1.3.7 Avaliação Experimental

A avaliação dos modelos foi realizada utilizando validação cruzada estratificada para problemas multirrótulo (`MultiLabelStratifiedKFold`) com cinco partições. O desempenho foi medido por meio das métricas Macro F1-Score, Micro F1-Score e Hamming Loss. Os resultados obtidos foram comparados para identificar as combinações de algoritmo, pré-processamento e estratégia de rotulagem com melhor desempenho.

1.3.8 Ferramentas Utilizadas

O desenvolvimento deste trabalho foi realizado utilizando a linguagem de programação Python, devido à ampla disponibilidade de bibliotecas voltadas para processamento de linguagem natural, aprendizado de máquina e integração com modelos de linguagem.

A coleta dos comentários foi realizada por meio da biblioteca `google_play_scraper`, utilizada para acessar informações e avaliações disponíveis na Google Play Store. A plataforma AppBrain foi utilizada como fonte complementar para obtenção de informações sobre popularidade, quantidade de instalações e categorias dos jogos analisados.

As etapas de processamento textual foram implementadas utilizando a biblioteca `spaCy`, responsável pela tokenização, lematização e identificação de classes gramaticais dos comentários. A biblioteca `NLTK` também foi empregada em atividades relacionadas ao processamento textual e manipulação de *stopwords*.

Para a modelagem de tópicos foi utilizada a biblioteca `Gensim`, responsável pela construção dos corpus, treinamento dos modelos LDA e cálculo das métricas de coerência utilizadas na avaliação dos tópicos gerados.

A rotulagem automática dos comentários foi realizada utilizando a plataforma `Ollama`, que permitiu a execução local dos modelos de linguagem empregados no trabalho. Foram utilizados os modelos `Phi-3 3.8B`, `Mistral 7B`, `Gemma 2 9B`, `Gemma 3 12B` e `Llama 3.1 8B`.

Os experimentos de aprendizado de máquina foram desenvolvidos utilizando a biblioteca `scikit-learn`, responsável pela implementação dos algoritmos de classificação, técnicas de vetorização textual, validação cruzada e métricas de avaliação. A biblioteca `iterative-stratification` foi utilizada para aplicação do método `MultilabelStratifiedKFold`, empregado na validação cruzada estratificada para problemas multirrótulo.

Por fim, as bibliotecas `Pandas`, `NumPy` e `Matplotlib` foram utilizadas para manipulação dos dados, análise exploratória e geração dos gráficos apresentados ao longo do trabalho.

Capítulo 2

Fundamentação Teórica

Neste capítulo será feita uma revisão de literatura, em que serão introduzidas as bases técnicas necessárias para o entendimento do trabalho a ser desenvolvido. As sessões a seguir detalham os conceitos de **Inteligência Artificial**, **Processamento de Linguagem Natural**, **Aprendizado de Máquina**, **Modelos de IA** e **Técnicas de Pré-Processamento**.

2.1 Inteligência Artificial (IA)

A Inteligência Artificial (IA), do inglês *Artificial Intelligence*, pode ser definida como o ramo da Ciência da Computação que se dedica ao desenvolvimento de comportamentos inteligentes em sistemas computacionais (LUGER, 2024). A definição de “inteligência”, contudo, permanece em debate, uma vez que se relaciona diretamente à inteligência humana, campo que envolve diversas questões filosóficas e cognitivas (RUSSELL; NORVIG, 2021).

De forma geral, a IA busca emular comportamentos humanos e racionais por meio da aplicação de técnicas estatísticas, lógicas e computacionais, com o objetivo de resolver problemas específicos (TURING, 1950; RUSSELL; NORVIG, 2021). Desde os trabalhos iniciais de Turing, passando pelas abordagens simbólicas até os métodos baseados em redes neurais profundas, a área tem avançado significativamente e se expandido para diversas aplicações práticas.

Entre as principais áreas de aplicação da IA destacam-se: a Visão Computacional, responsável pelo processamento e interpretação de imagens e vídeos, incluindo tarefas como re-

conhecimento de objetos e análise facial (SZELISKI, 2022); o Processamento de Linguagem Natural (PLN), voltado para a compreensão e geração de linguagem humana, sendo aplicado em sistemas de tradução automática, chatbots, análise de sentimentos e compreensão textual (JURAFSKY; MARTIN, 2025); e o Aprendizado de Máquina, que se refere à construção de modelos capazes de identificar padrões e realizar previsões a partir de dados, sendo amplamente utilizado em sistemas de recomendação, classificação e tomada de decisão automatizada (MURPHY, 2022).

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN), ou *Natural Language Processing* (NLP), é uma subárea da inteligência artificial voltada para a interação entre computadores e linguagem humana. O objetivo do PLN é permitir que máquinas compreendam, interpretem, gerem e manipulem linguagem natural de forma significativa (JURAFSKY; MARTIN, 2025).

O PLN engloba uma variedade de tarefas linguísticas, que vão desde a análise morfo-sintática até a compreensão semântica e pragmática. Entre as tarefas mais comuns destacam-se: classificação de texto, análise de sentimentos, sumarização automática, reconhecimento de entidades nomeadas, tradução automática e resposta a perguntas (TREVISIO et al., 2023).

A crescente disponibilidade de dados textuais, impulsionada pelas redes sociais, fóruns e plataformas digitais, consolidou o PLN como ferramenta essencial em aplicações modernas de mineração de texto. O sucesso dessas aplicações está fortemente relacionado à qualidade da representação textual, ou seja, à forma como textos são convertidos em estruturas que algoritmos computacionais podem processar (JURAFSKY; MARTIN, 2025).

Inicialmente, as representações mais utilizadas baseavam-se em abordagens simples como o modelo *bag-of-words*, que ignora a ordem e a estrutura gramatical das palavras, representando o texto como um vetor de frequências. Para melhorar essa representação, foram desenvolvidas técnicas como TF-IDF (*Term Frequency-Inverse Document Frequency*), que pondera a relevância de termos com base em sua frequência nos documentos e no corpus como um todo (JURAFSKY; MARTIN, 2025).

2.3 Modelagem de Tópicos

A modelagem de tópicos é uma técnica de aprendizado não supervisionado utilizada para descobrir estruturas semânticas latentes em grandes coleções de documentos textuais. Essa abordagem busca identificar grupos de palavras que frequentemente ocorrem juntas, permitindo representar cada documento como uma mistura de tópicos.

Entre os modelos mais utilizados para essa tarefa está o Latent Dirichlet Allocation (LDA), amplamente aplicado em mineração de texto, análise de sentimentos e classificação temática (BLEI; NG; JORDAN, 2003; STEVENS et al., 2012).

2.3.1 Latent Dirichlet Allocation (LDA)

O Latent Dirichlet Allocation (LDA) é um modelo probabilístico generativo que assume que cada documento é composto por uma distribuição de tópicos, e cada tópico é representado por uma distribuição de palavras.

O processo de geração do modelo considera que, para cada documento, são sorteados tópicos a partir de uma distribuição Dirichlet, e para cada tópico são selecionadas palavras também a partir de uma distribuição probabilística. Dessa forma, o LDA permite inferir tanto a distribuição de tópicos por documento quanto a distribuição de palavras por tópico.

Na prática, o modelo é amplamente utilizado por sua capacidade de interpretar grandes volumes de texto de forma não supervisionada, sendo eficiente em tarefas de exploração de dados textuais (BLEI; NG; JORDAN, 2003).

2.3.2 Coerência de Tópicos

A avaliação da qualidade dos tópicos gerados por modelos como o LDA pode ser realizada por meio de métricas de coerência. A coerência de tópicos mede o grau de similaridade semântica entre as palavras mais representativas de um mesmo tópico.

Diferentemente de métricas puramente estatísticas, como perplexidade, a coerência tem melhor correlação com a interpretabilidade humana dos tópicos gerados. Por esse motivo, ela

é amplamente utilizada na seleção de modelos de tópicos em aplicações práticas (RÖDER; BOTH; HINNEBURG, 2015; SYED; SPRUIT, 2020).

2.3.3 Seleção do Número de Tópicos (Elbow Method)

A escolha do número de tópicos é um dos principais desafios na modelagem de tópicos, uma vez que esse parâmetro influencia diretamente a granularidade dos resultados.

Uma abordagem comum para auxiliar essa escolha é o método do cotovelo (*elbow method*), que consiste em avaliar uma métrica de qualidade do modelo (como coerência) em função do número de tópicos. O ponto em que o ganho marginal começa a diminuir significativamente é considerado um bom candidato para o número ideal de tópicos.

Esse método é amplamente utilizado como heurística em problemas de aprendizado não supervisionado, especialmente quando não há uma verdade de base conhecida (SYED; SPRUIT, 2020).

2.4 Rotulagem Automática de Comentários com LLMs

A rotulagem de dados é uma etapa crítica em tarefas de aprendizado supervisionado, especialmente em problemas de classificação textual. No entanto, a anotação manual é frequentemente custosa, demorada e sujeita a inconsistências entre anotadores humanos.

Com o avanço dos modelos de linguagem de grande porte (Large Language Models – LLMs), tornou-se possível automatizar parte desse processo por meio de técnicas de inferência direta sobre texto, sem necessidade de treinamento supervisionado específico. Essa abordagem tem sido amplamente explorada em cenários de *data labeling*, *weak supervision* e geração de conjuntos de dados sintéticos (BROWN et al., 2020; WEI et al., 2022a; ZHANG et al., 2023b).

2.4.1 Prompt Engineering (Zero-shot)

O *prompt engineering* consiste na formulação de instruções textuais que guiam o comportamento de modelos de linguagem durante a geração de respostas. Em cenários *zero-shot*, o modelo

é capaz de executar tarefas sem exemplos explícitos de treinamento, baseando-se apenas na descrição da tarefa fornecida no prompt.

Essa abordagem é particularmente útil para tarefas de classificação, onde o modelo deve inferir categorias a partir de descrições textuais. A qualidade dos resultados depende fortemente da estrutura do prompt, da clareza das instruções e da definição adequada das classes possíveis (LIU et al., 2023; WEI et al., 2022b).

No contexto deste trabalho, o prompt foi utilizado para induzir modelos de linguagem a realizar a classificação multirrótulo de comentários, retornando diretamente os rótulos associados a cada instância textual.

2.4.2 Consenso entre Modelos de Linguagem

Apesar da capacidade dos LLMs de realizar tarefas de classificação em modo zero-shot, diferentes modelos podem produzir respostas inconsistentes devido a variações em treinamento, arquitetura e distribuição de probabilidade interna.

Para mitigar esse problema, estratégias baseadas em consenso entre múltiplos modelos têm sido utilizadas. Nessa abordagem, as previsões de diferentes LLMs são agregadas, e o rótulo final é definido com base em critérios de votação ou concordância entre os modelos.

Esse tipo de ensemble entre modelos de linguagem tende a aumentar a robustez das anotações automáticas, reduzindo ruídos e vieses individuais, especialmente em tarefas subjetivas como análise de sentimentos e classificação temática (WANG et al., 2023; ZHANG et al., 2023a).

2.5 Representação Vetorial de Texto

Algoritmos de aprendizado de máquina não são capazes de processar diretamente dados textuais em sua forma bruta. Dessa forma, é necessário transformar os textos em representações numéricas, conhecidas como vetores de características. Esse processo é fundamental em tarefas de Processamento de Linguagem Natural, pois permite que modelos estatísticos e de aprendizado supervisionado operem sobre o conteúdo textual.

Entre as abordagens mais utilizadas para essa transformação estão os modelos *Bag of Words* e TF-IDF, amplamente aplicados em tarefas de classificação de texto e mineração de opiniões (JURAFSKY; MARTIN, 2025; VOGEL, 2021).

2.5.1 Bag of Words

O modelo *Bag of Words* (BoW) representa um documento como um vetor baseado na frequência de ocorrência das palavras, ignorando completamente a ordem em que elas aparecem. Cada posição do vetor corresponde a um termo do vocabulário, e o valor associado indica sua contagem no documento.

Apesar de sua simplicidade, o BoW é amplamente utilizado por sua eficiência computacional e por fornecer uma representação adequada para muitos modelos tradicionais de aprendizado de máquina. No entanto, essa abordagem possui limitações importantes, como a perda de informação sintática e semântica, além da incapacidade de capturar relações contextuais entre palavras (MANNING; RAGHAVAN; SCHÜTZE, 2022).

2.5.2 TF-IDF

O TF-IDF (*Term Frequency-Inverse Document Frequency*) é uma extensão do modelo Bag of Words que busca atribuir pesos mais informativos aos termos. Essa técnica combina a frequência do termo em um documento com a sua frequência inversa no corpus, reduzindo o impacto de palavras muito comuns e pouco discriminativas.

Formalmente, o TF-IDF pode ser interpretado como uma medida de importância de um termo dentro de um documento em relação ao conjunto total de documentos. Dessa forma, termos frequentes em um documento, mas raros no corpus, tendem a receber maior peso.

Esse modelo é amplamente utilizado em sistemas de recuperação de informação e tarefas de classificação textual, sendo especialmente eficaz em combinação com algoritmos lineares de aprendizado supervisionado (JURAFSKY; MARTIN, 2025; SEBASTIANI, 2022).

2.6 Aprendizado de Máquina

O Aprendizado de Máquina (em inglês, *Machine Learning*) é um subcampo da Inteligência Artificial que se dedica ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados, sem necessidade de programação explícita para cada tarefa. Segundo (MITCHELL, 1997), um programa de computador é considerado capaz de aprender a partir de uma experiência E , em relação a uma tarefa T e uma medida de desempenho P , se seu desempenho em T , medido por P , melhora com a experiência E .

No contexto da classificação de textos, o aprendizado de máquina permite que modelos computacionais reconheçam estruturas linguísticas e semânticas, mapeando entradas textuais para categorias ou rótulos relevantes. Essa abordagem é especialmente eficaz quando há grandes volumes de dados rotulados disponíveis para treinamento supervisionado (MURPHY, 2022).

Existem três paradigmas principais em aprendizado de máquina: supervisionado, não supervisionado e por reforço.

2.6.1 Aprendizado Supervisionado

O Aprendizado Supervisionado é uma das abordagens centrais do aprendizado de máquina, caracterizando-se pelo uso de dados rotulados para treinar algoritmos capazes de realizar previsões ou classificações sobre novos exemplos. Neste paradigma, o modelo é apresentado a um conjunto de exemplos de treinamento, composto por pares de entrada e saída, com o objetivo de aprender uma função que generalize corretamente para dados ainda não vistos (MITCHELL, 1997; MURPHY, 2022).

Formalmente, o aprendizado supervisionado busca aproximar uma função $f : X \rightarrow Y$, onde X representa o espaço de entrada (por exemplo, comentários textuais) e Y o espaço de saída (por exemplo, conjuntos de rótulos representando aspectos dos jogos). Com base nos exemplos disponíveis, o algoritmo infere padrões que permitem prever os rótulos corretos para novas entradas (MURPHY, 2022).

Existem dois principais tipos de tarefas supervisionadas: *regressão*, na qual a saída é contínua, e *classificação*, em que a saída é discreta. A tarefa de classificação, por sua vez, pode

ser dividida em dois subtipos: Classificação Multiclasse, em que cada instância é atribuída a uma única classe, e Classificação Multirrótulo, na qual uma mesma instância pode estar associada simultaneamente a múltiplos rótulos. Esta última é particularmente útil em contextos nos quais os dados apresentam múltiplas características ou categorias sobrepostas, exigindo uma modelagem mais flexível e expressiva (LIU et al., 2022).

Diversos algoritmos de aprendizado supervisionado são aplicáveis a problemas de classificação, incluindo o *Naive Bayes*, *Support Vector Machine (SVM)*, Árvore de Decisão, *Random Forests* e Redes Neurais. A escolha do modelo mais adequado depende de diversos fatores, como dimensionalidade dos dados, correlação entre atributos e equilíbrio entre as classes (PALANIVINAYAGAM; EL-BAYEH; DAMASEVČIUS, 2023).

2.6.2 *One vs All*

A técnica *One-vs-All* (também conhecida como *One-vs-Rest* ou OvA) é uma estratégia amplamente utilizada para adaptar classificadores binários a problemas de classificação multiclasse ou multirrótulo. Em sua forma tradicional, consiste em treinar um classificador binário para cada classe, onde os exemplos pertencentes a essa classe são considerados positivos, e todos os demais, negativos (RIFKIN; KLAUTAU, 2004).

No contexto da classificação multirrótulo, essa técnica é especialmente útil, pois permite que cada rótulo seja tratado de forma independente, facilitando a implementação com modelos supervisionados convencionais. Assim, para um conjunto de L rótulos distintos, o modelo *One-vs-All* cria L classificadores binários, cada um responsável por prever a presença ou ausência de um rótulo específico (LIU et al., 2022).

A principal vantagem do OvA está na sua simplicidade e flexibilidade. Ele pode ser aplicado com qualquer algoritmo de classificação binária — como *Support Vector Machine (SVM)*, *Logistic Regression (LR)* ou redes neurais — e oferece boa escalabilidade em conjuntos de dados com número moderado de rótulos. Além disso, sua interpretação é direta: o resultado de cada classificador binário indica a relevância de um rótulo específico para a instância de entrada.

Entretanto, essa abordagem também possui limitações. Por tratar cada rótulo de forma

independente, ela ignora possíveis correlações entre rótulos, o que pode comprometer o desempenho em domínios onde tais dependências são relevantes. Além disso, a presença de classes desbalanceadas pode afetar negativamente a performance dos classificadores individuais (TAREKEGN; GIACOBINI; MICHALAK, 2021).

Apesar dessas limitações, o *One-vs-All* permanece como uma das abordagens mais adotadas em tarefas multirrótulo, sendo uma escolha natural em aplicações de processamento de linguagem natural, como a categorização de textos com múltiplos tópicos ou sentimentos simultâneos (LIU et al., 2022). Sua eficiência, combinada com técnicas adequadas de pré-processamento e balanceamento, pode resultar em classificadores robustos mesmo em cenários complexos.

2.6.3 *One-vs-One*

A técnica *One-vs-One* (OvO) consiste em construir um classificador binário para cada par possível de classes distintas no problema. Para um problema com C classes, isso gera $\frac{C(C-1)}{2}$ classificadores, cada um treinado para distinguir entre duas classes específicas, ignorando as demais. No momento da inferência, cada classificador disputa entre as duas classes que representa, e a classe vencedora geralmente é aquela que “recebe mais votos” ou apresenta o maior número de decisões favoráveis em todos os classificadores binários.

Comparativamente ao *One-vs-All* (OvA), OvO pode oferecer vantagens quando as fronteiras de decisão entre pares de classes são menos complexas ou mais lineares, reduzindo a dificuldade de separar uma classe “contra todas”, o que pode exigir limites mais complexos ou sofrer com desequilíbrios acentuados de classes. Entretanto, OvO também tem desvantagens: exige muito mais classificadores, o que pode implicar maior custo computacional tanto no treinamento quanto na inferência, especialmente para muitos rótulos ou classes.

Esse trade-off foi examinado recentemente, por exemplo, em “Applying Multi-CLASS Support Vector Machines: One-vs-One vs. One-vs-All on the UWF-ZeekDataFall22 Dataset” (KREBS et al., 2024), que encontrou que OvO superou OvA em várias métricas (como acurácia e F1), embora OvA fosse menos custoso. Outro exemplo relevante é o artigo “One-vs-One classification for deep neural networks” (PAWARA et al., 2020), que propõe uma variante de

OvO para redes neurais profundas, mostrando que em certas condições OvO pode levar a melhores fronteiras de decisão e desempenho comparativo em datasets com várias classes (“plant recognition”, etc.).

Portanto, a escolha entre OvO e OvA depende de fatores como número de classes, balanceamento de dados, complexidade das fronteiras entre classes e limitações computacionais. Em muitos casos, pode ser interessante testar ambas para ver qual se adapta melhor ao seu dataset específico.

2.7 Modelos de Inteligência Artificial

Nesta seção, serão apresentados três dos principais algoritmos utilizados para classificação de texto: *Support Vector Machine (SVM)*, *Random Forest(RF)* e *Logistic Regression (LR)*.

2.7.1 *Support Vector Machine (SVM)*

Support Vector Machine (SVM) é uma técnica de aprendizado supervisionado amplamente utilizada em tarefas de classificação e regressão, notadamente eficaz em problemas de alta dimensionalidade, como a classificação de texto (CORTES; VAPNIK, 1995; GUIDO et al., 2024). O principal objetivo de uma SVM é encontrar um hiperplano que maximize a margem entre duas classes de dados, garantindo que as instâncias de cada classe estejam separadas de maneira ótima no espaço de características.

O conceito de margem é central no funcionamento das SVMs. Dados dois conjuntos de pontos pertencentes a classes distintas, a SVM busca o hiperplano que não apenas separa os dados, mas o faz com a maior distância possível entre os pontos mais próximos das classes opostas, denominados vetores de suporte. Esse critério de maximização da margem contribui para a boa capacidade de generalização do modelo (VAPNIK, 1998; GUIDO et al., 2024).

Quando os dados não são linearmente separáveis no espaço original, a SVM utiliza a técnica do *kernel trick*, que consiste em mapear os dados para um espaço de maior dimensionalidade, onde a separação linear se torna possível. Os kernels mais comuns incluem o linear, polinomial,

Radial Basis Function (RBF) e sigmoide. Essa flexibilidade permite que a SVM se adapte a uma ampla variedade de domínios e formatos de dados (SCHÖLKOPF; SMOLA, 2002; MURPHY, 2022).

Na tarefa de classificação de texto, a SVM demonstra desempenho particularmente eficaz, pois lida bem com conjuntos de dados esparsos e com grande número de atributos, como vetores de frequência de termos ou representações baseadas em TF-IDF. Além disso, sua robustez frente a dados não balanceados e ruído textual a torna uma escolha comum em problemas de categorização de documentos (PALANIVINAYAGAM; EL-BAYEH; DAMASEVČIUS, 2023).

Para problemas de classificação multirrótulo, a SVM é frequentemente utilizada em conjunto com estratégias como *One-vs-All* (OvA) ou *Binary Relevance*, que permitem decompor o problema multirrótulo em múltiplas classificações binárias independentes (LIU et al., 2022). Cada classificador SVM é responsável por prever a presença ou ausência de um rótulo específico, permitindo a aplicação eficiente do modelo mesmo em cenários com múltiplas categorias simultâneas.

Apesar de sua eficácia, o uso da SVM exige cuidados com a escolha do kernel e com o ajuste dos hiperparâmetros, como o parâmetro de penalidade C e os parâmetros do kernel, que impactam diretamente a capacidade do modelo de generalizar sem superajustar os dados de treino. Técnicas como validação cruzada e busca em grade são comumente empregadas para otimizar esses parâmetros (GUIDO et al., 2024).

2.7.2 *Random Forest*

O algoritmo *Random Forest* é um método de aprendizado supervisionado baseado em árvores de decisão, proposto por (BIAU; SCORNET, 2024), que combina múltiplas árvores em um processo de agregação conhecido como *ensemble learning*. Sua construção é fundamentada na ideia de que uma coleção de classificadores fracos, quando combinados de maneira adequada, pode resultar em um classificador robusto e de alto desempenho.

Durante o treinamento, o Random Forest gera diversas árvores de decisão, cada uma treinada sobre um subconjunto aleatório dos dados originais, obtido por meio da técnica de *boots-*

trap sampling. Além disso, a cada divisão (ou *split*) em uma árvore, um subconjunto aleatório dos atributos é considerado, o que introduz ainda mais diversidade entre as árvores individuais. Essa combinação de amostragem de dados e variáveis reduz a correlação entre as árvores e contribui para a melhoria da generalização do modelo (BIAU; SCORNET, 2024).

A predição em um modelo *Random Forest* é feita por meio do voto majoritário das árvores, no caso de tarefas de classificação, ou pela média dos valores preditos, em tarefas de regressão. Essa abordagem resulta em modelos mais resistentes a sobreajuste (*overfitting*) e com desempenho competitivo em diversas tarefas supervisionadas (HAN et al., 2023).

No contexto de PLN, o *Random Forest* é frequentemente utilizado para classificação de textos, especialmente em representações vetoriais como TF-IDF, *bag-of-words* ou *embeddings*. Sua capacidade de lidar com atributos irrelevantes e interações não lineares o torna adequado para dados textuais de alta dimensionalidade, como ocorre em comentários de usuários (LI et al., 2022).

Para problemas de classificação multirrótulo, como a categorização de comentários em múltiplos aspectos de jogos (ex.: “jogabilidade”, “história”, “gráficos”), o *Random Forest* é geralmente aplicado em conjunto com estratégias de decomposição do problema, como *Binary Relevance* ou *One-vs-All*. Nessas abordagens, um modelo é treinado para cada rótulo de forma independente, tratando a presença de cada rótulo como uma tarefa binária (TAREKEGN; GIACOBINI; MICHALAK, 2021).

Embora o *Random Forest* não modele explicitamente as dependências entre rótulos, sua robustez frente a ruído e sua capacidade de capturar padrões complexos nos dados o tornam uma alternativa viável e eficaz para sistemas de classificação multirrótulo baseados em texto, especialmente em cenários com rótulos desbalanceados e vocabulário diverso.

2.7.3 *Logistic Regression (LR)*

A Regressão Logística, em inglês *Logistic Regression (LR)* é um algoritmo amplamente utilizado em tarefas de classificação supervisionada. Embora o termo “regressão” possa sugerir uma tarefa de previsão contínua, trata-se de um modelo de classificação binária que estima a proba-

bilidade de uma instância pertencer a uma determinada classe, com base em uma combinação linear de variáveis independentes (MURPHY, 2022).

O modelo logístico utiliza a função logística (sigmoide) para mapear a combinação linear das variáveis para um valor entre 0 e 1, interpretado como uma probabilidade. A equação fundamental do modelo pode ser expressa como:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}}$$

onde $\mathbf{x}$ representa o vetor de atributos da instância, $\mathbf{w}$ é o vetor de pesos aprendido durante o treinamento, e b é o viés (intercepto). O processo de aprendizado consiste na otimização dos parâmetros $\mathbf{w}$ e b com o objetivo de minimizar a função de perda logística, geralmente por meio de métodos de otimização convexa como gradiente descendente (MURPHY, 2022).

A Regressão Logística é especialmente eficaz em tarefas de classificação de texto, devido à sua capacidade de lidar com espaços vetoriais de alta dimensionalidade, como aqueles produzidos por representações TF-IDF ou *word embeddings*. Sua natureza linear, combinada com regularização (como L1 ou L2), oferece controle sobre o ajuste do modelo e evita o sobreajuste, o que é crucial em cenários com dados esparsos (LI et al., 2022).

No cenário de classificação multirrótulo, a Regressão Logística é comumente utilizada em conjunto com estratégias de decomposição, como *Binary Relevance* ou *One-vs-All*, em que um modelo logístico independente é treinado para cada rótulo. Isso permite que o modelo aprenda a prever a probabilidade de presença de cada rótulo de maneira individual, viabilizando aplicações em tarefas como categorização de comentários de usuários com múltiplos aspectos simultâneos (HAN et al., 2023).

Apesar de sua simplicidade, a Regressão Logística continua sendo uma escolha sólida para classificação textual, fornecendo resultados interpretáveis, eficientes e competitivos, especialmente quando combinada com técnicas adequadas de engenharia de atributos e pré-processamento textual.

2.8 Técnicas de Pré-Processamento

O pré-processamento textual é uma etapa fundamental no desenvolvimento de sistemas baseados em Processamento de Linguagem Natural (PLN), especialmente em tarefas de classificação de texto. Essa etapa tem como objetivo transformar os dados brutos em uma representação mais estruturada e adequada para algoritmos de aprendizado de máquina, reduzindo ruídos e padronizando a entrada (SIINO; TINNIRELLO; La Cascia, 2024).

Entre as técnicas mais comuns de pré-processamento, destaca-se inicialmente a normalização do Texto, que inclui a conversão de todos os caracteres para minúsculas (*lowercasing*), a remoção de pontuação, números e caracteres especiais, além da eliminação de espaços em branco redundantes. Essas operações visam reduzir a variabilidade lexical, facilitando a generalização dos modelos (JURAFSKY; MARTIN, 2025).

Outra prática essencial é a remoção de palavras vazias (*stopwords*), que consiste na exclusão de palavras de alta frequência, mas baixo valor semântico, como preposições, pronomes e artigos. A presença dessas palavras pode gerar ruído na modelagem, especialmente em tarefas de classificação, onde o conteúdo informativo do texto é crítico (SIINO; TINNIRELLO; La Cascia, 2024).

A radicalização das palavras também é amplamente utilizada, por meio de técnicas como *stemming* e *lemmatization*. O *stemming* aplica regras heurísticas para remover sufixos das palavras, reduzindo-as a suas raízes morfológicas, ainda que nem sempre linguisticamente corretas (SIINO; TINNIRELLO; La Cascia, 2024). Já a lematização busca identificar o lema correto de uma palavra, considerando sua classe gramatical e forma canônica, oferecendo maior precisão semântico-linguística (spaCy, 2025).

Para a representação vetorial dos textos, é comum aplicar filtros por classes gramaticais, como substantivos, verbos e adjetivos, que geralmente carregam maior densidade informacional. Essa filtragem pode ser realizada por meio de *Part-of-Speech Tagging* (POS), que atribui rótulos morfossintáticos às palavras com base em seu contexto na sentença (CHICHE; YITAGESU, 2022).

Por fim, dependendo da tarefa, pode-se aplicar técnicas mais específicas como a normalização

de acentuação, a correção ortográfica, ou até mesmo transformações linguísticas mais avançadas, como substituição de sinônimos ou detecção de neologismos. A escolha e a combinação das técnicas de pré-processamento devem ser guiadas pelas características do corpus, pelos objetivos da tarefa e pelo tipo de modelo a ser empregado.

No caso de dados extraídos de comentários de usuários, como em jogos digitais, o pré-processamento assume um papel ainda mais relevante, devido à presença frequente de abreviações, erros ortográficos, emoticons e linguagem informal. A correta aplicação dessas técnicas é determinante para a construção de representações textuais mais eficazes e, conseqüentemente, para o desempenho do sistema de classificação.

2.9 Validação de Modelos

A validação de modelos é uma etapa essencial no desenvolvimento de sistemas de aprendizado de máquina, pois tem como objetivo estimar a capacidade de generalização do modelo em dados não vistos. Em outras palavras, busca-se avaliar o desempenho do modelo de forma robusta, reduzindo o risco de sobreajuste (*overfitting*) e viés na avaliação.

Entre as abordagens mais utilizadas para esse fim está a validação cruzada, amplamente aplicada em problemas de classificação e regressão (KOHAVI, 1995; RODRIGUEZ; PEREZ; LOZANO, 2010; STEVENS, 2023).

2.9.1 Validação Cruzada

A validação cruzada (*cross-validation*) é uma técnica que consiste em dividir o conjunto de dados em k subconjuntos (folds). O modelo é treinado em $k - 1$ folds e avaliado no fold restante, repetindo o processo k vezes. Ao final, o desempenho é obtido pela média dos resultados em cada iteração.

Essa abordagem permite uma avaliação mais estável do modelo, pois cada amostra é utilizada tanto para treino quanto para teste em diferentes iterações. Entre as variantes mais comuns está a validação cruzada do tipo k -fold, amplamente utilizada na literatura devido

ao seu equilíbrio entre custo computacional e confiabilidade estatística (DEVELOPERS, 2024; BROWNE et al., 2020).

Apesar de sua eficiência, a validação cruzada pode apresentar limitações em cenários com forte desbalanceamento de classes, o que pode comprometer a representatividade das partições geradas.

2.9.2 Validação Cruzada Estratificada Multirrótulo

Em problemas multirrótulo, cada instância pode pertencer simultaneamente a múltiplas classes, o que torna a divisão dos dados mais complexa. Nesses casos, a validação cruzada tradicional pode gerar folds com distribuições de rótulos inconsistentes, prejudicando a avaliação do modelo.

Para mitigar esse problema, utiliza-se a validação cruzada estratificada multirrótulo, que busca preservar a distribuição dos rótulos em cada partição. Essa estratégia é uma extensão da estratificação clássica, adaptada para lidar com múltiplos rótulos por amostra.

Algoritmos como o `MultilabelStratifiedKFold` foram propostos especificamente para esse cenário, garantindo maior consistência estatística entre os folds e reduzindo o viés da avaliação em datasets desbalanceados (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011; TIITTANEN et al., 2022).

Essa abordagem é especialmente importante em aplicações reais de processamento de linguagem natural, onde a distribuição de classes tende a ser altamente desigual e correlacionada.

2.10 Otimização de Hiperparâmetros

Modelos de aprendizado de máquina possuem um conjunto de parâmetros ajustáveis que não são aprendidos diretamente durante o processo de treinamento, conhecidos como hiperparâmetros. Esses valores influenciam diretamente o desempenho do modelo, afetando sua capacidade de generalização e sua sensibilidade a dados de entrada.

A escolha adequada de hiperparâmetros é uma etapa fundamental no desenvolvimento de modelos preditivos, uma vez que diferentes configurações podem levar a variações significativas

no desempenho final. Por esse motivo, técnicas sistemáticas de otimização são amplamente utilizadas na literatura (BERGSTRA; BENGIO, 2012; SNOEK; LAROCHELLE; ADAMS, 2012; PROKHORENKOVA et al., 2021).

2.10.1 Grid Search

O *Grid Search* é uma estratégia de busca exaustiva de hiperparâmetros na qual todas as combinações possíveis de um conjunto pré-definido de valores são avaliadas.

Essa abordagem consiste em definir um espaço discreto de busca para cada hiperparâmetro e treinar o modelo para todas as combinações possíveis desses valores. O desempenho de cada configuração é então avaliado por meio de uma métrica específica, sendo a melhor combinação selecionada com base no melhor resultado médio obtido.

Embora o Grid Search garanta a exploração completa do espaço definido, ele pode se tornar computacionalmente custoso à medida que o número de hiperparâmetros e valores possíveis aumenta. Ainda assim, é amplamente utilizado devido à sua simplicidade e reprodutibilidade, especialmente em cenários com espaço de busca moderado (BERGSTRA; BENGIO, 2012).

2.11 Métricas de Avaliação

A escolha de métricas adequadas é essencial para avaliar corretamente o desempenho de modelos em tarefas de classificação multirrótulo. Diferentemente da classificação tradicional, onde cada instância pertence a uma única classe, na classificação multirrótulo cada instância pode estar associada simultaneamente a diversos rótulos. Isso demanda métricas específicas que consigam refletir tanto acertos parciais quanto erros múltiplos (HAN et al., 2023).

2.11.1 Precisão (*Precision*)

A Precisão mede a proporção de rótulos previstos como positivos que são realmente corretos. Em cenários multirrótulo, pode-se calcular a precisão para cada rótulo separadamente ou aplicar médias micro, macro ou ponderada, conforme o objetivo da análise (HAN et al., 2023). A

equação geral é dada por:

$$\text{Precisão} = \frac{VP}{VP + FP}$$

onde VP são os verdadeiros positivos e FP os falsos positivos.

2.11.2 Revocação (*Recall*)

A *Revocação*, também conhecida como sensibilidade, indica a capacidade do modelo de identificar corretamente os rótulos relevantes. É definida como:

$$\text{Revocação} = \frac{VP}{VP + FN}$$

onde VP são os verdadeiros positivos e FN representa os falsos negativos. Assim como a precisão, pode ser calculada por rótulo ou por agregações.

2.11.3 F1-Score

O *F1-Score* é uma métrica amplamente utilizada para avaliação de modelos de classificação, especialmente quando existe um desequilíbrio entre as classes ou quando se deseja balancear a precisão e a revocação em uma única medida. O F1-Score é definido como a média harmônica entre precisão e revocação:

$$F_1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}$$

Em problemas de classificação multirrótulo, existem diferentes formas de calcular o F1-Score global, variando conforme a agregação das métricas por classe. As principais são as versões *micro*, *macro* e *ponderada (weighted)* (HAN et al., 2023).

F1-Score Micro

A média *micro* agrega os verdadeiros positivos, falsos positivos e falsos negativos de todas as classes antes de calcular a precisão e a revocação. Isso significa que todas as decisões

(independentemente da classe) são ponderadas igualmente, favorecendo classes majoritárias.

$$F_1^{micro} = 2 \cdot \frac{\sum_l VP_l}{\sum_l VP_l + \sum_l FP_l + \sum_l FN_l}$$

Essa abordagem é útil quando se deseja avaliar o desempenho geral do modelo considerando todos os rótulos como igualmente importantes em termos de ocorrência.

F1-Score Macro

A média *macro* calcula o F1-Score individualmente para cada rótulo e, em seguida, computa a média aritmética desses valores. Isso dá o mesmo peso a todas as classes, independentemente da frequência com que aparecem nos dados.

$$F_1^{macro} = \frac{1}{L} \sum_{l=1}^L F_1^l$$

Embora seja sensível a classes com poucas amostras, o F1 macro é apropriado quando se deseja uma avaliação equilibrada entre rótulos frequentes e raros.

Considerações

A escolha entre as diferentes médias do F1-Score depende do objetivo da tarefa. Em cenários com classes desbalanceadas ou quando todos os rótulos são igualmente relevantes, a média macro pode ser mais informativa. Já a média micro pode ser útil para medir a performance global.

2.11.4 Hamming Loss

A *Hamming Loss* é uma métrica amplamente utilizada em classificação multirrótulo. Ela calcula a fração de rótulos incorretamente previstos (tanto falsos positivos quanto falsos negativos) em relação ao total de rótulos possíveis (TAREKEGN; GIACOBINI; MICHALAK, 2021). Sua fórmula é:

$$\text{Hamming Loss} = \frac{1}{N \times L} \sum_{i=1}^N \sum_{j=1}^L \mathbb{I}(y_{ij} \neq \hat{y}_{ij})$$

onde N representa o número de instâncias, L o número total de rótulos, y_{ij} o valor real e $\hat{y}_{ij}$ o valor previsto para o rótulo j na instância i . A função $\mathbb{I}$ vale 1 quando a predição está incorreta e 0 caso contrário. Quanto menor o valor de Hamming Loss, melhor o desempenho do modelo.

Capítulo 3

Trabalhos Relacionados

A classificação de textos é uma das tarefas centrais do Processamento de Linguagem Natural (PLN), consistindo na atribuição automática de rótulos a documentos textuais com base em seu conteúdo semântico. Com o avanço das técnicas de aprendizado de máquina e o aumento da disponibilidade de dados digitais, essa área tem evoluído significativamente, permitindo aplicações em diversos domínios, como análise de sentimentos, recomendação de conteúdo e mineração de opiniões.

Neste capítulo são apresentados trabalhos relacionados à classificação multirrótulo em textos, com ênfase em aplicações em língua portuguesa, além de abordagens envolvendo aprendizado supervisionado, representação textual e construção automática de conjuntos de dados. Por fim, são discutidas as lacunas identificadas na literatura que motivam este trabalho.

3.1 Classificação Multirrótulo em Reviews de Restaurantes Brasileiros

A tarefa de classificação multirrótulo tem sido amplamente explorada no contexto de análise de opiniões em plataformas digitais. Em particular, (MELO, 2021) realizaram uma análise abrangente de comentários em português coletados de seis plataformas de avaliação online (Facebook, Foursquare, Google Reviews, TripAdvisor, Yelp e Zomato) relacionados a restaurantes brasi-

leiros com estrela Michelin. O estudo identificou variações significativas entre as plataformas em termos de características linguísticas, estrutura dos comentários e utilidade das avaliações, evidenciando os desafios associados à análise de dados provenientes de redes sociais.

A partir desse corpus, (NETO; MELO, 2023) investigaram a aplicação de modelos de aprendizado supervisionado para classificação multirrótulo dos comentários, comparando algoritmos como Regressão Logística, *Random Forest* e estratégias de decomposição *One-vs-All*. Os resultados demonstraram que abordagens baseadas em representações TF-IDF combinadas com classificadores tradicionais são eficazes para identificar simultaneamente múltiplos aspectos nos textos, como qualidade do serviço, ambiente e experiência gastronômica.

Esses estudos evidenciam que, mesmo em cenários de linguagem natural informal e complexa, modelos clássicos de aprendizado supervisionado permanecem competitivos quando associados a boas estratégias de pré-processamento e representação textual.

3.2 Modelos Supervisionados para Classificação de Texto

Modelos de aprendizado supervisionado, como *Support Vector Machines*, *Random Forests* e Regressão Logística, são amplamente utilizados em tarefas de classificação textual devido à sua eficiência computacional, interpretabilidade e desempenho competitivo em representações vetoriais esparsas, como Bag of Words e TF-IDF.

Mesmo com o avanço de arquiteturas baseadas em transformadores, estudos recentes indicam que modelos tradicionais ainda apresentam resultados robustos em diversos cenários de PLN, especialmente quando aplicados a conjuntos de dados de tamanho moderado e com pré-processamento adequado.

3.3 Construção e Rotulagem Automática de Dados

A criação de conjuntos de dados rotulados é uma etapa essencial, porém frequentemente custosa em tarefas de aprendizado supervisionado. Nesse contexto, abordagens recentes têm explorado métodos automáticos e semi-automáticos de anotação de dados.

Entre essas estratégias, destaca-se o uso de modelagem de tópicos, como Latent Dirichlet Allocation (LDA), para identificar estruturas semânticas iniciais nos textos. Além disso, Modelos de Linguagem (LLMs) têm sido utilizados em abordagens *zero-shot* e *few-shot* para rotulagem automática, reduzindo a dependência de anotação manual.

Nesse cenário, abordagens de consenso entre múltiplos modelos têm ganhado destaque. O trabalho de (KAMEN; KAMEN, 2025) propõe o uso de um conjunto de LLMs como um comitê de decisão, no qual a classificação final é obtida por meio de votação majoritária. Os resultados demonstram que o consenso entre modelos melhora a robustez das predições e reduz inconsistências individuais, sendo uma estratégia simples e eficiente para cenários de classificação de conteúdo.

Essa linha de pesquisa reforça a ideia de que múltiplos modelos podem atuar como uma forma de “crowd virtual”, aumentando a confiabilidade dos rótulos gerados automaticamente e reduzindo o custo de anotação manual.

3.4 Síntese e Lacunas da Literatura

Apesar dos avanços na aplicação de PLN em análise de opiniões, observa-se que a maioria dos estudos se concentra em domínios como restaurantes, e-commerce e redes sociais, com predominância da língua inglesa.

No contexto brasileiro, ainda há uma lacuna relevante na aplicação de classificação multirótulo em domínios como jogos digitais, especialmente considerando abordagens que integrem construção automática de datasets, modelos supervisionados clássicos e técnicas modernas de rotulagem com LLMs.

Além disso, embora existam estudos sobre LLMs para anotação e métodos de agregação

de rótulos, ainda são escassas as abordagens que utilizam de forma estruturada a combinação entre modelagem de tópicos, LLMs e consenso por votação para construção de datasets em português.

Dessa forma, este trabalho se diferencia ao propor uma abordagem integrada que combina modelagem de tópicos (LDA), rotulagem automática baseada em modelos de linguagem e mecanismos de consenso por votação para construção do conjunto de dados, seguida da avaliação de modelos supervisionados clássicos para classificação multirrótulo de comentários de jogos em português.

Capítulo 4

Desenvolvimento

4.1 Obtenção do Dataset

O objetivo desta etapa foi construir um dataset de comentários de jogos digitais em português brasileiro que pudesse ser utilizado posteriormente no treinamento e avaliação de modelos de classificação multirrótulo.

4.1.1 Seleção dos Jogos

Inicialmente, foi utilizada a plataforma AppBrain para identificar jogos populares disponíveis na Google Play Store brasileira. Como critérios de seleção, foram considerados apenas jogos com mais de 5 milhões de instalações e pertencentes a gêneros distintos. A exigência de um elevado número de instalações teve como objetivo garantir uma quantidade expressiva de avaliações disponíveis para coleta, enquanto a diversidade de gêneros buscou reduzir vieses associados a um único tipo de jogo e aumentar a variedade temática dos comentários obtidos.

Ao final dessa etapa, foram selecionados 17 jogos para análise preliminar.

Posteriormente, a biblioteca `google_play_scraper` foi utilizada para obter informações detalhadas sobre cada jogo. Durante essa consulta com a função `app()`, foram aplicados filtros para idioma português (`lang='pt'`) e país Brasil (`country='br'`), garantindo que os dados coletados refletissem o contexto do público brasileiro.

Entre as informações retornadas pela biblioteca estava a quantidade de avaliações (*reviews*) associadas a cada jogo. Como cada avaliação está vinculada a um comentário de usuário, essa métrica foi utilizada como uma estimativa da quantidade de comentários potencialmente disponíveis para coleta.

Os jogos foram então ordenados de forma decrescente pela quantidade de avaliações disponíveis. Os dez primeiros colocados foram selecionados para compor o conjunto de coleta de comentários, pois apresentavam maior potencial para obtenção de dados em português brasileiro. A Tabela 4.1 apresenta os jogos inicialmente analisados e suas respectivas quantidades de avaliações.

Tabela 4.1: Jogos selecionados inicialmente e quantidade de avaliações disponíveis

Posição	Jogo	Quantidade de Avaliações
1	Free Fire	10.392.476
2	Roblox	3.753.114
3	Clash Royale	2.481.513
4	Subway Surfers	2.160.451
5	Clash of Clans	1.670.043
6	8 Ball Pool	1.580.316
7	Brawl Stars	1.489.966
8	Candy Crush Saga	991.009
9	eFootball	853.893
10	Coin Master	510.686
11	Minecraft	419.675
12	Grand Theft Auto: San Andreas	88.484
13	Block Blast!	57.571
14	Terraria	21.398
15	Stardew Valley	19.550
16	Last War: Survival Game	16.590
17	Geometry Dash	14.869

A partir dessa ordenação, foram selecionados os dez jogos com maior quantidade de avaliações para a etapa de coleta dos comentários.

Tabela 4.2: Jogos selecionados para coleta de comentários e seus gêneros principais

Jogo	Gênero Principal	Subgênero / Característica
Free Fire	Ação	Battle Royale / Tiro Multiplayer
Roblox	Simulação	Sandbox / Conteúdo Gerado por Usuários
Clash Royale	Estratégia	Tower Defense / Cartas
Subway Surfers	Ação	Runner Infinito
Clash of Clans	Estratégia	Construção e Batalha
8 Ball Pool	Esportes	Bilhar Multiplayer
Brawl Stars	Ação	MOBA / Tiro Multiplayer
Candy Crush Saga	Quebra-cabeças	Match-3
eFootball	Esportes	Futebol Multiplayer
Coin Master	Cassino	Cassino e Aventura

Pode-se observar que os jogos selecionados abrangem diferentes gêneros e estilos de jogabilidade, incluindo ação, estratégia, esportes, quebra-cabeças, simulação e cassino. Essa diversidade contribui para a obtenção de comentários relacionados a múltiplos aspectos da experiência do jogador, reduzindo o risco de que o conjunto de dados fique excessivamente associado às características de um único gênero de jogo.

4.1.2 Coleta dos Comentários

Após a seleção dos dez jogos com maior quantidade de avaliações disponíveis, foi realizada a coleta dos comentários utilizando a biblioteca `google_play_scraper`. Para essa etapa foi utilizada a função `reviews()`, responsável por recuperar avaliações publicadas na Google Play Store.

Durante a coleta foram aplicados filtros de idioma (`lang='pt'`) e país (`country='br'`), garantindo que apenas comentários em português brasileiro fossem obtidos. Adicionalmente, foi utilizado o parâmetro (`sort=Sort.MOST_RELEVANT`), que prioriza avaliações consideradas mais relevantes pela própria plataforma.

O objetivo inicial era coletar 200 mil comentários para cada um dos dez jogos selecionados. Após a coleta, os comentários de cada jogo foram ordenados de acordo com o tamanho de caracteres e foram selecionados os 20 mil comentários mais extensos. Essa estratégia foi adotada porque comentários maiores tendem a conter descrições mais detalhadas da experiência do

usuário, frequentemente abordando múltiplos aspectos do jogo em uma única avaliação. Dessa forma, espera-se obter informações mais ricas para as etapas posteriores de modelagem de tópicos e classificação multirrótulo.

Além disso, foi estabelecido um limite de 20 mil comentários por jogo para compor o dataset final. Essa decisão teve como objetivo evitar que jogos com maior volume de avaliações dominassem o conjunto de dados, mantendo uma distribuição balanceada entre os títulos analisados.

Durante a coleta verificou-se que nem todos os jogos possuíam 200 mil comentários disponíveis em português brasileiro. Ao final do processo foram obtidos 1.742.000 comentários, distribuídos entre os dez jogos selecionados. A Tabela 4.3 apresenta a quantidade de comentários recuperados para cada jogo.

Tabela 4.3: Quantidade de comentários coletados por jogo

Jogo	Comentários Coletados
Free Fire	200.000
Roblox	200.000
Clash Royale	200.000
Subway Surfers	200.000
Clash of Clans	200.000
8 Ball Pool	157.500
Brawl Stars	99.000
Candy Crush Saga	200.000
eFootball	85.500
Coin Master	200.000
Total	1.742.000

Pode-se observar que alguns jogos apresentaram quantidade de comentários inferior ao limite de coleta estabelecido, sendo o eFootball o título com menor volume de dados disponíveis, totalizando aproximadamente 85 mil comentários. Ainda assim, todos os jogos possuíam quantidade suficiente de comentários para que fossem selecionados os 20 mil textos mais extensos de cada título.

Ao final dessa etapa, foi construído um dataset balanceado contendo 200 mil comentários, distribuídos uniformemente entre os dez jogos selecionados, com 20 mil comentários provenientes de cada jogo.

4.1.3 Análise Exploratória do Dataset

Após a etapa de coleta e seleção dos comentários, foi realizada uma análise exploratória com o objetivo de compreender as características gerais do conjunto de dados construído.

Para avaliar as características do conjunto de dados obtido, foi realizada uma análise do tamanho dos comentários em número de caracteres. A Tabela 4.4 apresenta as principais estatísticas descritivas observadas.

Tabela 4.4: Estatísticas descritivas do tamanho dos comentários

Métrica	Valor
Quantidade de comentários	200.000
Menor comentário	96 caracteres
Maior comentário	2.259 caracteres
Média	376,08 caracteres
Mediana	392 caracteres

Observa-se que o tamanho dos comentários variou entre 96 e 2.259 caracteres. A média encontrada foi de 376,08 caracteres, enquanto a mediana foi de 392 caracteres. A proximidade entre média e mediana sugere uma distribuição relativamente equilibrada, embora exista a presença de comentários significativamente maiores que a maioria dos registros.

A Figura 4.1 apresenta a distribuição dos comentários por intervalos de tamanho.

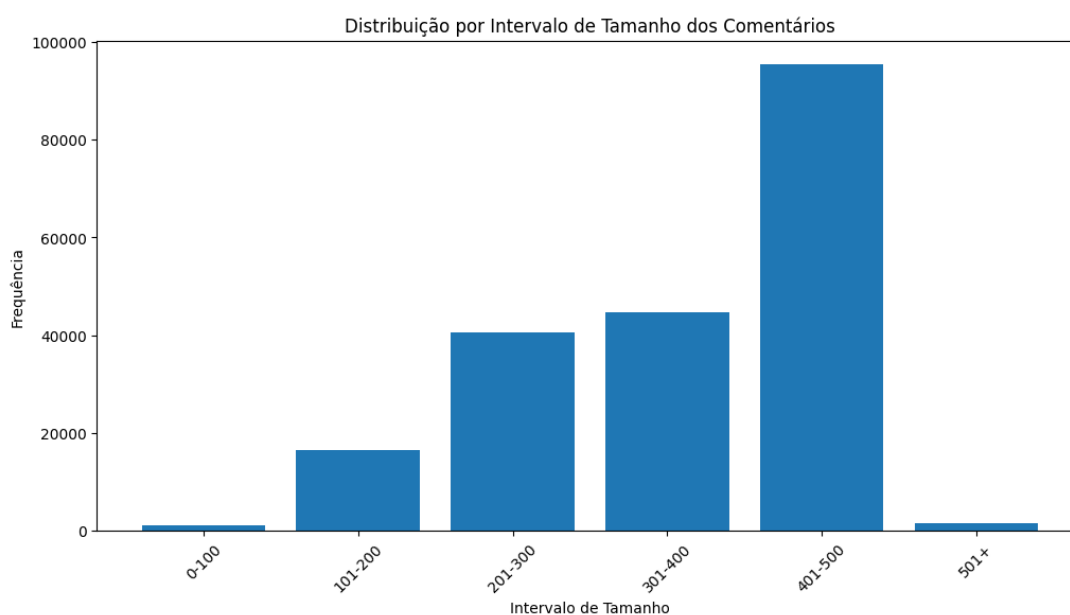


Figura 4.1: Distribuição dos comentários por intervalo de tamanho.

Tabela 4.5: Distribuição dos comentários por intervalo de tamanho

Intervalo (caracteres)	Quantidade	Percentual
0 – 100	1.163	0,58%
101 – 200	16.576	8,29%
201 – 300	40.585	20,29%
301 – 400	44.767	22,38%
401 – 500	95.412	47,71%
501+	1.497	0,75%
Total	200.000	100,00%

A Tabela 4.5 e a Figura 4.1 apresentam a distribuição dos comentários de acordo com seu tamanho em caracteres.

Observa-se que aproximadamente 47,7% dos comentários encontram-se no intervalo entre 401 e 500 caracteres, representando a faixa mais frequente do conjunto de dados. Somando-se os intervalos entre 201 e 500 caracteres, verifica-se que mais de 90% dos comentários estão concentrados nessa região da distribuição.

Por outro lado, comentários muito curtos são raros no conjunto final. Apenas 0,58% dos registros possuem até 100 caracteres, enquanto comentários com mais de 500 caracteres representam 0,75% do dataset. Esse comportamento confirma que o procedimento de seleção dos maiores comentários de cada jogo produziu um conjunto de dados composto majoritariamente por avaliações extensas e detalhadas.

A predominância de comentários longos é particularmente relevante para este trabalho, pois textos mais extensos tendem a apresentar múltiplos aspectos da experiência do usuário, como jogabilidade, gráficos, desempenho, progressão e monetização simultaneamente. Dessa forma, o dataset construído mostra-se adequado para tarefas de classificação multirrótulo, nas quais um mesmo comentário pode ser associado a mais de um tópico.

4.2 Definição dos Rótulos

Como não existia um conjunto de categorias previamente definido para os comentários coletados, foi utilizada modelagem de tópicos para identificar temas recorrentes no dataset e, a partir deles, construir os rótulos empregados na tarefa de classificação.

As subseções a seguir descrevem o processo utilizado para obtenção e consolidação desses rótulos.

4.2.1 Modelagem de Tópicos com LDA

Os experimentos de modelagem de tópicos foram realizados utilizando o algoritmo LDA (*Latent Dirichlet Allocation*), com apoio das bibliotecas `spaCy` e `Gensim` para processamento textual, construção dos corpus e treinamento dos modelos.

Pré-processamento dos Comentários

Todos os comentários foram processados utilizando o modelo `pt_core_news_sm` da biblioteca `spaCy`. Durante as primeiras execuções do LDA, observou-se que diversos tópicos eram dominados por termos específicos de determinados jogos, como “brawl”, “brawler”, “clash” e “vila”, além de abreviações comuns em comentários de aplicativos, como “vc”, “pq”, “q” e “ñ”. Para reduzir esse efeito, foi criada uma lista personalizada de *stopwords*, utilizada em todos os pré-processamentos.

Foram avaliadas cinco estratégias de pré-processamento:

- **Simples:** Retorna o lema das palavras, remove stopwords e termos não alfabéticos;
- **Substantivo:** Mesmas etapas do pré-processamento simples, porém mantendo apenas substantivos e removendo as stopwords personalizadas;
- **Substantivo e Adjetivo:** Mesmas etapas do pré-processamento simples, porém mantendo substantivos e adjetivos e removendo as stopwords personalizadas;
- **Substantivo, Adjetivo e Len:** Mesmas etapas do pré-processamento simples, porém mantendo substantivos e adjetivos, removendo as stopwords personalizadas, e termos com até dois caracteres;
- **Substantivo, Adjetivo e Remoção de Acentos:** Mesmas etapas do pré-processamento simples, porém mantendo substantivos e adjetivos, removendo as stopwords personalizadas, e normalizando caracteres acentuados;

Para cada estratégia, todos os comentários foram processados e armazenados para utilização nos experimentos subsequentes.

Construção do Corpus

Após o pré-processamento, foi criado um dicionário contendo os termos presentes em cada conjunto de comentários utilizando a estrutura `Dictionary` da biblioteca *Gensim*. Em seguida, os comentários foram convertidos para a representação *Bag of Words*.

Também foram avaliadas diferentes configurações dos parâmetros `no_below` e `no_above`, responsáveis pela remoção de termos excessivamente raros ou frequentes. Adicionalmente, alguns experimentos utilizaram *n-grams* para representar expressões compostas recorrentes.

Para reduzir o tempo de processamento, as estruturas intermediárias geradas foram armazenadas em cache e reutilizadas quando necessário.

Treinamento dos Modelos LDA

Os modelos foram treinados utilizando a implementação `LdaMulticore` da biblioteca *Gensim*. Para explorar diferentes configurações, foi realizada uma busca em grade considerando as combinações apresentadas na Tabela 4.6.

Tabela 4.6: Parâmetros avaliados na busca em grade dos modelos LDA

Parâmetro	Valores Avaliados
Número de tópicos (<i>n_topics</i>)	7, 8, 9, 10, 11, 12, 13, 14, 15
Pré-processamento	Simples; Substantivo; Substantivo e Adjetivo; Substantivo, Adjetivo e Filtro de Comprimento; Substantivo, Adjetivo e Remoção de Acentos
<i>no_below</i>	5, 10
<i>no_above</i>	0,3; 0,4
Uso de bigrams	Sim; Não

Para cada combinação de parâmetros foi treinado um modelo independente e seus tópicos foram armazenados para análise posterior.

Avaliação por Coerência

Após o treinamento, cada modelo foi avaliado utilizando a métrica de coerência c_v , calculada por meio da classe `CoherenceModel` da biblioteca *Gensim*.

Os valores de coerência foram utilizados como principal critério para comparação entre modelos e seleção das configurações mais adequadas para as etapas seguintes de definição dos rótulos.

4.2.2 Definição da Quantidade de Tópicos

Antes da definição dos rótulos finais, foi necessário determinar qual quantidade de tópicos produzia resultados mais coerentes para o conjunto de comentários analisado.

Inicialmente, foi realizada uma primeira rodada de experimentos utilizando apenas o pré-processamento Simples. Foram avaliadas quantidades de tópicos entre 7 e 15. Para cada quantidade, foram testadas oito combinações de parâmetros envolvendo os valores de `no_below`, `no_above` e utilização de *n-grams*, totalizando 72 modelos treinados.

Para cada quantidade de tópicos, foi selecionado o modelo que apresentou o maior valor de coerência. Os resultados obtidos são apresentados na Figura 4.2.

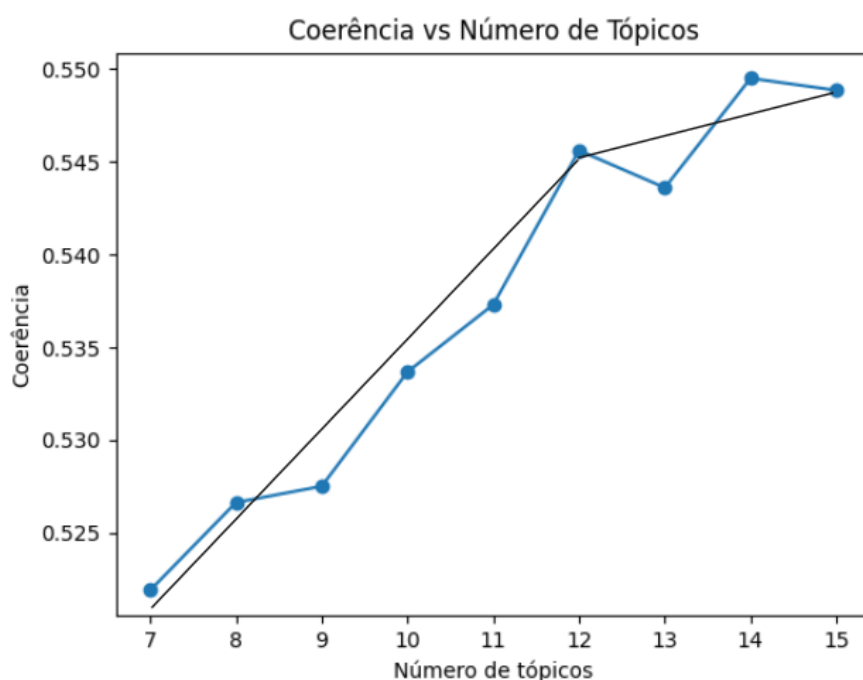


Figura 4.2: Maior coerência obtida para cada quantidade de tópicos.

Observa-se que a coerência cresce de forma consistente conforme o número de tópicos aumenta. Entretanto, a partir de 12 tópicos os ganhos tornam-se significativamente menores. Utilizando a regra do cotovelo (*elbow method*), identificou-se que o ponto de inflexão ocorre próximo de 12 tópicos, indicando um equilíbrio entre qualidade dos tópicos e complexidade do modelo.

Com base nessa análise, foi definido o valor de 12 tópicos para os experimentos subsequentes.

Após essa etapa, uma nova rodada de treinamento foi realizada utilizando os cinco métodos de pré-processamento descritos anteriormente. Foram avaliadas as quantidades de tópicos 7, 10, 12 e 15, combinadas com os demais parâmetros da grade de busca, totalizando 160 modelos treinados.

Para cada estratégia de pré-processamento foi gerado um gráfico contendo os valores de coerência obtidos para as diferentes quantidades de tópicos, conforme apresentado na Figura 4.3.

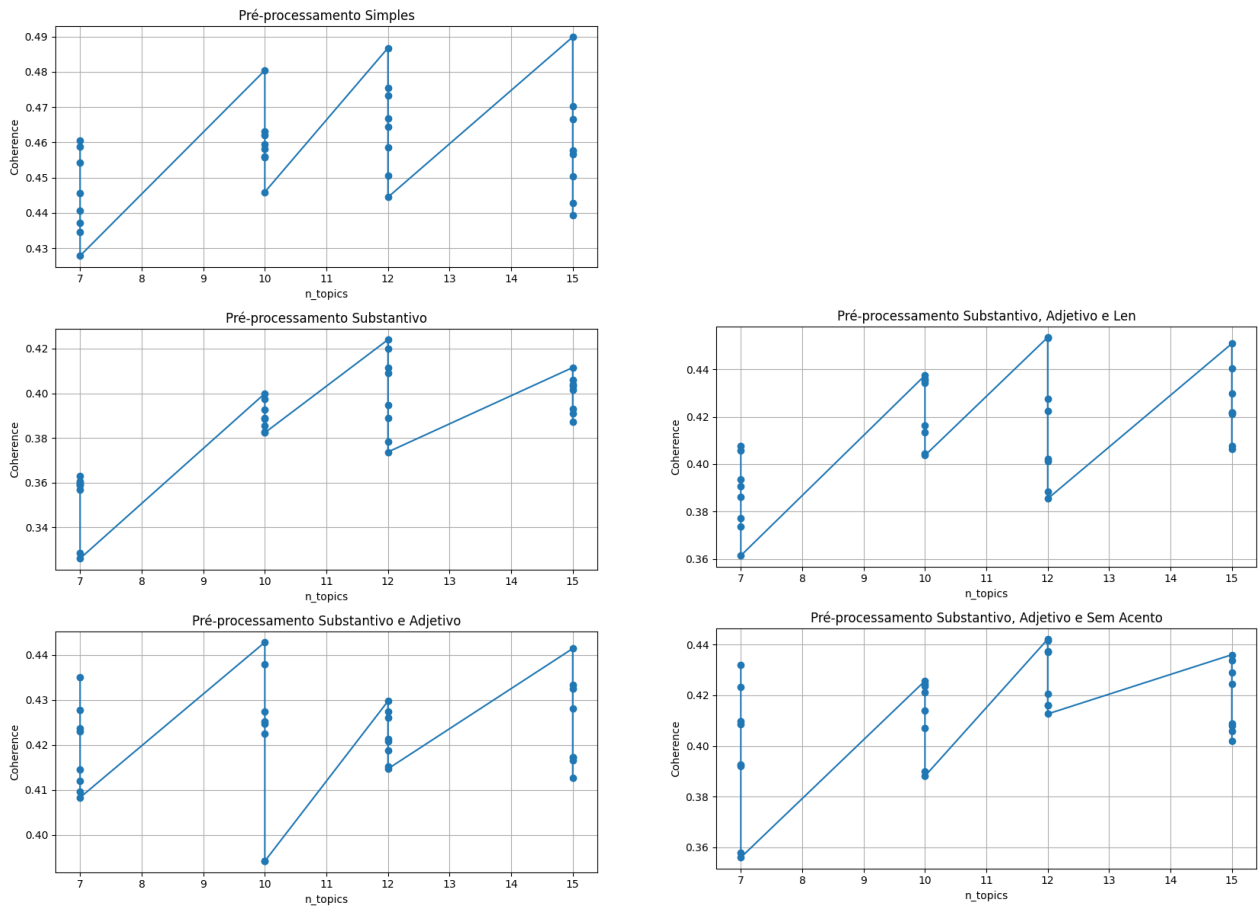


Figura 4.3: Resultados de coerência para os diferentes pré-processamentos avaliados.

Os resultados mostraram que os modelos com 12 tópicos estiveram consistentemente entre os melhores resultados obtidos.

Dessa forma, foi mantida a escolha de 12 tópicos para a etapa de interpretação manual dos tópicos gerados e definição dos rótulos finais utilizados neste trabalho.

4.2.3 Interpretação dos Tópicos e Definição dos Rótulos

Após a definição de 12 tópicos como configuração padrão, foi selecionado o modelo de maior coerência para cada uma das cinco estratégias de pré-processamento avaliadas. Em seguida, os tópicos gerados foram analisados manualmente a partir das palavras com maior peso em cada distribuição.

Durante essa análise observou-se que diversos tópicos apresentavam forte similaridade semântica entre diferentes pré-processamentos. Temas relacionados a monetização, problemas técnicos, progressão, qualidade do jogo e suporte ao usuário surgiram de forma recorrente em praticamente todos os modelos avaliados.

Também foram identificados tópicos excessivamente específicos de determinados jogos, como termos relacionados a futebol, Clash of Clans ou mecânicas particulares de alguns títulos. Além disso, alguns tópicos apresentaram baixa interpretabilidade, contendo predominantemente palavras genéricas, abreviações ou termos sem uma temática claramente definida. Esses tópicos foram classificados como ruído e descartados durante a definição dos rótulos finais.

A Tabela 4.7 apresenta os principais temas identificados ao longo dos cinco modelos avaliados.

Tabela 4.7: Temas recorrentes identificados nos modelos LDA.

Tema Identificado	Descrição
Avaliação Geral	Opiniões positivas ou negativas sobre a experiência geral do jogo.
Qualidade do Jogo	Comentários sobre gráficos, jogabilidade, mecânicas e experiência de uso.
Problemas Técnicos	Bugs, travamentos, desempenho e incompatibilidades.
Problemas de Conexão	Falhas de servidor, desconexões e instabilidade de rede.
Monetização	Preços, compras internas, sistema <i>pay-to-win</i> e custos do jogo.
Progressão e Recompensas	Evolução do jogador, troféus, níveis, moedas e recompensas.
Conteúdo do Jogo	Personagens, itens, skins, mapas e elementos de customização.
Comunidade e Interação Social	Amigos, chat, comunicação e interação entre jogadores.
Conta e Suporte	Recuperação de contas, suporte técnico e problemas de acesso.
Toxicidade e Frustração	Reclamações, experiências negativas e sentimentos de injustiça.

A consolidação dos temas foi realizada considerando a frequência com que apareciam nos diferentes modelos, a clareza semântica dos tópicos e sua relevância para a análise de comentários de jogos digitais. Tópicos excessivamente específicos de um único jogo ou que não apresentavam um tema predominante foram descartados.

Como resultado desse processo, foram definidos os dez rótulos utilizados na etapa de classificação multirrótulo dos comentários:

1. Avaliação Geral;
2. Qualidade do Jogo;
3. Toxicidade e Frustração;
4. Monetização;
5. Progressão e Recompensas;

6. Conteúdo do Jogo;
7. Comunidade e Interação Social;
8. Problemas Técnicos;
9. Problemas de Conta e Suporte;
10. Problemas de Conexão.

Esses rótulos foram posteriormente utilizados como categorias-alvo no processo de rotulagem automática dos comentários por modelos de linguagem e no treinamento dos classificadores multirrótulo.

4.3 Rotulagem Automática Utilizando LLMs

Após a definição dos rótulos finais, foi necessário atribuí-los aos comentários do dataset. Como a rotulagem manual de milhares de comentários demandaria um esforço elevado, optou-se pela utilização de Grandes Modelos de Linguagem (LLMs) para realizar a classificação automática dos comentários.

Com o objetivo de reduzir possíveis erros individuais dos modelos, foi adotada uma estratégia baseada em consenso entre múltiplas LLMs. Cada comentário foi classificado independentemente por cinco modelos distintos utilizando o mesmo conjunto de rótulos e o mesmo prompt de instrução. Posteriormente, as respostas foram processadas e combinadas para gerar diferentes versões do dataset rotulado.

4.3.1 Seleção dos Modelos

A execução das classificações foi realizada localmente utilizando a plataforma *Ollama*, que permite a execução de modelos de linguagem de código aberto sem dependência de serviços externos.

A escolha dos modelos buscou contemplar arquiteturas distintas, tamanhos variados e compatibilidade com a infraestrutura computacional disponível. Foram selecionados os seguintes modelos:

- `phi3:3.8b`;
- `mistral:7b`;
- `llama3.1:8b`;
- `gemma2:9b`;
- `gemma3:12b`.

Todos os modelos foram utilizados na configuração padrão disponibilizada pelo Ollama. A utilização de modelos distintos teve como objetivo aumentar a diversidade das classificações produzidas e permitir posteriormente a construção de mecanismos de consenso entre respostas.

Inicialmente pretendia-se rotular pelo menos 10 mil comentários. Entretanto, devido ao tempo necessário para execução dos modelos localmente e às limitações de hardware disponíveis, foram classificados 2.400 comentários, correspondendo a 240 comentários de cada um dos dez jogos presentes no dataset.

4.3.2 Engenharia de Prompt

A classificação foi realizada utilizando a abordagem *zero-shot*, sem exemplos prévios de comentários rotulados.

Durante o desenvolvimento da solução foram realizados diversos ciclos de engenharia de prompt com o objetivo de reduzir alucinações, evitar respostas fora do formato esperado e impedir a criação de categorias inexistentes. Ao todo, foram testadas doze versões diferentes do prompt.

O prompt final instruía explicitamente o modelo a utilizar apenas os rótulos fornecidos, limitar a classificação a no máximo três tópicos e retornar exclusivamente uma lista Python válida contendo os rótulos identificados.

Como exemplo, o prompt possuía a seguinte estrutura:

Listing 4.1: Template de prompt para classificação multirrótulo

```
Classifique o comentário usando APENAS os tópicos abaixo.
```

```
REGRAS OBRIGATÓRIAS:
```

- Responda SOMENTE com uma lista Python válida
- NÃO escreva explicações
- NÃO escreva "Comentário", "Classificação" ou qualquer texto adicional
- NÃO use quebras de linha
- Máximo de 3 tópicos
- Use SOMENTE os tópicos da lista

```
FORMATO EXATO:
```

```
["topico1", "topico2"]
```

```
Tópicos:
```

```
{topicos}
```

```
Comentário:
```

```
{comentario}
```

Para cada comentário, o mesmo prompt foi enviado individualmente para os cinco modelos selecionados.

4.3.3 Processamento das Respostas

Após a execução dos modelos, cada resposta retornada foi submetida a uma etapa de validação sintática. Inicialmente, foi realizada a tentativa de conversão da resposta para uma lista Python contendo apenas cadeias de caracteres.

Respostas que não seguiam o formato especificado foram descartadas, por exem-

plo ["Avaliação geral (positivo/negativo genérico)": "Positivo",.. Também foram descartadas respostas que, após processamento, não continham nenhum rótulo válido.

Durante a análise das classificações observou-se que, embora os modelos geralmente utilizassem os rótulos corretos, frequentemente geravam pequenas variações textuais dos nomes esperados. Ao todo foram identificadas 361 variações distintas de rótulos.

Por exemplo, alguns modelos retornavam categorias como Toxicidade e Frustração (Bugs presentes)” ou Toxicidade e Frustração (Eventos roubados e hackers)”, embora o rótulo esperado fosse apenas “Toxicidade e Frustração”.

Para corrigir esse problema foi criado um arquivo de mapeamento contendo as variações encontradas e seus respectivos rótulos válidos, conforme 4.8

Tabela 4.8: Exemplo de mapeamento de rótulos encontrados para os rótulos corretos

Rótulo Encontrado	Rótulo Correto
Avaliação geral (-)	Avaliação geral
Conexão durante partidas	Problemas de conexão
Conteúdo do jogo (HUD, contra squad, ilha de treinamento)	Conteúdo do jogo
Sistema de recompensa e progressão	Progressão e Recompensas
Sensibilidade	Nenhum

Após a aplicação desse processo de normalização, cada resposta passou a conter exclusivamente rótulos pertencentes ao conjunto definido na etapa anterior.

As 10 categorias que não apresentavam correspondência clara com nenhum dos rótulos definidos permaneceram sem mapeamento e foram descartadas.

4.3.4 Construção dos Datasets por Consenso

Ao final do processamento, cada comentário possuía até cinco classificações independentes, uma proveniente de cada modelo utilizado.

Para combinar essas classificações foi adotado um mecanismo de votação por consenso. Para cada rótulo identificado, foi contabilizada a quantidade de modelos que atribuíram aquele rótulo ao comentário.

Formalmente, seja X a quantidade de modelos que atribuíram determinado rótulo. O rótulo é incluído no resultado final quando sua frequência de ocorrência é maior ou igual a um valor mínimo de consenso definido previamente.

Como X pode assumir valores entre 1 e 5, foram construídos cinco datasets distintos:

- Consenso 1: rótulo presente em pelo menos 1 modelo;
- Consenso 2: rótulo presente em pelo menos 2 modelos;
- Consenso 3: rótulo presente em pelo menos 3 modelos;
- Consenso 4: rótulo presente em pelo menos 4 modelos;
- Consenso 5: rótulo presente nos 5 modelos.

Essa abordagem permitiu gerar conjuntos de dados com diferentes níveis de rigor. Valores menores de consenso tendem a aumentar a cobertura de rótulos atribuídos, enquanto valores maiores tendem a aumentar a confiabilidade das classificações ao exigir concordância entre múltiplos modelos.

Posteriormente, cada um desses datasets foi utilizado nos experimentos de treinamento e avaliação dos classificadores multirrótulo, permitindo analisar o impacto do nível de consenso na qualidade dos modelos gerados.

4.4 Treinamento dos Modelos de Classificação

Foram avaliados três algoritmos clássicos de aprendizado supervisionado: *Support Vector Machine*, *Logistic Regression* e *Random Forest*, além de uma rede neural do tipo *Multilayer Perceptron* (MLP).

As principais bibliotecas utilizadas nesta etapa foram `spaCy`, `NLTK` e `Scikit-Learn`. A avaliação dos modelos foi realizada por meio das métricas *Hamming Loss*, F1-Score Micro e F1-Score Macro.

4.4.1 Representação Vetorial dos Comentários

Como os algoritmos de aprendizado de máquina operam sobre dados numéricos, os comentários textuais precisaram ser convertidos para uma representação vetorial.

Para isso foi utilizado o método TF-IDF (*Term Frequency-Inverse Document Frequency*), implementado pela classe `TfidfVectorizer` da biblioteca **Scikit-Learn**. Essa técnica atribui pesos aos termos de acordo com sua frequência em um comentário e sua raridade no conjunto de documentos, permitindo destacar palavras mais relevantes para a classificação.

Em cada experimento, os comentários foram submetidos ao método de pré-processamento correspondente e posteriormente transformados em vetores TF-IDF. Essa representação foi utilizada como entrada para todos os algoritmos avaliados.

Antes do treinamento, os conjuntos de rótulos foram convertidos para representação binária utilizando a classe `MultiLabelBinarizer`. Dessa forma, cada comentário passou a ser representado por um vetor binário indicando a presença ou ausência de cada um dos rótulos definidos na etapa anterior.

Tabela 4.9: Exemplo de binarização multirrótulo

Rótulos	AG	QJ	TF	MO	PR	CJ	CI	PT	PCS	PC
Avaliação Geral, Monetização, Problemas Técnicos, Conteúdo do Jogo	1	0	0	1	0	1	0	1	0	0
Avaliação Geral, Monetização, Qualidade do Jogo, Problemas Técnicos	1	1	0	1	0	0	0	1	0	0
Avaliação Geral, Conteúdo do Jogo, Problemas Técnicos, Qualidade do Jogo, Monetização, Progressão e Recompensas	1	1	0	1	1	1	0	1	0	0
Avaliação Geral, Problemas Técnicos, Comunidade e Interação Social, Toxicidade e Frustração, Conteúdo do Jogo, Qualidade do Jogo	1	1	1	0	0	1	1	1	0	0
Comunidade e Interação Social, Problemas Técnicos, Toxicidade e Frustração, Avaliação Geral, Qualidade do Jogo	1	1	1	0	0	0	1	1	0	0

AG: Avaliação Geral; QJ: Qualidade do Jogo; TF: Toxicidade e Frustração; MO: Monetização; PR: Progressão e Recompensas; CJ: Conteúdo do Jogo; CI: Comunidade e Interação Social; PT: Problemas Técnicos; PCS: Problemas de Conta e Suporte; PC: Problemas de Conexão.

4.4.2 Estratégias de Pré-Processamento Textual

Foram definidas e avaliadas múltiplas estratégias de pré-processamento textual com o objetivo de analisar seu impacto no desempenho dos modelos de classificação. Essas estratégias variam desde abordagens mais simples de normalização até técnicas mais avançadas baseadas em Processamento de Linguagem Natural. Foram avaliadas nove estratégias de pré-processamento: Foram avaliadas nove estratégias de pré-processamento textual:

- **Simples:** converte todo o texto para minúsculas, remove espaços excedentes e elimina caracteres de pontuação;
- **Remoção de Stopwords:** realiza tokenização do texto e remove palavras consideradas muito frequentes no idioma (stopwords), mantendo apenas os tokens restantes;
- **Stemming:** realiza tokenização e aplica redução das palavras ao seu radical por meio de algoritmos de stemming;
- **Lematização:** processa o texto utilizando o modelo spaCy, realizando tokenização e transformação das palavras para sua forma canônica (lema);
- **Filtro Substantivo:** utiliza o processamento via spaCy e mantém apenas tokens classificados como substantivos (NOUN);
- **Filtro Adjetivo:** utiliza o processamento via spaCy e mantém apenas tokens classificados como adjetivos (ADJ);
- **Filtro Verbo:** utiliza o processamento via spaCy e mantém apenas tokens classificados como verbos (VERB);
- **Substantivo, Verbo e Adjetivo:** utiliza spaCy e mantém apenas tokens classificados como substantivos, verbos e adjetivos;
- **Pipeline Melhorado:** aplica lematização via spaCy, remove stopwords do modelo e de domínio, filtra apenas tokens alfabéticos e mantém exclusivamente substantivos e verbos, descartando termos irrelevantes previamente definidos.

4.4.3 Estratégia de Classificação Multirrótulo

Uma análise exploratória dos datasets revelou um desbalanceamento na distribuição dos rótulos, conforme ilustrado na Figura 4.4. Observa-se que alguns tópicos aparecem com frequência significativamente maior que outros.

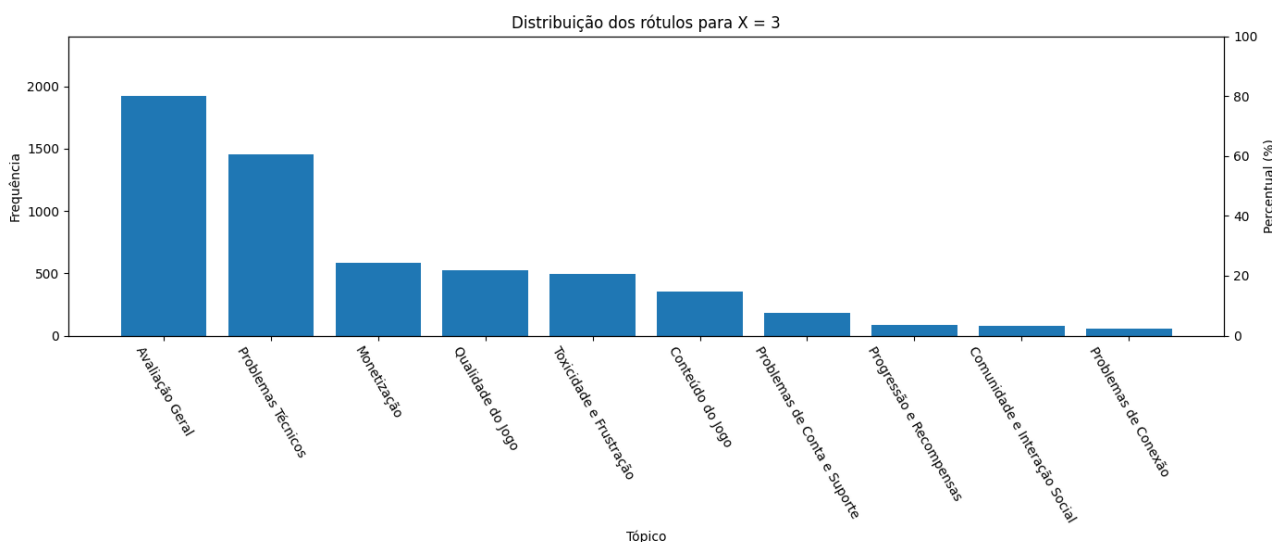


Figura 4.4: Distribuição dos rótulos no 3º dataset.

Além disso, como um mesmo comentário pode abordar múltiplos aspectos de um jogo simultaneamente, o problema foi tratado como uma tarefa de classificação multirrótulo.

Para possibilitar a utilização dos algoritmos tradicionais de classificação, foi adotada a estratégia *One-vs-Rest* (OvR), também conhecida como *One-vs-All*. Nessa abordagem, um classificador independente é treinado para cada rótulo do problema, aprendendo a distinguir exemplos positivos e negativos daquele tópico específico.

Durante a fase de predição, todos os classificadores são executados e os rótulos identificados são combinados para formar o resultado final do comentário.

4.4.4 Algoritmos Avaliados

Foram selecionados três algoritmos clássicos amplamente utilizados em tarefas de classificação textual:

- **Support Vector Machine (SVM)** utilizando a implementação `LinearSVC`;

- **Regressão Logística (LR)** utilizando a implementação `LogisticRegression`;
- **Random Forest (RF)** utilizando a implementação `RandomForestClassifier`.

Os modelos clássicos foram avaliados utilizando os hiperparâmetros padrão disponibilizados pela biblioteca `Scikit-Learn`. O objetivo dessa decisão foi estabelecer uma linha de base consistente para comparação entre algoritmos, reduzindo a influência de ajustes específicos de hiperparâmetros.

Para cada combinação de algoritmo e método de pré-processamento foi executado um processo completo de treinamento e validação, permitindo comparar o desempenho dos modelos sob diferentes representações textuais.

4.4.5 Validação Cruzada

A avaliação dos modelos foi realizada utilizando validação cruzada estratificada para problemas multirrótulo por meio da classe `MultilabelStratifiedKFold`.

Foi adotado um esquema de validação cruzada com $k = 5$ partições. Em cada iteração, quatro partições eram utilizadas para treinamento e uma para teste, garantindo que todas as amostras participassem do processo de avaliação.

Para cada divisão da validação cruzada foi executada a seguinte sequência:

1. Separação dos conjuntos de treinamento e teste;
2. Aplicação do pré-processamento textual;
3. Vetorização TF-IDF;
4. Treinamento do classificador utilizando a estratégia One-vs-Rest;
5. Geração das previsões;
6. Cálculo das métricas de avaliação.

Ao final das cinco execuções, os resultados foram agregados por meio da média aritmética das métricas obtidas em cada *fold*.

As métricas utilizadas para comparação dos modelos foram:

- **Hamming Loss;**
- **F1-Score Micro;**
- **F1-Score Macro.**

Dentre elas, o F1-Score Macro foi utilizado como principal critério para seleção dos melhores modelos, por considerar igualmente o desempenho em todos os rótulos, independentemente de sua frequência no dataset.

4.4.6 Treinamento da Rede Neural MLP

Além dos algoritmos clássicos, também foi avaliada uma rede neural do tipo *Multilayer Perceptron* (MLP), implementada por meio da classe `MLPClassifier` da biblioteca `Scikit-Learn`.

Diferentemente dos modelos clássicos, a MLP foi submetida a um processo de ajuste de hiperparâmetros devido à sua maior sensibilidade à configuração arquitetural e aos parâmetros de otimização. Para isso, foi definida uma grade de busca (*grid search*) abrangendo parâmetros relacionados à redução de dimensionalidade, taxa de aprendizado, regularização e arquitetura das camadas ocultas.

Antes do treinamento, os vetores TF-IDF foram projetados em um espaço de menor dimensionalidade por meio do `TruncatedSVD`, com o número de componentes variando entre 256 e 512.

A taxa de aprendizado inicial (`learning_rate_init`) foi testada com valores de 10^{-3} e 5×10^{-4} , enquanto o parâmetro de regularização L2 (`alpha`) foi avaliado com valores de 10^{-4} e 10^{-3} .

A arquitetura da rede neural foi variada por meio do parâmetro `hidden_layer_sizes`, contemplando desde arquiteturas simples com uma única camada contendo entre 1 e 256 neurônios, até arquiteturas mais profundas com duas camadas ocultas, incluindo combinações como (1,2), (10,5), (100,128) e (256,128).

Para cada combinação de hiperparâmetros, foi executado um processo de validação cruzada estratificada com cinco partições, utilizando a mesma estratégia aplicada aos modelos clássicos.

Em cada *fold*, foi construída uma *pipeline* composta pelas seguintes etapas:

1. Vetorização TF-IDF dos comentários;
2. Redução de dimensionalidade com `TruncatedSVD`;
3. Classificação utilizando `MLPClassifier`.

Ao final do treinamento, foram calculadas as métricas Hamming Loss, F1-Score Micro e F1-Score Macro para cada configuração de hiperparâmetros. Os resultados foram agregados por média entre os *folds*, permitindo a comparação consistente entre as diferentes arquiteturas testadas.

O F1-Score Macro foi utilizado como principal critério de seleção dos modelos, por considerar igualmente o desempenho em todas as classes, independentemente de sua frequência no conjunto de dados.

4.5 Análise dos Resultados

Esta seção apresenta a análise comparativa dos resultados obtidos nos cinco conjuntos de dados construídos a partir de diferentes níveis de consenso na rotulagem automática dos comentários (Dataset 1 a Dataset 5). A análise considera as métricas Micro F1, Macro F1 e Hamming Loss, bem como a influência dos modelos e estratégias de pré-processamento.

De forma geral, observa-se que os modelos baseados em abordagens lineares, especialmente a Regressão Logística (LR), apresentam desempenho mais consistente ao longo de todos os cenários avaliados. Em contraste, o Random Forest (RF) demonstra maior sensibilidade à variação da qualidade dos rótulos, com degradação mais acentuada principalmente em termos de Macro F1.

Tabela 4.10: Dataset 1 - Top 3 por modelo

Modelo	Pré-processamento	Micro F1	Macro F1	Hamming Loss
SVM				
SVM	Sem Stopwords	0.791160	0.697178	0.193807
SVM	Lemmatization	0.790177	0.703109	0.195914
SVM	Sem Espaços	0.790007	0.698531	0.195278
Logistic Regression				
LR	Stemming + Stopwords + Sem Espaços	0.797275	0.724740	0.191345
LR	Sem Stopwords	0.794879	0.715631	0.191118
LR	Stemming	0.793549	0.722293	0.195156
Random Forest				
RF	Stemming + Stopwords + Sem Espaços	0.788753	0.563699	0.189763
RF	Filtro Subst/Adj/Verbo	0.785517	0.574920	0.192806
RF	Junta tudo	0.784473	0.574077	0.194303
MLP				
MLP	Melhor configuração	0.796325	0.646577	0.184884
MLP	Segunda melhor	0.795054	0.643749	0.186574
MLP	Terceira melhor	0.794307	0.642495	0.187327

O Dataset 1 representa o cenário com menor rigor de consenso, no qual há maior cobertura de rótulos. Nesse contexto, os modelos atingem seus melhores desempenhos absolutos, com destaque para a Regressão Logística, que alcança os maiores valores de Micro F1 e Macro F1. O SVM apresenta desempenho próximo, indicando que, em cenários com menor ruído de rotulagem, modelos de margem também são competitivos.

Tabela 4.11: Dataset 2 - Top 3 por modelo

Modelo	Pré-processamento	Micro F1	Macro F1	Hamming Loss
SVM				
SVM	Lemmatization	0.787763	0.648215	0.136425
SVM	Stemming	0.785061	0.657828	0.139408
SVM	Stemming + Stopwords	0.784983	0.649542	0.139362
Logistic Regression				
LR	Sem Stopwords	0.784615	0.659620	0.140512
LR	Stemming + Stopwords + Sem Espaços	0.784177	0.676493	0.143073
LR	Stemming	0.781415	0.671680	0.145215
Random Forest				
RF	Junta tudo	0.759152	0.443370	0.140610
RF	Filtro Subst/Adj/Verbo	0.757040	0.437794	0.141574
RF	Stemming + Stopwords + Sem Espaços	0.751920	0.417202	0.143193
MLP				
MLP	Melhor configuração	0.779147	0.520010	0.135339
MLP	Segunda melhor	0.779033	0.519783	0.135380
MLP	Terceira melhor	0.777691	0.516323	0.135918

Tabela 4.12: Dataset 3 - Top 3 por modelo

Modelo	Pré-processamento	Micro F1	Macro F1	Hamming Loss
SVM				
SVM	Sem Espaços	0.751919	0.543760	0.114231
SVM	Lemmatization	0.751715	0.562284	0.115867
SVM	Sem Pre-processamento	0.750304	0.554844	0.115516
Logistic Regression				
LR	Stemming	0.745205	0.583373	0.121302
LR	Stemming + Stopwords + Sem Espaços	0.743728	0.601706	0.124881
LR	Sem Stopwords	0.742701	0.610175	0.125602
Random Forest				
RF	Junta tudo	0.718140	0.308594	0.120427
RF	Sem Stopwords	0.714384	0.287070	0.121481
RF	Filtro Subst/Adj/Verbo	0.714356	0.300899	0.121629
MLP				
MLP	Melhor configuração	0.750088	0.434539	0.111369
MLP	Segunda melhor	0.748968	0.436039	0.111778
MLP	Terceira melhor	0.747760	0.421843	0.112571

À medida que o nível de consenso aumenta (Datasets 2 e 3), observa-se uma leve redução no Micro F1, acompanhada de maior estabilidade relativa entre SVM e LR. Entretanto, o Macro F1 apresenta maior variação, sugerindo que o aumento do consenso afeta principalmente a capacidade dos modelos em capturar rótulos menos frequentes, intensificando o impacto do

desbalanceamento multirrótulo.

Tabela 4.13: Dataset 4 - Top 3 por modelo

Modelo	Pré-processamento	Micro F1	Macro F1	Hamming Loss
SVM				
SVM	Stemming + Stopwords + Sem Espaços	0.693284	0.447444	0.099792
SVM	Stemming	0.692543	0.454354	0.100250
SVM	Sem Pre-processamento	0.690782	0.415462	0.097917
Logistic Regression				
LR	Stemming + Stopwords + Sem Espaços	0.702546	0.509992	0.102208
LR	Stemming	0.694683	0.507915	0.105000
LR	Lemmatization	0.691812	0.481308	0.104542
Random Forest				
RF	Stemming + Stopwords + Sem Espaços	0.663775	0.197985	0.100042
RF	Junta tudo	0.659620	0.210284	0.101292
RF	Filtro Subst/Adj/Verbo	0.659566	0.201986	0.101167
MLP				
MLP	Melhor configuração	0.688648	0.313542	0.093664
MLP	Segunda melhor	0.687555	0.312877	0.093914
MLP	Terceira melhor	0.684278	0.301378	0.094667

Tabela 4.14: Dataset 5 - Top 3 por modelo

Modelo	Pré-processamento	Micro F1	Macro F1	Hamming Loss
SVM				
SVM	Stemming + Stopwords + Sem Espaços	0.623109	0.256707	0.067875
SVM	Lemmatization	0.623005	0.255349	0.066667
SVM	Stemming	0.622955	0.261561	0.067542
Logistic Regression				
LR	Stemming	0.642074	0.312407	0.070125
LR	Stemming + Stopwords + Sem Espaços	0.642058	0.310073	0.069917
LR	Lemmatization	0.640862	0.303033	0.069458
Random Forest				
RF	Filtro Subst/Adj/Verbo	0.509251	0.122932	0.068500
RF	Sem Stopwords	0.493848	0.117437	0.069125
RF	Junta tudo	0.484116	0.120489	0.070417
MLP				
MLP	Melhor configuração	0.601347	0.178240	0.062583
MLP	Segunda melhor	0.595528	0.176065	0.062833
MLP	Terceira melhor	0.595157	0.183865	0.063458

Nos datasets mais restritivos (Datasets 4 e 5), há uma queda mais acentuada no desempenho geral. O Random Forest apresenta as maiores perdas relativas, especialmente em Macro F1,

indicando baixa robustez em cenários com menor densidade de rótulos positivos. A Regressão Logística mantém maior estabilidade frente à redução da diversidade de rótulos.

Em relação ao pré-processamento textual, não se observa uma configuração universalmente superior. Estratégias leves, como remoção de stopwords e stemming, tendem a apresentar desempenho ligeiramente superior ou equivalente na maioria dos cenários. Por outro lado, abordagens mais restritivas baseadas em filtragem por classes gramaticais (exclusão de substantivos, verbos ou adjetivos isoladamente) resultam em degradação consistente de desempenho, indicando perda de informação semântica relevante para tarefas multirrótulo.

A análise do Hamming Loss reforça as observações anteriores, indicando aumento do erro por rótulo nos cenários de maior restrição de consenso. Esse comportamento sugere que a redução na densidade de rótulos positivos afeta diretamente a capacidade de generalização dos modelos, especialmente os baseados em árvores de decisão.

Por fim, os resultados indicam que o fator mais determinante para o desempenho dos modelos é o nível de qualidade dos rótulos (definido pelo consenso entre modelos), seguido pela escolha do algoritmo de aprendizado. As variações de pré-processamento exercem impacto secundário, sendo insuficientes para alterar significativamente o comportamento geral dos modelos. Modelos lineares, em especial a Regressão Logística, demonstram maior robustez ao longo de todos os cenários avaliados.

4.6 Conclusão

Este trabalho investigou o desempenho de diferentes modelos de aprendizado supervisionado na tarefa de classificação multirrótulo de comentários de jogos em língua portuguesa, considerando distintos níveis de qualidade de rotulagem automática obtidos por meio de consenso entre modelos baseados em LLMs. Além disso, avaliou-se o impacto de diferentes estratégias de pré-processamento textual sobre o desempenho dos classificadores.

Os resultados demonstram que o nível de consenso utilizado na construção dos datasets exerce influência direta no comportamento dos modelos. Em cenários com menor rigor de consenso, observa-se maior cobertura de rótulos e melhores valores absolutos de Micro F1 e Macro

F1. À medida que o consenso se torna mais restritivo, há uma redução progressiva na densidade de rótulos, o que impacta negativamente o desempenho dos modelos, especialmente em termos de Macro F1, evidenciando maior dificuldade na modelagem de classes menos frequentes.

Entre os algoritmos avaliados, a Regressão Logística apresentou o desempenho mais consistente ao longo de todos os cenários, mantendo bom equilíbrio entre Micro F1, Macro F1 e Hamming Loss. O SVM também apresentou desempenho competitivo, especialmente nos datasets com maior densidade de rótulos. Em contrapartida, o Random Forest demonstrou maior sensibilidade à variação na qualidade dos dados, com degradação mais acentuada em Macro F1, indicando menor adequação para cenários multirrótulo com distribuição esparsa de rótulos.

O modelo MLP apresentou desempenho intermediário, com resultados competitivos em alguns cenários, porém sem superioridade consistente em relação aos modelos lineares. Esse comportamento sugere que, para o conjunto de dados e representações utilizadas, o aumento da complexidade do modelo não se traduz necessariamente em ganhos de desempenho, especialmente quando há limitações na qualidade e cobertura dos rótulos.

Em relação ao pré-processamento textual, observou-se que abordagens simples, como remoção de stopwords e stemming, tendem a apresentar desempenho ligeiramente superior ou equivalente às demais configurações. Por outro lado, estratégias mais agressivas baseadas em filtragem por classes gramaticais resultaram em degradação consistente do desempenho, indicando perda de informação semântica relevante para tarefas multirrótulo.

De forma geral, conclui-se que o fator mais determinante para o desempenho dos modelos foi a qualidade dos rótulos, definida pelo nível de consenso na rotulagem automática, seguido pela escolha do algoritmo de aprendizado. As variações de pré-processamento exerceram impacto secundário, não sendo suficientes para alterar as tendências gerais observadas.

Os resultados também indicam a existência de um trade-off relevante entre qualidade de rotulagem e cobertura de dados. Em linha com o estudo de *Majority Rules: LLM Ensemble is a Winning Approach for Content Categorization*, o aumento do número de LLMs no processo de consenso tende a produzir rótulos mais consistentes, reduzindo inconsistências de anotação e aumentando a confiabilidade dos rótulos.

Entretanto, neste trabalho observou-se que o aumento do rigor do consenso reduz significativamente a quantidade de rótulos atribuídos. Essa redução impacta a densidade informacional dos datasets, afetando negativamente o desempenho dos modelos supervisionados, especialmente em cenários mais restritivos.

Dentre os cenários avaliados, o Dataset 3 apresentou o melhor equilíbrio entre qualidade e cobertura de rótulos, resultando no melhor compromisso entre desempenho e robustez dos modelos. Assim, este conjunto de dados mostrou-se o mais adequado para a tarefa proposta, com destaque para a Regressão Logística, que apresentou o comportamento mais estável entre os algoritmos avaliados.

Por fim, considerando o tamanho relativamente limitado do conjunto de dados rotulados (aproximadamente 2400 comentários), é plausível que a ampliação do volume de dados, aliada ao uso de um maior número de LLMs no processo de consenso, possa melhorar simultaneamente a qualidade da rotulagem e o desempenho dos modelos supervisionados.

Referências Bibliográficas

ABRAGAMES. *Relatório da Indústria de Jogos Digitais no Brasil*. [S.l.], 2023. Acesso em: 10 jun. 2025. Disponível em: <<https://www.abragames.org/uploads/5/6/8/0/56805537/abragames-pt.pdf>>.

AppBrain. *AppBrain Android Statistics*. 2026. <<https://www.appbrain.com>>. Acesso em: 19 mar. 2026.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. In: *Journal of Machine Learning Research*. [S.l.: s.n.], 2012. v. 13, p. 281–305.

BIAU, G.; SCORNET, E. Theory of random forests. *Annual Review of Statistics and Its Application*, v. 11, p. 141–163, 2024.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research*, v. 3, p. 993–1022, 2003.

BROWN, T. B. et al. Language models are few-shot learners. *NeurIPS*, 2020.

BROWNE, C. et al. Cross-validation methods in machine learning: a survey. *arXiv preprint*, 2020.

CHICHE, A.; YITAGESU, B. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, Springer, v. 9, n. 1, 2022.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995.

DEVELOPERS, S. learn. *Cross-validation: evaluating estimator performance*. 2024. <https://scikit-learn.org>.

GUIDO, R. et al. An overview on the advancements of support vector machine models in healthcare applications: A review. *Information*, MDPI, v. 15, n. 4, p. 235, 2024.

HAN, M. et al. A survey of multi-label classification based on supervised and semi-supervised learning. *International Journal of Machine Learning and Cybernetics*, v. 14, p. 697–724, 2023.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3. ed. Prentice Hall, 2025. Disponível em: <<https://web.stanford.edu/~jurafsky/slp3/>>.

- KAMEN, A.; KAMEN, Y. *Majority Rules: LLM Ensemble is a Winning Approach for Content Categorization*. 2025. Preprint / Technical Report. Work on LLM ensemble voting for content classification.
- KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI*, 1995.
- KREBS, F. et al. Applying multi-class support vector machines: One-vs-one vs. one-vs-all on the uwf-zeekdatafall22 dataset. *Electronics*, v. 13, n. 19, p. 3916, 2024. Disponível em: <<https://www.mdpi.com/2079-9292/13/19/3916>>.
- LI, Q. et al. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, v. 13, n. 2, p. 1–41, 2022.
- LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in nlp. *ACM Computing Surveys*, 2023.
- LIU, W. et al. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 44, n. 11, p. 7955–7974, 2022.
- LUGER, G. F. *Artificial Intelligence: Principles and Practice*. Cham: Springer Nature, 2024. ISBN 9783031574368.
- MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2022.
- MELO, T. de. Análise de comentários das plataformas online de restaurantes michelin no brasil. In: *A Produção do Conhecimento nas Ciências da Comunicação*, p. 226–238, 2021.
- MINAEE, S. et al. Deep learning based text classification: A comprehensive review. *ACM Computing Surveys*, v. 54, n. 3, p. 1–40, 2021.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.
- MURPHY, K. P. *Probabilistic Machine Learning: An Introduction*. Cambridge, MA: The MIT Press, 2022. ISBN 9780262046824. Disponível em: <<https://probml.github.io/pml-book/book1.html>>.
- NETO, J. A. A.; MELO, T. de. Exploring supervised learning models for multi-label text classification in brazilian restaurant reviews. *arXiv preprint*, 2023. Disponível em: <<https://arxiv.org/abs/2023.xxxxxx>>.
- PALANIVINAYAGAM, A.; EL-BAYEH, C. Z.; DAMASEVČIUS, R. Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*, MDPI, v. 16, n. 5, p. 236, 2023.
- PAWARA, P. et al. One-vs-one classification for deep neural networks. *Pattern Recognition*, v. 108, p. 107528, 2020. ISSN 0031-3203. Disponível em: <<https://doi.org/10.1016/j.patcog.2020.107528>>.
- PETERS, M. E. et al. Modern natural language processing: A survey. *arXiv preprint arXiv:2301.12345*, 2023. Disponível em: <<https://arxiv.org/abs/2301.12345>>.

- PGB. *Pesquisa Game Brasil 2026*. 2026. Disponível em: <<https://www.pesquisagamebrasil.com.br>>.
- PROKHORENKOVA, L. et al. Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, v. 31, 2021.
- RIFKIN, R.; KLAUTAU, A. In defense of one-vs-all classification. *Journal of Machine Learning Research*, v. 5, p. 101–141, 2004.
- RÖDER, M.; BOTH, A.; HINNEBURG, A. Exploring the space of topic coherence measures. *Proceedings of WSDM*, 2015.
- RODRIGUEZ, J. D.; PEREZ, A.; LOZANO, J. A. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4. ed. Hoboken, NJ: Pearson, 2021. ISBN 9780134610993.
- SCHÖLKOPF, B.; SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
- SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys*, 2022.
- SECHIDIS, K.; TSOUMAKAS, G.; VLAHAVAS, I. On the stratification of multi-label data. *ECML PKDD*, 2011.
- SIINO, M.; TINNIRELLO, I.; La Cascia, M. Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. *Information Systems*, Elsevier, v. 121, p. 102342, 2024.
- SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. In: *Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2012.
- spaCy. *Industrial-strength Natural Language Processing in Python*. 2025. <<https://spacy.io/>>. Accessed: 2025-06-07.
- STEVENS, K. Modern perspectives on cross-validation in machine learning. *Machine Learning Review*, 2023.
- STEVENS, K. et al. Exploring topic coherence measures. In: *Proceedings of NAACL*. [S.l.: s.n.], 2012.
- SYED, M.; SPRUIT, M. A survey of topic modeling and its applications. *IEEE Access*, 2020.
- SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2. ed. Cham: Springer, 2022. ISBN 9783030343712.
- TAREKEGN, A. N.; GIACOBINI, M.; MICHALAK, K. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, Elsevier, v. 118, p. 107965, 2021.

- TIITTANEN, H. et al. Novel split quality measures for multilabel stratified cross validation. *Applied Computing and Intelligence*, 2022.
- TREVISIO, M. et al. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, MIT Press, v. 11, p. 826–860, 2023.
Disponível em: <<https://aclanthology.org/2023.tacl-1.48/>>.
- TURING, A. M. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950.
- VAPNIK, V. N. *Statistical Learning Theory*. New York: Wiley, 1998.
- VOGEL, L. *Natural Language Processing with Python and Transformers*. [S.l.]: O'Reilly Media, 2021.
- WANG, X. et al. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023.
- WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.
- WEI, J. et al. Finetuned language models are zero-shot learners. *ICLR*, 2022.
- ZHANG, H. et al. Large language model ensembles for robust text classification. *arXiv preprint*, 2023.
- ZHANG, Y. et al. A survey on prompt engineering. *arXiv preprint*, 2023.