Benchmarking Agentic Capabilities of LLMs in Data Science: Prompt Analysis and Performance of Open-Weight Models

Hudson Diego Ferreira de Oliveira[1], **Cesar Augusto Tacla**[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPGCA-CT)
Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba – PR – Brazil

hudsondiego@alunos.utfpr.edu.br, tacla@utfpr.edu.br

Abstract. *This work analyzes three LLM benchmarks for data science: DS-1000, InfiAgent-DABench, and DataSciBench. The focus is on the evolution of evaluated tasks, from code generation in constrained settings to agent-oriented scenarios involving planning, tool use, code execution, and error correction. The analysis covers prompt complexity, the data science areas represented in each benchmark, and the performance of open-weight models. The methodology combines prompt difficulty analysis, topic modeling with BERTopic, and local evaluation of models in the 24B to 30B parameter range. The results show that newer benchmarks include more complex prompts, broader coverage of the data science workflow, and stronger dependence on capabilities associated with LLM-based agents. In the performance evaluation, Qwen3.6-27B and GLM-4.7-30B achieved competitive results in tasks that require tool use, code execution, and iterative correction.*

1. Introdução

Os benchmarks de modelos de linguagem de grande escala (LLMs, do inglês *Large Language Models*) tornaram-se essenciais para avaliar a capacidade desses modelos em múltiplos domínios [Chang et al. 2024], e recentemente têm ganhado destaque em tarefas de ciência de dados [Lai et al. 2023], sobretudo diante da expansão do paradigma de agentes e da crescente demanda por automação de processos de análise de dados [Hu et al. 2024].

Nesse contexto, este artigo investiga a estrutura, o escopo e o comportamento de três benchmarks de destaque e frequentemente citados na área: DS-1000 [Lai et al. 2023], InfiAgent-DABench [Hu et al. 2024] e DataSciBench [Zhang et al. 2026].

O objetivo deste trabalho é analisar benchmarks de ciência de dados sob a perspectiva de sistemas agênticos e de avaliação. Da perspectiva de sistemas agênticos, o estudo considera cenários em que LLMs deixam de ser avaliados apenas como geradores de texto ou código e passam a executar tarefas que envolvem planejamento de ações, uso de ferramentas, interação com ambientes isolados, como *sandboxes*, e correção iterativa em tempo real. Da perspectiva de avaliação, o artigo realiza uma meta-análise quantitativa e qualitativa de benchmarks que medem essas capacidades, examinando a dificuldade estrutural dos prompts e a cobertura das tarefas avaliadas, desde aquisição, limpeza e manipulação de dados até visualização, análise estatística e treinamento de modelos de aprendizado de máquina.

Este estudo visa contribuir para a avaliação do estado da arte de agentes e LLMs aplicados à ciência de dados e pode apoiar desenvolvedores e pesquisadores no desenvol-

vimento e na avaliação de agentes baseados em LLMs nesse domínio. Também é relevante para organizações que buscam reduzir custos e a dependência de APIs proprietárias e que precisam ampliar a privacidade no processamento de dados por meio da execução local de modelos.

A análise busca responder a três questões centrais: (i) como os prompts dos benchmarks se distribuem em termos de complexidade; (ii) quais áreas da ciência de dados são exploradas por esses benchmarks; e (iii) qual é o desempenho de modelos de pesos abertos nesses benchmarks. O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica e os trabalhos relacionados; a Seção 3 descreve a metodologia; a Seção 4 apresenta os resultados; e a Seção 5 traz as conclusões.

2. Fundamentação Teórica e Trabalhos Relacionados

2.1. LLMs em Arquiteturas de Agentes

Modelos de Linguagem de Grande Escala, ou LLMs, são modelos de aprendizado de máquina treinados em grandes volumes de texto para prever e gerar sequências de palavras [Brown et al. 2020]. Baseados em arquiteturas com mecanismos de atenção, os LLMs identificam relações entre termos em uma sequência textual e produzem respostas coerentes a partir de prompts fornecidos pelo usuário [Vaswani et al. 2017]. Essa capacidade também sustenta a geração de código, pois a sintaxe de linguagens como Python ou SQL pode ser representada como sequência textual estruturada, ampliando o uso dos LLMs para tarefas de programação [Chen et al. 2021].

Agentes são sistemas computacionais situados em um ambiente, capazes de perceber esse ambiente e agir sobre ele de forma autônoma para atingir objetivos definidos. Essa autonomia implica operar sem intervenção direta constante, tomando decisões a partir de suas percepções e objetivos. Além disso, agentes podem apresentar reatividade, ao responderem a mudanças no ambiente; proatividade, ao iniciarem ações orientadas a objetivos; e habilidade social, ao interagirem com outros agentes ou usuários [Wooldridge 2009].

A incorporação dos LLMs em arquiteturas de agentes ampliou seu papel de geradores de texto para sistemas capazes de coordenar etapas de raciocínio e ação [Wang et al. 2024]. O *framework* ReAct, por exemplo, combina raciocínio passo a passo com ações externas, permitindo que o agente planeje, decida o que fazer e execute etapas de forma iterativa [Yao et al. 2023]. Nesse contexto, o uso de ferramentas constitui um componente central, pois permite ao modelo invocar funções externas, como busca na web, execução de código em tempo real ou acesso a bancos de dados, para obter informações ou realizar cálculos que vão além do seu treinamento interno [Schick et al. 2023].

Essa combinação de geração de código, agentes e uso de ferramentas viabilizou a aplicação dos LLMs em tarefas de ciência de dados. Nesse contexto, esses modelos podem auxiliar na limpeza e transformação de conjuntos de dados, na análise exploratória, na construção de análises estatísticas, na geração de visualizações, na execução de código, na avaliação de métricas e na interpretação de resultados em fluxos analíticos [Hong et al. 2025].

2.2. Benchmarks para Avaliação de LLMs

Benchmarks de LLMs são conjuntos padronizados compostos por *datasets*, tarefas específicas, protocolos de avaliação e métricas de desempenho, permitindo a comparação sistemática e replicável entre modelos por meio de *scores* comparáveis [Raji et al. 2021]. Na avaliação de LLMs, esses instrumentos organizam o processo em torno de três dimensões principais: o que avaliar, isto é, as capacidades analisadas, como raciocínio, robustez, segurança ou uso como agentes; onde avaliar, envolvendo a escolha de *datasets* e tarefas; e como avaliar, referente aos protocolos e métricas empregados [Chang et al. 2024]. Embora essa estrutura favoreça avaliações mais controladas, benchmarks ainda apresentam limitações, como escopo restrito das tarefas, inadequação entre a capacidade avaliada e o desenho do benchmark, caráter estático das avaliações e possível contaminação dos dados de teste [Raji et al. 2021, Chang et al. 2024].

Com o uso crescente de LLMs em ciência de dados, tornou-se necessário avaliar o desempenho desses modelos nas diferentes etapas que compõem o domínio [Lai et al. 2023]. Tarefas de análise de dados envolvem a interpretação do problema, a seleção de bibliotecas, a escrita de código, a execução de procedimentos estatísticos, a produção de visualizações e a verificação dos resultados. Diante dessa complexidade, o paradigma de agentes foi incorporado aos LLMs com o objetivo de torná-los capazes de planejar ações, utilizar ferramentas, executar procedimentos e revisar resultados de forma iterativa [Hu et al. 2024]. Nesse contexto, surgiram benchmarks específicos para a área, voltados à avaliação dessas capacidades em cenários próximos aos fluxos reais de trabalho. Esses benchmarks permitem observar não apenas se o modelo produz uma resposta plausível, mas se executa etapas técnicas com consistência, correção e aderência ao objetivo da tarefa [Zhang et al. 2026].

Como trabalhos relacionados, o estudo de [Nikolakopoulos et al. 2025] avalia LLMs de pesos abertos executados localmente em tarefas de ciência de dados, utilizando Codestral 22B e Qwen2.5-Coder 14B em prompts com diferentes níveis de complexidade. Já [Hong et al. 2025] propõem e avaliam o Data Interpreter, um agente baseado em LLM para ciência de dados, testado no InfiAgent-DABench e com diferentes modelos base, incluindo Qwen-72B-Chat e Mixtral-8x7B.

2.3. Análise Exploratória de Prompts

Técnicas de análise de dados não estruturados podem ser aplicadas aos prompts de benchmarks de LLMs para examinar padrões textuais e temáticos. Nesse caso, cada prompt é tratado como um documento textual e convertido em uma representação vetorial, o que viabiliza comparações por similaridade conforme o modelo de espaço vetorial [Salton et al. 1975]. O TF-IDF (do inglês *Term Frequency-Inverse Document Frequency*) pondera os termos pela frequência no documento e pela raridade no conjunto analisado [Spärck Jones 1972]. Como essas representações podem ter alta dimensionalidade, a Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) pode ser usada para reduzir o espaço de representação e apoiar a visualização dos dados [Jolliffe 2002]. De forma relacionada, [Zhang et al. 2025] analisaram *datasets* de prompts de LLMs com técnicas de exploração quantitativa, tratando os prompts como objetos textuais passíveis de comparação e análise.

Além de técnicas como TF-IDF e PCA, métodos recentes de modelagem de

tópicos permitem explorar relações semânticas mais complexas entre documentos. O BERTopic combina *embeddings* de linguagem, redução de dimensionalidade, agrupamento e extração de termos representativos para identificar tópicos em coleções textuais [Grootendorst 2022]. Como trabalho relacionado, [Chiang et al. 2024] aplicam BERTopic a prompts de usuários do Chatbot Arena para analisar sua diversidade temática e identificar grupos de tópicos.

Até onde foi possível verificar na literatura consultada, não foram identificados trabalhos que analisem conjuntamente a complexidade dos prompts, a cobertura temática e o desempenho de modelos de pesos abertos em benchmarks de LLMs aplicados à ciência de dados.

3. Metodologia

3.1. Processamento de Dados

O conjunto de dados utilizado neste trabalho é composto por prompts de três benchmarks de ciência de dados: DS-1000, InfiAgent-DABench e DataSciBench. Os prompts são disponibilizados pelos autores em repositórios públicos, como GitHub e Hugging Face. Esses benchmarks foram selecionados por representarem uma evolução temporal recente da avaliação de LLMs em ciência de dados, abrangendo o período de 2023 a 2026, desde o DS-1000, um dos primeiros benchmarks mais notáveis da área, até o DataSciBench, benchmark mais recente entre os analisados. Além disso, os três conjuntos contemplam tarefas com *ground truth*, o que permite avaliação objetiva das respostas dos modelos. Os três benchmarks foram publicados em conferências com revisão por pares. Todos os prompts estão em inglês e foram padronizados antes das análises, com normalização textual e organização em uma estrutura comum, permitindo aplicar os mesmos procedimentos a cada benchmark. As características gerais dos conjuntos analisados são apresentadas na Tabela 1.

Tabela 1. Características dos benchmarks de Ciência de Dados analisados

Benchmark	Ano	Quantidade	Publicação	*Ground Truth*
DS-1000	2023	1000 prompts	ICML 2023	Sim
InfiAgent-DABench	2024	257 prompts	ICML 2024	Sim
DataSciBench	2026	222 prompts	ACL 2026	Sim

3.2. Análise da Dificuldade dos Prompts

Para caracterizar a dificuldade dos prompts, adotou-se um procedimento não supervisionado implementado em Python, baseado em sete características interpretáveis extraídas do texto de cada prompt, todas orientadas de modo que valores maiores indicassem maior dificuldade: (i) número de palavras do texto natural; (ii) densidade de restrições, contada por marcadores como *must*, *should*, *exactly* e *at least*, somada aos itens enumerados; (iii) número de termos técnicos pertencentes a um léxico fixo de ciência de dados; (iv) marcadores de sequenciamento procedural; (v) especificidade de entrada e saída; (vi) complexidade do código de referência, definida como $f_6 = L + 0{,}5C + 2I + 1{,}5D$, em que L é o número de linhas, C o número de chamadas, I o número de importações e D a profundidade máxima de indentação; e (vii) densidade linguística do texto natural, medida pelo índice de Flesch-Kincaid. A separação entre texto natural e código foi realizada por

expressões regulares, evitando sobreposição entre complexidade textual e complexidade estrutural.

Os pesos de f_6 foram definidos pelos autores, considerando que linhas representam a extensão do código; chamadas de função recebem menor peso por poderem ocorrer múltiplas vezes em uma mesma linha; importações recebem maior peso por indicarem o uso de bibliotecas externas, cada uma agregando novas dependências técnicas ao problema; e a profundidade máxima de indentação recebe peso intermediário por refletir maior complexidade de fluxo de controle.

Como os atributos apresentam escalas distintas, foi aplicada uma padronização global sobre o conjunto agregado dos três benchmarks, utilizando `StandardScaler`. Essa estratégia permitiu comparar DS-1000, InfiAgent-DABench e DataSciBench em uma mesma escala, evitando que atributos com maior amplitude numérica dominassem o escore final. O escore composto de dificuldade textual foi calculado pela média dos atributos padronizados por z-score, e os rótulos foram definidos por tercis globais da distribuição:

$$D_i = \frac{1}{m}\sum_{j=1}^{m} z_{ij}, \quad Classe_i = \begin{cases} \text{Fácil}, & D_i \leq Q_{33} \\ \text{Médio}, & Q_{33} < D_i \leq Q_{66} \\ \text{Difícil}, & D_i > Q_{66} \end{cases} \tag{1}$$

Em que D_i representa o escore do prompt i, z_{ij} o valor padronizado do atributo j, m o número de atributos considerados, e Q_{33} e Q_{66} os tercis da distribuição agregada. Como os cortes foram calculados globalmente, e não dentro de cada benchmark, as proporções de prompts fáceis, médios e difíceis tornam-se comparáveis entre os conjuntos avaliados. Por fim, foi aplicada a Análise de Componentes Principais (PCA) sobre os atributos padronizados. A técnica foi usada para projeção bidimensional e inspeção visual da distribuição das classes.

3.3. Modelagem de Tópicos

Para mapear a estrutura temática dos prompts, os três benchmarks foram unificados em um único corpus de 1.479 documentos, no qual cada prompt foi representado pela concatenação de seus campos textuais: enunciado, código de referência, restrições, conceitos e metadados. Cada documento foi convertido em uma representação vetorial densa de 1.024 dimensões por meio do modelo de *embeddings* de sentenças `mxbai-embed-large-v1`[1], escolhido por ser otimizado para compreensão de prompts, recuperação semântica e busca contextual, produzindo representações sensíveis a nuances técnicas. A modelagem de tópicos foi então conduzida com o BERTopic, cujo *pipeline* reduz a dimensionalidade dos *embeddings*, utilizando cinco componentes e métrica de cosseno, e emprega o HDBSCAN como método padrão de clusterização. O HDBSCAN (do inglês *Hierarchical Density-Based Spatial Clustering of Applications with Noise*) determina automaticamente o número de grupos e isola documentos atípicos como ruído [Campello et al. 2013]. Posteriormente, o BERTopic representa cada tópico por seus termos mais relevantes por meio do c-TF-IDF, uma variação do TF-IDF baseada em classes.

[1] `https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1`

3.4. Execução dos Benchmarks em Modelos de Pesos Abertos

A etapa de benchmark em modelos de pesos abertos foi conduzida em ambiente local, utilizando o LM Studio[2] com modelos quantizados em Q4_K_M. Foram selecionados quatro modelos na faixa aproximada de 24B a 30B parâmetros: Gemma 4 26B-A4B, Qwen3.6-27B, GLM-4.7 Flash e Devstral Small 2 24B. A escolha considerou a viabilidade de execução em uma GPU RTX 3090 com 24 GB de VRAM e o foco recente desses LLMs em tarefas de programação, uso de ferramentas e capacidades de codificação agêntica.

Além dos quatro modelos locais, foram utilizados dois modelos proprietários via API como referência comparativa. O Claude Opus 4.7 foi adotado como referência de modelo *flagship*, enquanto o GPT-5.4 mini foi incluído por representar uma alternativa mais compacta e comparável, em escala de uso, aos modelos locais na faixa de 30B parâmetros. Dessa forma, a avaliação combina modelos de pesos abertos executados localmente com referências proprietárias, permitindo observar a diferença de desempenho entre soluções locais e modelos comerciais em tarefas de ciência de dados.

Tabela 2. Modelos utilizados na etapa de benchmark.

Modelo	Lançamento	Tipo	Parâmetros	Execução
Gemma 4 26B-A4B	Abr. 2026	Open weights	26B-A4B	Local
Qwen3.6-27B	Abr. 2026	Open weights	27B	Local
GLM-4.7 Flash	Jan. 2026	Open weights	30B-A3B	Local
Devstral Small 2	Dez. 2025	Open weights	24B	Local
Claude Opus 4.7	Abr. 2026	Proprietário	–	API
GPT-5.4 mini	Mar. 2026	Proprietário	–	API

4. Resultados

4.1. Como os prompts dos benchmarks se distribuem em termos de complexidade?

A distribuição de complexidade foi analisada por meio de duas visualizações: uma projeção em PCA e um gráfico de barras por benchmark. O PCA permite observar a organização geral dos prompts no espaço reduzido (Figura 1), em que as componentes 1 e 2 representam combinações das características textuais e estruturais extraídas dos prompts. Já o gráfico de barras sintetiza a proporção de cada classe de complexidade em cada benchmark (Figura 2).

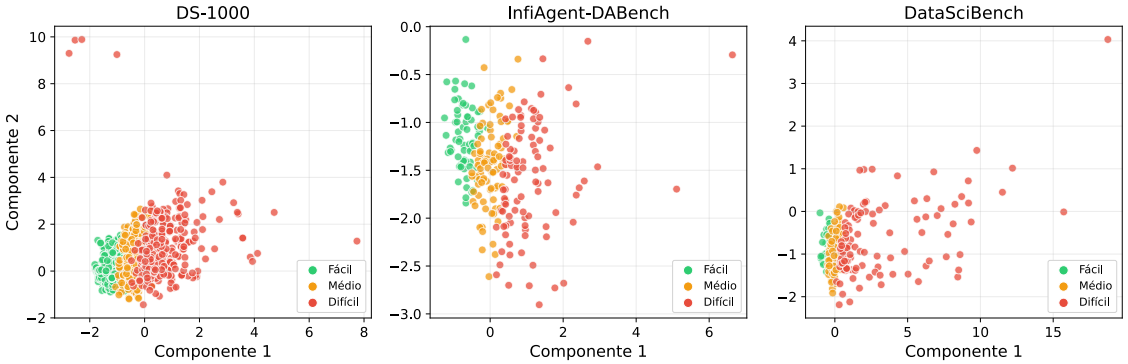

Figura 1. Projeção PCA dos prompts por complexidade

[2] https://lmstudio.ai/docs/app

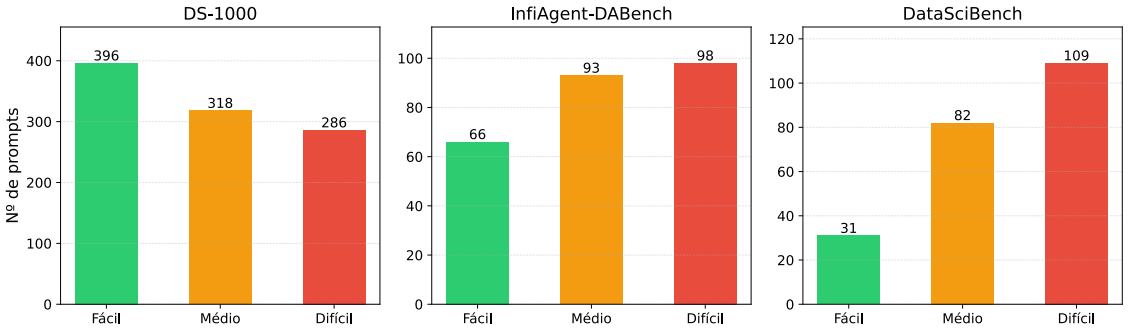

Figura 2. Distribuição dos níveis de complexidade dos prompts por benchmark

O **DS-1000** foi um dos primeiros benchmarks lançados para avaliar LLMs em tarefas de ciência de dados. O conjunto tem 1000 prompts voltados a operações de manipulação de dados, estatística básica e geração de código estruturado. Publicado em 2023, foi construído em um período no qual os modelos de linguagem ainda tinham limitações em raciocínio complexo, integração de bibliotecas e resolução de tarefas com várias etapas. Isso ajuda a explicar a presença de problemas mais curtos, bem definidos e com menor variação estrutural. Na classificação proposta, o DS-1000 concentrou 396 prompts fáceis, o que corresponde a 39,6% do conjunto, e 286 prompts difíceis, ou 28,6%. Esses valores indicam maior presença de prompts fáceis e médios, com menor ocorrência de tarefas de modelagem, otimização ou análise multietapa.

O **InfiAgent-DABench**, publicado em 2024, tem 257 prompts associados a 52 arquivos CSV e se apresenta como o primeiro benchmark projetado especificamente para avaliar *LLM-based agents* em tarefas de análise de dados. Diferente do DS-1000, o InfiAgent-DABench não avalia apenas a geração de código em um contexto delimitado. O modelo precisa interpretar a pergunta, planejar etapas, produzir código funcional, interagir com um *sandbox* Python e, quando necessário, corrigir falhas durante a execução por meio de *self-debugging*. Na classificação proposta, o benchmark apresentou 66 prompts fáceis, 93 médios e 98 difíceis, correspondendo a 25,7%, 36,2% e 38,1% do conjunto. A distribuição indica maior concentração nas classes média e difícil, o que é compatível com a avaliação de agentes baseados em LLMs em tarefas completas de análise de dados.

O **DataSciBench**, publicado em 2026, apresenta o perfil mais complexo entre os três benchmarks. Na classificação proposta, dos 222 prompts analisados, 109 foram classificados como difíceis, o que representa 49,1% do conjunto. A classe fácil corresponde a apenas 14,0%. Esse padrão é coerente com a forma como o artigo se apresenta, como um *LLM Agent Benchmark for Data Science*. Diferente do DS-1000, o DataSciBench busca avaliar capacidades mais amplas de LLMs em ciência de dados, com subtarefas, funções de avaliação e regras programáticas. Em relação ao InfiAgent-DABench, o DataSciBench cobre um conjunto mais amplo de tarefas, como mineração de padrões, interpretabilidade e geração de relatórios. Sua maior concentração de prompts difíceis está relacionada à presença de tarefas compostas e à necessidade de integrar planejamento, geração de código, execução e avaliação de resultados.

4.2. Quais áreas da ciência de dados são exploradas pelos benchmarks?

A modelagem de tópicos com BERTopic identificou 17 tópicos a partir dos 1.479 prompts dos três benchmarks analisados (Figura 3). Para facilitar a interpretação dos resultados,

os tópicos foram agrupados em quatro áreas centrais da ciência de dados. A redução foi realizada de forma independente pelos dois autores, com base nos termos de maior peso e na proximidade temática entre os tópicos, e as divergências foram discutidas até a obtenção de consenso. As áreas resultantes foram: Preparação e Manipulação de Dados, Análise Estatística, Machine Learning e Deep Learning e Visualização de Dados.

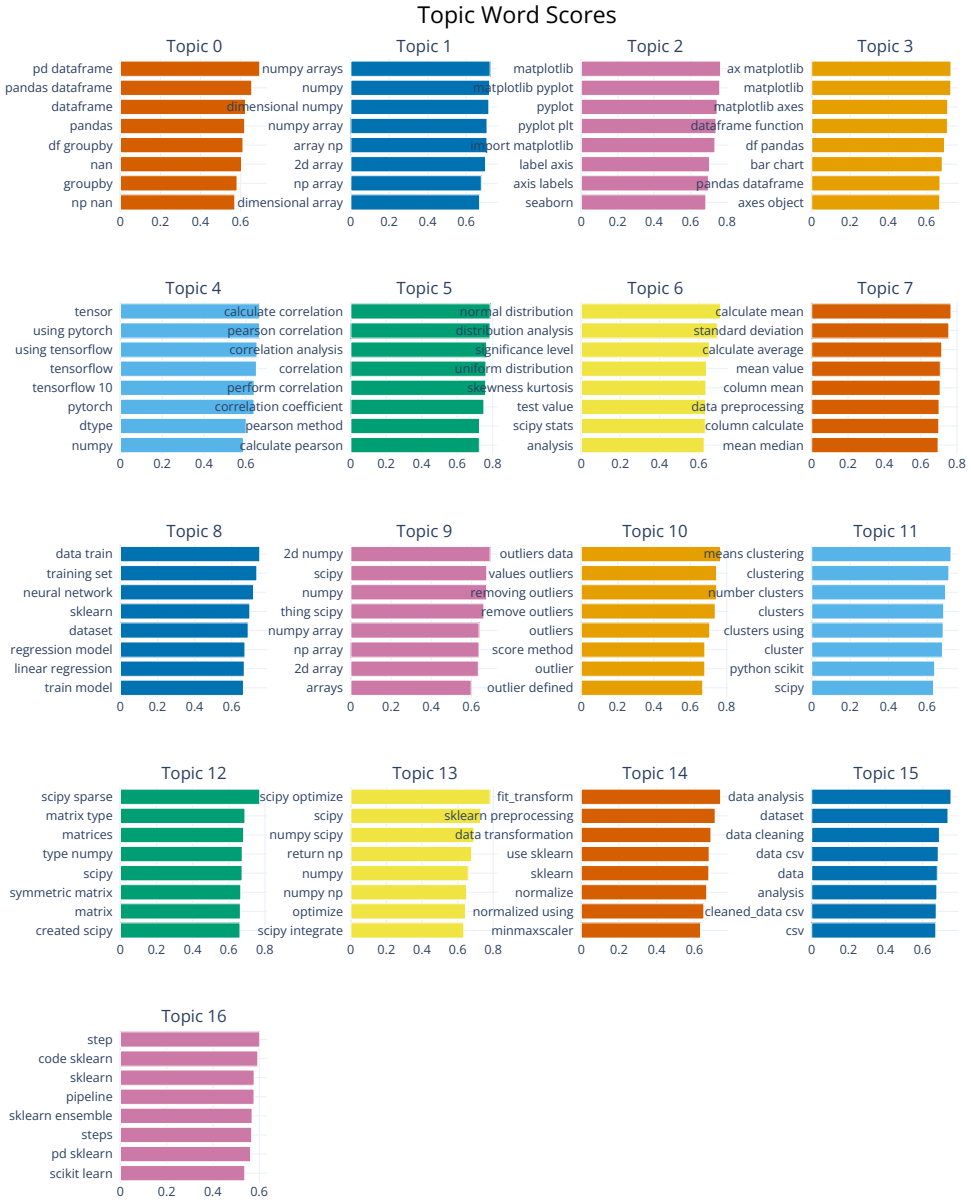

Figura 3. Termos mais representativos por tópico gerados pelo BERTopic

A área de **Preparação e Manipulação de Dados** reúne os Tópicos 0, 1, 9, 12 e 15, cujos termos dominantes, como `pd dataframe`, `pandas`, `numpy arrays`, `2d numpy`, `scipy sparse`, `matrix type`, `data analysis`, `dataset`, `data cleaning` e `data csv`, apontam para tarefas de leitura, transformação, estruturação, limpeza e organização de dados. Essa área representa a etapa inicial do fluxo analítico, na qual os dados são convertidos para formatos adequados ao processamento e à análise posterior.

A área de **Visualização de Dados** é composta pelos Tópicos 2 e 3, marcados por termos como `matplotlib`, `pyplot`, `plt`, `ax matplotlib`, `matplotlib axes` e `dataframe function`. Esses tópicos concentram prompts relacionados à construção e ao ajuste de gráficos, ao uso de bibliotecas de visualização e à representação visual de informações extraídas dos dados. A presença dessa área indica que os benchmarks também avaliam a capacidade dos modelos de comunicar resultados por meio de visualizações, não apenas de produzir cálculos ou código funcional.

A área de **Análise Estatística** reúne os Tópicos 5, 6, 7, 10 e 13, associados a termos como `calculate correlation`, `pearson correlation`, `normal distribution`, `significance level`, `calculate mean`, `standard deviation`, `outliers`, `removing outliers` e `scipy optimize`. Esses tópicos abrangem tarefas de cálculo estatístico, análise de distribuições, medidas de tendência central e dispersão, correlação, identificação de valores discrepantes e procedimentos de otimização. Trata-se de uma área voltada à interpretação quantitativa dos dados e à aplicação de métodos estatísticos em problemas analíticos.

A área de **Machine Learning e Deep Learning** compreende os Tópicos 4, 8, 11, 14 e 16, caracterizados por termos como `tensor`, `pytorch`, `tensorflow`, `data train`, `training set`, `neural network`, `sklearn`, `means clustering`, `number clusters`, `fit_transform`, `sklearn preprocessing` e `sklearn pipeline`. Esses tópicos indicam tarefas relacionadas ao treinamento de modelos, redes neurais, clusterização, pré-processamento para aprendizado de máquina e construção de *pipelines*. Essa área representa a parte dos benchmarks mais associada à modelagem preditiva e ao uso de bibliotecas especializadas para aprendizado de máquina.

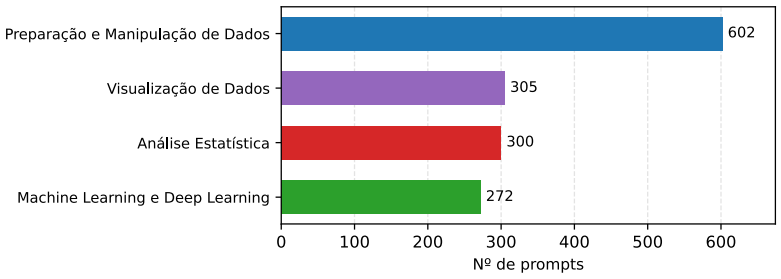

Figura 4. Distribuição dos prompts por área de ciência de dados nos benchmarks

Conforme ilustrado na Figura 4, observa-se que a área de Preparação e Manipulação de Dados apresenta o maior número de prompts. Esse predomínio decorre, sobretudo, da composição do benchmark DS-1000, que concentra grande parte de suas tarefas em operações com *DataFrames*, manipulação de colunas, limpeza e transformação de dados. Tal ênfase está relacionada ao contexto de sua criação em 2023, período no qual os modelos de linguagem ainda apresentavam limitações significativas para lidar com problemas complexos de ciência de dados, conduzindo o benchmark a priorizar tarefas mais simples e estruturadas. Como o DS-1000 é também o benchmark com maior volume de prompts no conjunto analisado, sua predominância temática acaba influenciando a distribuição global dos tópicos gerados pelo BERTopic.

4.3. Qual é o desempenho de modelos de pesos abertos nesses benchmarks?

Os benchmarks foram executados conforme os procedimentos indicados pelos autores em seus repositórios e materiais de apoio. Para os modelos de pesos abertos, as execuções ocorreram em ambiente local, com ajustes apenas operacionais para integração com os *endpoints*. Dessa forma, os resultados seguem os protocolos originais de cada benchmark e preservam a comparabilidade entre os modelos avaliados.

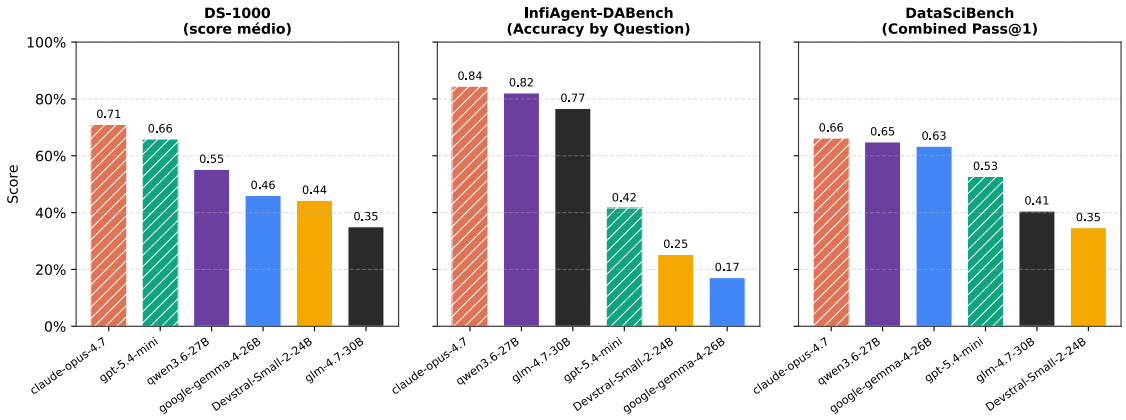

Figura 5. Desempenho dos LLMs nos três benchmarks

No **DS-1000**, o Claude Opus 4.7 teve o melhor desempenho, com 0,710, seguido pelo GPT-5.4-mini, com 0,659. Entre os modelos de pesos abertos, o Qwen3.6-27B obteve o maior resultado, com 0,552, seguido por Gemma-4-26B, Devstral-Small-2-24B e GLM-4.7-30B. Como o DS-1000 avalia principalmente geração de código em contextos delimitados, os modelos proprietários mantiveram vantagem, e o melhor modelo aberto ficou cerca de 22% abaixo do melhor modelo proprietário, como mostra a Figura 5.

No **InfiAgent-DABench**, o Claude Opus 4.7 alcançou 84,4% de *Accuracy by Question*. Entre os modelos de pesos abertos, Qwen3.6-27B teve 82,1% e GLM-4.7-30B teve 76,6%, enquanto GPT-5.4-mini, Devstral-Small-2-24B e Gemma-4-26B ficaram abaixo. Esse benchmark foi mais sensível ao formato agêntico, pois usa um fluxo ReAct com chamada à *sandbox* Python. Como a métrica exige acerto completo da questão, falhas na formatação da ação, no uso da ferramenta, na execução do código ou na resposta final reduzem a pontuação.

No **DataSciBench**, o Claude Opus 4.7 também obteve o melhor resultado, com *Combined Pass@1* de 0,662, mas a diferença para Qwen3.6-27B e Gemma-4-26B foi pequena, com 0,649 e 0,633. O GPT-5.4-mini ficou em posição intermediária, com 0,527, enquanto GLM-4.7-30B e Devstral-Small-2-24B tiveram os menores resultados. Esse resultado indica que, nesse benchmark, modelos de pesos abertos podem se aproximar de modelos proprietários quando lidam bem com instruções estruturadas, geração de código, execução e correção iterativa no fluxo do MetaGPT[3].

5. Conclusão

Os primeiros benchmarks, como o DS-1000, avaliam sobretudo capacidades reativas. Já benchmarks mais recentes, como o InfiAgent-DABench e o DataSciBench, exigem com-

[3] https://github.com/foundationagents/metagpt

portamento proativo, direcionado a objetivos, por meio de algoritmos de planejamento em múltiplas etapas conduzidos por iniciativa do próprio modelo, além de reatividade diante de mudanças de estado e mensagens de erro no ambiente de execução *sandbox*.

A análise temática mostra que, nos benchmarks mais recentes, aumenta a cobertura de áreas da ciência de dados, acompanhando a evolução dos próprios LLMs. Os conjuntos analisados já cobrem boa parte do ciclo analítico, com preparação e manipulação de dados, análise estatística, visualização, machine learning, deep learning, mineração de padrões, interpretabilidade e geração de relatórios. A maior concentração ainda ocorre em preparação e manipulação de dados, efeito associado ao volume do DS-1000, que é o maior benchmark do conjunto analisado.

Nos experimentos com modelos, os melhores resultados ocorreram nos modelos que lidaram melhor com uso de ferramentas, execução de código, formato ReAct e ciclos de correção. Qwen3.6-27B e GLM-4.7-30B tiveram desempenho competitivo em tarefas multietapas, especialmente quando a solução dependia de chamadas à *sandbox*, leitura de saídas intermediárias e ajustes no código gerado. Modelos com maior instabilidade na formatação da ação ou no uso da ferramenta tiveram queda de desempenho. Esse efeito foi mais evidente no InfiAgent-DABench, pois o benchmark depende de uma estrutura rígida de interação com a *sandbox*. Erros no formato da ação, na chamada da ferramenta ou na resposta final podem interromper a execução e comprometer a pontuação da tarefa.

Em termos práticos, os resultados indicam que LLMs de pesos abertos de médio porte podem ser utilizados como agentes locais em tarefas de ciência de dados, contribuindo para reduzir os custos e a dependência de APIs proprietárias. A execução local também amplia a privacidade e o controle das organizações sobre seus dados e sua infraestrutura. Além disso, o estudo apoia a escolha de modelos para a automação de fluxos de ciência de dados e pode orientar o desenvolvimento de futuros benchmarks.

Como trabalhos futuros, pretende-se aprofundar a análise do impacto de diferentes níveis de quantização sobre o desempenho dos modelos. Também se propõe avaliar modelos com menor número de parâmetros, a fim de investigar até que ponto LLMs mais compactos preservam desempenho competitivo em tarefas de ciência de dados. Outra direção consiste em comparar arquiteturas distintas, especialmente modelos densos e do tipo *Mixture of Experts* (MoE), além de ampliar os testes com benchmarks mais recentes.

Referências

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172.

Chang, Y., Wang, X., Wang, J., et al. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Chen, M., Tworek, J., Jun, H., et al. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chiang, W.-L., Zheng, L., Sheng, Y., et al. (2024). Chatbot arena: An open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24.

Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Hong, S., Lin, Y., Liu, B., et al. (2025). Data interpreter: An LLM agent for data science. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19796–19821.

Hu, X., Zhao, Z., Wei, S., et al. (2024). Infiagent-dabench: Evaluating agents on data analysis tasks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 19544–19572.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, 2 edition.

Lai, Y., Li, C., Wang, Y., et al. (2023). Ds-1000: A natural and reliable benchmark for data science code generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

Nikolakopoulos, A., Litke, A., Psychas, A., et al. (2025). Exploring the potential of offline LLMs in data science: A study on code generation for data analysis. *IEEE Access*, 13:64087–64114.

Raji, I. D., Denton, E., Bender, E. M., Hanna, A., and Paullada, A. (2021). AI and the everything in the whole wide world benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, NIPS '21.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schick, T., Dwivedi-Yu, J., Dessì, R., et al. (2023). Toolformer: Language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Wang, L., Ma, C., Feng, X., et al. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2 edition.

Yao, S., Zhao, J., Yu, D., et al. (2023). React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Zhang, D., Zhoubian, S., Cai, M., et al. (2026). DataSciBench: An LLM agent benchmark for data science. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 3685–3728.

Zhang, Y., Lin, Y., Khan, A., and Wan, H. (2025). Large language model prompt datasets: An in-depth analysis and insights. *arXiv preprint arXiv:2510.09316*.