Measuring Data Science Automation: A Survey of Evaluation Tools for AI Assistants and Agents

Anonymous authors
Paper under double-blind review

Abstract

Data science aims to extract insights from data to support decision-making processes. Recently, Large Language Models (LLMs) are increasingly used as *assistants* for data science, by suggesting ideas, techniques and small code snippets, or for the interpretation of results and reporting. Proper automation of some data-science activities is now promised by the rise of LLM *agents*, i.e., AI systems powered by an LLM equipped with additional affordances—such as code execution and knowledge bases—that can perform self-directed actions and interact with digital environments. In this paper, we survey the evaluation of LLM assistants and agents for data science. We find (1) a dominant focus on a small subset of goal-oriented activities, largely ignoring data management and exploratory activities; (2) a concentration on pure assistance or fully autonomous agents, without considering intermediate levels of human-AI collaboration; and (3) an emphasis on human substitution, therefore neglecting the possibility of higher levels of automation thanks to task transformation.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020) and their multimodal extensions first caught public prominence by powering capable chatbots that are now widely used as *assistants* to humans in several tasks, such as summarising documents (Liu et al., 2023c), performing translations (Zhu et al., 2023), and creating code snippets (Guo et al., 2023). These LLM assistants take instructions from a human in the form of a prompt and return an answer, with the human retaining control over planning and decision-making by determining the sequence of actions to follow and how much to rely on the assistant's output. Attention is now increasingly dedicated to "LLM *agents*" (Wang et al., 2024) that can autonomously and iteratively decide a sequence of actions to take and repeatedly interact with an external (digital) environment, being equipped with affordances and tools such as code execution (Huang et al., 2024b), internet access (Zhou et al., 2024), knowledge bases (Chen et al., 2024), and operating system control (Liu et al., 2023a).

In this paper we focus on how to *evaluate* assistants and agents for data science applications. Data science is the process of handling and analysing data to extract insights that support decision-making in science, business, or other contexts. Data science may deal with different modalities of data (tabular, images, audio, etc.) but it always involves writing and interpreting information represented as text, whether in the form of code or natural language, and processing image, such as to present information or models. This dominance of textual modalities, combined with the vast amount of relevant online material that LLMs leverage during both training and inference, makes data science well-suited for automation by LLMs. Indeed, since the early days of LLMs (Chandel et al., 2022), LLM assistants and agents have been used in data science applications. In this work, we survey evaluation tools measuring performance on tasks across the data science pipeline, offering, to our knowledge, the first comprehensive review of LLM evaluation for core data science tasks[1]. We do *not* overview the state of the art in developing LLM assistants or agents for data science, referring interested readers to other works (Sun et al., 2024).

[1]While many studies assess LLMs on tasks related to data science, such as coding (Jimenez et al., 2024) and planning (Valmeekam et al., 2023), our focus is on those that explicitly target data science.

Data science is highly multidisciplinary and involves a breadth of activities, combining fields such as statistics, machine learning, and data engineering with tasks such as understanding business needs, writing reports, and preliminary research. Therefore, to compare the data science automation evaluation tools in our survey, we adopt the widely used data science task taxonomy of Martínez-Plumed et al. (2019) and classify each evaluation tool by the activities it requires subjects to perform and explicitly evaluates. Moreover, in our survey we specifically consider the level of autonomy each evaluation targets—whether the subjects are agents, assistants, or intermediate forms, such as LLM agents operating under close human supervision that correct their actions as needed. Further, we also analyse the way in which the tasks are framed and evaluated, to understand if they measure the ability of AI systems to simply *substitute* humans or if instead they consider that the AI system can *transform* the task in deeper ways, such as by bringing functional improvements (for instance, LLMs may not need to produce visualisations to perform data exploration; see Sec. 2.1). By focusing on activities and autonomy, we make the following findings:

- Most evaluation works individually target a (small) subset of data science activities (Tables 2 and 3); a few works cover multiple activities (Secs. 4.2 and 4.3), but none cover all of them. Taken together, the surveyed works cover the landscape of data-science activities in a biased fashion, chiefly over-representing "goal-oriented" activities, such as data preprocessing, producing plots in specified formats, or building predictive models for predetermined targets. Only a few studies (Cheng et al., 2023; Sahu et al., 2025; Majumder et al., 2024) give prominence to open-ended, exploratory aspects of data science (such as interpreting client needs within a business context, creatively exploring datasets and proposing potential uses) or data management(Yu et al., 2018; 2019b;a; Lei et al., 2024).
- Most studies focus either on assistants following human-defined actions or fully autonomous agents, overlooking more realistic scenarios of intermediate LLM–human collaboration. Exceptions include Li et al. (2024; 2025b;a) (agents with simulated human users) and Yu et al. (2019a) (assistants aiding users in clarifying data tasks). Due to this near-binary focus, we organise the surveyed works according to the focus on assistants (Sec. 3) and agents (Sec. 4), highlighting when they assess intermediate autonomy.
- Evaluations often implicitly assume that AI will substitute humans without functionally changing the tasks, either in assuming the steps by which a task is solved are the same a human would follow (Yu et al., 2019a; Zhang et al., 2024b; Chen et al., 2024) or by scoring the task referring to human-produced output, despite there not being a single ground truth (Song et al., 2025; Huang et al., 2024b; Hu et al., 2024; Jing et al., 2024; Chen et al., 2024). Other works instead reward agents that transform the task to solve it by scoring the final output (Pietruszka et al., 2024; Cheng et al., 2023; Li et al., 2025b; Chan et al., 2024; Lei et al., 2024; Majumder et al., 2024; Sahu et al., 2025).

The paper is structured as follows: Sec. 2 discusses some fundamental concepts on technological transformation, assistance and autonomy, LLM evaluation, and data science automation and its activities. Secs. 3 and 4 dive deep into works evaluating LLM assistants and agents, respectively. Sec. 5 summarises the challenges in data science evaluation and suggests future directions for more effective and comprehensive evaluation.

2 Background and related work

2.1 Levels of technological transformation

A naive perception of how technology transforms processes and activities is automation by *substitution*: a human performing a task is replaced by a machine without functionally transforming the task. To evaluate this, a sample of tasks representing how humans tackle an activity is collected and the machine is tested on them (Eriksson et al., 2025). The Substitution-Augmentation-Modification-Redefinition (SAMR) model (Puentedura, 2006; Hamilton et al., 2016) identifies substitution as the lowest level of transformation and outlines subsequent levels with progressively greater degrees of transformation: augmentation, where the machine substitutes the human with some functional improvement; modification, where the task is significantly redesigned to allow automation; redefinition, in which the whole activity is redesigned, even creating new

tasks. Much of the debate around AI-powered automation focuses on the two bottom levels (augmentation and substitution), but the real penetration of AI technology is happening at the top levels of modification and redefinition (Brynjolfsson, 2022; Brynjolfsson et al., 2025), which hold the potential to achieve higher levels of automation. Indeed, we did not "automate away the jobs of lamplighters by building robots capable of carrying ladders and climbing lampposts" (Frey & Osborne, 2023). Importantly, activities that have already been substituted can be iteratively transformed further as technology improves. Evaluating progress is therefore much more complex than if substitution was the only force at play: a robot substituting human lamplighters in carrying and climbing ladders would have scored highly in turning on gas lamps, but the redefinition afforded by electricity led to automating street lights, rendering robotic lamplighter needless. Similar considerations apply when evaluating AI progress in automating complex processes composed of many activities, such as data science: for instance, LLMs may not need to produce high quality visualisations to perform successful data exploration, as they may be able to directly interpret large tables of data. To effectively evaluate modification and redefinition, AI evaluation should allow AI systems to perform activities differently from humans by rewarding the achievement of broad objectives.

2.2 Assistance and autonomy

Related yet orthogonal to the SAMR model is the distinction between assistance and autonomy (Shneiderman, 2020): in an assistive situation, a human uses the technology while retaining control of the process and only having some well-defined parts automated or improved. Assisted driving or writing are good examples: the process becomes more efficient and safe because of the use of technology. On the contrary, in an autonomous situation, the technology performs the task independently and has more freedom to choose how. Of course, autonomy exists on a spectrum: intermediate levels include, for example, technology operating independently while a human oversees the sequence of steps and retains the ability to halt operations. In relation to this, Cihon et al. (2024) defined levels of agent features relevant to autonomy. Their classification assigns high autonomy to agents acting fully autonomously, whereas the intermediate and lower levels correspond to agents consulting humans either at termination or at each step. This aligns with our understanding of autonomy levels, which additionally includes an even lower level where a human assigns a specific task to an assistant. Hence, holistic AI evaluation should take into account quality of the result and the level of human labour, which limits the impact of the technology in the long term. Importantly, for all levels of autonomy, the technology can perform the task in a way that places the automation at any level of the SAMR hierarchy.

2.3 LLM evaluation

The area of AI evaluation (Burden et al., 2025) mostly relies on tasks encapsulated in input-output benchmarks with a reference output for each example. For LLMs, these input–output pairs are most often Q&A examples used to evaluate assistants (Chang et al., 2023; Guo et al., 2023) or autonomously acting agents (Wang et al., 2024; Yehudai et al., 2025). The use of natural language gives a perspective of breadth, but fails to measure realistic human-LLM interaction (Guo et al., 2023) and hence real-world impact (Burden et al., 2025). This agrees with our findings that evaluations for data science mostly fail to capture intermediate levels of LLM-human collaboration and concentrate on evaluating substitution rather than higher levels of transformation (Sec. 2.1). Chang et al. (2023) highlighted human-in-the-loop testing and evaluations in an open environment as future directions, and Wang et al. (2024) identified a shift towards end-to-end tasks requiring human evaluators and versatile metrics, yet most evaluations today only consider a subset of tasks in the data science pipeline, with a few exceptions (Sec. 4.3).

A few studies evaluate truly long-horizon scenarios or quantify the human effort they still require. Wang et al. (2023) and Park et al. (2023) showed that agents can sustain hours-to-days of open-ended play or social simulation, but both exposed failure modes that need periodic human nudges. Quantitatively, Liu et al. (2023a) found that commercial models needed a median of 2.4 human corrections per task on a general agent benchmark, whereas open-source models needed 5–8. Recently, Kwa et al. (2025) showed that autonomous agents are progressively conquering tasks that take humans longer to complete when considering a fixed success rate (e.g., 50%), but performance still progressively degrades on tasks requiring more than 10 seconds.

2.4 Data science automation

Automating data science was a topic of research even before LLMs became commonplace. Bie et al. (2021) argued that the technical and domain knowledge required to solve data science tasks motivated efforts toward automation. The authors categorised data science tasks into four main quadrants, defined by two axes—degree of open-endedness and dependence on domain context—highlighting that model-building activities are more easily automated (e.g., through AutoML approaches, Hutter et al. 2019; Gijsbers et al. 2024) due to their lower open-endedness and context dependence. They also identified three forms of automation: mechanisation, composition, and assistance. Assistance corresponds to our interpretation of the term while mechanisation and composition can be grouped under our umbrella of automation (Sec. 2.2), but differ in focusing respectively on small parts of the process or on the overall pipeline; in our work, we do not make this distinction and instead rely on the activities of Martínez-Plumed et al. (2019) to identify how many elements of the pipeline each evaluation work covers. After Bie et al. (2021) published their survey, numerous works built LLM agents to automate data science. Their evolution, capabilities, and applications across the data science pipeline are reviewed by Sun et al. (2024); the authors, however, did not address LLM *evaluation* for data science, which is the focus of our work. More recently, (Hu et al., 2025) proposed a taxonomy for the data ecosystem: Data Management, which includes data collection, data storage, and data preprocessing; Data Analysis, which includes model evaluation, data interpretation, and decision making; and Data Visualisation. This taxonomy partly overlaps with that in Martínez-Plumed et al. (2019) which we use in our work (Sec. 2.5), but misses some of the most exploratory aspects.

2.5 The activities of data science

Table 1: Data-science activities and brief definition (complete definitions in Appendix A).

Activity (abbr.)	**Brief definition**
Goal-oriented (CRISP-DM)	
Business Understanding (BU)	Define the problem and draft a plan that meets business requirements
Data Understanding (DU)	Collect and explore data to spot useful subsets, insights, or issues
Data Preparation (DP)	Build the final analysis dataset via selection, cleaning, and transformation
Modelling (M)	Apply modelling techniques, tune their parameters and evaluate models
Evaluation (E)	Check that the business objectives are met, with no overlooked issues
Deployment (Dep)	Deliver the model's outputs in a usable form (report, integration, etc.)
Exploratory	
Goal Exploration (GE)	Identify business goals that could be addressed with data
Data Source Exploration (DSE)	Discover new, valuable data sources
Data Value Exploration (DVE)	Judge the potential value that can be extracted from the data
Result Exploration (RE)	Connect data-science results back to business goals
Narrative Exploration (NE)	Craft meaningful (visual or textual) stories from the data
Product Exploration (PE)	Devise services or applications that turn extracted value into products
Data-management	
Data Acquisition (Acq)	Obtain or generate relevant data (e.g., via sensors or apps)
Data Simulation (Sim)	Simulate systems to generate data and explore causal "what-if" scenarios
Data Architecting (Arch)	Design the logical/physical layout and integration of data sources
Data Release (Rel)	Make data accessible through databases, APIs, or visualisations

Many taxonomies of data science activities exist (see Martínez-Plumed et al., 2019, Sec 2). One of the most popular is CRISP-DM (Cross Industry Standard Process for Data Mining, Chapman, 2000), which considers projects as *goal-oriented*, with a pre-defined objective that can be approached by "mining" data through an approximately sequential process, from problem framing to solution delivery. However, Martínez-Plumed

et al. (2019) argues that this goal-oriented, pre-collected-data perspective ignores many tasks of modern data science, where exploration is essential and the data takes centre stage rather than serving as a fixed backdrop. Consequently, Martínez-Plumed et al. (2019) proposes a sequence of *exploratory* activities that underscore the less prescriptive nature of data science, re-framing it as an investigative endeavour; and *data management* activities that treat data as dynamic rather than static. We provide a list of the activities introduced in Martínez-Plumed et al. (2019) and a concise definition in Table 1. Note that not all modern data-science projects include every activity, nor is the order of activities fixed as in the CRISP-DM framework. Instead, each project follows its own "trajectory" in the space of data-science tasks (Martínez-Plumed et al., 2019).

3 Evaluating LLM assistants in data science

In this section, we focus on evaluations of LLMs as assistants, namely prompting them in a fixed, predetermined manner without letting them independently determine the sequence of steps. Table 2 shows the surveyed papers and the activities (Sec. 2.5) they cover; a double tick marks an activity that is explicitly assessed, whereas a single tick marks one that is vital for completing the tasks but not directly assessed.

Table 2: Data science activities covered by the surveyed LLM assistants evaluation works. See Sec. 2.5 for definition of the acronyms. A double tick refers to an activity explicitly evaluated, while a single tick refers to an activity necessary for succeeding in the tasks but not explicitly evaluated.

	Goal-oriented						**Exploratory**						**Data Management**			
Papers	BU	DU	DP	M	E	Dep	GE	DSE	DVE	RE	NE	PE	Acq	Sim	Arch	Rel
ARCADE (Yin et al., 2022)	-	✓	✓✓	-	-	-	-	-	✓	-	-	-	-	-	-	-
CERT (Zan et al., 2022)	-	-	✓✓	-	-	-	-	✓	✓	-	-	-	-	-	-	-
CoSQL (Yu et al., 2019a)	-	✓	-	-	-	✓✓	✓✓	-	-	-	-	-	-	-	✓✓	-
DS-1000 (Lai et al., 2023)	-	✓	✓✓	✓✓	-	-	-	-	-	-	-	-	-	-	-	-
DS-Bench (Ouyang et al., 2025)	-	✓	✓✓	✓✓	-	-	-	-	-	-	-	-	-	-	-	-
DSP (Chandel et al., 2022)	-	✓	✓✓	-	-	-	-	-	-	-	-	-	-	-	-	-
FeatEng (Pietruszka et al., 2024)	-	✓✓	✓✓	✓	-	-	-	✓	✓	-	-	-	-	-	-	-
GPT4-DA (Cheng et al., 2023)	✓	✓✓	✓	✓✓	-	✓✓	-	-	✓✓	✓✓	✓✓	-	-	-	-	-
HardML (Pricope, 2025)	-	✓	✓	✓	-	-	-	-	✓	-	✓	-	-	-	-	-
LIDA (Dibia, 2023)	-	✓✓	✓✓	-	✓	-	✓✓	✓	✓✓	-	✓✓	-	-	-	✓	✓✓
SParC (Yu et al., 2019b)	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓✓	-
Spider (Yu et al., 2018)	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓✓	-
Spider 2.0-Lite (Lei et al., 2024)	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓✓	-
Spider 2.0-Snow (Lei et al., 2024)	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	✓✓	-
StatLLM (Song et al., 2025)	-	✓	✓✓	✓✓	✓	-	-	-	-	-	-	-	-	-	-	-

First, many works evaluate LLMs used to generate code for specific steps of data science, such as preprocessing data given a template, fixing bugs, or producing visualisations given instructions or prerequisites. In particular, **ARCADE** (Yin et al., 2022) and **CERT** (Zan et al., 2022) focus on Data Preparation and related activities with specific Python libraries. ARCADE is a benchmark consisting of 1,082 coding problems involving data wrangling and Exploratory Data Analysis (EDA), defined as Jupyter notebooks, that require Python's `Pandas` library; for example, a problem could involve extracting min and max values from a dataframe and answer questions such as "In which year was the most played game added?". CERT instead introduces two benchmarks (PandasEval and NumpyEval), each consisting of 101 tasks manually reworked for coherence and consistency from StackOverflow[2] problems tagged as relevant to `Pandas` and `NumPy` respectively; for problems whose solution is a function, 20 test cases are included, while the correctness of the predicted variable is checked for the other problems. Relatedly, **DSP** (Chandel et al., 2022) contains problems instantiated in 306 pedagogical Jupyter notebooks with 92 associated datasets, covering data manipulation, cleaning, and wrangling (parts of Data Preparation). Similarly to CERT, the correctness of the task is automatically graded with test cases. An example problem would be "Show the correlation between

[2] `https://stackoverflow.com/questions`

population density in 2023 and 2050, rounded to 2 decimals". Similarly, **DS-1000** (Lai et al., 2023) consists of 1,000 coding problems extracted from StackOverflow, spanning Python libraries such as `NumPy`, `Pandas`, `SciKit-Learn`, `TensorFlow`, `matplotlib`, `SciPy`, and `PyTorch`. The problems are manually perturbed to circumvent the issue of memorisation in LLMs and cover Data Preparation and Modelling. The problems are scored through multi-criteria execution-based evaluation metrics that rely on test cases and surface-form constraints to check for the presence of specific APIs. See Fig. 1 for an example problem and evaluation set-up. Ouyang et al. (2025) extend DS-1000 to create **DS-Bench** by adding `Seaborn`, `Keras` and `LightGBM`. To build the benchmark, they first define a broad task scope and collect Python seed code from GitHub using DS-1000's reference code and corresponding StackOverflow answers. An automated LLM pipeline then transforms each snippet to avoid memorisation. Candidates are filtered by properties (compilability, stars, API calls) and functionality (must pass at least one LLM-generated test). For each surviving candidate, an LLM generates 200 test cases and a problem description—complete with an introduction, function signature, input/output formats and examples. After manual review, 1,000 problems are selected. Performance is measured with pass@k: a problem is solved if any k generated samples pass the unit tests. Instead, **FeatEng** (Pietruszka et al., 2024), evaluates LLMs' ability to produce a Python function for engineering data features (thus addressing Data Understanding and Preparation) suitable for downstream modelling tasks. The authors select datasets based on their popularity on Kaggle and ensuring broad domain coverage, reaching a total of 101 tasks. Notably, and in contrast to the older works described above, where questions admitted fixed ground truths, performance is measured in terms of the reduction in error of a model trained on the extracted features compared to a baseline model trained on the original, untransformed data. Therefore, this allows AI models to go beyond simply substituting humans and reach higher levels of task transformation (Sec. 2.1). **StatLLM** (Song et al., 2025) instead focuses on statistical Modelling, assessing LLMs' ability to generate code to solve a dataset of 207 statistical analysis tasks assembled from various public online resources, including descriptive statistics, hypothesis testing, regression and ANOVA, generalised linear models, survival analysis, model selection, and non-parametric statistics; tasks might require the LLM to run a specific model on a variable in a given dataset, or to plot a variable. Uniquely, the LLM has to generate SAS code; evaluation is carried out using Natural Language Processing (NLP) metrics to compare the generated code against a human gold standard, thus being grounded in *substitution* (Sec. 2.1).

Considering Data Management activities, **Spider** (Yu et al., 2018), **SParC** (Yu et al., 2019b) and **CoSQL** (Yu et al., 2019a) (all from the same research group) evaluate conversational database querying systems translating natural language into SQL queries (part of Data Architecting but also requiring Data Understanding). These works build on the same 200 databases from 138 domains: Spider consists of 10,181 manually crafted questions and 5,693 unique SQL reference queries and evaluates the generated queries with matching of SQL components or the overall query to the reference one, or with the accuracy of the execution. SParC expands Spider, which contains only single-turn questions, by simulating multi-turn interactions and therefore introducing context dependence: annotators chained Spider tasks together in a conversational flow resulting in 4,298 question sequences with 12,726 questions. Performance is evaluated in terms of exact set match (per turn), and interaction match (full sequence accuracy); however, this does *not* evaluate the ability of the AI system to interact with a user successfully. This is done in CoSQL, which also includes task where the system must identify ambiguous questions needing clarifications and unanswerable queries (accuracy is evaluated using dialogue act labels). This makes CoSQL unique in addressing intermediate levels of automation for assistants (Sec. 2.2). The clarifications are then included in the context the system uses to determine the correct SQL query, scored using exact match or component match. However, despite being inserted in the context of a conversation, only one system answer at a time is evaluated, therefore still considering the paradigm of *substitution* (Sec. 2.1). Natural language summaries of the query output produced by the system are also evaluated (with the BLEU score). Overall, CoSQL comprises over 30,000 dialogue turns and 10,000 annotated SQL queries, derived from approximately 3,000 dialogues collected by having users interact with a mock interface controlled by an expert and simulating real-world database query scenarios. Finally, the same authors recently introduced (Lei et al., 2024), **Spider 2.0-Lite**, consisting of 547 test instructions mapped to 158 real databases hosted on BigQuery, Snowflake and SQLite and solely scored based on execution accuracy, and **Spider 2.0-Snow**, re-hosting the same 547 questions on Snowflake to spotlight one dialect while keeping identical self-contained evaluation.

Moving to the exploratory aspects of data science, **LIDA** (Dibia, 2023) introduces a system generating data visualisation and infographics by prompting LLMs in a structured manner to provide a summary of the dataset (Data Understanding), formulate data exploration goals (Data Value Exploration), generate code specifications for the visualisations (Goal Exploration), and generate stylised graphics based on the previous output (Narrative Exploration). This also covers aspects of Data Release as it involves making data accessible through visualisations. The system is accompanied by an evaluation tool, based on 57 datasets sourced from the `Vega` datasets[3] repository; two metrics are used: visualisation error rate, computed as the percentage of generated visualisations that result in code compilation errors; and visualisation quality, in which GPT-4 (Achiam et al., 2023) is tasked with assessing the quality of the generated visualisations across 6 dimensions: code accuracy, data transformation, goal compliance, visualisation type, data encoding, and aesthetics.

While the above works focus on single steps of the data science pipeline, Cheng et al. (2023) evaluates GPT-4 as a data analyst on end-to-end data mining problems (excluding several exploratory steps and the entirety of data management). In particular, they provide GPT-4 with a database schema (Data Understanding) and a real-world business question (Business Understanding) and tasks it with extracting the relevant data (Data Preparation, Data Value Exploration), conducting Modelling, generating visualisations and producing an analysis (Deployment, Narrative Exploration, Result Exploration). GPT-4 is embedded within a framework (referred to as **GPT-4DA**), in which it is first prompted to generate code that is executed to produce graphs and a text file containing the generated data, and then prompted again to generate an analysis comprising five insights derived from the textual data (excluding the figures). They devise three evaluation metrics for the generated figures (correctness of data and information, chart type, and aesthetic), and four evaluation metrics for the generated insights (correctness of data and information, alignment with question, complexity, and fluency). By using these broad metrics, GPT-4 is free to solve the task in ways different from what humans would do, thus reaching higher levels of transformation (Sec. 2.1). They test this pipeline on the NvBench dataset (Luo et al., 2021) and employ six human professionals to evaluate GPT-4 (using a rubric detailing the above metrics) and a professional (human) data analyst as baseline. While involving humans leads to more comprehensive understanding of performance, it also makes running the evaluation more costly and less reproducible.

Data science involves additional skills other than coding. For example, domain knowledge in data science is essential. To evaluate this, Pricope (2025) introduce **HardML**, a benchmark of 100 multiple-choice questions, designed to challenge experienced data science professionals, assessing advanced reasoning skills and domain knowledge. The questions are original, handcrafted, and may include multiple correct answers; they span various topics such as natural language processing, computer vision, statistics and statistical modelling, classical machine learning, and cover activities such as Data Understanding and Preparation, Modelling, Data Value Exploration and Narrative Exploration. An example question is: "An AI company just shipped a new foundational language model. They claim they have trained it for 2.79M H800 hours on 14.8T tokens. Upon further research, looking at Nvidia card specs, you find 3,026 TFLOPs/s of FP8 performance with sparsity, or typically half this (1.513e15 FLOPs/s) without sparsity. Moreover, you find out that they used FP8 FLOPs without structured sparsity. Given that the model has 37B activated parameters, roughly what hardware utilization did they achieve? Select the closest." Importantly, whilst several activities are evaluated by the benchmark, each question only targets a single activity; moreover, the majority of the questions focus on reasoning capabilities and coding for machine learning and various aspects of deep learning engineering.

From this overview, it is evident how nearly all LLM assistant evaluation works focus on code generation, and how there is a concentration on the goal-oriented activities of Data Understanding, Data Preparation and Modelling (Table 2), with lack of evaluation works for exploratory and, even more, data management activities, except for Data Value Understanding, and a few works touching on Narrative Exploration, Goal Exploration, Data Source Exploration, Result Exploration, Data Architecting, and Data Release.

4 Evaluating LLM agents in data science

In this section, we consider evaluations of LLM agents, which augment LLMs with a set of affordances and allow them to determine the sequence of steps they go through by iterative prompting. We also consider works evaluating agents with (simulated) user interaction. Table 3 shows the papers we overview and the

[3] `https://github.com/vega/vega-datasets`

Table 3: Data science activities covered by the surveyed LLM agent evaluation works. See Sec. 2.5 for definition of the acronyms. A double tick refers to an activity explicitly evaluated, while a single tick refers to an activity necessary for succeeding in the tasks but not explicitly evaluated.

	Goal-oriented						**Exploratory**						**Data Management**			
Papers	BU	DU	DP	M	E	Dep	GE	DSE	DVE	RE	NE	PE	Acq	Sim	Arch	Rel
BLADE (Gu et al., 2024)	-	✓✓	✓✓	✓✓	-	✓	-	-	✓	-	✓✓	-	-	-	-	✓
BiasBenchmark (Li et al., 2025b)	-	✓✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CSR-Bench (Xiao et al., 2025)	-	-	✓	✓	✓	✓✓	-	-	-	-	-	-	-	-	-	-
Data-Copilot (Zhang et al., 2024a)	-	✓	✓✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-
DA-Code (Huang et al., 2024b)	-	✓✓	✓✓	✓✓	✓	-	-	-	✓	-	-	-	-	-	-	-
DiscoveryBench (Majumder et al., 2024)	✓	✓	✓	✓	-	✓	✓✓	✓	✓	-	✓✓	-	-	✓	-	-
DSBench (Jing et al., 2024)	-	✓	✓	✓✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-	-
DS-Eval (Zhang et al., 2024b)	✓	✓✓	✓✓	✓✓	-	✓✓	-	-	-	-	✓✓	-	-	-	-	-
IDA-Bench (Li et al., 2025a)	-	✓	✓	✓✓	-	✓✓	-	-	-	-	-	-	-	-	-	-
InfiAgent-DABench (Hu et al., 2024)	-	✓	✓	✓✓	-	-	-	-	✓	-	✓	-	-	-	-	-
InsightBench (Sahu et al., 2025)	✓	✓	✓	✓	✓	✓✓	✓	-	✓	✓✓	✓✓	-	-	-	-	✓✓
MLAgentBench (Huang et al., 2024a)	-	✓	-	✓✓	✓	✓	-	-	-	✓	✓	-	-	-	-	-
MLE-Bench (Chan et al., 2024)	-	✓	✓	✓✓	-	✓	-	-	✓	-	✓	-	-	-	-	-
MLGym (Nathani et al., 2025)	-	✓	✓	✓✓	✓	-	-	-	✓	-	-	-	-	-	-	-
RE-Bench (Wijk et al., 2024)	-	-	-	✓✓	-	-	-	-	-	-	-	-	-	-	-	-
ScienceAgentBench (Chen et al., 2024)	-	✓✓	✓✓	✓✓	-	✓✓	✓	-	✓	-	✓✓	-	-	-	-	-
Spider 2.0 (Lei et al., 2024)	-	✓✓	-	-	-	-	✓✓	-	-	-	-	-	-	-	✓✓	-
SUPER (Bogin et al., 2024)	-	-	-	✓	✓	✓✓	-	-	-	-	-	-	-	-	-	-
Tapilot-Crossing (Li et al., 2024)	-	✓	✓✓	✓✓	-	✓	✓	-	-	-	✓	-	-	-	-	-

activities they cover (Sec. 2.5); a double tick marks an activity that is explicitly assessed, whereas a single tick marks one that is vital for completing the tasks but not directly assessed.

4.1 Works targeting specific goal-oriented activities

Many works aim to evaluate LLM agents on individual goal-oriented activities of the data science pipeline. Starting from Data Understanding, Li et al. (2025b) introduce **BiasBenchmark**, which evaluates the ability to detect biases in datasets. To build the benchmark, they select 5 datasets from prior bias mitigation research and 100 demographic-related features or their combinations. They then craft (using an LLM playing the role of a user) possible bias detection queries, including intentionally ambiguous questions. During evaluation, clarification questions posed by the agent are automatically answered by the LLM-based user simulator according to the original task specifications, ensuring consistency and reproducibility. The agent must quantify the bias level according to a 5-level scale, which is then compared to the ground truth obtained by measuring five widely-used bias detection metrics. However, scoring the final bias level allows to agent to determine it in potentially novel ways, therefore going above human substitution (Sec. 2.1). They also evaluate the agent's intermediate process, by developing an agent-based automated evaluation system that looks at the evaluated agent's logs and produces performance rating levels for five aspects: user communication, task planning, tool invocation, dynamic plan adjustment and result analysis. Interestingly, the first one (user communication) scores the ability of the agent to ask clarification questions to the (simulated) user who set the task.

Instead, **Data-Copilot** (Zhang et al., 2024a) introduces an LLM agent for data wrangling that, given a dataset schema, independently explores potential user requests and generates modular code to address them, which is then leveraged in the deployment stage. To benchmark it, the authors release 547 test requests drawn from 173 human seeds plus a larger 3547-request self-exploration pool. The tasks rely on financial data and touch upon Data Value Exploration, Data Understanding and Preparation, and Modelling. Each test case comes with a human-curated answer table and four manual labels for dataset analysis: task difficulty, request rationality, expression ambiguity, answer accuracy. System performance is measured with GPT-4-based Pass@1 scoring against the gold tables (plus an image check) and with the number of tokens used.

Moving towards Modelling, **MLE-Bench** (Chan et al., 2024) and **RE-Bench** (Wijk et al., 2024) are both learning engineering benchmarks, but differ in the complexity of the tasks and scenarios: MLE-Bench encompasses 75 tasks sourced from Kaggle[4], whose deterministic scoring functions are taken from the corresponding Kaggle competitions—as they score the final result, this allows agents to solve the task in ways potentially different from humans, going beyond substitution (Sec. 2.1); however, because these functions vary across tasks, each score is compared against a snapshot of the (human) leaderboard. Instead, RE-Bench includes 7 environments each presenting a unique Machine Learning (ML) task focused on optimising either the loss function or the run-time; the value of these scoring functions is manually inspected for evaluation, and evaluators need to have access to a reference solution. Relatedly, **MLAgentBench** (Huang et al., 2024a) comprises 13 tasks specified by a goal, occasionally constraints or specific instructions, starter files, and an evaluator; the tasks are collected and adapted from recent Kaggle challenges, CLRS (Veličković et al., 2022), BabyLM[5]; the starter files consist of data, description of data and metric, and initial code; each task has its own goal metric to improve on, whose measure is used for automated evaluation. For an overview of MLAgent workflow and evaluation, see Fig. 6. Instead, Nathani et al. (2025) introduce **MLGym**, an environment to train LLM agents on ML tasks using reinforcement learning. Given a task description, an initial codebase, and actions and observations history, the agent generates an action (shell commands executed by the environment) to accomplish research objectives iteratively; the execution feedback can then be used to refine the agent. MLGym is equipped with a benchmark consisting of 13 tasks spanning data science, game theory, computer vision, NLP, and reinforcement learning, and selected from sources such as Kaggle's House Price Prediction[6], 3-SAT (Cook, 1971), CIFAR-10's image classification[7], and more; they require the agent to perform Data Understanding, Modelling and Evaluation. As the various tasks have different performance metric, they score each agent by a quantity that reflects how closely, on average across a range of tolerance levels, it matches the best performer on every individual task. MLGym differs from MLAgentBench for the larger complexity of its tasks. Finally, (Li et al., 2025a) introduce **IDA-Bench**, which attempts to evaluate LLMs on their ability to perform guided predictive Modelling tasks; the benchmark includes an LLM-simulated user with domain knowledge and subjective insights who interacts with an agent to provide instructions throughout a multi-turn iterative process; the agent is then tested on adapting its goal and following instructions. The tasks are obtained from Kaggle; an LLM distils reference insights in natural language format from an optimal solution. These, together with information such as hyperparameters, serve as a task-specific template for the simulated user, which requests the agent to perform certain steps, without offering all insights up-front, and offers clarifications when asked. Results of the trained model are evaluated against a ground truth using task-specific evaluation functions; and compared with a human baseline, obtained by running the notebook the simulated user has access to. They also determine the ability to interact by considering how the prediction accuracy changes by increasing the number of interactions. Fig. 3 shows an example "trajectory" of the guided data analysis process.

Next, **CSR-Bench** (Xiao et al., 2025) and **SUPER** (Bogin et al., 2024) test whether agents can correctly deploy code from a project repository when given instructions—an important, though not exclusive, part of the data-science Deployment stage (Sec. 2.5). CSR-Bench and SUPER both transform GitHub repositories into end-to-end "run-the-code" challenges in which an autonomous LLM agent must parse documentation, install dependencies, debug failures, and produce a outcomes assessed by an automatic completion metric. CSR-Bench supplies 100 diverse repositories, each constituting one comprehensive task that typically involves environment setup, data and model acquisition, model training, inference, and evaluation. In contrast, SUPER targets reproducibility in machine-learning and NLP research across 801 repositories, organised into three nested subsets—*expert* (45 manually authored full-pipeline problems with human gold standards), *masked* (152 focused subtasks derived from the expert set), and *auto* (604 GPT-4-o-generated tasks created from repository `README`s)—each accompanied by task-specific metrics or expected outputs for evaluation.

Finally, considering agents performing Data Management tasks, **Spider 2.0** (Lei et al., 2024), the most recent iteration of Spider (Yu et al., 2018), is a benchmark of 632 real-world text-to-SQL workflow problems derived from enterprise-level database use cases; the agent's answers are evaluated using completion rate, accuracy, and coherence, therefore allowing agents to solve the task in ways different from pure human

[4] `https://www.kaggle.com/`
[5] `https://babylm.github.io/`
[6] `https://www.kaggle.com/datasets/zafarali27/house-price-prediction-dataset`
[7] `https://www.kaggle.com/code/faressayah/cifar-10-images-classification-using-cnns-88`

substitution (Sec. 2.1). Spider 2.0 differs from previous benchmarks by the same authors (Yu et al., 2018; 2019b;a, discussed in Sec. 3), in its more complex set-up: the tasks do not consist of pre-prepared inputs (question and database schema) or expected outputs (predicted SQL), but a real project codebase and a database interface; the agent interacts with the codebase through command scripts, as well as SQL queries.

4.2 Evaluating multiple activities explicitly

Some works target a broader spectrum of data science activities and evaluate each explicitly. To start with, **DA-Code** (Huang et al., 2024b), **InfiAgent-DABench** (Hu et al., 2024) and **DSBench** (Jing et al., 2024) all predominantly consider Data Preparation and Modelling, and mostly score the agent-produced solution by closely comparing it with reference ones, thus being anchored in the "substitution" paradigm (Sec. 2.1). In particular, DA-Code consists of 500 tasks sourced from Kaggle, GitHub, and other sources, each primarily covering exploratory data analysis (which roughly includes Data Understanding and Data Value Exploration), Data Preparation, or Modelling—thus, even though the overall benchmark consider multiple activities, each task is more narrow. DA-Code includes a variety of data structures and requires the use of SQL, Python, and Bash. Each task is accompanied by a single canonical artefact (table, chart, text file or hidden test-set labels) created by experienced annotators except for predictive modelling tasks. For grading, a solution is stripped down to the elements explicitly constrained by the instructions (such as required columns, the numeric data underlying a plot, or specified visual metadata) before applying a strict equality check against the reference artefact. For machine-learning tasks, the grader instead computes the task-specific metric (e.g. F1, MAE, Silhouette) on the hidden labels and awards partial credit in proportion to performance above baseline. Relatedly, InfiAgent-DABench introduces DAEval, a dataset of 257 GPT-4 generated closed-form questions, such as "Is there a linear relationship between the GDP per capita and the life expectancy score in Happiness_rank.csv? Conduct linear regression and use the resulting coefficient of determination (R-squared) to evaluate the model's goodness of fit ... [omitted for brevity]", derived from `csv` files sourced from GitHub repositories, with respective gold-standard answers generated by OpenAI's Advanced Data Analysis[8]. The benchmark covers a broad range of tasks, such as feature engineering, correlation analysis, data preprocessing, distribution analysis, summary statistics (all representing Data Preparation and Understanding), and machine learning (Modelling). The evaluation relies on calculating the portion of questions for which all subquestions exactly match the reference solution. Finally, DSBench obtains tasks from ModelOff[9] and Kaggle and split them into two categories: data analysis, 466 tasks characterised by long text context, various modalities, and a wide scope for solutions, and evaluated in terms of accuracy by an LLM which compares the responses to a human solution; and data modelling, 74 tasks requiring the LLM to build a predictive model with performance scored by the ability of the agent to generate and submit a bug-free model. Beyond Data Understanding, Data Preparation and Modelling, some tasks also cover Evaluation, Deployment, and exploratory activities.

Moving to a broader range of activities, **DSEval** (Zhang et al., 2024b) contains chains of inter-dependent problems (based on data from StackOverflow, Pandas-exercises[10], LeetCode[11], and Kaggle) where each highlights a different stage of the data-science lifecycle—Data Understanding and Preparation, Modelling, or interpretation (belonging to Deployment and Narrative Exploration)—while re-using the runtime context left by the previous problems. By doing so, agents must solve the overall task by following the same steps that humans would follow; thus, DSEval only evaluates agents' substitution ability rather than their potential to transform tasks (Sec. 2.1). For each problem, they employ custom validator modules to check correctness against the solution or run unit tests. Relatedly, **Tapilot-Crossing** (Li et al., 2024) obtains a set of tasks by simulating (with LLMs) a company setup composed of an administrator and data scientist solving a client's problem making use of an AI Chatbot Agent; they then manually filter those interactions where the Chatbot Agent produced correct code and obtain 1024 interactions where agents are asked to write code to solve a problem or answer a multiple-choice question. Overall, these tasks cover Data Understanding and Preparation, Data Value Exploration, Modelling, Deployment, Results and Narrative Exploration (by converting plots into answers or summarising findings in prose). They test agents both in a "normal" mode,

[8] `https://openai.com/blog/chatgpt-plugins#code-interpreter`
[9] `https://corporatefinanceinstitute.com/resources/financial-modeling/modeloff-guide/`
[10] `https://github.com/guipsamora/pandas_exercises`
[11] `https://leetcode.com/`

where all requirements and details are specified by the user, and in an "action" mode, where the agent has to perform an action such as asking for clarification, updating code based on user-reported error, and others. For code updating and normal turns, the agent's interactivity is disabled, effectively falling back to an assistant setup, while in the other cases the agent may iteratively call a sandboxed Python executor. Further, for the "clarification" tasks, the agent has to pose follow-up questions that are answered by a LLM-simulated user, making this one of a few benchmarks (Li et al., 2025b;a) that evaluate interactivity. An example of the different interaction modes can be found in Fig. 4. Overall, however, scoring is purely based on the outcome (thus not judging the quality of the interaction): the agent's artefact is compared to a gold reference with task-specific comparators. Finally, **ScienceAgentBench** (Chen et al., 2024) builds on 102 tasks from scientific peer-reviewed publications, validated by subject experts; each task includes a data-driven discovery goal, information on the data, expert-provided knowledge, and a reference Python program. The questions are challenging, such as "Develop a drug-target interaction model with the DAVIS dataset to repurpose the antiviral drugs for COVID", or "Analyze Toronto fire stations and their service coverage to identify coverage gaps". The performance of the LLM agent on each task is scored against 3 metrics: Program Evaluation (itself consisting of: Valid Execution Rate, Success Rate, API Cost and embedding similarity computed by CodeBERT Zhou et al., 2023); Figure Evaluation (using GPT-4o); and Rubric-Based Evaluation based on 5 fundamental steps (Data Loading, Data Processing, Modelling or Visualisation, Output Formatting, and Output Saving). Therefore, this mostly evaluates substitution (Sec. 2.1) both by closely referring to human-provided solutions and by splitting the task in the same sequence of steps humans would follow.

4.3 End-to-end tasks scored by their final result

A few works instead evaluate agents on end-to-end questions—involving formulating plans, generating code and plots, and producing coherent results and insights—and score the final output of the task (in contrast to individual steps as in Sec. 4.2). This naturally allows to reward higher levels of task transformation beyond mere human substitution (Sec. 2.1). First, **InsightBench** (Sahu et al., 2025) includes 100 tabular datasets of 500 synthetically-generated entries each, organised in structures obtained from a real-world enterprise data management platform. When generating the synthetic data, a set of insights is manually "planted" in them. The insights (a total of 475) are divided into four families: descriptive, consisting of plots that describe the data; diagnostic, analysing the cause behind trends; predictive, consisting of visualisations that summarise model predictions; prescriptive, that explain actionable insights. The LLM agents are evaluated based on how many insights they recover, when provided with the dataset and an open-ended goal formulated by non-expert users, such as "Analyse incident trends in the data.csv file". In particular, Llama-3-Eval, a technique inspired by G-Eval (Liu et al., 2023b) which uses Llama-3 (Dubey et al., 2024), is used to compare the agent-produced insights with the reference ones, both at a summary level and at a deeper description level. The tasks require touch upon multiple data science activities—from Goal Exploration to Data Understanding, Value Exploration and Preparation, and to Modelling and Narrative Exploration—but only the final insights are directly evaluated (Deployment, Narrative Exploration, and Result Exploration). Similarly, **BLADE** (Gu et al., 2024) and **DiscoveryBench** (Majumder et al., 2024) both challenge LLM agents to explore a complex dataset with a vaguely defined goal, such as "Are soccer players with a dark skin tone more likely than those with a light skin tone to receive red cards from referees?" (Gu et al., 2024). However, differently from InsightBench, they consider scientific datasets and focuses on agents' ability to integrate statistical knowledge with understanding of data from a broad range of scientific domains. Both BLADE and DiscoveryBench include end-to-end scientific data-analysis tasks that begin with genuine research questions and necessitate multistep solution workflows. BLADE includes 12 carefully curated datasets collected from statistical textbooks, research papers, and crowd-sourced studies, while DiscoveryBench offers 264 real-world tasks drawn from published studies plus 903 synthetic tasks spanning 48 domains. Tasks cover Data Understanding, Preparation, statistical and Machine Learning Modelling, Narrative Exploration, as well as Deployment, Data Value Exploration and various degrees of domain understanding (Business Understanding and Data Understanding). Both BLADE and DiscoveryBench grade solutions automatically with LLMs so that multiple defensible workflows can receive credit. Their emphases, however, diverge: BLADE checks if each analytical step—conceptual variable selection, admissible transformations, model family, hyper-parameters—corresponds to one of multiple expert-produced solutions (to account for alternatives), which however still limits the amount by which the agent can transform the task; DiscoveryBench

instead scores at the level of the final context, variables and relationship identified, using GPT-4 to judge the semantic match between the agent's claim and without considering how the former was obtained. An overview of BLADE and DiscoveryBench can be found in Figs. 2 and 5.

From Table 3, we can see that agents have been evaluated on more data science activities than assistants (Table 2), particularly considering goal-oriented activities (with the exception of Business Understanding); however, there is still a lack of evaluations for data management and, to a lesser extent, exploratory activities. Additionally, of all the surveyed works, only BiasBenchmark (Li et al., 2025b), Tapilot-Crossing (Li et al., 2024), and IDA-Bench (Li et al., 2025a) evaluate agents in a collaborative framework with (simulated) users.

5 Challenges and future directions

Our analysis shows that most evaluation works focus either on assistance, looking at isolated tasks that require LLMs to provide an answer on a single-turn basis (without access to tools and under human supervision) or on full automation, where LLMs are wrapped in agents that act autonomously. A few notable exceptions exist (Yu et al., 2019a; Li et al., 2024; 2025b;a), which primarily rely on other LLMs to simulate human users. While this approach ensures cost-effective evaluation and reproducibility, exploratory tasks may lead the agent to attempt novel solutions, such that the simulated user may be unable to assist it as the most suitable answer may not be within its knowledge base. This can be partly addressed by employing humans to answer such unprecedented queries and progressively enriching the knowledge base. Furthermore, informative evaluations should also quantify the trade-off between autonomy and reliability/performance; among the surveyed works, only Li et al. (2025a) advances in this direction.

We found data management and exploratory activities remain mostly uncovered. This is due to 1) the inherent difficulty of scoring exploratory activities, which lack a fixed ground truth, and 2) the complex real-world interactions that certain exploratory activities (such as Business Understanding and Goal Exploration) and data management activities (Acquisition and Simulation) demand. To address this issue, simulated environments where data management and client interaction can occur should be developed, analogous to related developments in scientific research evaluation (Jansen et al., 2024; Cerrato et al., 2024). Such environments would enable the evaluation of agents or assistants that function holistically as data scientists by understanding business requirements, facilitating data collection, and adapting customer requests through data exploration. Simultaneously, realistic evaluation should progress toward end-to-end tasks that do not depend on strict ground truth comparisons or simple activity-specific metrics. Instead, these evaluations should reward insight generation (such as the works in Sec. 4.3) in potentially original ways, thereby properly incentivising systems that fundamentally redefine activities rather than focusing solely on human substitution.

Thus, we propose the following actions to improve data science evaluation of AI systems:

- More comprehensive benchmarks covering most activities in Table 1, considering intermediate steps and preparatory activities, for evaluating substitution-focused approaches.
- Greater emphasis on incorporating human assistance (either real or simulated) in the evaluation, and developing methods to quantify the trade-off between autonomy and reliability.
- Development of comprehensive simulated environments that enable testing AI systems as holistic data scientists performing data collection and client interaction activities.
- Evaluations incorporating end-to-end tasks and broad objectives that allow and reward systems that redefine activities and propose original solutions differing from the reference ones.
- Field studies to validate the measurements obtained through these evaluation tools by comparing them to the real-world impact of human-AI collaborations.

There has been enormous progress in data science automation, compared to the state of the art just a few years ago (De Bie et al., 2022). It is in open-ended tasks, the use of domain context and human-AI collaboration where data science automation is lagging behind, but upcoming tools may be able to conquer these domains: we must make sure our evaluations allow us to properly track progress.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Made laine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Jo hannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Li, Rachel Lim, Molly Lin, Stephanie Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report. 2023. URL https://arxiv.org/pdf/2303.08774.

Tijl De Bie, Luc de Raedt, José Hernández-Orallo, Holger H. Hoos, Padhraic Smyth, and Christopher K. I. Williams. Automating data science. *Communications of the ACM*, 65:76–87, 2021. URL https://dl.acm.org/doi/10.1145/3495256.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. SUPER: Evaluating Agents on Setting Up and Executing Tasks from Research Repositories. *ArXiv*, abs/2409.07440, 2024. URL https://arxiv.org/abs/2409.07440.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper%5Ffiles/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Erik Brynjolfsson. The turing trap: The promise & peril of human-like artificial intelligence. *Daedalus*, 151 (2):272–287, 2022.

Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. Generative AI at work. *The Quarterly Journal of Economics*, pp. qjae044, 2025.

John Burden, Marko Tešić, Lorenzo Pacchiardi, and José Hernández-Orallo. Paradigms of AI Evaluation: Mapping Goals, Methodologies and Culture. *arXiv preprint arXiv:2502.15620*, 2025. URL https://doi.org/10.48550/arXiv.2502.15620.

Mattia Cerrato, Nicholas Schmitt, Lennart Baur, Edward Finkelstein, Selina Jukic, Lars Münzel, Felix Peter Paul, Pascal Pfannes, Benedikt Rohr, Julius Schellenberg, et al. Science-Gym: A simple testbed for AI-driven scientific discovery. In *International Conference on Discovery Science*, pp. 229–243. Springer, 2024.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal A. Patwardhan, Lilian Weng, and Aleksander Mkadry. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *ArXiv*, abs/2410.07095, 2024. URL https://arxiv.org/abs/2410.07095.

Shubham Chandel, Colin B. Clement, Guillermo Serrato, and Neel Sundaresan. Training and Evaluating a Jupyter Notebook Data Science Assistant. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022. URL https://github.com/microsoft/DataScienceProblems.

Yu-Chu Chang, Xu Wang, Jindong Wang, Yuanyi Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Weirong Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qian Yang, and Xingxu Xie. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15:1–45, 2023. URL https://dl.acm.org/doi/10.1145/3641289.

Peter Chapman. CRISP-DM 1.0: Step-by-step data mining guide. 2000. URL https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. *ArXiv*, abs/2410.05080, 2024. URL https://arxiv.org/abs/2410.05080.

Liying Cheng, Xingxuan Li, and Lidong Bing. Is GPT-4 a Good Data Analyst? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9496–9514, 2023. URL https://doi.org/10.48550/arXiv.2305.15038.

Peter Cihon, Merlin Stein, Gagan Bansal, Sam Manning, and Kevin Xu. Measuring AI Agent Autonomy: Towards a Scalable Approach With Code Inspection. In *Workshop on Socially Responsible Language Modelling Research*, 2024. URL https://openreview.net/forum?id=VulxpvCNoA.

Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pp. 151–158, New York, NY, USA, 1971. Association for

Computing Machinery. ISBN 9781450374644. doi: 10.1145/800157.805047. URL https://doi.org/10.1145/800157.805047.

Tijl De Bie, Luc De Raedt, José Hernández-Orallo, Holger H. Hoos, Padhraic Smyth, and Christopher K. I. Williams. Automating data science. *Commun. ACM*, 65(3):76–87, February 2022. ISSN 0001-0782. doi: 10.1145/3495256. URL https://doi.org/10.1145/3495256.

Victor Dibia. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), Volume 3: System Demonstrations*, pp. 113–126, 2023. URL https://aclanthology.org/2023.acl-demo.11/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Bap tiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Ro main Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia,

Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 Herd of Models. *ArXiv*, abs/2407.21783, 2024. URL `https://arxiv.org/abs/2407.21783`.

Maria Eriksson, Erasmo Purificato, Arman Noroozian, Joao Vinagre, Guillaume Chaslot, Emilia Gomez, and David Fernandez-Llorca. Can We Trust AI Benchmarks? An Interdisciplinary Review of Current Issues in AI Evaluation. *arXiv preprint arXiv:2502.06559*, 2025. URL `https://doi.org/10.48550/arXiv.2502.06559`.

Carl Benedikt Frey and Michael Osborne. Generative AI and the future of work: a reappraisal. *Brown J. World Aff.*, 30:161, 2023.

Pieter Gijsbers, Marcos LP Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. Amlb: an automl benchmark. *Journal of Machine Learning Research*, 25(101):1–65, 2024. URL `https://www.jmlr.org/papers/volume25/22-0493/22-0493.pdf`.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, Yikun Zhang, Tianmai M. Zhang, Lanyi Zhu, Mike A. Merrill, Jeffrey Heer, and

Tim Althoff. BLADE: Benchmarking Language Model Agents for Data-Driven Science. *arXiv*, 2024. URL https://arxiv.org/abs/2408.09667v2.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. Evaluating Large Language Models: A Comprehensive Survey. *ArXiv*, abs/2310.19736, 2023. URL https://arxiv.org/abs/2310.19736.

Erica R Hamilton, Joshua M Rosenberg, and Mete Akcaoglu. The substitution augmentation modification redefinition (SAMR) model: A critical review and suggestions for its use. *TechTrends*, 60:433–441, 2016.

Chuxuan Hu, Dwip Dalal, and Xiaona Zhou. A Dataset-Centric Survey of LLM-Agents for Data Science. *OpenReview*, 2025. URL https://openreview.net/pdf?id=W4hexmqgoN.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. InfiAgent-DABench: Evaluating Agents on Data Analysis Tasks. *ArXiv*, abs/2401.05507, 2024. URL https://arxiv.org/abs/2401.05507.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024a. URL https://doi.org/10.48550/arXiv.2310.03302.

Yiming Huang, Jianwen Luo, Yan Yu, Yitong Zhang, Fangyu Lei, Yifan Wei, Shizhu He, Lifu Huang, Xiao Liu, Jun Zhao, and Kang Liu. DA-Code: Agent Data Science Code Generation Benchmark for Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 13487–13521, 2024b. URL https://arxiv.org/abs/2410.07331.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in Neural Information Processing Systems*, 37:10088–10116, 2024.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-Bench: Can Language Models Resolve Real-World GitHub Issues? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. DSBench: How Far Are Data Science Agents to Becoming Data Science Experts? *ArXiv*, abs/2409.07703, 2024. URL https://arxiv.org/abs/2409.07703.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, et al. Measuring AI Ability to Complete Long Tasks. *arXiv preprint arXiv:2503.14499*, 2025.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen tau Yih, Daniel Fried, Sida Wang, and Tao Yu. DS-1000: A Natural and Reliable Benchmark for Data Science Code Generation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. URL https://doi.org/10.48550/arXiv.2211.11501.

Fangyu Lei, Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Hongcheng Gao, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating Language Models on Real-World Enterprise Text-to-SQL Workflows. *ArXiv*, abs/2411.07763, 2024. URL https://arxiv.org/abs/2411.07763.

Hanyu Li, Haoyu Liu, Tingyu Zhu, Tianyu Guo, Zeyu Zheng, Xiaotie Deng, and Michael I. Jordan. IDA-Bench: Evaluating LLMs on Interactive Guided Data Analysis. *arXiv preprint arXiv:2505.18223*, 2025a. URL https://arxiv.org/abs/2505.18223. Submitted on 23 May 2025.

Haoxuan Li, Mingyu Derek Ma, Jen tse Huang, Zhaotian Weng, Wei Wang, and Jieyu Zhao. BIASINSPECTOR: Detecting Bias in Structured Data through LLM Agents. 2025b. URL https://arxiv.org/abs/2504.04855.

Jinyang Li, Nan Huo, Yan Gao, Jiayi Shi, Yingxiu Zhao, Ge Qu, Yurong Wu, Chenhao Ma, Jian-Guang Lou, and Reynold Cheng. Tapilot-Crossing: Benchmarking and Evolving LLMs Towards Interactive Data Analysis Agents. *ArXiv*, abs/2403.05307, 2024. URL https://arxiv.org/abs/2403.05307.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Yuxian Gu, Hangliang Ding, Kai Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Shengqi Shen, Tianjun Zhang, Sheng Shen, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as Agents. *ArXiv*, abs/2308.03688, 2023a. URL https://arxiv.org/abs/2308.03688.

Yang Liu, Dan Iter, Yichong Xu, Shuo Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In *Conference on Empirical Methods in Natural Language Processing*, 2023b. URL https://doi.org/10.48550/arXiv.2303.16634.

Yixin Liu, Alexander R. Fabbri, Pengfei Liu, Dragomir R. Radev, and Arman Cohan. On Learning to Summarize with Large Language Models as References. In *North American Chapter of the Association for Computational Linguistics*, 2023c. URL https://doi.org/10.48550/arXiv.2305.14239.

Yuyu Luo, Jiawei Tang, and Guoliang Li. nvBench: A Large-Scale Synthesized Dataset for Cross-Domain Natural Language to Visualization Task. *ArXiv*, abs/2112.12926, 2021. URL https://arxiv.org/abs/2112.12926.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards Data-Driven Discovery with Large Language Models. *ArXiv*, abs/2407.01725, 2024. URL https://arxiv.org/abs/2407.01725.

Fernando Martínez-Plumed, Lidia Contreras-Ochando, Cesar Ferri, José Hernández-Orallo, Meelis Kull, Nicolas Lachiche, María José Ramírez-Quintana, and Peter Flach. CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE transactions on knowledge and data engineering*, 33(8):3048–3061, 2019. URL https://ieeexplore.ieee.org/abstract/document/8943998.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A New Framework and Benchmark for Advancing AI Research Agents. 2025. URL https://doi.org/10.48550/arXiv.2502.14499.

Shuyin Ouyang, Dong Huang, Jingwen Guo, Zeyu Sun, Qihao Zhu, and Jie M. Zhang. DS-Bench: A Realistic Benchmark for Data Science Code Generation. 2025. URL https://doi.org/10.48550/arXiv.2505.15621.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Michał Pietruszka, Łukasz Borchmann, Aleksander Jędrosz, and Paweł Morawiecki. Can models help us create better models? Evaluating LLMs as data scientists. *arXiv*, 2410.23331v1, 2024. URL https://arxiv.org/abs/2410.23331v1.

Tidor-Vlad Pricope. HardML: A Benchmark for Evaluating Data Science and Machine Learning Knowledge and Reasoning in AI. *arXiv*, 2025. URL https://arxiv.org/abs/2501.15627v1.

Ruben R Puentedura. Transformation, technology, and education. *[Retrieved May], http://hippasus.com/resources/tte/*, 2006.

Gaurav Sahu, Abhay Puri, Juan Rodriguez, Amirhossein Abaskohi, Mohammad Chegini, Alexandre Drouin, Perouz Taslakian, Valentina Zantedeschi, Alexandre Lacoste, David Vazquez, Nicolas Chapados, Christopher Pal, Sai Rajeswar Mudumba, and Issam Hadj Laradji. InsightBench: Evaluating Business Analytics Agents Through Multi-Step Insight Generation. *arXiv*, 2025. URL https://arxiv.org/abs/2407.06423v3.

Ben Shneiderman. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6):495–504, 2020. URL https://doi.org/10.1080/10447318.2020.1741118.

Xinyi Song, Lina Lee, Kexin Xie, Xueying Liu, Xinwei Deng, and Yili Hong. StatLLM: A Dataset for Evaluating the Performance of Large Language Models in Statistical Analysis. 2025. URL https://doi.org/10.48550/arXiv.2502.17657.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. A Survey on Large Language Model-based Agents for Statistics and Data Science. *ArXiv*, abs/2412.14222, 2024. URL https://arxiv.org/abs/2412.14222.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023. URL https://doi.org/10.48550/arXiv.2206.10498.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Mikhail Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS Algorithmic Reasoning Benchmark. In *International Conference on Machine Learning*, 2022. URL https://doi.org/10.48550/arXiv.2205.15659.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi (Jim) Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Trans. Mach. Learn. Res.*, 2024, 2023. URL https://doi.org/10.48550/arXiv.2305.16291.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024. URL https://link.springer.com/article/10.1007/s11704-024-40231-1.

Hjalmar Wijk, Tao R. Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Jun Koba Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. *ArXiv*, abs/2411.15114, 2024. URL https://arxiv.org/abs/2411.15114.

Yijia Xiao, Runhui Wang, Luyang Kong, Davor Golac, and Wei Wang. CSR-Bench: Benchmarking LLM Agents in Deployment of Computer Science Research Repositories. 2025. URL https://doi.org/10.48550/arXiv.2502.06111.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on Evaluation of LLM-based Agents, 2025. URL https://arxiv.org/abs/2503.16416.

Pengcheng Yin, Wen-Ding Li, Kefan Xiao, A. Eashaan Rao, Yeming Wen, Kensen Shi, Joshua Howland, Paige Bailey, Michele Catasta, Henryk Michalewski, Oleksandr Polozov, and Charles Sutton. Natural Language to Code Generation in Interactive Data Science Notebooks. *ArXiv*, abs/2212.09248, 2022. URL https://arxiv.org/abs/2212.09248.

Tao Yu, Rui Zhang, Kai-Chou Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Z Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. *ArXiv*, abs/1809.08887, 2018. URL https://arxiv.org/abs/1809.08887.

Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander R. Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter S. Lasecki, and Dragomir R. Radev. CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases. *ArXiv*, abs/1909.05378, 2019a. URL https://arxiv.org/abs/1909.05378.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, He Yang Er, Irene Z Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir R. Radev. SParC: Cross-Domain Semantic Parsing in Context. *ArXiv*, abs/1906.02285, 2019b. URL https://arxiv.org/abs/1906.02285.

Daoguang Zan, Bei Chen, Dejian Yang, Zeqi Lin, Minsu Kim, Bei Guan, Yongji Wang, Weizhu Chen, and Jian-Guang Lou. CERT: Continual Pre-Training on Sketches for Library-Oriented Code Generation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2369–2375, 2022. URL https://github.com/microsoft/PyCodeGPT.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Data-Copilot: Bridging Billions of Data and Humans with Autonomous Workflow. *arXiv*, 2024a. URL https://arxiv.org/abs/2306.07209v7.

Yuge Zhang, Qiyang Jiang, Xingyu Han, Nan Chen, Yuqing Yang, and Kan Ren. Benchmarking Data Science Agents. In *Annual Meeting of the Association for Computational Linguistics*, 2024b. URL https://doi.org/10.48550/arXiv.2402.17168.

Shuyan Zhou, Uri Alon, Sumit Agarwal, and Graham Neubig. CodeBERTscore: Evaluating code generation with pretrained models of code. *arXiv preprint arXiv:2302.05527*, 2023.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oKn9c6ytLx.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. Multilingual Machine Translation with Large Language Models: Empirical Results and Analysis. *ArXiv*, abs/2304.04675, 2023. URL https://arxiv.org/abs/2304.04675.

A Appendix: definition of data-science activities from Martínez-Plumed et al. (2019)

We reproduce here for convenience the definition of the data-science activities as used in Martínez-Plumed et al. (2019).

The original stages of the CRISP-DM (Cross Industry Standard Process for Data Mining, Chapman, 2000) framework are as follows:

- **Business Understanding**: Understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

- **Data Understanding**: Beginning with an initial data collection and proceeding with activities to familiarize oneself with the data, identify data quality problems, discover initial insights, and detect interesting subsets for hypothesis formation.
- **Data Preparation**: Encompassing all activities required to construct the final dataset from the initial raw data. This includes selecting tables, records, and attributes, as well as transforming and cleaning data for modelling.
- **Modelling**: Selecting and applying various modelling techniques while calibrating their parameters to optimal values. Some techniques have specific data requirements, often necessitating a return to the data preparation phase.
- **Evaluation**: Evaluating the constructed model(s) to ensure they properly achieve business objectives. The steps taken in the modelling process are reviewed to confirm that no important business issues have been overlooked.
- **Deployment**: Applying the model in a way that is useful for the customer, such as generating a report, implementing a repeatable data mining process, or integrating it into decision-making systems. While the customer typically executes deployment, the analyst ensures that all necessary steps are understood.

As argued in Martínez-Plumed et al. (2019), this framework assumes a well-defined business goal and pre-collected data. Additionally, it follows a fairly linear process, similar to mining metal in a known location. Thus, it is goal-oriented and process-centric, with data serving as an essential ingredient rather than the focal point. However, in exploratory data science, data takes centre stage, akin to prospecting rather than direct mining. Martínez-Plumed et al. (2019) introduces the following additional exploratory activities:

- **Goal Exploration**: Identifying business goals that can be achieved through data-driven approaches.
- **Data Source Exploration**: Discovering new and valuable data sources.
- **Data Value Exploration**: Assessing the potential value that can be extracted from the data.
- **Result Exploration**: Relating data science results to business goals.
- **Narrative Exploration**: Extracting meaningful stories, whether visual or textual, from data.
- **Product Exploration**: Identifying ways to transform extracted data value into services or applications that provide new and valuable benefits to users and customers.

Furthermore, Martínez-Plumed et al. (2019) critiques the CRISP-DM model for representing data as a static entity within the process, assuming that data has already been collected and merely needs understanding and preparation for modelling. However, modern data science projects often involve dynamic data management activities, including:

- **Data Acquisition**: Obtaining or generating relevant data, such as through the installation of sensors or applications.
- **Data Simulation**: Simulating complex systems to produce useful data and explore causal relationships (e.g., “what-if” scenarios).
- **Data Architecting**: Designing the logical and physical layout of data and integrating different data sources.
- **Data Release**: Making data accessible through databases, interfaces, and visualisations.

B Appendix: examples of tasks

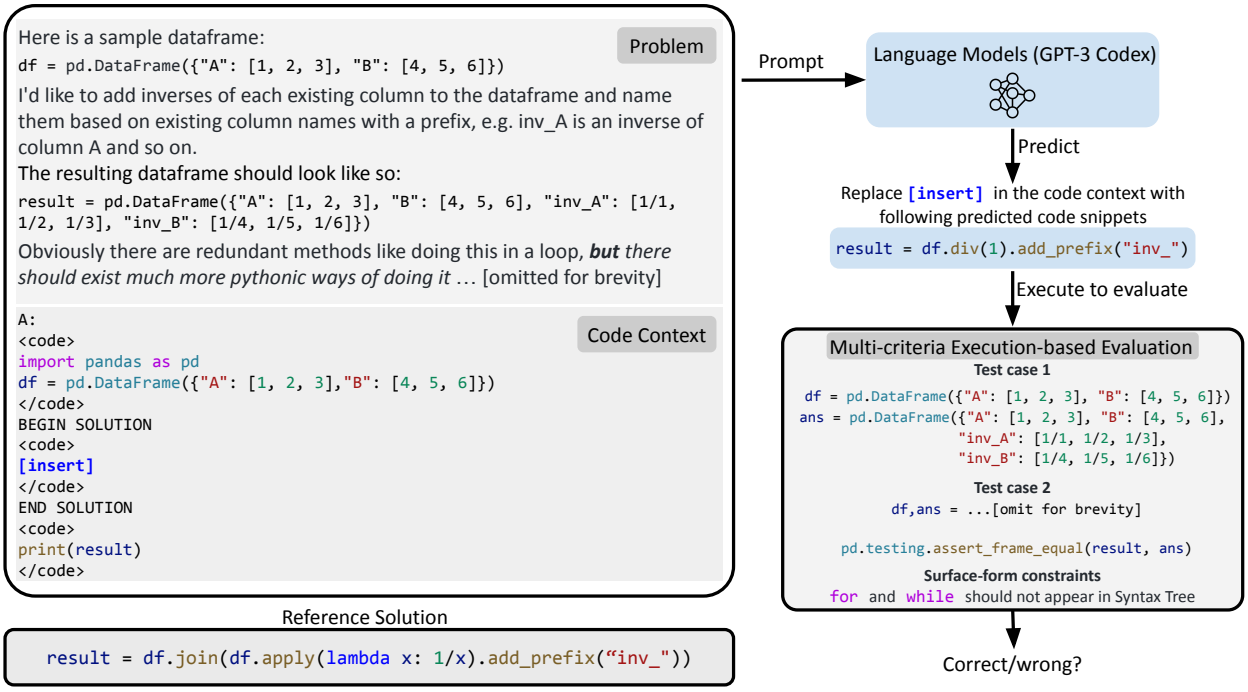

Figure 1: Lai et al. (2023): "An example problem in DS-1000. The model needs to fill in the code into [insert]in the prompt on the left; the code will then be executed to pass the multi-criteria automatic evaluation, which includes the test cases and the surface-form constraints; a reference solution is provided at the bottom left."

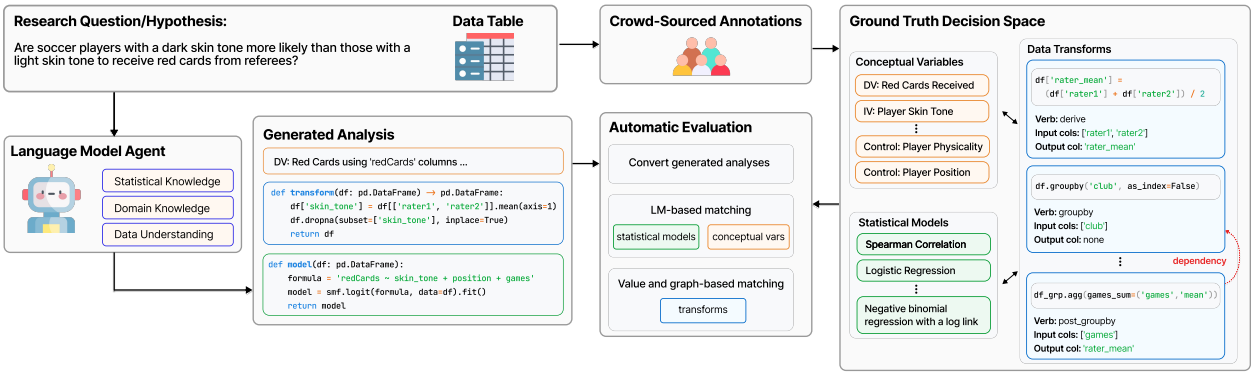

Figure 2: Gu et al. (2024): "Overview of BLADE. Given a research question and dataset, LM agents generate a full analysis containing the relevant conceptual variables, a data transform function, and a statistical modeling function (boxes 1-4-5). BLADE automatically evaluates this against the ground truth (box 6)."

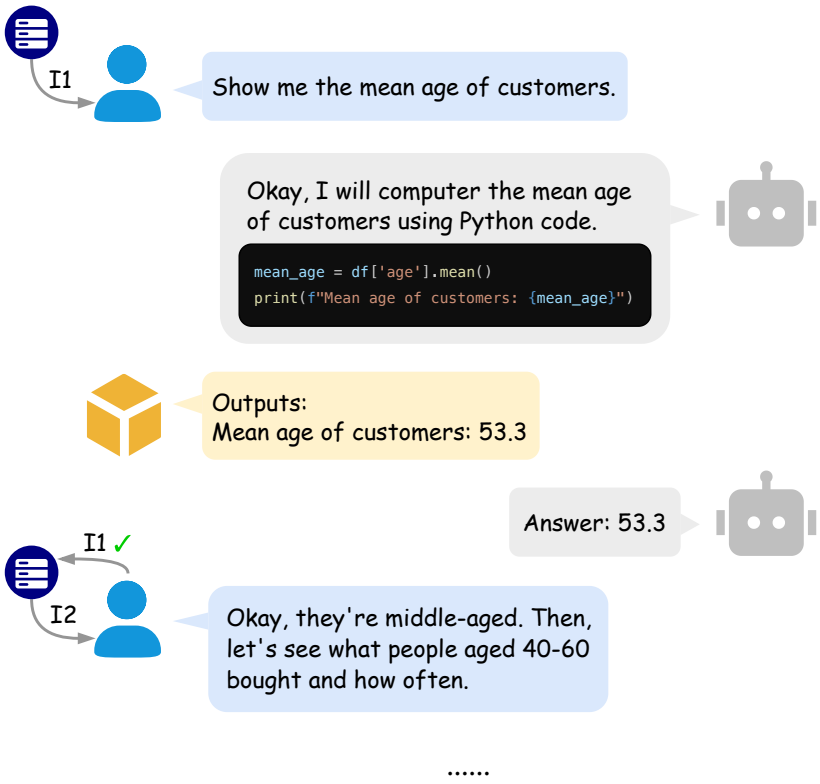

Figure 3: Li et al. (2025a): "Example task trajectory for Walmart sale prediction, showcasing the iterative interaction between the simulated user providing instructions and the agent executing code within the sandbox to achieve the analysis goal."

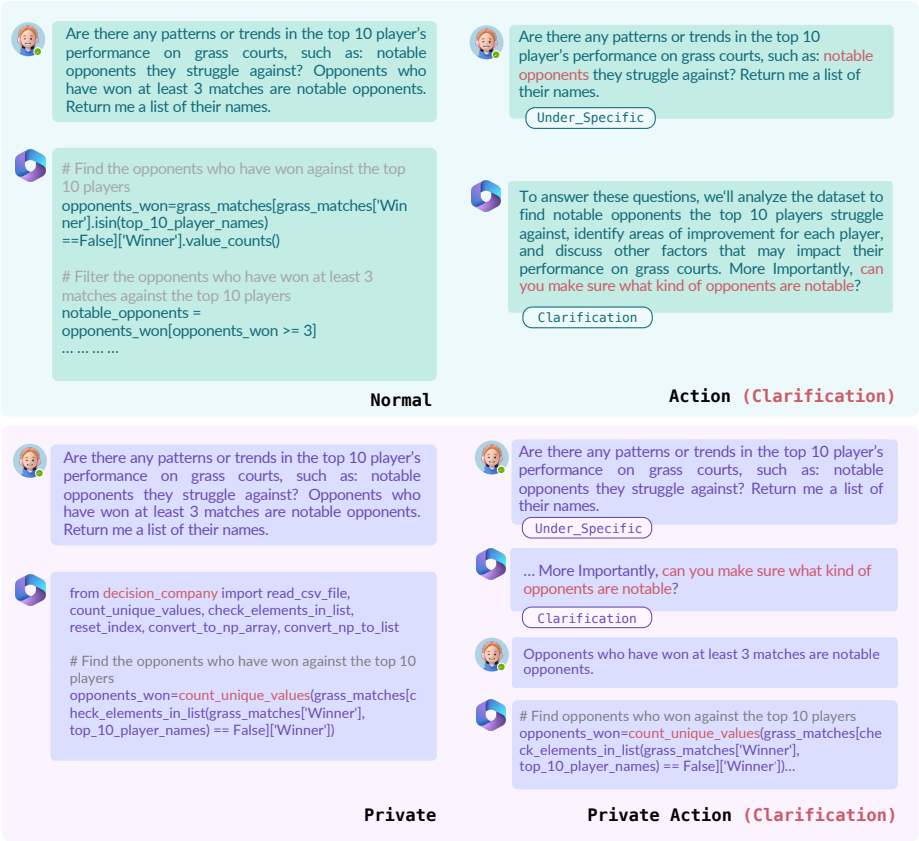

Figure 4: Li et al. (2024): The figure shows the "Normal" mode, with the agent being provided all the relevant information and tasked with writing code to address the task, and "Action" mode, where the agent has to take a specific action (in this case, asking for clarification). "Private" refers to tasks requiring the use of bespoke software libraries to which the agent has access to.

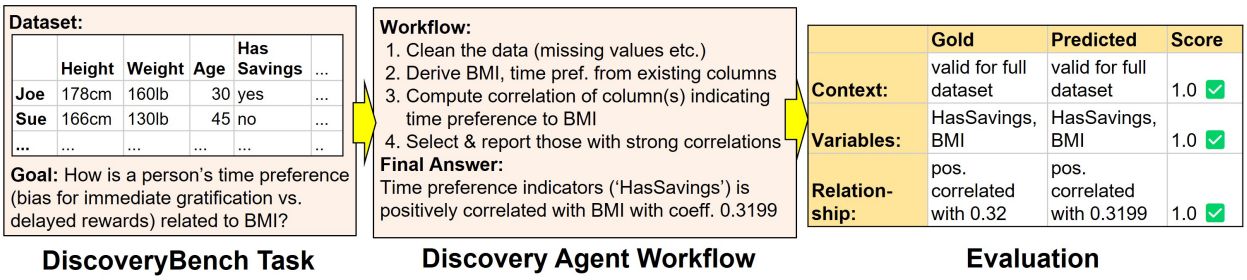

Figure 5: Majumder et al. (2024): "Each DISCOVERYBENCH task consists of a goal and dataset(s) (left). Solving the task requires both statistical analysis and scientific semantic reasoning, e.g., deciding which analysis is appropriate for the domain, and mapping goal terms to column names (center). A faceted evaluation allows open-ended final answers to be rigorously evaluated (right)."

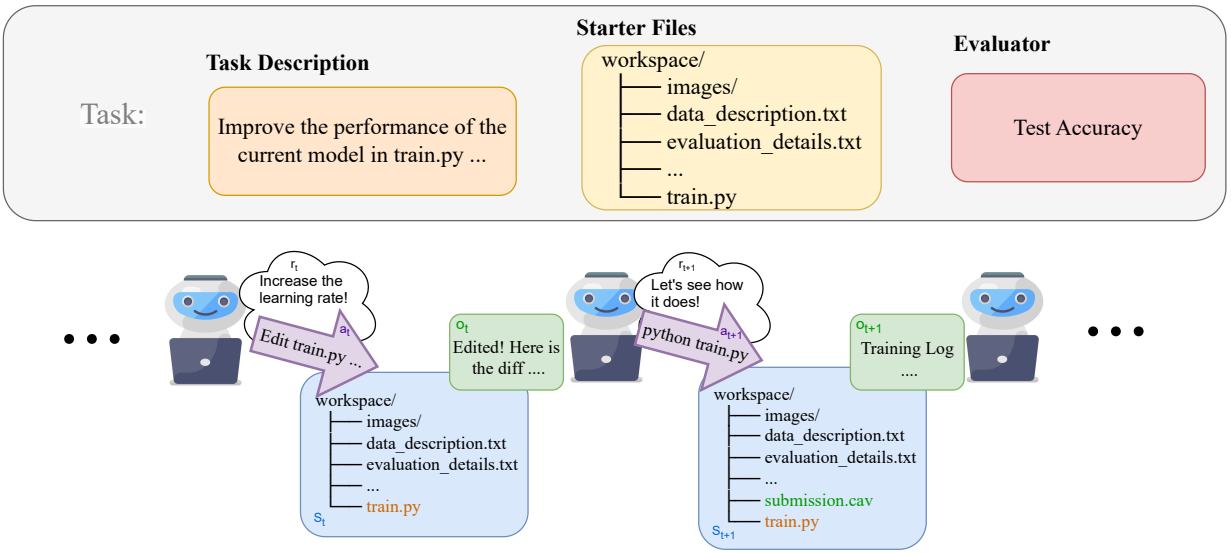

Figure 6: Huang et al. (2024a): "Overview of MLAgentBench. Each environment in MLAgentBench includes a task description, a set of starter files, and an evaluator. An agent can read/write files and execute Python code repeatedly, eventually producing a final file (e.g., test predictions in submission.csv). The agent is evaluated based on the quality of this file.."