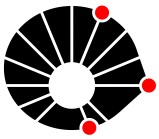

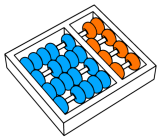

Um Estudo Comparativo entre Aprendizagem Supervisionada e Contextualizada na Classificação de Teses e Dissertações

Gabriel Pavani Giro *Renilson Conceição de Luna Junior*
Paulo César Polastri *Flávia Linhalis* *Julio Cesar dos Reis*

Technical Report - IC-26-1 - Relatório Técnico
July - 2026 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Um Estudo Comparativo entre Aprendizagem Supervisionada e Contextualizada na Classificação de Teses e Dissertações

Gabriel Pavani Giro Renilson Conceição de Luna Junior Paulo César Polastri
Flávia Linhalis Julio Cesar dos Reis

Junho de 2026

Resumo

Este trabalho apresenta um estudo comparativo entre modelos clássicos de aprendizado de máquina (Árvore de Decisão, *Support Vector Machine* (SVM) e *Multilayer Perceptron* (MLP)) e o Modelo de Linguagem de Grande Porte LLaMA 3, aplicados à classificação automática de teses e dissertações acadêmicas por área do ensino de ciências da natureza (Física, Química, Biologia e Ciências Gerais). Os modelos clássicos foram treinados sobre representações TF-IDF, enquanto o LLaMA 3 foi avaliado nos regimes *zero-shot* e *few-shot*. Os modelos clássicos alcançaram precisão e F1-Score em torno de 86–87%, com dificuldade sistemática na distinção entre Biologia e Ciências Gerais. No regime *zero-shot*, a LLM obteve a acurácia máxima de 78,8%, inferior à dos modelos supervisionados. No regime *few-shot*, com apenas um exemplo por classe no prompt, a LLM atingiu acurácia de 94%, superando todos os modelos clássicos. Os resultados indicam que LLM em regime *few-shot* constituem uma alternativa viável e de alto desempenho para a classificação textual em português, mesmo em ambientes com recursos computacionais limitados.

1 Introdução

A classificação de um grande volume de teses e dissertações científicas é uma tarefa fundamental para a organização e análise do conhecimento produzido em diversas áreas, como no ensino de ciências. Tradicionalmente, esse processo é realizado manualmente por especialistas que analisam campos como título, resumo e palavras-chave para atribuir um rótulo a cada trabalho. Contudo, quando o número de documentos ultrapassa a casa das centenas ou milhares, a classificação manual torna-se um processo consideravelmente demorado e de custo operacional elevado. A automatização do processo de classificação por meio de modelos de aprendizado de máquina (ML, do inglês *Machine Learning*) e Modelos de Linguagem de Grande Porte (LLM, do inglês *Large language Model*) surge como uma alternativa vantajosa para esse problema. A automação oferece escalabilidade, atualização em tempo real e redução de custos, liberando os pesquisadores para se dedicarem a outras etapas da pesquisa.

Teses e dissertações apresentam desafios particulares para a tarefa de classificação automática. Diferentemente de textos tradicionais, esse tipo de documento emprega linguagem altamente especializada, cujos jargões podem assumir significados distintos conforme a área do conhecimento. Soma-se a isso o fato de que a fronteira entre áreas é frequentemente difusa, o que torna a categorização intrinsecamente ambígua. Do ponto de vista dos dados, há escassez de conjuntos rotulados disponíveis publicamente, além do desbalanceamento entre áreas; algumas disciplinas concentram volumes muito maiores de produção acadêmica do que outras.

Este estudo realiza uma análise comparativa para avaliar a viabilidade e a efetividade de diferentes abordagens de classificação automática de texto. Objetivamos comparar o desempenho de modelos clássicos de aprendizado de máquina, como Árvore de Decisão, *Support Vector Machine* (SVM) e *Multilayer Perceptron* (MLP), com o de uma LLM (LLaMA 3) em diferentes regimes de aprendizado. Especificamente, com relação ao LLM, este trabalho avalia a efetividade de técnicas de engenharia de *prompt*, como o *zero-shot prompting* — no qual o modelo realiza a tarefa sem exemplos prévios — e o *few-shot learning*, no qual o modelo generaliza a partir de um pequeno número de exemplos. O presente estudo foi desenvolvido considerando as limitações de desbalanceamento de dados identificadas na literatura, adotando medidas de balanceamento entre as teses disponíveis. Buscamos não apenas solucionar um problema prático de categorização de grandes volumes de documentos, mas também aprofundar o conhecimento sobre as vantagens e limitações de abordagens clássicas quando comparadas a abordagens mais recentes, fornecendo uma base para futuras pesquisas na área.

A seguinte pergunta norteia esta investigação: **Classificadores tradicionais são mais efetivos do que aprendizagem contextualizada (*prompts* LLM) para classificar teses e dissertações na área de ensino das ciências da natureza?**

Do ponto de vista metodológico, este estudo concentra-se na classificação automática de teses e dissertações na área de Ensino de Ciências da Natureza. Utilizamos um conjunto de dados rotulado com trabalhos acadêmicos classificados em diferentes áreas do conhecimento. Os modelos supervisionados foram treinados a partir de representações textuais. Em paralelo, o modelo LLaMA 3 foi avaliado em regimes de aprendizagem contextualizada (*zero-shot* e *few-shot*), dispensando treinamento supervisionado específico para a tarefa. O desempenho das diferentes abordagens foi analisado por meio de métricas consolidadas e de classificação multiclasse, o que permite uma comparação sistemática entre métodos baseados em aprendizado supervisionado e em inferência contextual. As métricas de avaliação utilizadas para a comparação foram principalmente a precisão, que mede a proporção de acertos positivos, e o *recall*, que avalia a capacidade do modelo de identificar todas as instâncias relevantes.

Os resultados obtidos evidenciam diferenças relevantes entre as abordagens avaliadas. Os modelos supervisionados apresentaram desempenho consistente, alcançando valores de precisão e F1-Score próximos de 86–87%, embora tenham demonstrado dificuldades recorrentes na distinção entre classes semanticamente próximas, especialmente Biologia e Ciências Gerais[1]. Em contraste, o modelo LLaMA 3 apresentou desempenho inferior no regime *zero-shot*, mas alcançou acurácia de 94% quando utilizado em regime *few-shot* com apenas um exemplo por classe, superando todos os classificadores tradicionais avaliados. Esses achados sugerem que a aprendizagem contextualizada baseada em LLM pode capturar relações semânticas complexas que não são adequadamente representadas por abordagens tradicionais baseadas na frequência de termos. Os resultados demonstram a viabilidade prática do uso de LLM executadas localmente para tarefas de classificação científica em português, mesmo em ambientes com recursos computacionais limitados.

O restante deste relatório está organizado da seguinte maneira. A Seção 2 apresenta o referencial teórico que fundamenta a pesquisa, discutindo conceitos relacionados à classificação automática de textos, técnicas tradicionais de aprendizado de máquina e o uso de Large Language Models em tarefas de Processamento de Linguagem Natural. Em seguida, a Seção 3 descreve os procedimentos metodológicos adotados no estudo, incluindo a caracterização do conjunto de dados utilizado na pesquisa, as etapas de pré-processamento aplicadas aos dados textuais, os modelos de classificação

[1]O termo Ciências Gerais é usado para classificar trabalhos em que não há abordagem de temas e conteúdos de ensino específicos de uma determinada área, quando essa abordagem é genérica, generalizada ou indeterminada [EAEC 2025] Isso acontece principalmente em trabalhos aplicados no Ensino Fundamental, onde as disciplinas de Física, Química e Biologia são trabalhadas na disciplina de Ciências.

avaliados (Árvore de Decisão, *Support Vector Machine*, *Multilayer Perceptron* e LLaMA 3), bem como seus respectivos parâmetros e estratégias de treinamento. A Seção 4 apresenta os resultados experimentais obtidos para os modelos clássicos e os resultados obtidos com a LLM, enquanto a Seção 5 promove uma análise crítica dos resultados obtidos, com destaque para o desempenho dos modelos clássicos diante da sobreposição semântica entre as classes, o impacto da engenharia de prompt no desempenho das LLM, a viabilidade do uso de LLM quantizadas em contextos com recursos computacionais limitados e as limitações do estudo. Por fim, a Seção 6 apresenta as conclusões da pesquisa, sintetizando os principais achados do trabalho e indicando perspectivas para pesquisas futuras.

2 Fundamentação Teórica

A classificação automática de textos constitui uma das tarefas fundamentais no campo do Processamento de Linguagem Natural (PLN), com aplicações que abrangem desde a organização de documentos até a análise de sentimentos e categorização de conteúdo científico [Aggarwal e Zhai 2012]. Esta seção apresenta os fundamentos teóricos que embasam as abordagens exploradas neste estudo, abordando tanto os métodos clássicos de aprendizado de máquina quanto as técnicas contemporâneas baseadas em Modelos de Linguagem de Grande Porte (LLM).

2.1 Aprendizado de Máquina para Classificação de Textos

O aprendizado de máquina (*Machine Learning* ou ML) refere-se ao conjunto de técnicas computacionais que permitem que sistemas aprendam padrões a partir de dados sem serem explicitamente programados para cada tarefa específica [Mitchell 1997]. No contexto da classificação textual, os algoritmos de aprendizado supervisionado são treinados com exemplos rotulados, nos quais cada documento está associado a uma categoria predefinida, permitindo que o modelo aprenda a mapear características textuais para rótulos de classe [Sebastiani 2002].

2.1.1 Representação Vetorial de Textos

Para que algoritmos de aprendizado de máquina possam processar textos, é necessário converter documentos em representações numéricas. Uma das técnicas mais consolidadas é o *Term Frequency-Inverse Document Frequency* (TF-IDF), que atribui pesos às palavras com base em sua frequência no documento e sua raridade no corpus completo. Formalmente, o peso $TF - IDF$ de um termo t em um documento d é calculado por:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \tag{1}$$

onde $TF(t, d)$ representa a frequência do termo no documento e $IDF(t) = \log \frac{N}{df(t)}$, sendo N o número total de documentos e $df(t)$ o número de documentos que contêm o termo t. Essa abordagem privilegia termos que são informativos para documentos específicos, reduzindo o peso de palavras muito comuns [Ramos 2003].

2.1.2 Pré-processamento Textual

O pré-processamento de textos é uma etapa crítica que impacta diretamente o desempenho dos modelos de classificação [Vijayarani, Ilamathi e Nithya 2015]. As principais técnicas incluem a tokenização, que segmenta o texto em unidades lexicais (*tokens*) que podem ser processadas individualmente; a remoção de *stopwords*, que elimina palavras funcionais de alta frequência e baixo

valor discriminativo, como artigos, preposições e conjunções [Manning, Raghavan e Schütze 2008]; o *stemming*, que reduz palavras às suas raízes morfológicas através de heurísticas de truncamento, permitindo que variações flexionais sejam tratadas como uma única entidade [Porter 1980]; e a *lemmatization*, processo mais sofisticado que reduz palavras à sua forma canônica (lema) considerando o contexto morfossintático, embora seja computacionalmente mais custoso [Plisson, Lavrac e Mladenic 2004].

2.2 Modelos Clássicos de Aprendizado Supervisionado

Os modelos clássicos de aprendizado de máquina são algoritmos estabelecidos que funcionam bem em tarefas de classificação, especialmente quando há dados rotulados disponíveis e uma separação clara entre as classes. Nos estudos apresentados, foram utilizados e comparados três modelos clássicos: a **Árvore de Decisão (*Decision Tree*)**, o ***Support Vector Machine* (SVM)** e o ***Multilayer Perceptron* (MLP)**.

2.2.1 Árvore de Decisão

Um classificador por árvore de decisão é um algoritmo de aprendizado supervisionado que emprega uma estrutura hierárquica, semelhante a uma árvore, para tomar decisões e classificar dados [Ho 2024]. A árvore é construída dividindo recursivamente o conjunto de dados em subgrupos menores com base nos atributos mais informativos. O processo de divisão continua até que um critério de parada seja alcançado, como atingir uma profundidade máxima.

Dessa forma, para medir a qualidade dessas divisões, são utilizados critérios como a Impureza de Gini (Gini *Impurity*) ou a entropia. O critério de Impureza é definido para um nó t com as classes $\{1, 2, ..., K\}$. A Impureza de Gini quantifica o grau de desordem em um conjunto de dados, calculando a probabilidade de um elemento escolhido aleatoriamente ser classificado incorretamente. A Impureza de Gini é definida como:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2 \tag{2}$$

onde p_k representa a proporção de instâncias da classe k no nó t [Breiman et al. 1984]. Um valor de Gini igual a zero indica pureza completa (todas as instâncias pertencem à mesma classe), enquanto valores próximos a 1 indicam alta heterogeneidade.

Alternativamente, o critério de entropia, baseado na teoria da informação, pode ser empregado:

$$Entropia(t) = 1 - \sum_{k=1}^{k} p_k log_2(p_k) \tag{3}$$

As árvores de decisão apresentam vantagens significativas, incluindo interpretabilidade intuitiva e capacidade de modelar relações não lineares. Entretanto, são suscetíveis a *overfitting* quando não adequadamente regularizadas através de parâmetros como profundidade máxima e número mínimo de amostras por folha [Hastie, Tibshirani e Friedman 2009].

2.2.2 *Support Vector Machine* (SVM)

O *Support Vector Machine* é um algoritmo de aprendizado supervisionado desenvolvido por [Vapnik 1995] que busca encontrar o hiperplano ótimo que maximiza a margem de separação entre classes no espaço de atributos. Para problemas linearmente separáveis, o SVM resolve o problema de otimização:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ sujeito a } y_i \left(w^T x_i + b \right) \geq 1, \forall i \tag{4}$$

onde w representa o vetor de pesos, b o termo de viés, x_i as instâncias de treinamento e $y_i \in \{-1, +1\}$ os rótulos de classe [Cortes e Vapnik 1995].

Para lidar com dados não linearmente separáveis, o SVM emprega o conceito de margem suave, introduzindo variáveis de folga ξ_i e um parâmetro de regularização C que controla o *trade-off* entre a maximização da margem e a minimização do erro de classificação [Bennett e Campbell 2000]:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \mathrm{C} \sum_{i=1}^{n} \xi_i \tag{5}$$

O SVM é particularmente eficaz em espaços de alta dimensionalidade, como os gerados pela vetorização TF-IDF [Joachims 1998], mas seu desempenho pode ser limitado quando há forte sobreposição semântica entre as classes [Denil e Trappenberg 2011].

2.2.3 *Multilayer Perceptron* (MLP)

O *Multilayer Perceptron* constitui uma arquitetura de rede neural artificial composta por múltiplas camadas de neurônios artificiais totalmente conectados [Rumelhart, Hinton e Williams 1986]. Cada neurônio aplica uma transformação não linear aos sinais de entrada através de uma função de ativação. A arquitetura típica inclui:

- Camada de entrada: Recebe a representação vetorial do texto.
- Camadas ocultas: Transformam progressivamente as representações através de combinações lineares e funções de ativação não lineares.
- Camada de saída: Produz as probabilidades de classe através de uma função softmax.

Para um neurônio j em uma camada l, a ativação é calculada como:

$$a_j^{(l)} = f \left(\sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)} \right) \tag{6}$$

onde $w_{ij}^{(l)}$ representa os pesos sinápticos, $b_j^{(l)}$ o termo de viés, e $f(\cdot)$ a função de ativação. Funções de ativação comuns incluem ReLU (*Rectified Linear Unit*) [Nair e Hinton 2010], Sigmoid, Softmax (utilizada na camada de saída para classificação multiclasse), entre outras.

O treinamento de MLP é realizado através do algoritmo de retropropagação (*backpropagation*), que calcula gradientes da função de perda em relação aos parâmetros da rede [Rumelhart, Hinton e Williams 1986]. Otimizadores adaptativos como Adam (*Adaptive Moment Estimation*) ajustam dinamicamente as taxas de aprendizado para cada parâmetro, acelerando a convergência [Kingma e Ba 2015].

Para mitigar o *overfitting*, técnicas de regularização são essenciais, destacando-se o *dropout*, que desativa aleatoriamente uma fração de neurônios durante o treinamento [Srivastava et al. 2014], e o *early stopping*, que interrompe o treinamento quando o desempenho em um conjunto de validação começa a deteriorar [Prechelt 1998].

2.3 Modelos de Linguagem de Grande Porte

Os Modelos de Linguagem de Grande Porte representam uma mudança paradigmática no processamento de linguagem natural, baseando-se em arquiteturas de redes neurais profundas, particularmente a arquitetura *Transformer*, proposta por [Vaswani et al. 2017]. Diferentemente dos modelos clássicos que requerem engenharia manual de atributos e treinamento específico para cada tarefa, as LLM são pré-treinadas em vastos *corpora* textuais e podem ser adaptadas para diversas tarefas através de técnicas de *prompting* [Brown et al. 2020].

2.3.1 Arquitetura *Transformer* e Mecanismo de Atenção

A arquitetura *Transformer* fundamenta-se no mecanismo de *self-attention* (auto-atenção), que permite ao modelo capturar dependências contextuais de longo alcance sem as limitações das redes recorrentes [Vaswani et al. 2017]. O mecanismo de atenção calcula representações ponderadas dos *tokens* de entrada, onde os pesos são determinados pela relevância contextual:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

onde Q (*queries*), K (*keys*) e V (*values*) são projeções lineares dos *embeddings* de entrada, e d_k é a dimensionalidade das *keys*. O *multi-head attention* permite que o modelo aprenda múltiplas relações contextuais simultaneamente [Vaswani et al. 2017].

2.3.2 Engenharia de *Prompt* (Aprendizagem Contextualizada)

A engenharia de *prompt* refere-se ao processo de projeto e otimização das instruções textuais fornecidas às LLM para maximizar o desempenho em uma tarefa específica. Elementos críticos incluem:

- Clareza e especificidade: instruções inequívocas sobre o formato de saída esperado;
- Contexto relevante: informações de *background* que orientem a interpretação;
- Restrições explícitas: limitações sobre o conjunto de respostas válidas; e
- Estruturação: organização lógica que facilite o processamento pelo modelo.

A qualidade do *prompt* impacta significativamente o desempenho, podendo resultar em variações de dezenas de pontos percentuais em acurácia.

Paradigmas de Aprendizado: *Zero-Shot* e *Few-Shot Learning*

As LLM manifestam capacidades emergentes de aprendizado *in-context*, permitindo a adaptação a novas tarefas sem atualização de parâmetros [Brown et al. 2020]. Essa é uma das principais vantagens das LLM – sua aptidão para generalizar em tarefas com poucos ou nenhum exemplo, através de técnicas como *zero-shot learning* e *few-shot learning*. Isso elimina a necessidade de treinamento supervisionado extenso para cada tarefa específica, tornando-as apropriadas para problemas com baixa disponibilidade de dados rotulados.

No *zero-shot learning*, o modelo recebe apenas uma descrição textual da tarefa (*prompt*) e deve realizar a classificação baseando-se exclusivamente em seu conhecimento pré-treinado

[Radford et al. 2019]. Por exemplo, o *prompt* pode solicitar: "Classifique o seguinte resumo científico nas categorias: Física, Química, Biologia ou Ciências Gerais. Resumo: [texto] Categoria:". Um exemplo do uso do paradigma *Zero-Shot Learning* é ilustrado na Figura 1.

```
Classifique o seguinte resumo científico nas categorias:
Física, Química, Biologia ou Ciências Gerais.

Resumo: [texto]
Categoria:
```

Figura 1: Exemplo do paradigma *Zero Shot Learning*.

Já no *few-shot learning*, o modelo recebe alguns exemplos (tipicamente 1 a 5) de cada classe dentro do *prompt*, permitindo que aprenda o padrão desejado por analogia [Brown et al. 2020]. A eficácia do *few-shot learning* está relacionada à capacidade das LLM de realizar inferência bayesiana implícita sobre a tarefa, ajustando suas previsões com base nos exemplos fornecidos [Xie et al. 2022]. Um exemplo do uso do paradigma *Few-Shot Learning* é ilustrado na Figura 2.

```
Exemplo 1 - Física: [texto] → Física
Exemplo 2 - Química: [texto] → Química
...
Classifique: [novo texto] →
```

Figura 2: Exemplo do paradigma *Few Shot Learning*.

LLaMA: *Large Language Model* Meta AI

O LLaMA (*Large Language Model Meta AI*) constitui uma família de modelos autorregressivos de linguagem desenvolvida pela Meta AI Research, cuja filosofia de projeto prioriza a democratização do acesso a modelos de linguagem de alta performance através da eficiência computacional e da disponibilização em código aberto [Touvron et al. 2023]. Diferentemente de outros LLM proprietários que requerem infraestrutura computacional massiva, a arquitetura LLaMA foi otimizada para alcançar desempenho competitivo com modelos significativamente maiores, mantendo requisitos de hardware mais modestos. Esta característica torna os modelos LLaMA particularmente adequados para aplicações em ambientes com recursos limitados, como instituições acadêmicas e organizações de pesquisa que não dispõem de *clusters* de GPUs de alto desempenho.

A família LLaMA é composta por modelos com diferentes escalas de parâmetros, variando de 7 bilhões a 70 bilhões na versão LLaMA 2, permitindo que usuários selecionem a configuração mais apropriada conforme as restrições computacionais e as demandas de desempenho de suas aplicações específicas [Touvron et al. 2023]. A versão LLaMA 3 com 8 bilhões de parâmetros, utilizada neste estudo, representa um equilíbrio estratégico entre capacidade expressiva e viabilidade de execução em hardware convencional, oferecendo desempenho robusto em tarefas de compreensão e geração de linguagem natural sem exigir aceleradores de hardware especializados.

Do ponto de vista arquitetural, o LLaMA implementa diversas otimizações em relação à arquitetura *Transformer* padrão. O modelo utiliza a normalização RMSNorm (*Root Mean Square*

Layer Normalization) em vez da LayerNorm tradicional, resultando em maior estabilidade durante o treinamento e redução do custo computacional [Zhang e Sennrich 2019]. Adicionalmente, emprega a função de ativação SwiGLU (*Swish-Gated Linear Unit*) nas camadas *feed-forward*, que demonstra desempenho superior em comparação com ativações ReLU convencionais [Shazeer 2020]. O mecanismo de atenção foi aprimorado através da incorporação de *Rotary Position Embeddings* (RoPE), uma técnica que codifica informações posicionais de forma mais eficiente e permite melhor generalização para sequências de comprimento variável [Su et al. 2021].

Por meio da quantização é possível executar o modelo em um hardware convencional. A quantização é uma técnica de compressão que reduz a precisão numérica dos parâmetros do modelo, tipicamente de 16 bits (FP16) ou 32 bits (FP32) para 4 ou 8 bits [Gholami et al. 2022]. No esquema de quantização Q4_0 e Q4_K_M, os pesos são representados com 4 bits, resultando em redução significativa do consumo de memória (aproximadamente 75% comparado a FP16), diminuição dos requisitos de largura de banda de memória, e possibilidade de execução em hardware sem GPU dedicada [Dettmers et al. 2022]. Embora a quantização introduza uma degradação marginal na precisão devido à perda de informação numérica, estudos demonstram que modelos quantizados mantêm grande parte de sua capacidade de inferência, especialmente em tarefas de classificação e geração de texto [Frantar et al. 2023].

O processo de treinamento do LLaMA envolveu corpus textuais massivos e multilíngues, totalizando trilhões de *tokens* provenientes de fontes diversificadas, incluindo CommonCrawl, C4, Github, Wikipedia, livros, artigos científicos ArXiv e código-fonte StackExchange [Touvron et al. 2023]. Esta diversidade de dados de treinamento confere ao modelo uma compreensão ampla de diferentes domínios de conhecimento e estilos textuais, facilitando sua aplicação em tarefas especializadas através de técnicas de *prompting*. Particularmente relevante para este estudo é a presença de conteúdo científico no corpus de pré-treinamento, o que potencialmente contribui para a capacidade do modelo de compreender e classificar textos acadêmicos com terminologia técnica e estrutura característica de teses e dissertações.

2.4 Métricas de Avaliação para Classificação Multiclasse

A avaliação rigorosa de modelos de classificação requer métricas que capturem diferentes aspectos do desempenho. Para cada classe c, define-se [Sokolova e Lapalme 2009]:

- Verdadeiros Positivos (TP): Instâncias corretamente classificadas como pertencentes à classe c;
- Falsos Positivos (FP): Instâncias incorretamente classificadas como pertencentes à classe c;
- Falsos Negativos (FN): Instâncias da classe c classificadas em outras categorias.

As métricas fundamentais são:

- Precisão (*Precision*): Mede a proporção de classificações positivas que estavam de fato corretas. É útil para entender quantos falsos positivos o modelo gerou.

$$Precision_c = \frac{TP_c}{TP_c + FP_c} \tag{8}$$

- Revocação/*Recall*: Mede a capacidade do modelo de identificar todas as instâncias positivas relevantes para uma determinada classe. Também é conhecida como taxa de verdadeiros positivos.

$$Recall_c = \frac{TP_c}{TP_c + FN_c} \tag{9}$$

- F1-*Score*: Corresponde à média harmônica entre a precisão e o *recall*, fornecendo uma métrica única que equilibra ambos os valores. É especialmente útil em cenários com desbalanceamento de classes ou custos desiguais para falsos positivos e falsos negativos.

$$F1_c = 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c} \tag{10}$$

- Acurácia Global: Proporção de previsões corretas sobre o total de instâncias de determinada classe.

$$Acuracia = \frac{\sum_c TP_c}{N} \tag{11}$$

Para agregação em problemas multiclasse, utiliza-se *macro-average*, ou seja, média aritmética simples das métricas por classe, tratando todas as classes igualmente, e *weighted-average*, ou seja, média ponderada pela frequência de cada classe [Sokolova e Lapalme 2009].

A matriz de confusão complementa a análise quantitativa. Trata-se de uma tabela que possibilita visualizar o desempenho de um algoritmo de classificação, mostrando os acertos e erros para cada classe. Ela permite identificar quais classes estão sendo mais confundidas entre si, visualizando os padrões de erro e permitindo identificar confusões sistemáticas entre pares de classes específicos [Stehman 1997].

2.5 Validação Cruzada e Generalização

A validação cruzada *k-fold* é uma técnica de avaliação que particiona o conjunto de dados em k subconjuntos (*folds*) de tamanho aproximadamente igual [Kohavi 1995]. O modelo é treinado k vezes, utilizando cada vez $k-1$ *folds* para treinamento e o *fold* restante para validação. A métrica final corresponde à média das k avaliações, fornecendo uma estimativa mais robusta e menos enviesada do desempenho em dados não vistos.

Essa abordagem é particularmente importante para *datasets* de tamanho limitado, maximizando o uso dos dados disponíveis tanto para treinamento quanto para avaliação, ao mesmo tempo que reduz a variância das estimativas de desempenho [Arlot e Celisse 2010].

3 Metodologia

A Figura 3 apresenta as etapas da metodologia.

3.1 Conjunto de dados

O conjunto de dados é uma planilha com 9.363 registros, contendo dados de teses e dissertações na área de Ensino de Ciências da Natureza, defendidas entre 1972 e 2019. A coleta dos dados foi realizada pelo Grupo do Projeto EAEC [EAEC 2025], com dados do Catálogo de Teses e Dissertações da CAPES e do Banco de Dados de Teses e Dissertações (BDTD). Cada registro contém descritores de base institucional e descritores de base específica.

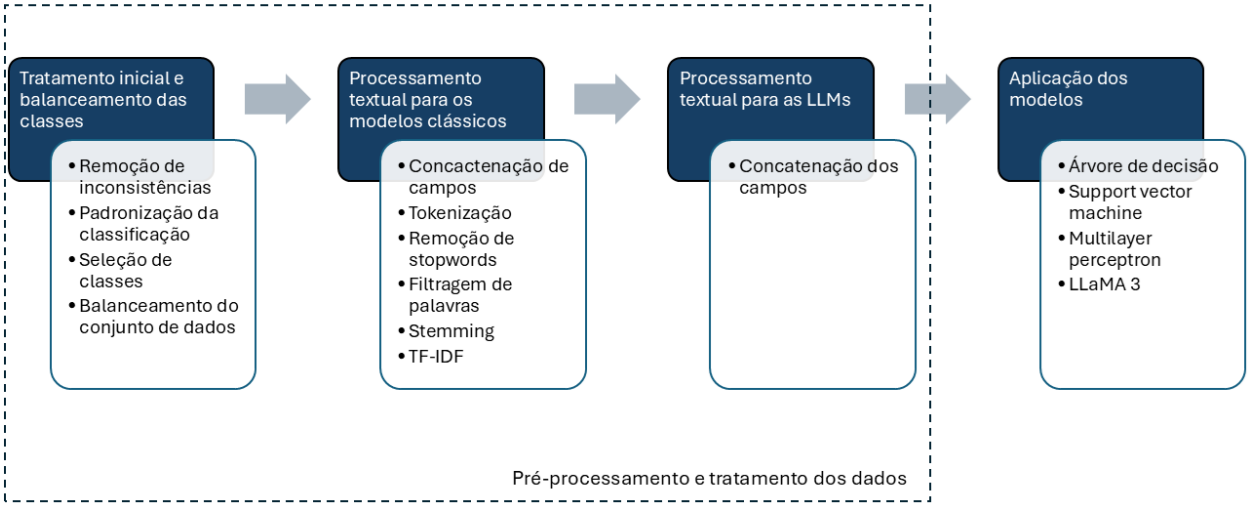

Figura 3: Etapas da metodologia.

Os descritores de base institucional são metadados utilizados em portais acadêmicos para descrever as teses e as dissertações. Esses incluem: Autor(a), Orientador(a), Ano de Defesa, Grau Acadêmico, Programa de Pós-Graduação em que o trabalho foi defendido, Área da CAPES em que o programa de pós-graduação está vinculado, Instituição sede do programa de pós-graduação, Cidade sede do programa de pós-graduação, Unidade Federativa da instituição sede do programa de pós-graduação, Natureza Administrativa, Título da Tese/Dissertação, Resumo, Palavras-chave e link para o arquivo PDF [EAEC 2025].

Os descritores de base específica são incluídos a critério dos interesses do pesquisador sobre o campo de conhecimento a ser estudado [Megid-Neto e Carvalho 2018, p.14]: Os descritores de base específica do *dataset* que utilizamos são referentes ao contexto educacional, por isso são chamados de descritores de base educacional. Esses descritores são os seguintes: Modalidade de Ensino, Nível Educacional, Área de Conhecimento das Ciências da Natureza e Foco Temático Principal [EAEC 2025]. Diferentemente dos descritores de base institucional, os descritores de base educacional foram classificados manualmente pela Equipe de pesquisadores do Projeto EAEC.

Vale reforçar que o objetivo deste estudo é comparar modelos de aprendizado de máquina para que a classificação dos descritores de base educacional possa ser feita automaticamente, com mínimas intervenções humanas. Neste relatório, descrevemos os resultados para a classificação automática do descritor **Área de Conhecimento das Ciências da Natureza**, que pode ser classificado com os rótulos: Astronomia, Biologia, Física, Geociências, Química e Ciências-geral.

Para apoiar a classificação automática da área de conhecimento, utilizamos os descritores **Título da Tese/Dissertação, Resumo e Palavras-chave**. O *dataset* original foi, portanto, reduzido a um subconjunto de dados de interesse.

3.2 Pré-processamento e Tratamento dos Dados

O procedimento metodológico iniciou-se com a etapa crítica de tratamento e pré-processamento dos dados brutos, provenientes do *dataset* contendo título, resumo e palavras-chaves dos 9.363 registros de teses previamente rotuladas manualmente. Essa fase foi essencial para garantir a qualidade e a viabilidade da aplicação dos modelos de classificação. O processo foi dividido em duas subetapas principais: tratamento e padronização do conjunto de dados, e processamento textual específico para os modelos clássicos de aprendizado de máquina (ML).

3.2.1 Tratamento Inicial e Balanceamento de Classes

O tratamento inicial dos dados concentrou-se na limpeza e na padronização da base de dados. As inconsistências e os ruídos presentes foram mitigados através das seguintes ações:

1. **Remoção de Inconsistências e Limpeza**: Foram removidas linhas corrompidas, vazias ou que continham campos em excesso.

2. **Seleção de Classes**: Áreas de ciências minoritárias, com baixa quantidade de instâncias, foram removidas do *dataset*. Essa seleção visou concentrar o conjunto de dados nas áreas que continham a maior parte das informações. Ao final desta etapa, restaram quatro classes principais: Física (f), Química (q), Biologia (b) e Ciências Gerais (c).

3. **Balanceamento do Conjunto de Dados**: Para evitar o viés de treinamento em classes mais frequentes, foi realizado o balanceamento do *dataset*. Todas as classes finais foram **niveladas pela classe com a menor quantidade de instâncias**, que correspondia a Química (999 instâncias).

4. **Padronização da Classificação**: Observou-se a falta de padronização nos rótulos de área de conhecimento (e.g., "fis", "física", ou "f"). Para solucionar essa questão, todas as classificações foram **simplificadas e encurtadas para apenas uma única letra minúscula** (f para Física, c para Ciências Gerais, q para Química, b para Biologia).

A decisão de nivelar todas as classes pela menor delas (Química, com 999 instâncias) foi adotada para evitar que os modelos supervisionados desenvolvessem viés em favor das classes majoritárias durante o treinamento. Embora essa estratégia reduza o volume total de dados disponível, ela garante que as métricas de avaliação reflitam o desempenho real de cada classe de forma equitativa, sem que classes mais frequentes inflem artificialmente os resultados globais.

3.2.2 Processamento Textual para Modelos Clássicos

Uma etapa adicional de pré-processamento foi aplicada ao texto das teses, especificamente para o treinamento dos classificadores clássicos (Árvore de Decisão, SVM e MLP). Essa etapa, ilustrada na Figura 4, foi essencial antes da vetorização do texto e incluiu:

1. **Concatenação de Campos**: Os campos textuais "título", "resumo" e "palavras-chave" das teses foram **unidos em um único campo**. Essa estratégia foi implementada para **aumentar a densidade de informações** semânticas disponíveis para os modelos.

2. **Tokenização**: O texto unificado foi segmentado em unidades lexicais (tokens), utilizando separadores padrão (tais como vírgula, ponto final, espaço, etc).

3. **Remoção de *Stopwords***: Palavras de alta frequência e baixo valor semântico (e.g., "de", "para", "com") foram eliminadas utilizando uma lista padrão da biblioteca NLTK, com suporte para o português brasileiro.

4. **Filtragem de Palavras Irrelevantes**: Além das *stopwords*, foram **removidas palavras específicas que se mostraram ambíguas ou pouco informativas** para a diferenciação entre classes (e.g., "ensino", "professores", "pesquisa", "dados"), pois apareciam em virtualmente todos os documentos.

5. **Stemming**: O processo de *stemming* foi aplicado para **reduzir cada palavra ao seu radical truncado**, visando reduzir ruídos e redundâncias no conjunto de dados. A alternativa de *lemmatization* foi considerada, mas descartada devido à baixa precisão observada em ferramentas para o idioma português, além que exigir mais recursos computacionais. A opção pelo *stemming* em detrimento da lemmatização foi motivada por dois fatores práticos: a baixa precisão das ferramentas de lemmatização disponíveis para o português brasileiro e o custo computacional adicional que esse processo impõe. Como o objetivo da etapa é reduzir a dispersão vocabular, o *stemming* mostrou-se suficiente para a tarefa, sem a necessidade de análise morfossintática completa que a lemmatização exigiria.

6. **Vetorização com TF-IDF**: Por fim, os textos processados foram convertidos em representações numéricas por meio da técnica **Term Frequency-Inverse Document Frequency (TF-IDF)**. O TF-IDF atribui um peso maior a palavras que são mais representativas de cada documento em relação ao corpo total de textos (corpus), preparando os dados para a entrada nos classificadores clássicos. A vetorização por TF-IDF foi escolhida por ser uma técnica consolidada para representação de textos em tarefas de classificação supervisionada, especialmente eficaz quando combinada com algoritmos como o SVM [Joachims 1998]. Sua principal vantagem neste contexto é atribuir maior peso a termos que são informativos para documentos específicos e reduzir a influência de termos que aparecem em praticamente todos os documentos do corpus, como verbos e substantivos genéricos que persistiram após a remoção de *stopwords*.

Figura 4: Fluxograma do pré-processamento textual.

3.2.3 Processamento Textual para LLM

De forma similar ao tratamento realizado para os modelos clássicos de ML, **o título e as palavras-chave** foram concatenadas em um único campo, de modo a formarem um corpo textual contínuo.

Isso foi feito considerando o fato de LLM darem respostas com base no contexto geral do texto, e não trabalhando com palavras individualmente, como é o caso do TF-IDF para modelos clássicos. Dessa forma, **o modelo utilizado consegue extrair a semântica do texto**, a fim de aumentar a qualidade da classificação.

3.3 Aplicação, avaliação e ajuste dos modelos

Após o tratamento dos dados, os modelos foram treinados e avaliados. Os resultados obtidos da avaliação dos modelos permitiu o ajuste de parâmetros para melhorar, otimizar a performance e extrair características do texto que agregassem na separabilidade de classes. Com isso, cada algoritmo foi ajustado para sua eficiência máxima, dentro das limitações do modelo, dos dados e da máquina em que foram executados, conforme descrito a seguir:

Árvore de decisão: O modelo foi implementado utilizando a biblioteca **scikit-learn**. Os **hiperparâmetros** do modelo foram definidos como: `criterion='gini'`, `max_depth=10`, `min_samples _split=10`, `min_samples_leaf=2`, `class_weight='balanced'` e `random_state=42`. Para avaliação, foi empregada a técnica de **validação cruzada k-fold**, com k = 5, para obter uma estimativa mais robusta do desempenho.

Support Vector Machine: Foram testados três tipos de *kernel* — linear, radial (RBF) e polinomial —, que determinam como o algoritmo projeta os dados em espaços de maior dimensão para encontrar o hiperplano de separação ideal. O *kernel* linear obteve o melhor desempenho, resultado esperado para dados textuais representados por TF-IDF, nos quais a separação entre classes tende a ser aproximadamente linear no espaço de alta dimensão. O parâmetro de regularização C, que controla o equilíbrio entre maximizar a margem de separação e minimizar os erros de classificação no conjunto de treinamento, foi testado em valores entre 1 e 30. O valor ótimo obtido foi C=1, o menor do intervalo testado, indicando que uma margem mais ampla, ainda que com maior tolerância a erros individuais, generalizou melhor do que margens mais rígidas, o que é consistente com a sobreposição semântica presente entre as classes.

Multilayer Perceptron: A arquitetura utilizada consistiu em três camadas ocultas com 64, 32 e 16 neurônios, respectivamente, com redução progressiva de capacidade para forçar a compressão das representações aprendidas. A função de ativação ReLU foi adotada nas camadas ocultas por sua eficiência computacional e por mitigar o problema do desaparecimento do gradiente em redes mais profundas. Para reduzir o *overfitting*, foram aplicadas camadas de *dropout* após cada camada oculta, com taxas de 0,5, 0,3 e 0,3 respectivamente — taxas mais altas nas camadas iniciais, onde a representação é mais densa, e menores nas camadas finais. A camada de saída continha 4 neurônios com função de ativação `softmax`, que converte os valores de saída em probabilidades para cada uma das quatro classes. O modelo foi compilado com o otimizador Adam e taxa de aprendizado de 0,0001, valor conservador escolhido para garantir convergência estável. O treinamento foi configurado para até 25 épocas com *early stopping*, mecanismo que interrompe o treinamento automaticamente quando o desempenho no conjunto de validação deixa de melhorar, prevenindo o sobreajuste sem necessidade de definir antecipadamente o número ideal de épocas.

Large Language Model: No regime *zero-shot*, foi utilizado o modelo LLaMA3:8b em sua versão base. Essa versão foi escolhida por preservar o comportamento do modelo tal como resultante do pré-treinamento, sem ajustes adicionais para seguir instruções, o que permite avaliar a

capacidade de generalização intrínseca da LLM a partir exclusivamente do conhecimento adquirido durante o treinamento. No regime *few-shot*, foi utilizado o modelo LLaMA3:8b-instruct-q4_k_m, versão ajustada para seguir instruções (*instruction-tuned*) e quantizada no esquema Q4_K_M. A variante *instruction-tuned* é mais adequada para o regime *few-shot* pois foi otimizada para interpretar e seguir estruturas de *prompt* com exemplos, produzindo respostas mais consistentes com o formato solicitado. A quantização Q4_K_M reduz os pesos do modelo para 4 bits de precisão, resultando em redução de aproximadamente 75% no consumo de memória em relação à versão em meia-precisão (FP16), com qualidade superior às quantizações mais agressivas como Q4_0 e Q3 [Dettmers et al. 2022]. Ambas as versões foram executadas localmente por meio da ferramenta de código aberto Ollama, sem dependência de serviços externos, o que torna a abordagem replicável em ambientes acadêmicos com recursos computacionais limitados.

4 Resultados

Nesta seção, são apresentados os resultados obtidos por cada um dos modelos de classificação, tanto os clássicos quanto a LLM. A análise comparativa é realizada com base nas métricas de avaliação e nas matrizes de confusão, culminando em uma discussão que consolida os achados de todos os experimentos.

4.1 Desempenho dos Modelos Clássicos

4.1.1 Árvore de Decisão (*Decision Tree*)

Após a aplicação da validação cruzada (k-fold com k=5), o modelo de Árvore de Decisão alcançou uma **média macro de 86% tanto para precisão quanto para *recall***, o que representa um desempenho sólido.

Na Figura 5, pode-se ver que o modelo teve poucos falsos-positivos e falsos-negativos para a classificação das teses de Química (q), alguns falsos-positivos para Física (f) e maior confusão entre Biologia (b) e Ciências gerais (c).

A análise por classe, detalhada na Tabela 1, revela que o modelo performou de maneira excelente para as classes de Química (precisão de 94%) e Física (precisão de 93%). No entanto, o desempenho foi inferior para Biologia e Ciências Gerais.

Classe	**Precisão**	**Recall**
Biologia (b)	0,83	0,78
Ciências Gerais (c)	0,73	0,82
Física (f)	0,93	0,89
Química (q)	0,94	0,92

Tabela 1: Resultados do modelo Árvore de Decisão por classe.

4.1.2 *Support Vector Machine* (SVM)

Para o SVM, foram testados os kernels linear, radial e polinomial. O **SVM Linear obteve o melhor desempenho, com um F1-Score de 87%**. O resultado ótimo foi alcançado com o

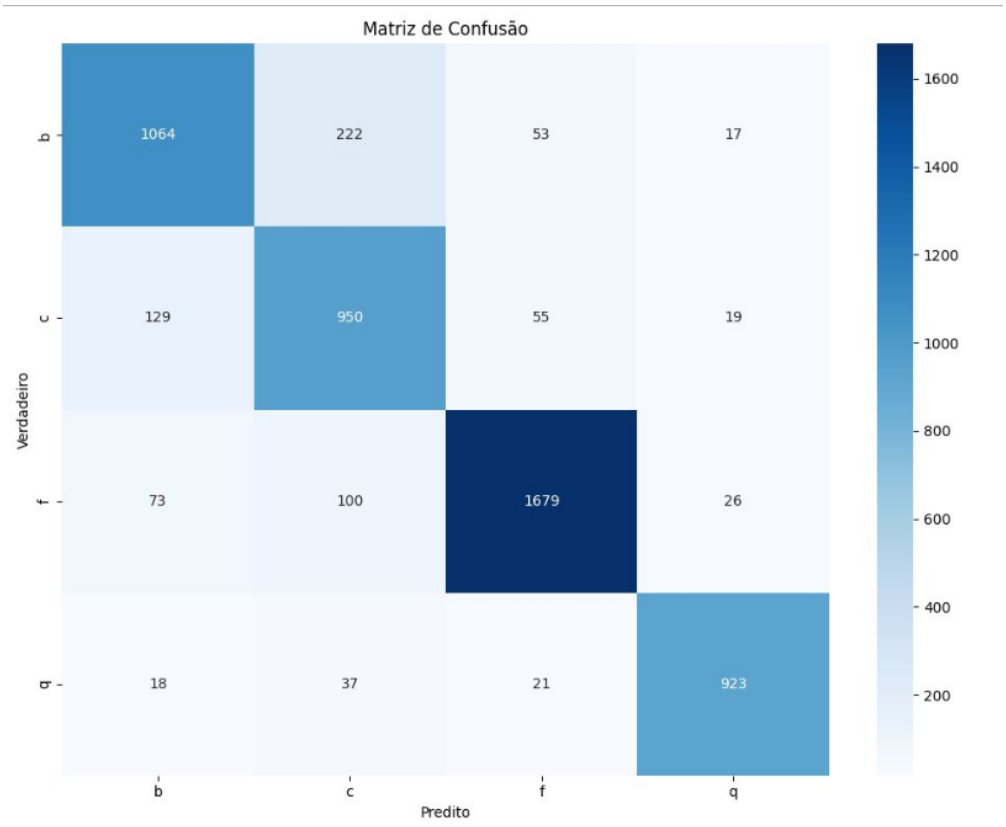

Figura 5: Matriz de confusão da Árvore de Decisão.

menor valor de margem (C=1), o que sugere uma alta sobreposição entre os dados das diferentes classes.

Conforme ilustrado na Figura 6, a precisão foi alta para Física, Química e Biologia (acima de 90%). O desempenho para a classe Ciências Gerais foi inferior, conforme evidenciado pela matriz de confusão na Figura 6.

4.1.3 *Multilayer Perceptron* (MLP)

O desempenho do MLP foi semelhante ao do SVM Linear. A precisão para as classes dominantes (Física, Química e Biologia) também ultrapassou 90%. A matriz de confusão (Figura 7) revelou o mesmo padrão verificado nos demais modelos clássicos, com maior concentração de erros na classe Ciências Gerais.

4.2 Desempenho da *Large Language Model* (LLaMA 3)

4.2.1 Regime *Zero-Shot*

Nesta abordagem, o modelo recebeu apenas instruções detalhadas, sem exemplos, para realizar a classificação. Foram desenvolvidos três *prompts* progressivamente mais complexos, com orientações semânticas e restrições de formato.

Conforme a Tabela 2, o melhor resultado neste regime foi obtido pelo Prompt 2, com acurácia de 78,8%, valor inferior aos 86% alcançados pela Árvore de Decisão. O Prompt 3, mais detalhado,

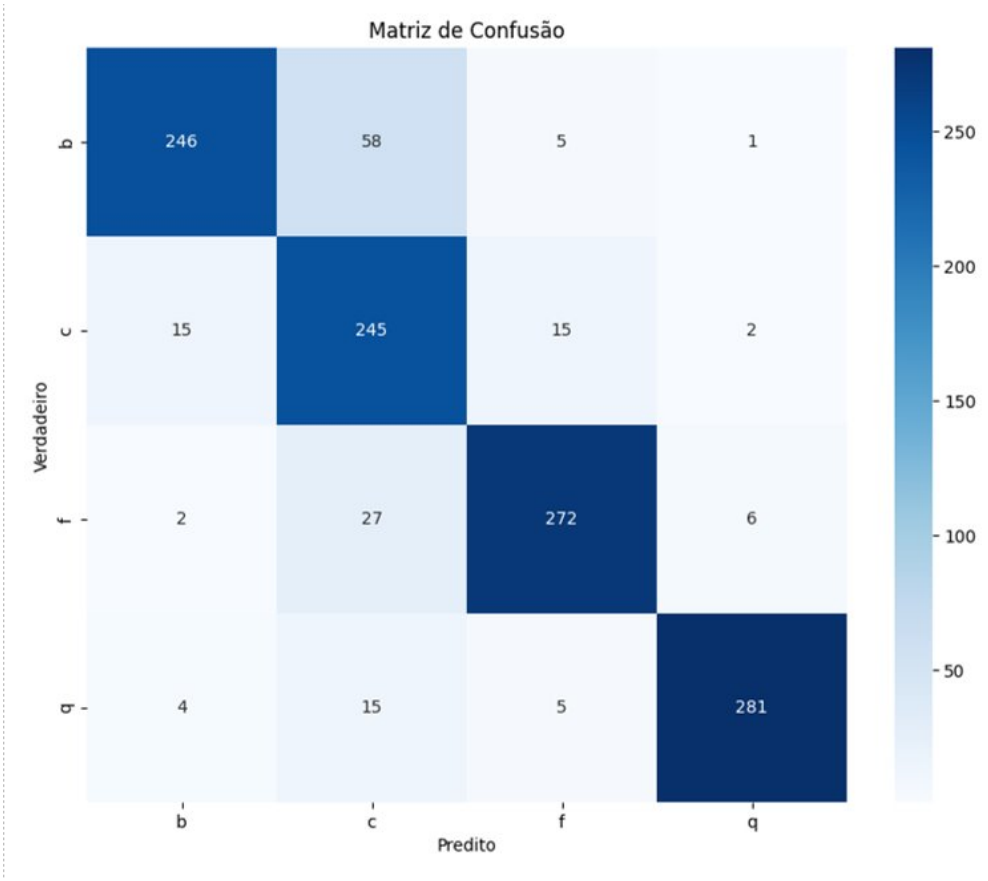

Figura 6: Matriz de confusão para o modelo SVM linear.

obteve acurácia de 74,8%.

Tabela 2: Comparação de desempenho: Árvore de Decisão vs. LLM (*Zero-Shot*).

Modelo	**Acurácia**	**Precisão**	***Recall***
Árvore de Decisão	86%	86%	86%
LLM (*Prompt* 1)	58,41%	75%	58%
LLM (*Prompt* 2)	78,8%	83%	79%
LLM (*Prompt* 3)	74,8%	-	-

4.2.2 Regime *Few-Shot*

Nesta abordagem, o modelo foi instruído com um *prompt* que continha um único exemplo de classificação para cada uma das quatro classes. O objetivo era avaliar se o modelo conseguiria generalizar a partir dessa pequena amostra.

O LLaMA 3 quantizado, operando em regime *few-shot*, atingiu acurácia global de 94%, com F1-Scores entre 92% e 96% para todas as classes. A matriz de confusão (Figura 8) indica redução expressiva dos erros em relação aos modelos clássicos, inclusive para a classe Ciências Gerais.

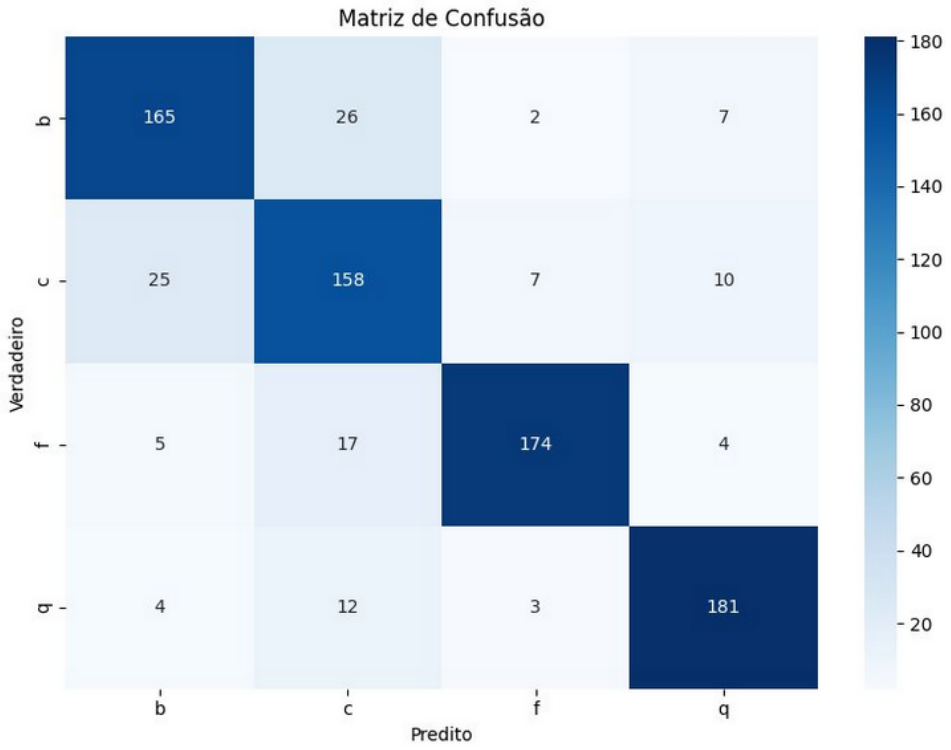

Figura 7: Matriz de confusão para o modelo MLP.

5 Discussão

A pergunta central desta investigação, *"Classificadores tradicionais são mais efetivos do que aprendizagem contextualizada (prompts LLM) para classificar teses e dissertações na área de ensino das ciências da natureza?"* foi respondida a partir dos resultados obtidos, embora com nuances que exigem uma reflexão aprofundada sobre as condições experimentais e as características intrínsecas dos dados.

Os modelos supervisionados clássicos (Árvore de Decisão, SVM Linear e MLP) apresentaram desempenho sólido e consistente, com precisão e F1-Score em torno de 86–87%. Esses valores estão alinhados com a literatura que aponta o SVM como particularmente eficaz em espaços de alta dimensionalidade gerados pela vetorização TF-IDF [Joachims 1998]. Contudo, o padrão de erro foi sistemático e comum aos três modelos: a classe Ciências Gerais concentrou a maior parte das confusões, sendo frequentemente classificada como Biologia ou vice-versa.

Esse comportamento não deve ser atribuído a uma falha específica dos algoritmos, mas a um limite intrínseco da representação TF-IDF diante de classes semanticamente adjacentes. Como apontam Denil e Trappenberg [Denil e Trappenberg 2011], o desempenho do SVM é reduzido quando há forte sobreposição semântica entre classes — condição verificada estruturalmente neste *dataset*, uma vez que conteúdos de Biologia são usualmente tratados dentro da disciplina de Ciências no Ensino Fundamental. A filtragem de palavras genéricas para o contexto trabalhado, como "ensino" e "professores", durante o pré-processamento atenuou parcialmente esse problema, sem, no entanto, eliminá-lo, o que indica que a limitação reside na natureza semântica dos próprios dados e não na escolha dos hiperparâmetros dos modelos.

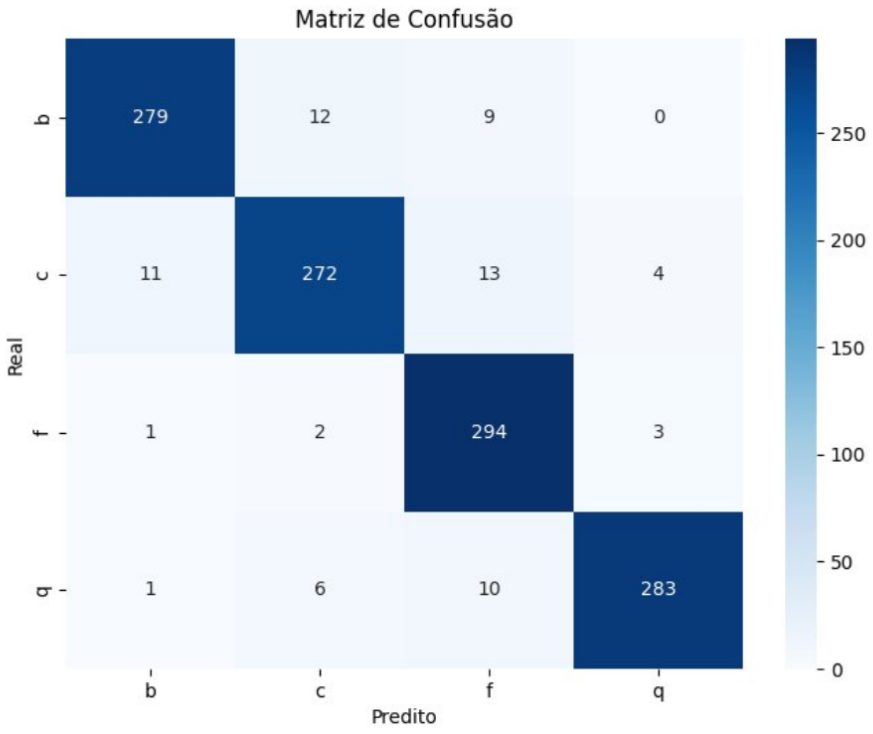

Figura 8: Matriz de confusão para o modelo LLaMA3 8b:instruct-q4_k_m.

Os resultados obtidos no regime *zero-shot* evidenciam um aspecto crítico do uso de LLM em tarefas de classificação: a capacidade do modelo não depende exclusivamente de sua arquitetura, mas também da qualidade da instrução fornecida. O fato de o *prompt* mais detalhado (Prompt 3) ter obtido desempenho inferior ao Prompt 2 contraria a intuição de que instruções mais elaboradas produzem necessariamente melhores resultados. Esse achado é consistente com as observações de Brown e colegas [Brown et al. 2020] sobre a sensibilidade das LLM à formulação dos *prompts*, e reforça que a engenharia de *prompt* constitui, ela mesma, uma variável experimental que deve ser tratada com rigor metodológico.

O salto de desempenho verificado entre os regimes *zero-shot* (acurácia máxima de 78,8%) e *few-shot* (acurácia de 94%) merece atenção especial. A diferença de aproximadamente 15 pontos percentuais, obtida pela adição de um único exemplo por classe no *prompt*, ilustra empiricamente a capacidade de inferência *in-context* das LLM descrita por Xie e colegas [Xie et al. 2022]. Para uma tarefa com classes de fronteiras semanticamente difusas, como a investigada neste estudo, o exemplo concreto funciona como uma âncora interpretativa que o modelo não consegue derivar a partir de instruções abstratas isoladas.

Um resultado de relevância prática destacada é que o modelo LLaMA 3 com 8 bilhões de parâmetros, quantizado no esquema Q4_K_M e executado localmente sem acelerador de hardware dedicado, superou todos os modelos clássicos avaliados quando operando em regime *few-shot*. Isso tem implicações diretas para instituições acadêmicas e de pesquisa que não dispõem de infraestrutura computacional robusta: é possível obter desempenho competitivo em tarefas de classificação textual sem recorrer a modelos proprietários ou a *clusters* de GPUs de alto desempenho. Embora a quantização introduza degradação marginal em relação à versão em precisão completa [Dettmers et al. 2022, Frantar et al. 2023], os resultados obtidos indicam que tal perda é desprezível para a tarefa de classificação investigada.

Algumas limitações do presente estudo devem ser reconhecidas. Em primeiro lugar, o balanceamento do *dataset* pelo menor número de instâncias disponíveis — 999 por classe, determinado pela área de Química — reduziu o volume total de dados para treinamento, o que pode ter penalizado os modelos clássicos de forma desproporcional, uma vez que as LLM não dependem desse conjunto para operar em regime *few-shot*. Em segundo lugar, a avaliação do LLaMA 3 em regime *few-shot* foi conduzida com apenas um exemplo por classe; diferentes volumes de exemplos não foram testados, o que impede conclusões sobre a curva de aprendizado *in-context* para esta tarefa específica. Por fim, os *prompts* foram desenvolvidos e refinados de forma iterativa, introduzindo um risco de sobreajuste do *prompt* ao conjunto de avaliação — fenômeno análogo ao *overfitting* em modelos supervisionados, porém menos visível e de difícil quantificação.

6 Conclusão

A classificação automática de documentos acadêmicos permanece um desafio relevante devido à sobreposição semântica entre áreas do conhecimento e à disponibilidade limitada de dados rotulados em domínios especializados. Neste estudo, comparamos modelos supervisionados clássicos (Árvore de Decisão, SVM e MLP) com o modelo LLaMA 3 para a classificação de teses e dissertações na área de Ciências da Natureza. Os resultados mostraram que os classificadores tradicionais alcançaram desempenho consistente, com precisão e F1-Score entre 86% e 87%, mas apresentaram dificuldades recorrentes na distinção entre classes semanticamente próximas, especialmente Biologia e Ciências Gerais. O LLaMA 3, por sua vez, obteve desempenho inferior no regime *zero-shot*, porém atingiu acurácia de 94% no regime *few-shot* com apenas um exemplo por classe, superando todos os modelos clássicos sem necessidade de treinamento supervisionado ou de infraestrutura computacional especializada. Esses resultados indicam que as LLM em regime *few-shot* constituem uma alternativa promissora para a classificação de textos acadêmicos em português, especialmente em cenários com restrições de dados rotulados e de recursos computacionais, abrindo oportunidades para investigações futuras envolvendo diferentes quantidades de exemplos contextuais, estratégias de *fine-tuning* e a incorporação de atributos contextuais adicionais. Trabalhos futuros irão avaliar o impacto do volume de exemplos no *few-shot*, o uso de LLM em regime *fine-tuned* no domínio específico, e a incorporação de atributos contextuais adicionais — como ano de publicação, instituição e programa de pós-graduação — que possam aprimorar a distinção entre classes semanticamente próximas.

Referências

[Aggarwal e Zhai 2012]AGGARWAL, C. C.; ZHAI, C. *Mining Text Data*. [S.l.]: Springer Science & Business Media, 2012.

[Arlot e Celisse 2010]ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. *Statistics Surveys*, v. 4, p. 40–79, 2010.

[Bennett e Campbell 2000]BENNETT, K. P.; CAMPBELL, C. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, v. 2, n. 2, p. 1–13, 2000.

[Breiman et al. 1984]BREIMAN, L. et al. *Classification and Regression Trees*. [S.l.]: CRC Press, 1984.

[Brown et al. 2020]BROWN, T. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, v. 33, p. 1877–1901, 2020.

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

[Denil e Trappenberg 2011]DENIL, M.; TRAPPENBERG, T. A characterization of the combined effects of overlap and imbalance on the svm classifier. *arXiv preprint*, Jan 2011.

[Dettmers et al. 2022]DETTMERS, T. et al. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, v. 35, p. 30318–30332, 2022.

[EAEC 2025]EAEC. *Projeto EAEC – Estado da Arte da Pesquisa em Educação em Ciências no Brasil – Relação dos Descritores Analíticos – versão dezembro de 2025"*. 2025. Https://www.cedoc.fe.unicamp.br/catalogo. Acessado em abril de 2026.

[Frantar et al. 2023]FRANTAR, E. et al. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2023.

[Gholami et al. 2022]GHOLAMI, A. et al. A survey of quantization methods for efficient neural network inference. In: *Low-Power Computer Vision*. [S.l.]: Chapman and Hall/CRC, 2022. p. 291–326.

[Hastie, Tibshirani e Friedman 2009]HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. [S.l.]: Springer, 2009.

[Ho 2024]HO, J. *Machine Learning: Supervised Learning*. 2024. https://johoblogs.medium.com/machine-learning-supervised-learning-3f6bdc00c5bb. Acessado em 2025.

[Joachims 1998]JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: SPRINGER. *European Conference on Machine Learning*. [S.l.], 1998. p. 137–142.

[Kingma e Ba 2015]KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015.

[Kohavi 1995]KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 1995. v. 14, n. 2, p. 1137–1145.

[Manning, Raghavan e Schütze 2008]MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008.

[Megid-Neto e Carvalho 2018]MEGID-NETO, J.; CARVALHO, L. M. Pesquisas de estado da arte: fundamentos, características e percursos metodológicos. In: ESCHENHAGEN, G. M. L. et al. (Ed.). *Construcción de problemas de investigación: diálogos entre el interior y el exterior*. Medellin: Universidad Pontifica Bolivariana / Universidad de Antioquia, 2018. p. 97–113.

[Mitchell 1997]MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw Hill, 1997.

[Nair e Hinton 2010]NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814.

[Plisson, Lavrac e Mladenic 2004]PLISSON, J.; LAVRAC, N.; MLADENIC, D. A rule based approach to word lemmatization. In: *Proceedings of IS*. [S.l.: s.n.], 2004. v. 3, p. 83–86.

[Porter 1980]PORTER, M. F. An algorithm for suffix stripping. *Program*, v. 14, n. 3, p. 130–137, 1980.

[Prechelt 1998]PRECHELT, L. Early stopping-but when? In: *Neural Networks: Tricks of the Trade*. [S.l.]: Springer, 1998. p. 55–69.

[Radford et al. 2019]RADFORD, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog*, v. 1, n. 8, p. 9, 2019.

[Ramos 2003]RAMOS, J. Using tf-idf to determine word relevance in document queries. In: *Proceedings of the First Instructional Conference on Machine Learning*. [S.l.: s.n.], 2003. v. 242, n. 1, p. 29–48.

[Rumelhart, Hinton e Williams 1986]RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986.

[Sebastiani 2002]SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys*, v. 34, n. 1, p. 1–47, 2002.

[Shazeer 2020]SHAZEER, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[Sokolova e Lapalme 2009]SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, v. 45, n. 4, p. 427–437, 2009.

[Srivastava et al. 2014]SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, v. 15, n. 1, p. 1929–1958, 2014.

[Stehman 1997]STEHMAN, S. V. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, v. 62, n. 1, p. 77–89, 1997.

[Su et al. 2021]SU, J. et al. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[Touvron et al. 2023]TOUVRON, H. et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Vapnik 1995]VAPNIK, V. *The Nature of Statistical Learning Theory*. [S.l.]: Springer, 1995.

[Vaswani et al. 2017]VASWANI, A. et al. Attention is all you need. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2017. p. 5998–6008.

[Vijayarani, Ilamathi e Nithya 2015]VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, v. 5, n. 1, p. 7–16, 2015.

[Xie et al. 2022]XIE, S. M. et al. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2022.

[Zhang e Sennrich 2019]ZHANG, B.; SENNRICH, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, v. 32, 2019.