POLITECNICO DI TORINO

MSc in Management Engineering

Benchmarking LLMs for Decision-Making in Project Management: Insights from a Company Context

Supervisors

Giovanni Zenezini

Filippo Maria Ottaviani

Candidate

Giulia Villa

March 2026

A chi mi ha lasciata volare,

senza paura di perdermi.

A chi è stato casa,

anche lontano da casa.

A chi mi aspetta,

sempre.

E a chi mi ha visto muovere i primi passi

e, da lassù,

continua a viaggiare con me.

Abstract

The rapid adoption of Large Language Models (LLMs) has generated growing interest in their potential application to Project Management, a domain where decision-making demands precision, operational efficiency, and adaptability. While general-purpose evaluation frameworks such as MMLU and HELM are widely employed, existing literature reveals a notable absence of systematic benchmarks tailored specifically to project management contexts. This thesis addresses this methodological gap by developing a comprehensive set of benchmarks designed to evaluate the reliability, computational efficiency, and practical utility of LLMs in managerial settings. The research is structured around two primary questions: (RQ1) which combinations of datasets, evaluation metrics, and prompting techniques facilitate the construction of meaningful benchmarks for project management tasks; (RQ2) which language model currently achieves the optimal balance among accuracy, computational speed, and economic cost. The overarching aim is to determine whether LLMs can function as valid tools for supporting managerial decision-making processes. To address these questions, a multi-tiered methodology was developed based on a "difficulty pyramid" dataset that progresses from single-choice questions through numerical problems requiring exact solutions, culminating in complex tasks that demand explicit step-by-step reasoning. The benchmarks incorporate various prompting approaches (Zero-Shot, Role Prompting, Chain-of-Thought) and assess multiple performance dimensions including accuracy, cost, response latency, token consumption, and reasoning transparency. The Analytic Hierarchy Process (AHP) was applied to integrate these heterogeneous metrics into a unified comparative index, though the survey-derived weights inherently reflect subjective stakeholder priorities. Experimental testing across eight state-of-the-art models uncovered systematic performance variations. GPT-5 and GPT-5 mini demonstrate the most robust and consistently high accuracy across the majority of benchmarks, albeit at the expense of substantially elevated computational costs and extended response times. Conversely, Gemini 2.5 Flash achieves a more favorable equilibrium among accuracy, cost-efficiency, and latency. Meanwhile, DeepSeek-V3.1, the Claude model family, and Gemini 2.5 Flash-Lite exhibit less stable performance patterns, though they remain competitive in terms of processing speed and reduced operational costs. A critical finding concerns prompting methodology. Implicit Chain-of-Thought prompting, achieved by adding *"Let's think step by step"* without mandating visible

reasoning, failed to enhance accuracy and occasionally diminished it, particularly in cognitively demanding tasks. By contrast, explicit reasoning (as evaluated in Benchmark 5) yielded marked improvements, demonstrating that transparency in the reasoning process strengthens model reliability. The comparison across question formats revealed that LLMs exhibit superior performance on single-choice tasks, where predefined response options serve as cognitive anchors, whereas they encounter greater difficulty with numerical problems requiring autonomous generation of correct values. Overall, this thesis establishes that LLMs can effectively support managerial decision-making within project management contexts, contingent upon adoption strategies informed by rigorous benchmarking that reconciles accuracy with operational efficiency. The work advances theoretical understanding by introducing a replicable, domain-specific evaluation framework and by proposing a systematic taxonomy for analyzing reasoning errors that differentiates between interpretation failures and planning deficiencies. From a practical standpoint, it provides actionable guidance for formulating queries that optimize reasoning quality and minimize errors, as well as for selecting models through informed trade-offs among accuracy, cost, and latency. Future research directions include extending this framework to additional professional domains, diversifying question types, developing interactive benchmark scenarios, and evaluating model robustness under dynamic and uncertain project management conditions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Existence

Modern projects are becoming more complicated, necessitating the integration of multiple tools and methodologies by managers to successfully address difficulties. This increasing complexity, paired with stringent time and financial limits, necessitates the exploration of novel solutions that can aid decision-making and enhance efficiency. In light of this, Large Language Models (LLMs) have recently gained traction in both academic research and managerial practice. Companies and organizations are experimenting with their usage in data analysis, forecasting, knowledge management, and automated decision-making processes. LLMs are particularly attractive in Project Management due to their potential to improve efficiency, predictive accuracy, and adaptability to changing circumstances. However, their actual performance in managerial settings remains to be fully established. Although LLMs promise quick, scalable, and flexible support, questions remain regarding their accuracy, robustness, and reliability in intricate, real-world decision-making. For managers, this raises a concrete dilemma: whether to implement these technologies without strong proof of where they deliver real added value, risking wasting both expense and time. For academics, the lack of defined and reproducible benchmarks tailored to project management contexts renders it impossible to objectively evaluate LLMs. These limitations underscore the need for research that can bridge empirical experimentation and methodological rigor.

1.2 Problem Importance

This topic has profound implications for both practitioners and researchers. For managers and companies, adopting Large Language Models without detailed evaluation may lead to wasteful investments, planning errors, cost overruns, and, ultimately, lost competitiveness. In dynamic and global supply chain scenarios, the ability to make accurate and fast decisions frequently determines success or failure. When an LLM makes errors in projections, simulations, or trade-off calculations, the economic and operational consequences can be severe.

For academics and researchers, developing methodologically robust benchmarks is critical to advancing comprehension of these models' real strengths and weaknesses in applied settings. Such benchmarks allow us to develop hypotheses about when generative technologies add value, establish which evaluation metrics are genuinely significant, and understand how prompting techniques affect performance.

In conclusion, solving this issue is essential to preventing expensive managerial errors and establishing a strong scientific basis for the ethical and successful application of large language models in Project Management.

1.3 Old and State-of-the-Art Literature Recap

From the early conception of Turing's test of machine intelligence (Russell & Norvig, 2016), artificial intelligence has developed into a vast field that currently encompasses generative AI, deep learning, and machine learning (Pahuja et al., 2025; Banh & Strobel, 2023).

Supervised, unsupervised, and reinforcement learning established the foundation for Neural Networks (Goodfellow, Bengio, et al., 2017), facilitating Deep Learning (Dol & Geetha, 2021) and the advent of Generative AI capable of generating realistic material (Banh & Strobel, 2023).

The Transformer architecture signified a pivotal transformation in natural language processing (Bengesi et al., 2023), resulting in the development of Large Language Models (LLMs) such as *BERT* and *GPT* (Haleem et al., 2022; Bengesi et al., 2023), subsequently refined through reinforcement learning from human feedback (RLHF) (Christiano et al., 2017), in-context learning (Brown et al., 2020), and prompt engineering (Clavié et al., 2023; White et al., 2023). Prompting techniques, including zero-shot (Wei et al., 2022), few-shot (Brown et al., 2020), chain-of-thought (Sivarajkumar et al., 2024), and ReAct (Yao, Zhao, et al., 2022), have enhanced reasoning powers; nonetheless, limitations such as bias (Ferrara, 2023; Schramowski

et al., 2022), hallucinations(Ji et al., 2023; Susarla et al., 2023), and lack of transparency (Janiesch et al., 2021; Meske et al., 2022) persist.

These advancements have expanded applications in IT (Kshetri et al., 2024), healthcare(S. Liu et al., 2023; Savage, 2023), marketing (Brand et al., 2023), and management sectors, including supply chain (Jackson et al., 2024) and project management (Prieto et al., 2023).

As Large Language Models' capabilities and usage rose, so did the need for systematic evaluation. Benchmarks give objective and repeatable assessments of performance, identifying strengths, weaknesses, and hazards across tasks and domains (Chang et al., 2024).

Benchmarks ranged from general-purpose datasets like *MMLU* (Hendrycks et al., 2021), *AGIEval* (Zhong et al., 2023), and *HELM* (Liang et al., 2023) to reasoning-focused tasks like *HotpotQA* (Yang et al., 2018), *2WikiMultiHopQA* (Ho et al., 2020), and *FanOutQA* (Zhu et al., 2024), as well as domain-focused frameworks like *EconLogicQA* (Quan & Z. Liu, 2024), *FinEval* (Guo et al., 2025).

Other contributions dealt with organizational contexts, with benchmarks for inventory management (Z. Li et al., 2024) and business process management (Busch & Leopold, 2024), and conversational quality, with *LLM-EVAL* (Lin & Y.-N. Chen, 2023) rating open-domain dialogues across several dimensions.

Despite these gains, challenges remain, ranging from prompt sensitivity (Ferrara, 2023) and benchmark gaming (Balloccu et al., 2024) to linguistic narrowness (Mushtaq et al., 2025) and a lack of standardized documentation (McIntosh et al., 2024).

Recent research reveals both significant gains and ongoing limitations in Large Language Models. Frontier models have outstanding capabilities, but they still struggle with complicated reasoning, domain transfer, and extended context management (Guo et al., 2025; Lunardi et al., 2025).

While advancements like Multi-Agent reasoning (P. Chen et al., 2024) and Chain-of-Thought prompting (Wei et al., 2022) help to minimize some of the shortcomings, they are still limited by scale and design.

These findings underscore the significance of transitioning to transparent, context-aware evaluation frameworks that are matched with real-world managerial targets, ensuring that improvements in LLM performance transfer into actual benefit in domains such as Project and Supply Chain Management.

1.4 Gap

Despite the rapid evolution of Generative Artificial Intelligence and the growing adoption of Large Language Models in organizational contexts such as Supply Chain Management and Project Management, the existing literature still lacks systematic benchmarks tailored to these domains.

This gap is an important obstacle because without domain-specific benchmarks, it is not possible to accurately determine whether Large Language Models can support decision-making processes, enhance forecasting reliability, or contribute to the reduction of project delays and cost overruns in supply chains.

Moreover, the absence of structured evaluation frameworks constrains both theoretical progress in learning how these models work in managerial contexts and practical recommendations for firms considering their applications.

Therefore, further investigation is needed to develop and apply dedicated benchmarks that capture the distinctive needs of Supply Chain and Project Management, allowing robust comparisons among models and supporting their effective and responsible incorporation into business operations and practices.

1.5 Objective(s)

This thesis aims to determine whether Large Language Models can be considered reliable tools for facilitating managerial decision-making in Project Management.

The objective is to analyze findings not solely based on technical performance, but also by assessing factors that represent the real operational needs of managers, to see whether these models can offer solid support in both strategic and routine decision-making.

To achieve this goal, the study develops and implements objective benchmarks for evaluating Large Language Models in this domain. Every design decision in the creation of the benchmarks is explicitly justified to ensure transparency and replicability, allowing the results to be independently reproduced. The investigation is guided by two distinct research questions:

- RQ1: Which combinations of datasets, evaluation metrics, and prompting techniques enable the development of valuable benchmarks to assess LLM performance in Project Management settings?

- RQ2: Which Large Language Model currently exhibits the best overall performance, offering a comparative framework to guide managerial choices?

Finally, the study closes methodological gaps and offers researchers and managers a trustworthy reference point for assessing the function of Large Language Models in the Project Management domain.

1.6 Structure

This thesis is organized as follows. Chapter 1 introduces the research problem, discussing its existence and importance, providing a brief recap of the previous state of the art, highlighting existing gaps, and presenting the objectives of the study. Chapter 2 presents a comprehensive literature review, covering the emergence of Generative AI and Large Language Models (LLMs), as well as the current approaches and challenges in benchmarking these models. Chapter 3 details the research methodology, including the formulation of research questions and the exploratory framework, the construction of domain-specific benchmarks, covering datasets, evaluation metrics, prompting techniques, and the creation of final benchmarks, and the implementation and testing of benchmarks, including the selection of LLMs and statistical significance testing. Chapter 4 reports the results, presenting both benchmark-level performance and cross-benchmark comparisons, and also includes the findings from a survey conducted. Chapter 5 discusses the results, highlighting their theoretical and practical implications/the contribution to the theory and practice. Lastly, Chapter 6 summarizes the study, outlines its (de)limitations, and suggests possible streams of future research.

Chapter 2

Literature Review

2.1 Introduction to Generative AI

This section outlines the evolution of Artificial Intelligence, tracing its development up to the emergence of contemporary Generative AI. The concept of Large Language Models will be introduced, with particular attention to prompting techniques and the current limitations. Finally, the section will conclude with an examination of Generative AI and Large Language Model applications in organizational contexts, with a specific focus on *Supply Chain* and *Project Management*.

2.1.1 From Artificial Intelligence to Generative AI

The term *Artificial Intelligence* (*AI*) refers to a machine's ability to perform tasks that would typically require human cognitive abilities (Gignac & Szodorai, 2024), including language comprehension, complex pattern recognition, experiential learning, and autonomous decision-making (Banh & Strobel, 2023; Winston, 1993). In 1950, Alan Turing introduced a test to determine whether a machine is capable of exhibiting intelligent behavior (Russell & Norvig, 2016). According to his operational criterion, a machine is deemed intelligent if, when interacting through written language, it is able to convince a human interlocutor that they are conversing with another human being (Jiang et al., 2022).

Over time, artificial intelligence has evolved and is now an umbrella term encompassing various subfields and methodologies (Pahuja et al., 2025; Banh & Strobel, 2023)(Figure 2.1).

Machine Learning (*ML*) is a part of this model, and it is now widely recognized as one of the foundational pillars of modern Artificial Intelligence (Lv, 2023). ML focuses on the

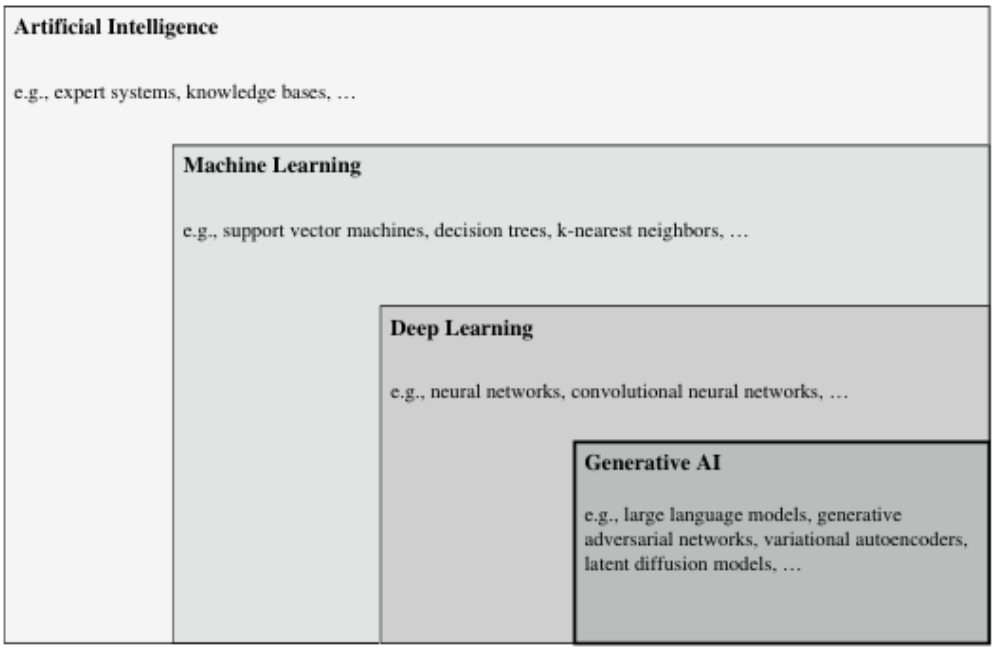


Figure 2.1: Generative AI and other AI concepts (Banh & Strobel, 2023).

development of algorithms that can identify patterns in data and improve their performance over time, without the need for explicit reprogramming for each new task (Brynjolfsson & Mitchell, 2017; Dol & Geetha, 2021). This ability to generalize from experience allows AI systems to adapt to dynamic environments and tackle complex, data-driven problems across various domains (Lv, 2023).

Machine learning methods can be categorized into different types, depending on the nature of the training data and the specific objectives of the algorithm (Mohri et al., 2012). The most important method in ML is *Supervised Learning*. In this approach the algorithm is trained on a labeled dataset: each input instance is associated with a corresponding output label (Cunningham et al., 2008). The model learns the mapping between inputs and outputs, and it is then able to make predictions on new unseen test data. This is the most frequently applied approach in tasks such as classification, regression, and ranking (Mohri et al., 2012). Unfortunately, *Supervised learning* by definition relies on a human supervisor to provide an output example for each input example. Due to this, many researchers have shifted their focus toward studying *Unsupervised learning* (Goodfellow, Pouget-Abadie, et al., 2020). In this case the data provided to the model are unlabeled. The algorithm relies on its internal mechanisms to autonomously identify patterns or correlations within the data (Dol & Geetha, 2021). This type of learning is often used for tasks such as clustering (Tyagi et al., 2022) but it's also used in generative

modelling (Goodfellow, Pouget-Abadie, et al., 2020). Lastly, in *Reinforcement Learning*, the algorithm, referred to as an agent, interacts with an environment and learns through a system of rewards and penalties, following a trial-and-error process (Pahuja et al., 2025; Mohri et al., 2012).

These learning paradigms provide the foundation for the implementation of *Artificial Neural Networks*, computational models inspired by the structure of the human brain (Goodfellow, Bengio, et al., 2017). When these models consist of multiple hidden layers, the approach is referred to as *Deep Learning* (*DL*) (Figure 2.2). Deep Learning (DL) is a subfield of Artificial Intelligence (AI) that enables systems to learn and classify objects by interpreting data in a manner inspired by the human brain. It is particularly effective in making predictions and informed decisions based on current data (Dol & Geetha, 2021).

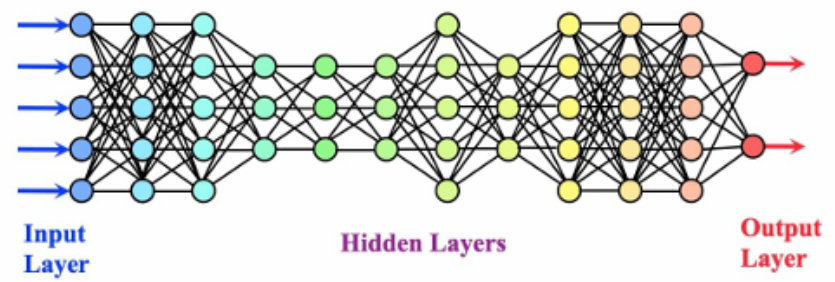


Figure 2.2: Example of Deap Neural Network (Horzyk et al., 2023).

The evolution of Deep Neural Networks, supported by advances in computational power and the availability of large-scale datasets, has enabled the development of increasingly sophisticated models (Lecun et al., 2015). In recent years, this progress has paved the way for the rise of *Generative Artificial Intelligence* (*GenAI*), which represents a fundamental shift from a purely predictive and discriminative paradigm toward a generative one (Banh & Strobel, 2023). In this new paradigm, the objective is not merely to analyze or classify data, but to autonomously and realistically generate novel content.

2.1.2 Generative AI and LLMs

Over the years, the shift of scientific interest from discriminative to generative models has fostered the development of numerous architectures that have transformed fields such as natural language processing and the generation of images and videos (e.g., *VAE, GAN, diffusion models* and *Transformer*) (Bengesi et al., 2023; Pahuja et al., 2025). The *Transformer* architecture, in

particular, signaled a significant change in the field of Generative Artificial Intelligence.

Transformers, which were first presented by a group of Google researchers under the direction of Vaswani in the 2017 paper *"Attention Is All You Need"* (Vaswani et al., 2017), have revolutionized the state-of-the-art in a variety of tasks, particularly in *Natural Language Processing (NLP)* (Bengesi et al., 2023). The innovation of the Transformer lies in its *attention* and *self-attention* mechanisms (Shen et al., 2023), which enable it to evaluate the importance of various input sequence elements, such as words in a sentence or pixels in an image, in a similar way to how people concentrate on particular words when attempting to comprehend a sentence (Bengesi et al., 2023; P. Chen et al., 2024).

The success of this model became evident with the introduction of architectures such as *BERT (Bidirectional Encoder Representations from Transformers)* (Devlin2018) developed by researchers at *Google*, and *GPT (Generative Pre-trained Transformer)* by *OpenAI* (Haleem et al., 2022; Bengesi et al., 2023). These models, more generally, belong to the family of *Large Language Models (LLMs)*, which refers to large pre-trained transformer models that are typically trained for prediction tasks, where the objective is to predict the next word given some textual input (Pahuja et al., 2025; Chang et al., 2024).

Beyond the self-attention mechanism, the progressive evolution of Large Language Models has been accompanied by the introduction of several key innovations that have significantly enhanced their capabilities. Among these, *Reinforcement Learning from Human Feedback (RLHF)* has played a particularly important role. By incorporating human judgments into the *fine-tuning* process, this approach allows guiding the model's behavior more precisely, helping to align its outputs with human preferences and expectations (Christiano et al., 2017).

Equally relevant is the development of *in-context learning*, a capability that allows LLMs to perform complex tasks without the need for additional training. Instead, the model learns to interpret and respond appropriately to the information provided within the prompt itself, demonstrating an impressive ability to generalize across tasks simply by leveraging contextual cues (Brown et al., 2020).

Finally, the emergence of *prompt engineering* has transformed the way users interact with these systems. Rather than writing code, users can now shape model behavior through carefully crafted natural language inputs. In this sense, prompt engineering represents a new kind of programming, one that is accessible and intuitive, yet capable of eliciting highly sophisticated outputs from the model (Clavié et al., 2023; White et al., 2023).

2.1.3 LLMs Prompt Techniques

In the following, some prompting techniques taken from the literature will be discussed.

Zero-Shot

The *Zero-Shot* prompt is the simplest type of prompt (Wei et al., 2022). It consists of providing the model with only a textual description of the task to be performed, without including explicit input-output examples (Sivarajkumar et al., 2024). In this approach, the LLM relies only on its pre-trained knowledge to interpret and complete the task (Reynolds & McDonell, 2021). Some researchers (Reynolds & McDonell, 2021) have shown that well-designed zero-shot prompts can achieve strong performance, sometimes even outperforming Few-Shot prompts (Sivarajkumar et al., 2024). However, in tasks such as comprehension of the language, answering questions, and inference of natural language, Few-Shot prompting generally leads to better performance (Wei et al., 2022).

One-Shot and Few-Shot

When designing prompts for LLM models, it can be advantageous to incorporate clear examples within the input provided (Reynolds & McDonell, 2021). In a *One-Shot* prompt, the model is given a single illustrative example of the task, followed by a new instance to solve. The *Few-Shot* prompt, by contrast, involves presenting the model with several examples, typically ranging from two to five or more, before the test prompt (Brown et al., 2020). These examples help to establish context and are particularly valuable in handling more complex tasks (Sivarajkumar et al., 2024). They are especially effective when aiming to guide the model toward a particular format or structure in its output. In fact, research shows that providing examples that closely align with the nature of the target task improves the performance of the model (Y. Li, 2023).

Role Prompting

This technique involves explicitly assigning a role to the model, instructing it to act as, for example, a professor, an expert, or a student (Kong et al., 2023). The role context helps the model adjust the tone, style, and level of expertise in its responses. Assigning a functional identity to the LLM is an effective way to guide the model's behavior toward answers that are

more relevant and consistent with the intended communicative goal (Zhao et al., 2025).

Chain of Thought (CoT)

The *Chain-of-Thought (CoT)* technique is based on explicitly prompting the model to break down a problem into successive logical steps, thereby simulating a step-by-step reasoning process. It is particularly useful for tasks that require deduction, calculations, or multi-step problem solving (Sivarajkumar et al., 2024). Making the reasoning chain explicit not only enhances the transparency of the reasoning process but also allows for the diagnosis of potential intermediate errors. The CoT technique has often been associated with Few-Shot prompting (Wei et al., 2022) and, more recently, with Zero-Shot prompting (Kojima et al., 2022). In the *Zero-Shot CoT* the prompt is augmented with a simple instruction such as *"Let's think step by step"*, without providing specific examples. This minimal modification has proven surprisingly effective in improving model performance in the absence of additional data (Y. Li, 2023). (Figure 2.3)

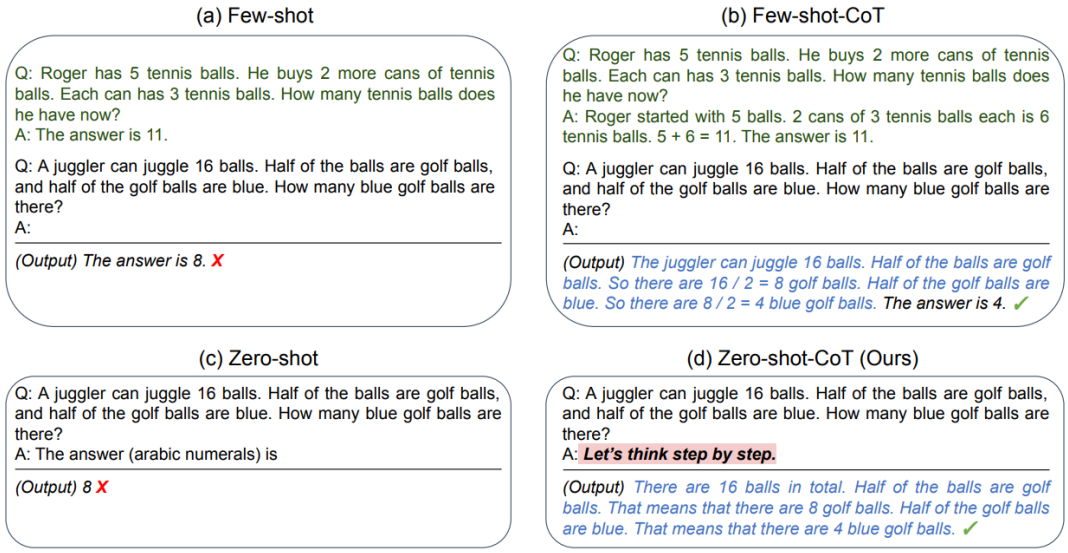


Figure 2.3: Positivie impact of the CoT prompting technique in Zero-Shot and Few-Shot cases (Kojima et al., 2022).

Self-Consistency

This strategy addresses the variability in the outputs generated by LLMs through a process of multiple sampling. The model is executed several times with the same prompt, producing different reasoning paths. Among the various responses obtained, the most frequent or most

consistent one is selected. This mechanism leverages the principle that correct reasoning paths tend to converge towards the same solution, whereas incorrect ones produce more dispersed outcomes (X. Wang et al., 2022). Based on the intuition that complex tasks can be solved through multiple reasoning pathways leading to a correct outcome (Stanovich & West, 2000), this technique is frequently combined with Chain-of-Thought prompting to address complex problems.

Tree of Thoughts (ToT)

Going beyond the linearity of the Chain of Thought, the *Tree of Thoughts* technique enables the model to explore multiple reasoning branches simultaneously. Each *"thought"* is treated as a node within a logical tree, from which new trajectories may emerge. This approach is particularly well suited to solving complex, open-ended problems, where the deliberate exploration of alternatives enhances the quality of the final decision(Yao, Yu, et al., 2023).(Figure 2.4)

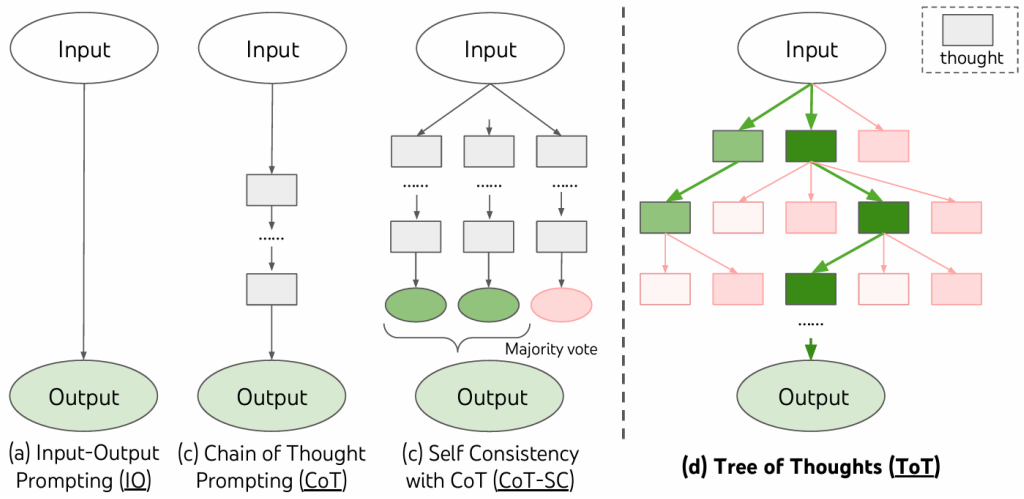


Figure 2.4: Comparison of various approaches to problem solving with LLMs (Yao, Yu, et al., 2023).

Reason e Act (ReAct)

The *ReAct* technique combines linguistic reasoning with the execution of actions. In this framework, the model alternates between phases of reasoning and operational phases (acting), such as consulting external sources or interacting with digital tools. This paradigm, which mirrors human behavior in problem solving, is one of the foundational components of recent LLM-based

agents, enabling them to interact dynamically with their environment to complete complex tasks(Yao, Zhao, et al., 2022).

2.1.4 Limitations

Although Large Language Models have reached significant milestones in recent years, they still present limitations that compromise the overall quality of their outputs. The most critical issues include *bias*, the risk of *hallucinations*, and the *lack of transparency and explainability* in their decision-making processes.

Bias

The performance of Generative AI systems is strongly influenced by the quality of the training data. As highlighted in the literature, GenAI models are prone to bias causing biased decisions, disadvantages, and discriminations (Ferrara, 2023; Schramowski et al., 2022). Such biases may emerge during the training phase, due to datasets that are non-representative, imbalanced, or incorrectly labeled, but can also appear during inference, when algorithmic choices such as overfitting introduce distortions not present in the original data. These dynamics make it challenging to ensure fairness and reliability in different applications (Banh & Strobel, 2023).

Hallucinations

A recurring limitation of LLMs is their tendency to produce hallucinations, namely outputs that are coherent and convincing but factually incorrect. *'Hallucinations [...] manifest themselves in confidently generated results that seem plausible but are unreasonable with respect to the source of information'*(Ji et al., 2023; Susarla et al., 2023). This phenomenon is mainly related to the probabilistic nature of generative models and to the use of training data containing contradictory or unreliable information (Dziri et al., 2022). The result is text that may deviate from reality, thus reducing user trust in the reliability of the system (Banh & Strobel, 2023; Pahuja et al., 2025).

Lack of Trasparency and Explainability

A further challenge is represented by the opacity of these systems. ML models function as black boxes (Janiesch et al., 2021; Meske et al., 2022), since it is rarely possible to trace how a given

output was produced. This absence of interpretability prevents users from fully validating or understanding model behavior, which is particularly critical in areas where accountability and decision traceability are required (Banh & Strobel, 2023).

2.1.5 Applications in Supply Chain and Project Management

Over the past few years, Generative Artificial Intelligence, and especially Large Language Models, has significantly reshaped the way organizations work. The ability of these technologies to combine analytical capabilities, predictive modeling, and creativity enables the automation of repetitive tasks, the improvement of output quality, and the reduction of execution times (Pahuja et al., 2025; Banh & Strobel, 2023).

From a business and industry perspective, applications cover a wide range of use cases. In the software and IT sector, tools such as *GitHub Copilot*, powered by OpenAI Codex, help developers write code, reducing completion times by up to 56% (Pahuja et al., 2025). In digital services, *Microsoft Bing* integrates ChatGPT to provide contextual responses in web searches, while in the marketing domain, GenAI is used for the generation of personalized content and offerings and the optimization of the sales lead generation process (Kshetri et al., 2024). In the financial sector, applications range from automated analysis of financial statements and transactions to the generation of forecasts for the stock and currency markets (George et al., 2023). LLM and GenAI also play an important role in the healthcare sector, supporting medical imaging diagnostics, the discovery of new drugs, and patient communication, thus contributing to the reduction of development times for therapies and clinical protocols (S. Liu et al., 2023; Savage, 2023).

In addition to these cross-sector applications, GenAI and LLMs are increasingly being applied in domains with high managerial complexity, such as Supply Chain Management and Project Management.

Supply Chain

In the field of Supply Chain and Operations Management (SCOM), Generative AI is demonstrating transformative potential in multiple decision-making areas. According to the framework proposed by Jackson et al. (2024), the capabilities of *learning*, *perception*, *prediction*, *interaction*, *adaptation*, *reasoning*, and *creativity* offered by GenAI can be applied in at least thirteen strategic domains, including *demand forecasting, inventory management, supply chain*

design, production planning and control, quality management, and *supply chain risk management*. (Figure 2.5)

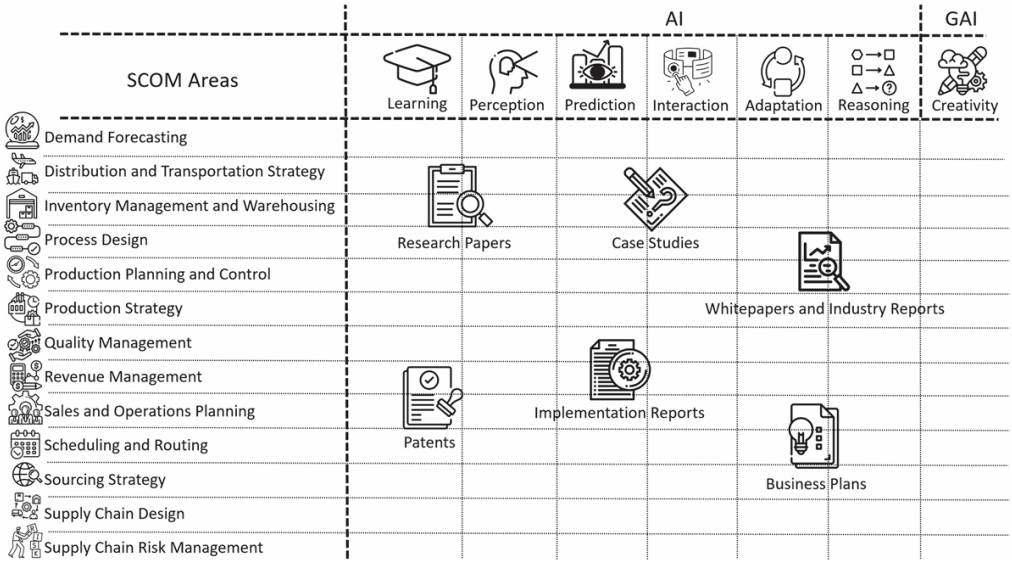


Figure 2.5: SCOM areas (Jackson et al., 2024).

In this regard, Skórnóg & Kmiecik (2023) demonstrate how ChatGPT can be employed for material forecasting in the manufacturing sector, in some cases achieving more accurate results than commonly used models for demand forecasting in business operations, such as ARIMA.

Other examples concern organizations that have integrated GenAI into their systems. *Walmart* has adopted Pactum AI, a generative chatbot-based system, to automate supplier negotiations (Hoek et al., 2022); *Maersk* has implemented GenAI to optimize logistics planning and improve resilience (Handley, 2023); while *DHL* is experimenting with ChatGPT to automate communications and warehouse operations (Moller, 2023). *Instacart* employs a conversational assistant powered by OpenAI to facilitate orders and personalize recommendations (Zhuang, 2023), and *Amazon Business* leverages AI models to analyze purchasing data and suggest more cost-effective alternatives.

Finally, another emerging development concerns the integration of LLM with optimization systems, as exemplified by *Microsoft's OptiGuide* framework (Y. Li, 2023), which translates requests in natural language (e.g., *'What happens if I use supplier B instead of A?'*) into queries for mathematical solvers, returning intelligible results and intuitive visualizations. This approach facilitates communication between planners and complex systems, enhancing decision-making transparency and reducing response times (Y. Li, 2023).

Project Management

In Project Management, although LLMs such as ChatGPT are not yet ready to replace professional software, studies such as that by Prieto et al. (2023) highlight their usefulness in generating coherent project plans and rapidly adjusting operational sequences in response to changing requirements. A notable example is the LLM-Project initiative (Zhen et al., 2024), in which LLMs, trained on Standard Operating Procedures (SOPs) and simulated data, were able to produce *Work Breakdown Structures (WBS)* complete with time coding (*Finish–Start*, *Start–Start* relationships) and resource allocation.

Further insights are provided by the study of Cinkusz et al. (2024), which introduces *CogniSim*, a framework that integrates *cognitive agents* powered by Large Language Models within the *Scaled Agile Framework (SAFe)* to strengthen software project management. Simulations revealed measurable improvements in various metrics, including task completion times, quality of deliverables, and communication coherence.

Looking ahead, the strategic adoption of GenAI and LLMs in Supply Chain and Project Management goes beyond improving operational efficiency: it paves the way for more resilient, transparent, and adaptive supply chains, where human–machine collaboration becomes a key driver of competitiveness.

Such widespread applicability, however, calls for a careful assessment of its ethical, security, and labor-related implications. While GenAI can boost productivity and create new professional roles (e.g., prompt engineers), it also introduces risks associated with data quality, the protection of sensitive information, and the potential replacement of low-skilled jobs (Einola & Khoreva, 2023).

2.2 Introduction to Benchmarking

This section covers the evolution of benchmarking from its origins to its widespread adoption across fields such as computing, finance, and management. It outlines the key lifecycle stages and discusses the fundamental principles that underpin the design of high-quality benchmarks.

2.2.1 Benchmark Definition

The term *benchmark* comes from measurement science, where it originally referred to a physical mark used as a reference point for leveling operations in geodesy(Zairi & Leonardo, 1996).

Over time, this concept evolved into a broader idea of a standardized reference for performance comparison, and has since been adopted across several disciplines, including computing, finance, and management (Zhan, 2022).In computer science, the first formal benchmarks were introduced in the early 1960s by the Auerbach Corporation to measure system speed through predefined routines. A primary limitation of these initial benchmarks was that their findings were not acquired through direct execution on the systems under examination, but instead derived from performance metrics published by vendors, so diminishing their impartiality and comparability (B. C. Lewis & A. E. Crews, 1985; Zhan, 2022).

The initial step in this endeavor was workload modeling, which entailed choosing a representative subset of programs from the diverse array of tasks commonly performed by users. The concept was that, by concentrating on a meticulously selected sample, one might emulate the overall behavior of real workloads while maintaining a manageable review process.Then, to compare performance on real-world jobs, researchers suggested application benchmarks, which are real programs running on different systems. Although they were more representative than abstract metrics, they were expensive and difficult to apply across diverse architectures(B. Lewis & A. Crews, 1985). The idea of synthetic benchmarks was developed in order to overcome these restrictions. Instead of running complete apps, synthetic benchmarks created smaller programs that mimicked the key functions of actual applications. These benchmarks allowed for more realistic and economical system comparisons by removing specifics while maintaining performance-critical features(Y. Liu, Khandagale, et al., 2021).

Together, these approaches established the foundation for performance evaluation in computing and continue to influence modern benchmark design. In parallel, the concept of benchmarking took root in the management sector. Xerox Corporation pioneered competitive benchmarking in the late 1970s, systematically analysing competitors' products, processes, and organizational practices to identify and adopt superior methods(Zairi & Leonardo, 1996). Over time, this evolved into a broader quality improvement strategy based on comparing internal operations with industry best practices. Across disciplines, the benchmark has emerged as an important scientific and engineering tool: it defines quantifiable objectives, establishes standard conditions, and allows for consistent performance comparison(Zairi & Leonardo, 1996).

2.2.2 Benchmark Design

According to the literature, a high-quality benchmark must go through four critical lifecycle stages: *design, implementation, documentation, and maintenance*(Figure 2.6).

Insights from domains such as transistor hardware, environmental science, and bioinformatics identify four fundamental characteristics of good benchmarks (Reuel et al., 2024).

- First, tasks should be planned for downstream *utility*, reflecting real-world conditions and use cases.
- Second, to ensure *validity*, benchmarks should use large test sets, avoid bias from gold standards, and be periodically updated to prevent overfitting (Y. L. Liu et al., 2024; Reuel et al., 2024).
- Third, *score interpretability* requires benchmarks to clearly define their purpose, scope, and procedures, avoiding misleading or absolute statements.
- Finally, *accessibility* promotes reproducibility through open data and code (Bartz-Beielstein et al., 2020; Reuel et al., 2024).

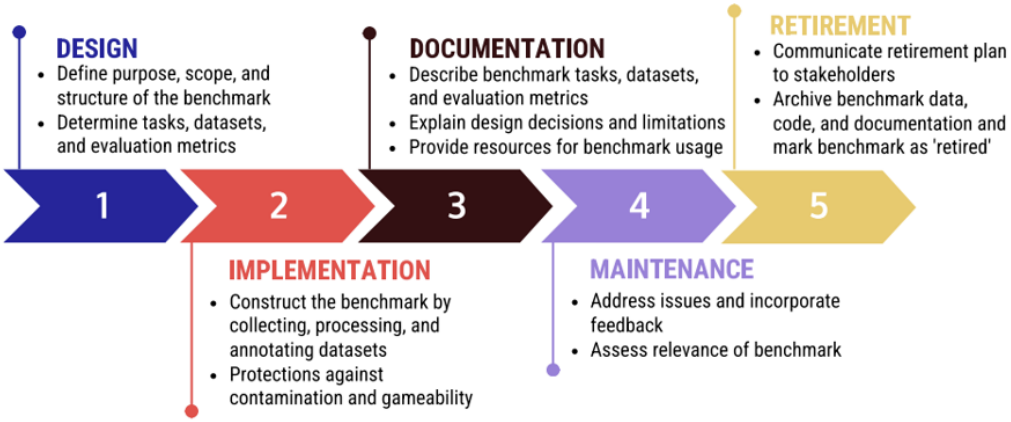


Figure 2.6: Five stages of the benchmark lifecycle (Reuel et al., 2024).

2.3 Benchmarking Large Language Models (LLMs)

As Large Language Models (LLMs) are increasingly utilized in fields such as education, healthcare, marketing, and finance, apprehensions about their reliability and influence have escalated reinforcing the need for rigorous and systematic evaluation recognized by both academic and

industrial stakeholders (Chang et al., 2024). Benchmarks fulfill this purpose by giving objective and repeatable performance measurements, allowing for the identification of strengths, shortcomings, and potential risks. Effective evaluation goes beyond simply assessing accuracy; it provides insights for optimizing human–AI interaction workflows, establishes safeguards for deployment in domains such as healthcare, and verifies system robustness in specialized tasks where errors may have high costs. For example, in market research, LLMs might complement traditionally high-priced methods such as conjoint studies, which assess how consumers value different product attributes through trade-off analysis, or focus groups, allowing for rapid, cost-effective, and iterative testing of marketing or pricing strategies prior to product launch(Brand et al., 2023). As models develop in size and ability, benchmarks must evolve to include not only task-specific skills, but also resilience, trustworthiness, and domain relevance(Busch & Leopold, 2024).

This chapter investigates how the evaluation of LLMs has been addressed in the literature. First, it covers the main types of tasks and datasets that are commonly used for benchmarking. Second, it examines the evaluation metrics employed to measure model performance. Third, it summarizes the findings from existing benchmark studies. Finally, it analyzes the limitations and open challenges discovered in several contributions.

2.3.1 Task Types and Datasets

Task Types

Benchmarks in Natural Language Processing have traditionally focused on generic and constrained tasks(Busch & Leopold, 2024). These include question answering, where models are asked to provide accurate answers based on a given passage or dataset; sentiment analysis, which assesses the ability to identify the emotional tone of a text, such as positive or negative reviews (Kumar et al., 2023); and natural language inference, which evaluates whether a model can determine whether one sentence logically follows from another (Miralles-González et al., 2025).

While these standardized exercises have helped to assess development, they do not fully capture the complexities of how LLMs are used in everyday or domain-specific contexts(Miller & Tang, 2025).This gap, evident when models score highly on benchmark datasets but underperform in real-world applications requiring contextual adaptation (Kiela et al., 2021), has

driven the creation of more sophisticated benchmarks designed to stress-test reasoning, interaction, and applied knowledge.

The most popular general-purpose benchmarks are *MMLU* (Hendrycks et al., 2021), which uses almost exclusively multiple-choice questions to assess knowledge in 57 academic and professional subjects; *AGIEval* (Zhong et al., 2023), which draws on standardized exams and employs multiple-choice and fill-in-the-blank formats; and *HELM* (Liang et al., 2023), which utilizes a combination of multiple-choice, short-answer, and free-text tasks to provide a more comprehensive assessment. These formats were specifically intended to minimize subjectivity and guarantee reproducible scoring, with multiple-choice and cloze questions providing unambiguous correctness standards, while free-text tasks add more open-ended evaluation to capture broader model abilities.

Other datasets explore more difficult goals beyond these all-purpose benchmarks, expanding on preexisting frameworks.*HotpotQA* (Yang et al., 2018) and *2WikiMultiHopQA* (Ho et al., 2020), for instance, examine whether models can respond to queries that call for integrating fragments of data from several sources rather than depending solely on a single finding. By constructing reasoning paths that are longer and less linear, *FanOutQA* (Zhu et al., 2024) makes this process even more difficult, requiring models to pass through an average of seven intermediate steps before arriving at the right answer.

Moreover, new benchmarks have been developed to assess performance in specific domains. Using multiple-choice questions to capture consistency in economic logic, *EconLogicQA* (Quan & Z. Liu, 2024) assesses sequential thinking in economics by asking models to predict and order interconnected economic events across numerous situations.The finance sector is the topic of *FinEval* (Guo et al., 2025), which assesses LLMs' proficiency in handling domain-specific knowledge and reasoning tasks using both multiple-choice and real-world case-based scenarios that mimic financial decision-making.

In addition to domain-specific reasoning, conversational quality has been an important area of evaluation. *LLM-EVAL* (Lin & Y.-N. Chen, 2023) offers a unified multi-dimensional framework for analyzing open-domain conversations and automatically assigns scores for appropriateness, grammar, relevance, and content quality.

Furthermore, organizational contexts have been covered in recent contributions. The multi-agent framework for inventory management by Z. Li et al. (2024) and the BPM benchmark by Busch & Leopold (2024) are two examples that extend evaluation toward fields directly re-

lated to Supply Chain and Project Management. However, systematic benchmarks specifically designed for Supply Chain and Project Management remain to be developed.

Datasets

In the literature, benchmark datasets are built using various methodologies based on the skills to be evaluated.

Some benchmarks are based on real examinations, such as AGIEval (Zhong et al., 2023), which gathers items from standardized tests and employs only objective formats to guarantee trustworthy scoring. Another example is the warehousing study by (Franke et al., 2025), where undergraduate exams originally in German were translated into English to make them accessible to the international research community and then administered to LLMs for comparison.

As demonstrated in FanOutQA, where students created intricate "fan-out" questions that required information from multiple Wikipedia articles and were then broken down into smaller questions that could be answered from single sources, another method entails creating new datasets through manual annotation (Zhu et al., 2024).

Lastly, ZhuJiu (Zhang et al., 2023)combines both approaches: it incorporates publically accessible datasets and adds newly created datasets produced using a ChatGPT-based self-instruction pipeline, with manual seeding and evaluation to prevent leakage and guarantee fairness.

These examples demonstrate the various ways to dataset compilation, but benchmark resources are smaller and more static than the massive, heterogeneous corpora needed for LLM training. The design of training and evaluation datasets also plays a central role in shaping benchmark outcomes. LLM development is based on vast and diverse collections of knowledge data, including books, journals, and websites, as well as structured data and multimodal sources such as images, audio, and video. How well a model generalizes across many contexts depends on the quality and diversity of these datasets, but benchmark datasets frequently reduce this richness to small, static samples (Miao et al., 2024). Furthermore, to avoid redundancy, bias, or toxicity, data management is based on organized pre-processing pipelines that include collection, filtering, deduplication, standardization, and review. Every step affects the model's ultimate capability.

For example, the data collection step necessitates identifying task-specific needs, selecting credible sources, and assuring privacy and legal compliance. Filtering stages frequently use

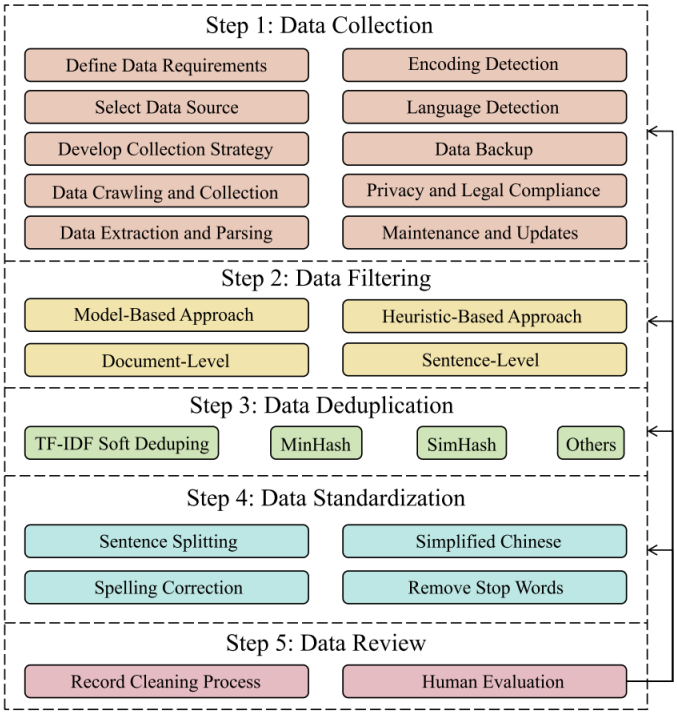


Figure 2.7: Preprocessing pipeline for pre-training corpora(Y. Liu, Cao, et al., 2024).

heuristic or model-based methods to filter low-quality, dangerous, or irrelevant content. Deduplication methods like *TF-IDF (Term Frequency-Inverse Document Frequenc) Soft Deduping* are used to remove redundant or too similar text segments, lowering noise in the corpus. Sentence segmentation, encoding correction, spelling normalization, and stop word removal are all part of the standardization process, which aims to provide cleaner and more consistent input. Finally, both automated and manual review systems ensure that errors or biases found earlier in the process are iteratively remedied. These procedures heavily influence model quality and fairness, but benchmarks seldom represent them, instead relying on static and simplified datasets that neglect the dynamic and curated character of genuine training corpora (Y. Liu, Cao, et al., 2024).

2.3.2 Evaluation Metrics

The literature shows that Large Language Model evaluation techniques can be broadly categorized into three different categories (Chang et al., 2024).

1) Metrics-Based Evaluation

The first approach for LLM assessment is *metrics-based evaluation*, which relies on predetermined quantitative criteria to measure model performance on existing datasets, providing objective and repeatable results. The most frequently used metrics in LLM benchmarks are *Accuracy* and the *F1-Score*.

Accuracy

Accuracy is defined as the proportion of the number of correct instances, both true positives and true negatives, out of to the total number of cases. It reflects the likelihood of randomly encountering a correctly classified occurrence, whether positive or negative. Equation 2.1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \approx \frac{Number\ of\ correct\ instances}{Number\ of\ total\ instances} \tag{2.1}$$

F1-Score

F1-Score, also known as the *F-measure*, is the harmonic mean of *precision* and *recall*, giving equal weight to both. Precision and recall are defined as the probability of finding a truly relevant instance while predicting a positive, and the probability of finding the right instance when predicting correctly. Equation 2.2

$$F1{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.2}$$

- Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. In other words, it answers the question: *"When the model predicts positive, how often is it correct?"*. Equation 2.3

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

- Recall measures the proportion of actual positive instances that were correctly identified by the model. It answers the question: *"Of all the real positives, how many did the model capture?"*. Equation 2.4

$$Precision = \frac{TP}{TP + FN} \tag{2.4}$$

While these metrics are valued for their simplicity and interpretability, they also have limitations. Accuracy can be misleading in highly imbalanced datasets, while the F1-score does not account for true negatives and may bias in favor of a majority class (Powers, 2015).

2) LLM-Based Evaluation

A second approach is *LLM-based evaluation*, also known as the *LLM-as-a-Judge* paradigm. In this setting, high-performing models such as GPT-4 are used to evaluate the outputs of other LLMs This method typically involves techniques such as prompt engineering, few-shot learning, and labeled responses, supported by repeated trials to enhance accuracy and stability(Gu et al., 2025). It can take three forms:

- *Pairwise comparison*, in which the judge chooses the better of two outputs or declares a tie;
- *Single answer grading*, in which the judge assigns a direct score to a single output;
- *Reference-guided grading*, in which the grading decision is based on a reference solution, which is especially useful in fields like mathematics.

Each method comes with trade-offs. Pairwise comparison provides robust relative judgments but scales poorly as the number of models grows. Single answer grading is more scalable but risks overlooking subtle quality differences. Reference-guided grading helps address domain-specific challenges but heavily depends on high-quality reference data.

Less reliance on human assessors, faster benchmarking cycles, and outputs that are interpretable and full of explanations are just a few benefits of the LLM-as-a-Judge. However, it is still susceptible to flaws such as verbosity bias (favoring solutions that are longer but equally correct), position bias (favoring responses in specific positions), and potential self-enhancement bias (favoring responses from the same LLM serving as judge) (Shi et al., 2025). The significance of continuous human monitoring in automated grading is further highlighted by the fact that LLM judges have the ability to improperly assess math or reasoning problems, even ones that they could solve correctly on their own (Zheng et al., 2023).

3) Human Evaluation

Lastly, Human evaluation is a fundamental aspect of LLM benchmarking, as it integrates subjective human judgment into the evaluation of model results. Many studies engage experts, stu-

dents, or professionals to evaluate model replies, which are often scored on a numerical scale (e.g., 1 to 5) to assess dimensions such as accuracy, completeness, or clarity. For instance, in a study conducted by (Mehri & Eskenazi, 2020), six researchers specialized in conversational AI rated system outputs across multiple qualitative aspects, such as understandability, naturalness, context maintenance, interestingness, and knowledge usage, before aggregating them into an overall quality score on a 1–5 scale.

Less frequently, human evaluation takes the form of academic grading, in which LLM responses get evaluated using the same criteria as university exams. A pertinent case is the study by (Franke et al., 2025), in which a faculty researcher scored ChatGPT's answers to three warehouse exams using the official sample solutions and the same grading system as students. By contrast, comparative evaluation takes a more natural approach, putting models in one-on-one arenas where human assessors directly compare their results, as popularized by Chatbot Arena (Zheng et al., 2023; Zhang et al., 2023). Chatbot Arena is a crowdsourced benchmarking tool that allows models to compete anonymously in head-to-head matches (Figure 2.8). Users engage with two unidentified models simultaneously, asking the same question and voting on their preferred response. Model identities are revealed only after voting, which mitigates evaluator bias. Unlike benchmarks with predefined prompts, Chatbot Arena allows users to ask unrestricted, spontaneously occurring inquiries, allowing for evaluation across a wide range of real-world use cases and interests (Zheng et al., 2023).

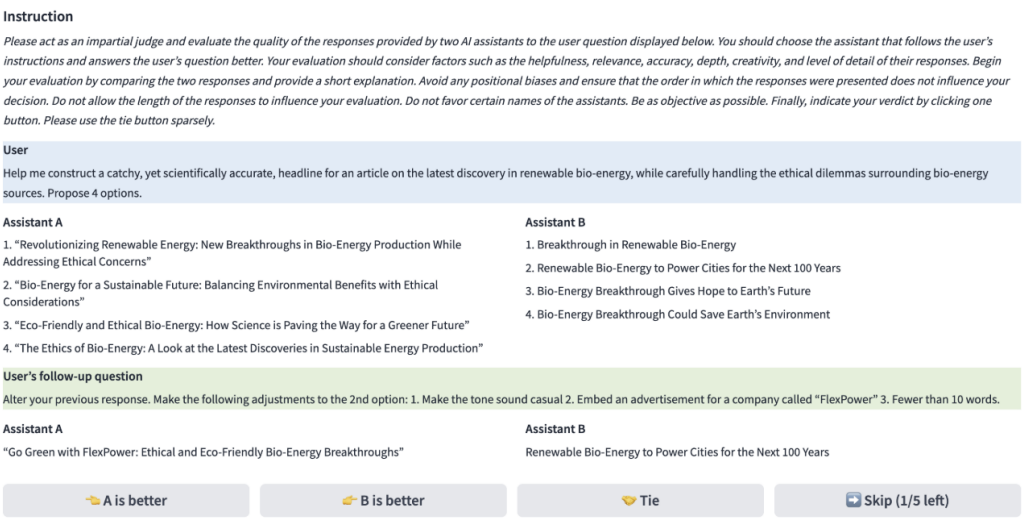


Figure 2.8: Chatbot Arena normal voting interface (Zheng et al., 2023).

The difference between structured evaluations (numerical scales or academic grading) and crowdsourced comparative ones emphasizes their complementary roles. While controlled set-

tings provide consistency and comparability, arena-based evaluation gives practical validity and alignment with real user expectations. However, both remain partial: structured evaluation is limited in variety and expensive to scale (Y. Wang et al., 2023), while crowdsourced votes could be noisy or biased (Zhang et al., 2023).

2.3.3 Challenges and Limitations

Evaluating Large Language Models remains challenging, as existing benchmarks often struggle to capture their true real-world performance and usefulness. Stability is a critical concern, since even minor changes to a prompt can lead to drastically different outcomes (Dam et al., 2024). Furthermore, designing fair assessments is complicated by ethical issues such as bias, privacy violations, and potential misuse, particularly in high-stakes industries where errors can have dire repercussions. A significant number of individuals utilizing AI lack technical expertise and engage with it across many text-based contexts(Figure 2.9).

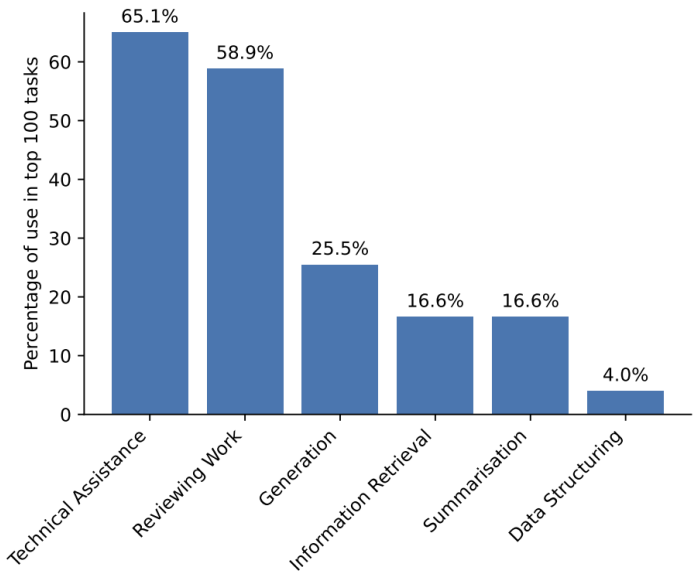


Figure 2.9: Prevalence of AI capabilities across the top 100 occupational tasks (Miller & Tang, 2025).

However, the majority of benchmarks assess limited tasks that are straightforward to evaluate, such as coding or recalling facts. Consequently, there is a gap between what benchmarks measure and how people actually use AI, since prevalent activities such as reviewing and refining written work are not included (Miller & Tang, 2025).

Moreover, most assessments neglect crucial aspects such as time savings, clarity, and sim-

plicity of integration into current workflows, prioritizing correctness over efficiency, interpretability, and contextual relevance (Eriksson et al., 2025). Beyond these limitations, existing benchmarks such as *EconLogicQA* (2024)(Quan & Z. Liu, 2024), *PredictaBoard* (2025) (Pacchiardi et al., 2025), *FanOutQA* (2024)(Zhu et al., 2024), frequently ignore linguistic and cultural diversity, preferring English over other languages such as Chinese.

Narrowness overlooks cultural nuances and alternative valid solutions shaped by different social, religious, or political contexts, thereby limiting inclusiveness and generalizability(Mushtaq et al., 2025). *ZhuJiu* was presented as the first comprehensive Chinese benchmark for LLMs in order to rectify this discrepancy. Although its uptake remains limited compared to English-centric frameworks, it provides both Chinese- and English-based evaluations and constitutes a step toward culturally grounded assessment (Zhang et al., 2023). These limitations are not only linguistic but also methodological.

Many evaluation methods rely on static forms, like multiple-choice questions or single-turn dialogue prompts, which fail to replicate the dynamic, multi-turn nature of real-world human-AI interactions, where consistency, coherence, and adaptability are essential (McIntosh et al., 2024). A related and ongoing issue is differentiating genuine reasoning from technical optimization, as models may learn to exploit benchmark-specific patterns or overfit to test structures rather than demonstrate real comprehension. This phenomenon, known as *benchmark gaming*, can artificially inflate outcomes and misrepresent a model's true capabilities, especially when evaluation datasets overlap with training data (Balloccu et al., 2024). Such concerns undermine the validity of benchmark results and may foster to overconfidence in deployment decisions.

The way benchmarks are used and interpreted is another limitation. Their proper application necessitates a thorough comprehension of methodological limitations and design choices. However, this knowledge is frequently underreported or ignored. This has resulted in cases where benchmarks such as *MMLU* (Hendrycks et al., 2021) or *BBQ* (Parrish et al., 2022) are applied inconsistently or their results are accepted at face value without taking into account the underlying assumptions.

These challenges are further compounded by the lack of standardized documentation for LLM benchmarks. At present, no specific frameworks exist to ensure consistent reporting of benchmark design, datasets, metrics, and evaluation assumptions, although some tools are available for characterizing AI datasets. The absence of such *benchmark metadata* makes it difficult for practitioners, regulators, and academics to evaluate benchmarks, choose appropri-

ate ones, and interpret results in light of real-world dangers(Reuel et al., 2024). The literature highlights several attempts to bridge this documentation gap. For instance, Sokol et al. (2025) presented *BenchmarkCards*, a structured framework designed to standardize the reporting of benchmark design, assumptions, metrics, and limitations, with the aim of improving transparency and alignment with intended use cases. Addressing these shortcomings requires the development of more comprehensive, transparent, and context-aware benchmarking methodologies that more accurately capture the diverse applications of LLMs in real-world contexts.

2.4 Benchmark Results Across LLMs

This section will present the results of benchmark analyses on several Large Language Models. It will show how models perform on a wide range of tasks and datasets, compare their strengths and weaknesses, and highlight developing patterns in capabilities and reliability.

2.4.1 Strengths and Limitations of LLM Performance

Recent benchmarks reveal heterogeneous outcomes that demonstrate both the benefits and limits of contemporary LLMs in various sectors.Across benchmarks, evidence shows that LLMs achieve strong results in structured and reference-based tasks but face difficulties with multi-step reasoning, domain-specific knowledge, and sophisticated real-world applications,as highlighted by recent benchmarks like *FanOutQA*(Zhu et al., 2024) and *EconLogicQA* (Quan & Z. Liu, 2024).In this latter benchmark, GPT-4-Turbo has the highest accuracy in both 1-shot and 5-shot settings, with GPT-4 following closely behind. This suggests that larger frontier models in sequential economic reasoning have a distinct benefit.

More broadly, LLM strengths emerge most clearly in standardized formats such as the natural language understanding tasks originally codified by *GLUE* (A. Wang et al., 2019), or multiple-choice question answering as exemplified by *MMLU* (Hendrycks et al., 2021), where task boundaries are explicit and scoring criteria are objective. However, performance drops considerably when tasks involve integrating information across multiple documents, sustaining logical coherence over long contexts, or applying technical expertise within specialized domains (Guo et al., 2025)); in the *FanOutQA* benchmark, GPT-4-Turbo and Claude 2.1 performed best overall, particularly in the evidence-provided setting, though major obstacles remain for smaller or less specialized models (Zhu et al., 2024).

These gaps imply that, while current LLMs excel at surface-level recognition and recall, they remain limited in deeper reasoning, contextual adaptation, and specialized competence. This discrepancy explains why models that rank highly on benchmark leaderboards may not necessarily prove reliable in professional or educational settings(Mishra & Arunkumar, 2021; Talby, 2025),as demonstrated by (Lunardi et al., 2025), who showed that linguistic variance in prompts can considerably affect accuracy even when leaderboard rankings stay unchanged.

Evidence from applied domains supports this view: although LLMs perform consistently well on routine knowledge tasks, they significantly underperform in quantitative reasoning and domain transfer. For example, studies of economic reasoning (Quan & Z. Liu, 2024) and warehousing applications (Franke et al., 2025)show that human participants often retain a comparative advantage.

At the same time, advances in prompting techniques like chain-of-thought or multi-agent prompting suggest that cognitive constraints are shaped not only by task complexity but also by the strategies used to structure reasoning.Nevertheless, smaller-scale models do not appear to benefit from these methods to the same extent (Wei et al., 2022).

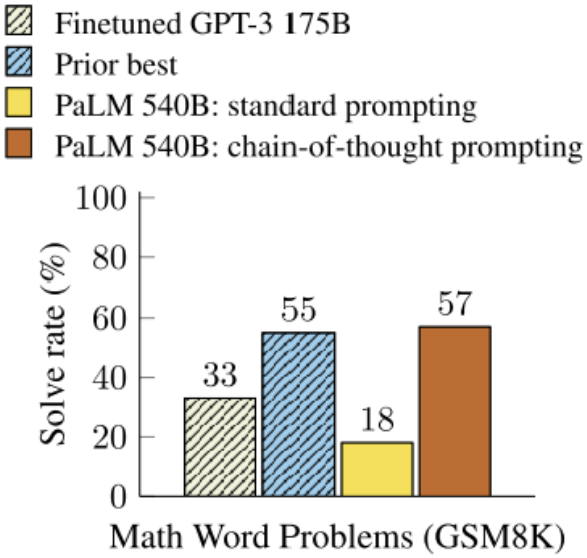


Figure 2.10: Impact of Chain-of-Thought prompting on mathematical problem-solving (Wei et al., 2022).

Building on this perspective, recent studies have introduced collaborative multi-agent and multi-path reasoning frameworks in which multiple independent instances of the same model act as agents, each tasked with a distinct reasoning role before exchanging their perspectives. By mimicking a collaborative approach to problem-solving, this technique allows models to take into account several points of view before reaching a decision(Z. Li et al., 2024). For ex-

ample, *Minstrel* (M. Wang et al., 2024) leverages structured prompt generation through agent collaboration to coordinate distinct reasoning paths, while *CoMM* (P. Chen et al., 2024) distributes complementary reasoning techniques across multiple agents and integrates their outputs to enhance robustness. Empirical evidence shows that such approaches improve performance in complex domains such as moral or ethical reasoning, where agent-to-agent dialogue helps balance conflicting opinions, even though highly technical disciplines like physics continue to reveal persistent problems (P. Chen et al., 2024).

In conclusion, these findings show that domain-specific benchmarks highlight the limitations of LLMs' applied competence, whereas advances in prompting,ranging from structured reasoning chains to collaborative multi-agent interaction, provide partial possibilities for closing these gaps. However, such methods remain constrained by scale and design, suggesting that strengthening reasoning capabilities requires not only improved data and evaluation practices but also new frameworks for orchestrating the cognitive processes of models.

Chapter 3

Research Methodology

This chapter illustrates the methodology adopted for the development of the research. After presenting the research questions and the exploratory framework that guided the work, the methodological choices related to the construction of the benchmarks will be described, along with the implementation and testing procedures that enabled the collection of results, which are discussed in the following chapter.

3.1 Research Questions and Exploratory Framework

In the previous chapter, the theoretical foundations of Generative AI, and more specifically Large Language Models, were analyzed in two distinct managerial domains: Project Management and Supply Chain Management. This analysis highlighted numerous contributions already available in the literature, bringing to light both the potential of LLMs in automating and supporting decision-making processes, as well as the methodological limitations that still remain. In particular, a clear gap emerged regarding the absence of systematic benchmarks that allow for comparable and replicable evaluation of LLM performance in real-world project and supply chain management contexts.

Building on this observation, the present research aims to contribute to filling this methodological gap. However, in order to ensure a more focused and coherent approach, from this chapter onward the analysis is narrowed exclusively to the Project Management domain, leaving the development of benchmarks for Supply Chain as a direction for future research.

The decision to focus on Project Management made it possible to design specific benchmarks, built on datasets, evaluation metrics, and prompting strategies, capable of reflecting the

actual decision-making needs of companies in this sector.

In this perspective, the purpose of the present section is to introduce the research questions guiding the study and to define the exploratory framework that served as a methodological reference for benchmark design.

3.1.1 Research Questions

The research questions stem from two complementary needs: on the one hand, companies require reliable and accurate tools to support decision-making in sensitive areas such as project management; on the other hand, the literature has highlighted the absence of systematic frameworks for evaluating LLMs, which may represent a valuable tool in the project management context.

Based on these premises, the main research questions were formulated as follows:

- **RQ1**: Which combinations of datasets, evaluation metrics, and prompting techniques enable the construction of meaningful benchmarks for assessing LLM performance in project management contexts? $\rightarrow$ This question seeks to identify the most suitable methodological configurations to transform LLM experimentation into a systematic, replicable, and comparable process.

- **RQ2**: Which LLM currently demonstrates the best performance? $\rightarrow$ This question aims to determine, on the basis of the developed benchmarks, which model best integrates the main evaluation criteria, providing a comparative overview useful for guiding managerial selection.

- **RQ3**: Are LLMs truly valid tools for supporting managerial decision-making? $\rightarrow$ The goal is to interpret the results not only in terms of technical performance but also by assessing aspects that respond to managers' real operational needs, in order to establish whether LLMs can serve as effective support in both strategic and day-to-day decisions.

3.1.2 Exploratory Framework

The exploratory framework is built around the concept of the benchmark as an instrument for integrated evaluation. It is structured into three main elements:

1. **Dataset**: consisting of a set of carefully selected questions that reflect common project management challenges to simulate decision-making scenarios.

2. **Evaluation metrics**: representing the criteria for measuring the performance of LLMs, including not only accuracy indicators but also other performance measures relevant to managerial use.

3. **Prompting techniques**: serving as the means through which the interaction with the models is shaped, guiding their behavior and optimizing their effectiveness in different scenarios.

The exploratory logic assumes that the interaction among these three elements generates different benchmark configurations. Each combination makes it possible to observe how LLMs respond to specific tasks. The practical implementation will then provide the results necessary to answer the third research question, namely how these benchmarks can effectively reflect the ability of LLMs to support managerial decision-making in complex project management settings. The analysis of performance in real scenarios will allow not only for the comparison of different configurations but also for assessing their applicability in practice, with the aim of identifying the most effective strategies and improving the decision-making process. Furthermore, the systematic comparison of results will make it possible to address the second research question, which seeks to determine which LLM currently represents the most effective solution.

3.2 Benchmark Construction

Once the research objectives have been clearly defined, the next phase concerns the construction of the benchmarks. This section therefore illustrates the procedures adopted for dataset creation, the criteria employed in the selection of evaluation metrics, and the prompting techniques considered.

3.2.1 Dataset

QUESTION TYPE

In the design of a benchmark aimed at evaluating the performance of Large Language Models (LLMs), a crucial methodological aspect concerns the selection of the types of questions to

be proposed. The structure of the questions, in fact, influences both the nature of the skills elicited,such as calculation, reasoning, and planning, and the measurability of the results, affecting aspects such as objectivity of grading, reproducibility, and inter-rater reliability. In the current literature on LLM benchmarks, there is a clear predominance of datasets based on multiple-choice questions. Tests such as MMLU (Massive Multitask Language Understanding), or similar evaluation tools, are primarily built on multiple-choice tasks, where the model must identify the correct answer within a set of alternatives. This approach has evident advantages: it allows for standardised evaluation, reduces interpretative ambiguity, and makes results easily comparable across different models. However, due to their highly structured nature, such benchmarks tend to explore only a limited portion of model capabilities,particularly those related to pattern recognition or the retrieval of already encoded knowledge, while neglecting more complex aspects such as autonomous quantitative reasoning or the handling of articulated application scenarios. However, these skills are fundamental in concrete project management applications. To overcome these limitations, the present research has chosen not to rely exclusively on the multiple-choice format, but to include heterogeneous types of questions, in order to construct a benchmark that is more comprehensive and representative of the real challenges an LLM may encounter in project management applications. The main types of questions considered during the benchmark design phase are reported in Table 3.1.

ID	Question Type	Description
Q1	Close question – single-choice	A multiple-choice question with a finite set of options (typically 3–5), of which only one is correct.
Q2	Close question – multiple-choice	A multiple-choice question in which two or more options may be correct.
Q3	True/false	A closed-ended question presenting a statement to be answered by indicating whether it is true or false.
Q4	Numerical answer	A question requiring an exact numerical response, usually derived from a calculation or quantitative data.
Q5	Open question	A question requiring a discursive or argumentative response, without predefined options.
Q6	Case study	A realistic and complex scenario requiring critical analysis and problem solving through a set of related questions.

Table 3.1: Question types with their descriptions

Q1: Close question - Single-choice

Single-choice questions are among the most traditional formats used in evaluation. In this case, the model is given a finite set of options, usually three to five, with only one correct answer. This format offers objectivity in assessment, allows for automated grading, and minimizes ambiguity. Another strength of the format is flexibility: single-choice questions can be theoretical, aimed at testing definitions or conceptual knowledge, or numerical, where the model has to perform a calculation and select the correct answer from among the options. This two-sidedness makes them particularly well-suited to combining the evaluation of conceptual knowledge with basic quantitative skills. At the same time, there are some problems that remain, such as the possibility of guessing the correct answer, the over dependence on the quality of distractors, and the danger of cueing, when unconscious linguistic cues make the correct option more identifiable.

Q2: Close question: Multiple-choice

This type allows two or more options to be correct simultaneously, making it possible to assess more articulated knowledge compared to the single-choice format. Unlike single-choice, it is poorly suited to testing numerical skills, as its emphasis lies mainly on theoretical or conceptual knowledge. Moreover, some critical issues emerge: identifying the exact subset of correct answers may be ambiguous, the evaluation process is more complex, requiring decisions on whether to assign partial credit, apply an all-or-nothing approach, or use differentiated weighting, and the increased cognitive load does not always correspond to a real gain in informational value.

Q3: True/false questions

The true/false format represents the simplest modality. The model is presented with a statement and asked to determine its truthfulness. The construction and correction of such items are immediate and easily automatable, but the format has evident limitations. The most obvious is the high probability of a correct response by chance (50%), which drastically reduces the discriminatory power of the test. Furthermore, the presence of negations or ambiguous linguistic formulations can lead to misleading evaluations that do not accurately reflect the model's actual competence.

Q4: Numerical answer questions

Numerical answer questions require the model to produce a precise value derived from a calculation or a formula. They provide a high degree of objectivity, since the expected output is a unique number that can be directly compared with the correct solution. This type is particularly relevant in the field of project management, where activities such as estimating project duration through critical path analysis, calculating earned value metrics (CPI, SPI), determining resource allocation, or assessing project costs rely on numerical results. The main criticalities concern formatting issues (for example, the use of decimal separators or measurement units), rounding, and the need for clear and consistent normalization criteria for results.

Q5: Open-ended questions

Open-ended questions are characterised by the absence of formal constraints: when presented with a theoretical prompt, the model is required to produce a discursive answer, support an argument, or provide an explanation. This format highlights argumentative ability, logical coherence, and the capacity to connect different concepts. However, the very lack of constraints also represents the main limitation. Evaluation inevitably becomes more subjective, reducing reproducibility of results; moreover, the analysis and correction of responses demand significant time and resources. Finally, the stylistic variability typical of different LLMs can further complicate comparison, as formally different answers may contain substantially similar content, or fluent texts may conceal conceptual errors.

Q6: Case studies

Case studies represent the most complex type, and the one closest to real-world scenarios. In this format, the model is not required to identify a single answer, but rather to analyse an articulated problem, formulate hypotheses, and propose motivated solutions. This type enables the evaluation of advanced skills such as strategic reasoning, the ability to manage trade-offs, and decision-making consistency. At the same time, evaluation is complex and requires structured rubrics and the intervention of human assessors.

Methodological choice: from Bloom's taxonomy to the "difficulty pyramid" (Q1 → Q4 → Q4+)

A fundamental starting point for the construction of the benchmark was to identify a theoretical framework capable of guiding the definition of question complexity levels. In this regard, Bloom's taxonomy (Figure 3.1) represents a particularly useful tool. It describes cognitive processes as a hierarchy, ranging from the simple recall of information (Remember) to the production of new knowledge and solutions (Create). The intermediate levels — Understand, Apply, Analyze — are especially relevant in managerial and operational contexts, as they reflect the progression from recognizing basic concepts, to applying them in concrete situations, and finally analyzing them critically (Elkins et al., 2024).

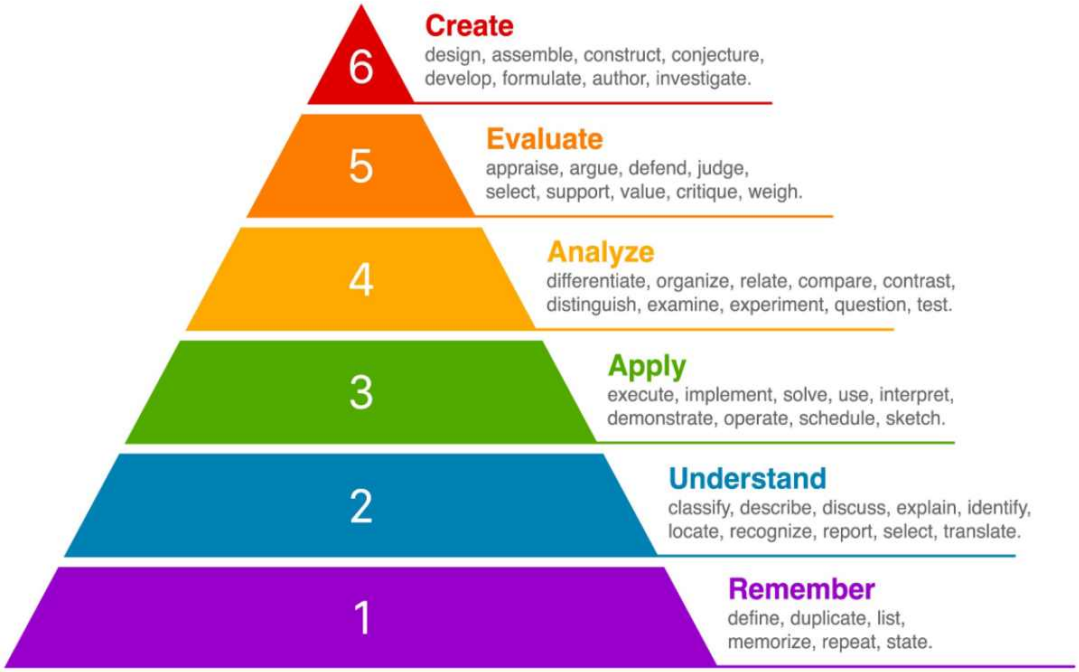


Figure 3.1: Bloom taxonomy

Based on this framework, it was necessary to select, among the different types of questions potentially suitable for a benchmark (as described in the previous section), a subset consistent with the project management context while remaining methodologically sound. The aim was to overcome the limitations of existing benchmarks, which rely almost exclusively on multiple-choice questions. In this perspective, three types of questions were selected to progressively reflect the different levels of Bloom's taxonomy:

- **Single-choice questions (Q1)**: testing basic knowledge and immediate recognition or comprehension skills, positioned at the lower levels of Bloom's hierarchy (Remember/Understand).
- **Numerical answer questions (Q4)**: requiring the application of formulas, manipulation of numerical data, and independent production of a result, corresponding to the Apply level.
- **Numerical answer questions with reasoning (Q4+)**: corresponding to the Analyze level, as they require not only the correct calculation but also the explicit explanation of the procedure, formulas used, and assumptions adopted.

To make this progression clearer and more operational, the logic of Bloom's taxonomy was

translated into a simplified representation adapted to the objectives of this work: the “difficulty pyramid” (Figure 3.2).

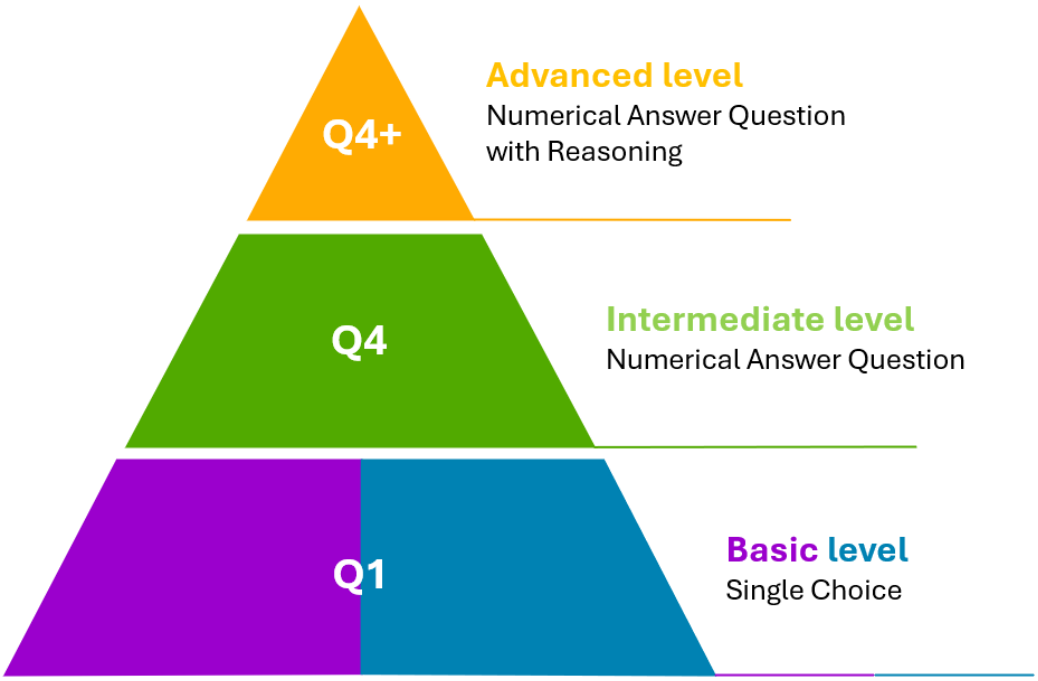


Figure 3.2: Difficulty pyramid

This pyramid, inspired by Bloom but tailored to the needs of the benchmark, organizes the question types into three progressive levels of difficulty:

- **Basic Level — Q1: Single Choice** Placed at the base of the pyramid, this represents the starting point of the evaluation. Single-choice questions provide an optimal compromise between ease of administration and objectivity of assessment. The presence of predefined options reduces ambiguity and makes it possible to test both theoretical knowledge and numerical skills, thus establishing a solid baseline reference.

- **Intermediate Level — Q4: Numerical Answer Question** At this level, the model is no longer guided by predefined options but must independently produce a numerical output. This introduces a higher degree of complexity and makes it possible to assess active skills such as logical-mathematical rigor and accurate calculation ability, both central to managerial and operational applications.

- **Advanced Level — Q4+: Numerical Answer Question with Reasoning** The final level combines the requirement of producing a numerical result with the obligation to make

the reasoning process explicit: applied formulas, logical steps, assumptions, and measurement units. This distinction makes it possible to separate calculation errors from conceptual gaps and reflects more closely the needs of project management contexts, where traceability and justification of the calculation process are as essential as the final result.

The transition from Bloom's taxonomy to the difficulty pyramid thus enables the integration of a general theoretical framework with a targeted application model. The outcome is a benchmark capable of going beyond the limits of traditional tests, providing a more realistic, structured, and context-relevant evaluation of project management challenges.

Exclusion of other question types

The definition of the Q1–Q4–Q6 triad simultaneously implied the exclusion of the remaining identified types. This decision did not stem from an underestimation of their potential, but rather from the need to guarantee methodological coherence, robustness of evaluation, and comparability of results.

Q2 – Multiple-choice questions with more than one correct answer

The multiple-answer format was excluded primarily because it introduces an excessive cognitive load, disproportionate to the actual informational value gained, particularly in a domain such as Project Management, where clarity and verifiability of responses are essential. In addition, the higher complexity and arbitrariness involved in defining evaluation criteria risk undermining the methodological soundness of the benchmark.

Q3 – True/false questions

Binary statements were excluded as they are overly simplistic and weakly discriminative. The 50% chance of a correct response drastically reduces the statistical robustness of the test, while sensitivity to negations or linguistic nuances can produce misleading results, not always related to the model's actual level of knowledge or reasoning. In this format, the noise introduced tends to outweigh the useful information.

Q5 – Open-ended questions

Open-ended questions have the advantage of highlighting the discursive and argumentative abilities of the model but are of limited usefulness in the project management context, where the true challenge lies in assessing logical reasoning and problem-solving rather than the mere exposition of theoretical knowledge. Moreover, their evaluation inevitably requires human intervention, reducing reproducibility and comparability of results. For these reasons, this type was excluded in favour of more controllable and objective formats.

Q6 – Case study

Case studies were excluded because, although they represent a form of assessment closely aligned with real-world project management scenarios, they introduce a level of methodological complexity that is difficult to reconcile with the construction of a systematic and replicable benchmark. Their heterogeneity makes it challenging to define standardized evaluation criteria, increasing the risk of results that are not easily comparable across models. Moreover, analyzing a case study almost always requires a subjective interpretative process, involving human judgment in scoring, which reduces both replicability and automation in the evaluation framework. For these reasons, this question type was set aside in favor of more controllable and objective formats, while still acknowledging its relevance for future experimental or applied investigations.

DATASET CONSTRUCTION

Once the types of questions to be included in the pyramid of difficulty had been defined, the next step was the construction of the dataset, designed to coherently reflect the three selected categories: single-choice (Q1), numerical answer (Q4), and numerical answer with reasoning (Q4+). The dataset development phase is central, as the coherence of the questions, their level of difficulty, and their adherence to the application domain largely determine the reliability of the benchmark and, consequently, the robustness of the conclusions drawn. In defining the questions, a heterogeneous approach was adopted, integrating both academic and professional sources:

- Teaching materials from lectures and exercises at Politecnico di Torino (Italy), a leading academic institution in engineering education and project management research;

- Teaching and examination materials from Eindhoven University of Technology (TU/e, the Netherlands), a university internationally recognized for its strong focus on systems engineering, data-driven project management, and applied research in technology-driven organizations.
- Question banks and bodies of knowledge from internationally recognized professional certifications and standards, including IPMA (Introductory Certificate in Project Management), PMI-RMP (PMI Risk Management Professional), CAPM (Certified Associate in Project Management), and PMP (Project Management Professional), together with their respective practice examinations. These frameworks define globally accepted competencies, process models, and performance domains, ensuring alignment with industry expectations for certified project managers.

The diversification of sources made it possible to construct a coherent and well-balanced dataset, capable of integrating different dimensions, conceptual knowledge, quantitative skills, and complex reasoning ability, and of organically reflecting the multi-level structure defined in the pyramid of difficulty.

Database of single-choice questions (Q1)

For the first type, single-choice questions, a database of 300 items was created, divided into:

- 200 theoretical questions, aimed at verifying the knowledge of concepts, definitions, and standard rules of project management, without requiring calculations;
- 100 numerical questions, which instead involve calculations or applications of formulas related to project management, in order to also assess logical and quantitative reasoning skills.

The questions were drawn from multiple sources: 90 from Politecnico di Torino, 75 from Eindhoven University of Technology (TU/e), and 135 from internationally recognized professional certifications and standards. This distribution ensures not only cultural and academic variety but also robustness, as the questions reflect different didactic and methodological perspectives. Examples of included questions:

- **Theoretical question (Q1):** *On the project that you're managing, you've noticed a decreasing trend in deliverable quality over time. How can Continuous Improvement help address this issue?*

a) *Increase the project budget to enhance quality instantly.*

b) *Ignore the trend, as it may rebound to normal without intervention.*

c) *Through the cycle of Plan, Do, Check, and Act (PDCA) to improve the quality in a measured way.*

d) *Replace team members with new ones assuming they have better skills.*

- **Numerical question (Q1):** *Your team is working on a software development project which has three time estimates: Optimistic is 10 days, Pessimistic is 30 days, and most likely is 15 days. What should be the expected time using PERT formula?*

  a) *18 days*

  b) *16 days*

  c) *15 days*

  d) *20 days*

Database of numerical answer questions (Q4)

For the second type, numerical answer questions, a database of 100 items was created, drawn from different sources: 62 from Politecnico di Torino, 23 from Eindhoven University of Technology (TU/e), and 15 from professional Project Management certifications. A methodologically relevant aspect is that these 100 questions coincide exactly with the numerical questions already used in the single-choice database (Q1). This choice was made to enable a direct comparison between two different administration modes: in the case of Q1, the model has four numerical options to choose from, thus being guided towards the solution; in the case of Q4, instead, the options disappear, and the model is required to calculate the correct result autonomously, without any external constraints or hints. This makes it possible to evaluate whether LLM performance depends on the ability to recognise the correct value among proposed alternatives, or on the actual ability to compute it independently. The questions were structured across three levels of difficulty:

- **Simple level.** Questions based on elementary formulas, with essential data and no superfluous information.

  – *Example: "You plan to deliver 10 features in 5 sprints. After 3 sprints, 4 features are completed. What is the SPI?"*

- **Medium level.** Questions with more articulated data requiring intermediate logical steps.
  - *Example: "On day 65, a software development project shows a physical progress of 70%, a delay of 5 days compared to the plan, and an ACWP of €500000.The original budget was €700000, and the planned duration was 90 days.The contract is a cost-plus fixed-fee type with a fixed fee of €150000, a 50/50 cost-sharing agreement for extra costs or savings between contractor and client, and a penalty of €1000 per day of delay.What is the revised Cost Estimate at Completion (CEAC)?"*
- **Difficult level.** Questions simulating complex business scenarios, with texts rich in information, not all relevant. The model must select the pertinent data and carry out chained reasoning.
  - *Example: "You are the Project Manager of a consulting company responsible for a reorganization project for a key client. The project started on May 2 and has a planned duration of 9 months.*

    Activity details:

    * A – As-is analysis: 1 month, no predecessor
    * B – Requirements & needs definition: 1 month, predecessor A
    * C – Gap analysis: 1 month, predecessor B
    * D – Re-organizational model: 3 months, predecessor C
    * E – Job definition: 2 months, predecessor C
    * F – HR assessment: 1 month, predecessor E
    * G – Deployment on pilot department: 1 month, predecessor D
    * H – Full deployment and coaching: 2 months, predecessor G

    The payment scheme is Cost Plus Fixed Fee (CPFF) with:

    * Fixed fee: 500000€
    * Delay penalty: 10000€ per week (including partial weeks)
    * Indirect costs reimbursed: 20000€/month

    As of August 31, the project status is as follows:

    * As-is analysis: BAC 20 k€, WS 100%, WP 100%, ACWP 24000€

* Requirements & needs definition: BAC 100000€, WS 100%, WP 100%, ACWP 90000€
* Gap analysis: BAC 40000€, WS 100%, WP 90%, ACWP 30000€
* Re-organizational model: BAC 120000€, WS 50%, WP 60%, ACWP 30000€
* Job definition: BAC 80000€, WS 30%, WP 0%, ACWP 0€
* HR assessment: BAC 90000€, WS 0%, WP 0%, ACWP 0€
* Deployment on pilot department: BAC 30000€, WS 0%, WP 0%, ACWP 0€
* Full deployment and coaching: BAC 50000€, WS 0%, WP 0%, ACWP 0€

Determine the final profit achievable by project end."

The three-tiered difficulty structure allows for the analysis not only of calculation accuracy but also of the ability of LLMs to handle complex texts and isolate relevant from redundant information,an essential skill in decision-making contexts within project management.

Database of numerical answer questions with reasoning (Q4+)

For the Q4+ level, conceived as an extension of basic numerical questions, a specific database was created based on the material already included in Q4. Specifically, only medium- and high-difficulty questions were selected, as they were deemed more suitable for eliciting structured reasoning in addition to the calculation of results. The overall database comprises 50 items,30 of medium difficulty and 20 of high difficulty, while maintaining the same distribution of sources already used for the other datasets: Politecnico di Torino, Eindhoven University of Technology (TU/e) and professional Project Management certifications The defining element of this level does not concern the content of the questions themselves but the mode of response expected: the model is no longer required merely to provide the correct numerical value but must also accompany it with a structured explanation of the logical procedure followed.

3.2.2 Evaluation Techniques

After defining the types of questions on which to test the language models and constructing the corresponding datasets, it was necessary to identify the evaluation techniques to

be employed in analysing the generated responses. The definition of metrics represents a central element in the design of the benchmark, as the reliability of the results and the possibility of conducting meaningful comparisons across different models depend directly on their robustness. As recalled in Section 2.3.2, the literature has identified several evaluation approaches, which can be grouped into three main categories: metric-based evaluation, human-based evaluation, and LLM-based evaluation, as reported in the following table.

		Evaluation Techniques	Definition
Metrics-based Evaluation	**E1**	Accuracy	Percentage of correct answers over the total number of questions.
	E2	F1 Score	Harmonic mean of Precision and Recall. Evaluate the quality of a classifier when it is important to consider both type I errors (false positives) and type II errors (false negatives). Precision: proportion of correctly predicted positive cases over all predicted positives. Recall: proportion of correctly predicted positive cases over all true positives in the dataset.
	E3	Latency	Response time of a model from the reception of an input to the generation of the complete output.
	E4	Token used	Total number of tokens processed by a model in an interaction, including both the input (prompt) and the output (generated response).
	E5	Cost	Economic expenditure required for the execution of the model, calculated as a function of the total tokens used (input + output) according to the provider's pricing.
Human-based Evaluation	**E6**	Human grade	Annotators assign a score from 1 to 5 based on predefined evaluation criteria.
	E7	Human comparative Judgment	Evaluation methodology based on pairwise comparison: human judges compare two responses generated by different models and select the better one. The process follows a tournament-style format, allowing rankings among models to be derived.
LLM-based Evaluation	**E8**	LLM as a judge	An LLM is employed as an evaluator to assess responses generated by other models (or by itself), based on prompts that define the evaluation criteria.

Table 3.2: Evaluation techniques and their definitions

The first category, metric-based evaluation, relies on automatically computable quantitative indicators such as accuracy, F1-score, response time (latency), number of tokens used, and computational cost. These metrics have the advantage of ensuring objectiv-

ity, reproducibility, and ease of comparison, thus enabling a standardized assessment of model performance. However, they mainly capture the surface aspects of responses (formal correctness, computational efficiency) without fully representing the quality of reasoning or the depth of content.

The second category, human-based evaluation, involves the direct intervention of human annotators, who assign scores to each response according to predefined criteria (human grade) or compare two outputs in pairs, selecting the one deemed superior (human comparative judgment). This approach makes it possible to capture qualitative dimensions that are difficult to measure through automatic metrics, such as argumentative coherence, clarity of exposition, or contextual relevance. On the other hand, human evaluation entails higher costs in terms of time and resources, while also introducing elements of subjectivity and reducing the reproducibility of results.

Finally, the third category, LLM-based evaluation, designates a language model itself as the evaluator, judging responses generated by other models (or by itself) according to criteria specified in the prompt. This technique combines execution speed with the ability to capture more nuanced qualitative aspects. Nevertheless, it raises critical concerns regarding reliability, potential bias, and dependence on the formulation of the evaluation prompt.

Application matrix of evaluation techniques

After identifying the main evaluation techniques, it was necessary to define systematically their application across the different types of questions included in the difficulty pyramid. To this end, a correspondence matrix between evaluation techniques and question types was constructed, as reported in the following table, representing a fundamental methodological step. This approach makes it possible to restrict the analysis to combinations that are genuinely meaningful, thereby avoiding, on the one hand, redundant or uninformative applications, and on the other, the use of metrics inconsistent with the nature of the question. For completeness, the matrix also includes question types not selected in the difficulty pyramid, together with the related methodological considerations. In this way, the table does not merely present the combinations adopted in the present research, but instead provides a broader and comparative view of the possible alternatives.

QUESTION TYPE	E1	E2	E3	E4	E5	E6	E7	E8
Q1	X		X	X	X			
Q2		X	X	X	X			
Q3	X		X	X	X			
Q4	X		X	X	X	X		
Q5			X	X	X	X	X	X
Q6			X	X	X	X		X

Table 3.3: Evaluation techniques applied to different question types

Legend	
Accuracy	E1
F1 Score	E2
Latency	E3
Token used	E4
Cost	E5
Human-grade	E6
Human comparative Judgment	E7
LLM as a judge	E8
Close question - single-choice	Q1
Close question - multiple-choice	Q2
True/false	Q3
Numerical answer	Q4
Open question	Q5
Case study	Q6

Table 3.4: Legend of evaluation techniques (E) and question types (Q)

In the matrix, the "X" marks indicate the combinations considered methodologically appropriate. Each choice was guided by a careful reflection on the relationship between the characteristics of the question and the ability of the metric to provide useful information.

– **Accuracy (E1).** This metric was associated with question types characterized by objective and unambiguous answers (Q1, Q3, Q4). In these cases, correctness can

be verified without margins of ambiguity, making accuracy a simple yet reliable measure. For multiple-answer questions (Q2), however, accuracy proves less representative, as it does not distinguish between completely wrong responses and partially correct ones.

- **F1-score (E2).** This metric was applied exclusively to Q2, where multiple answers can simultaneously be correct. Unlike accuracy, which evaluates responses in a binary way (all correct or all wrong), F1-score is able to recognize partially correct answers. In practice, this metric assigns an intermediate score when the model identifies only part of the correct options or includes both correct and incorrect ones. In this way, F1 provides a more nuanced and faithful measure of the overall quality of the response compared to accuracy alone, which in such cases would simply return a value of zero.
- **Latency, Token used, and Cost (E3–E4–E5).** These metrics were considered transversal, as they measure aspects of computational efficiency and economic sustainability regardless of the question's content. For this reason, they were applied to all question types (from Q1 to Q6). Their inclusion was deemed essential, since a model capable of providing correct answers but with excessive execution times or disproportionate costs would be unsuitable for concrete use in project management processes.
- **Human grade (E6).** The use of human evaluators was limited to contexts where subjective judgment adds real value. This applies to numerical questions (Q4) of medium-to-high difficulty, where answers may show slight deviations yet remain methodologically valid, and especially to case studies (Q6), which require qualitative evaluations of aspects such as reasoning consistency, plausibility of assumptions, or clarity of exposition. For closed and objective questions, on the other hand, human intervention would have been redundant and difficult to justify.
- **Human comparative judgment and LLM as a judge (E7–E8).** These techniques were reserved for open questions and complex scenarios (Q5 and Q6), where no univocal solution exists. Comparing multiple outputs or employing an LLM as evaluator allows for capturing qualitative nuances and stylistic differences that cannot be measured with standard metrics. For closed-ended questions, their use would instead have been excessively resource-intensive and of limited added value.

Once the general mapping was completed, the selection of the combinations effectively adopted in the present research focused exclusively on the question types identified in the difficulty pyramid (Q1, Q4, Q4+). For these categories, the metrics considered most appropriate were highlighted in green.

For single-choice questions (Q1), the selected metrics were E1 (accuracy), E3 (latency), E4 (token used), and E5 (cost). Accuracy represents the most immediate and objective measure of correctness, while the other metrics were chosen to monitor operational dimensions such as execution time, token consumption, and associated costs, essential elements for assessing efficiency.

For numerical questions (Q4), the metrics adopted were E1 (accuracy), E3 (latency), E4 (token used), and E5 (cost). Accuracy ensures an objective measure of the correctness of the returned value, while the other three metrics allow monitoring of operational aspects related to execution times, resource consumption, and economic sustainability. Together, these dimensions provide a comprehensive evaluation of both the model's effectiveness and its computational efficiency.

For numerical questions with reasoning (Q4+), it was deemed necessary to complement the metrics already used for Q4 (E1, E3, E4, E5) with human evaluation (E6 – human grade). In this case, it is not sufficient to verify the numerical correctness of the output: it becomes crucial to assess the quality of the reasoning provided, the coherence of logical steps, the relevance of assumptions, and the correctness of the formulas employed. These aspects, which cannot be quantified through automatic measures, require human intervention to ensure a complete and reliable evaluation.

Analytic Hierarchy Process (AHP)

In addition to the techniques described above, the Analytic Hierarchy Process (AHP) was adopted with the aim of synthesizing model performance into a single comparative measure, integrating heterogeneous dimensions such as accuracy, latency, and cost. For each benchmark, starting from the results of these three metrics, AHP was applied to derive a final ranking that reflects the overall set of criteria in a balanced manner. The application of the method involved two main steps:

– **Assignment of preferences.** To compare the models, a preference scale from 1

to 10 was defined, where a model was considered preferable to another depending on the metric under consideration (for example, a lower cost was judged preferable to a higher one, while higher accuracy was preferred to lower accuracy). At first, the possibility of assigning scores based solely on each model's rank was considered. However, this approach proved inaccurate, as it did not account for the actual distance between values: models that were very close would have been penalized in the same way as models that were far apart, with the risk of underestimating or overestimating real differences.

To overcome this limitation, the range between the maximum and minimum values of each metric was divided into ten equal-width classes (quantiles). Each class was associated with a score from 1 to 10, proportional to the observed gap. Formally, letting A be the absolute difference between two models for a given metric, and M_{max} and M_{min} the maximum and minimum values observed, the preference classes were defined as follows:

$$\Delta \in (0, 0.1\,(M_{\text{max}} - M_{\text{min}})) \Rightarrow 1$$

$$\Delta \in [0.1\,(M_{\text{max}} - M_{\text{min}}), 0.2\,(M_{\text{max}} - M_{\text{min}})) \Rightarrow 2$$

$$\dots$$

$$\Delta \in [0.9\,(M_{\text{max}} - M_{\text{min}}), +\infty) \Rightarrow 10$$

In this way, the scale not only reflected the relative ranking but also incorporated the magnitude of the actual difference. For example, two models with response times of 10.5 and 11 seconds received a very low preference score (class 1), while two models with latencies of 10.5 and 55 seconds fell into a high class, more realistically highlighting the superiority of the faster model.

– **Determination of metric weights.** The second step concerned the assignment of relative importance to accuracy, latency, and cost. To this end, a survey was conducted among a group of *Amazon Project Managers* based in Luxembourg, who routinely deal with planning, scheduling, and resource allocation in complex initiatives. They were invited to assign each criterion an importance score between 1 and 7, according to the traditional scale used in AHP. The questionnaire included three

main questions:

* **Accuracy** – How important do you think it is that the answer provided by the LLM is correct?
* **Cost** – How important do you think the cost of generating the answer is? (Considering that an answer to a complex question can vary from $0.01 to $0.10)
* **Latency** – How important do you think execution time is to generating the answer? (Considering that an answer to a complex question can vary from a few seconds to 6 minutes)

The aggregation of the results made it possible to derive the final weights to be applied in the multicriteria synthesis process, which were then used to build the AHP rankings of the different models.

The choice of AHP was motivated by the need for a tool capable of integrating objective data and managerial preferences within a coherent and transparent methodological framework. Compared to other multi-criteria methods, it allows for balancing trade-offs among different criteria, actively involving decision makers in the definition of priorities, and providing a final result in the form of a ranking of language models. Such a ranking serves as a practical reference for identifying the model most suitable for real operational scenarios, as it balances answer accuracy, execution speed, and economic sustainability.

3.2.3 Prompt Techniques

Once the dataset structure and the evaluation metrics have been defined, the next step is to understand how LLMs can interact with them.

The literature highlights that one of the key features of LLMs is their ability to interpret prompts expressed in natural language and adapt their responses according to the specific request. Consequently, in order to provide a tool capable of maximizing LLM performance, it is essential to assess different prompting techniques.

The table below reports the prompting techniques identified in the literature, together with a brief description to facilitate the reading of this chapter. Table 3.5

Prompt Techniques	Description
Zero-shot	The model is provided with only a textual description of the task to be performed, without including any explicit input-output examples.
One-shot / Few-shot	The model is provided with one or few illustrative examples of the task, followed by a new instance to solve.
Role prompting	A functional identity is assigned to the model (professor, expert, etc.) to adjust the tone, style, and level of expertise in its responses.
Chain-of-Thought (CoT)	The model is exhorted to solve the problem step-by-step, explaining the logical steps.
Self-consistency	The model is executed several times on the same prompt. The most frequent or most consistent response is selected.
Tree of Thoughts (ToT)	The model explores multiple reasoning branches simultaneously.
ReAct	The model alternates between phases of reasoning and operational phases (acting), such as consulting external sources or interacting with digital tools.

Table 3.5: Prompt techniques with their descriptions

Building on this comprehensive overview of prompting techniques, it was necessary to evaluate which approaches were most suitable for meeting the specific objectives of this research. In particular, returning to the primary aim of the study, the focus was placed on identifying prompting strategies that could best support management in real operational contexts.

From this perspective, the *Zero-Shot* approach proved to be more appropriate than the *One-Shot* and *Few-Shot* alternatives. A manager typically expects the model to provide a solution to a problem without relying on predefined examples, either due to limited domain-specific knowledge or constraints of time and resources. Including examples within the prompt does not reflect this scenario, whereas *Zero-Shot* prompting represents a more realistic condition. Moreover, from a computational standpoint, *Zero-Shot* enables the rapid processing of large volumes of data and questions, while respecting the time and resource limitations of this research.

Another technique well suited to the analyzed scenario is *Role Prompting*. As highlighted in the literature, this approach not only allows for the definition of a specific conversa-

tional tone but also leads to improved performance. In this study, the instruction *"You are a Project Manager"* was added to the prompt, so that responses would reflect a technical style aligned with a managerial perspective. Since this is a stylistic choice supported by well-established findings in the literature, role prompting was applied consistently across all analyzed scenarios.

In order to investigate possible improvements in response performance, and following the direction suggested by several academic contributions, an additional benchmark was designed for the single-choice (Q1) and numerical-answer (Q4) scenarios by introducing the *Chain of Thought (CoT)* technique. The prompt was enriched with the instruction *"Let's think step by step"*, intended to stimulate a gradual reasoning process before reaching the final solution. In these cases (Q1 and Q4), the CoT remains *implicit*: the reasoning unfolds internally, but the intermediate steps are not displayed in the answer. This configuration was chosen to examine how performance changes under such conditions and, more specifically, to test a setting aligned with situations in which managers primarily need a concise result without additional explanatory material.

In contrast, the numerical-answer scenario with reasoning (Q4+), *CoT* is required *explicitly*: the response must include not only the final value but also the logical progression leading to it. This option reflects an essential requirement in the field of Project Management, where the quality of a decision is assessed not only on the outcome but also on the reasoning that supports it, allowing potential weaknesses in the decision-making process to be identified.

Self-Consistency is often used in combination with *Chain of Thought*. In more complex tasks that require advanced reasoning, multiple logical pathways may emerge, and this technique allows the consideration of several responses generated from different reasoning chains. *Self-consistency* helps validate the robustness of the answers by comparing the various solutions produced by the model and selecting those that are the most frequent or consistent.

However, in the present study, this technique was not adopted as it would have resulted in a significant increase in computational and processing costs, without aligning with predefined analytical objectives. Therefore, it is considered an avenue for future research, where it can be explored to assess potential benefits in terms of accuracy and reliability

of the responses.

Unlike *Self-Consistency*, *Tree of Thoughts (ToT)* develops along multiple reasoning paths and autonomously selects the branch leading to the final result. However, this technique demands considerable computational resources, both in terms of processing power and memory, to handle multiple decision pathways, backtracking activities, and alternative explorations. Such requirements reduce its scalability and limit its applicability in contexts characterized by resource constraints or the need for rapid responses. For these reasons, this study chose not to adopt the technique, leaving its potential use to future research developments.

Similar reasoning applies to the *ReAct* technique, which alternates between the reasoning and acting phases, but also requires substantial computational effort. Its complexity limited its application at this stage of the research, though it may be considered in later phases where technological resources and contextual conditions are more favorable.

3.2.4 Final Benchmarks

The following Table 3.6 summarizes the details of each benchmark, including the datasets employed, the evaluation techniques applied, and the prompting strategies adopted. This overview provides a clear representation of the methodological choices made in each test scenario.

BENCHMARK	QUESTION	EVALUATION	PROMPTING
Benchmark 1	Single choice	Accuracy, Latency, Token, Cost	Zero-shot, Role prompting
Benchmark 2	Single choice	Accuracy, Latency, Token, Cost	Zero-shot, Role prompting, CoT
Benchmark 3	Numerical answer	Accuracy, Latency, Token, Cost	Zero-shot, Role prompting
Benchmark 4	Numerical answer	Accuracy, Latency, Token, Cost	Zero-shot, Role prompting, CoT
Benchmark 5	Numerical answer with reasoning	Human grade, Latency, Token, Cost	Zero-shot, Role prompting, CoT

Table 3.6: Benchmarks with question type, evaluation criteria, and prompting techniques

General structure of the benchmarks

All benchmarks shared a common methodological framework, based on four main metrics: *Accuracy, Latency, Cost* (calculated as a function of tokens used). Among these, *Accuracy* represented the central indicator and was analyzed at multiple levels:

- *Overall Accuracy*, measuring the global correctness of each model's answers;
- Accuracy by question type (*theoretical* and *numerical*), in order to highlight potential differences in behavior depending on content nature;
- Accuracy by difficulty level (*Easy, Medium, Hard*) for numerical questions only, with the aim of observing how model performance varied as task complexity increased.

These analyses made it possible to distinguish not only overall performance but also the models' sensitivity to question type and difficulty level. Finally, to integrate the set of evaluation criteria, the *Analytic Hierarchy Process (AHP)* was applied, enabling the synthesis of results into a comparative ranking of models that balances accuracy, time efficiency, and computational cost.

Benchmark 1

The first benchmark considered *Single-Choice* questions formulated in a *Zero-shot* setting with the addition of *Role Prompting* to steer the model toward behavior consistent with the decision-making context. The main objective was to establish a performance baseline in theoretical and numerical classification scenarios.

Benchmark 2

The second benchmark retained the same question type as B1 but added the *Chain-of-Thought* instruction. In this case, reasoning remained implicit (not shown in the final answer), allowing assessment of whether prompting step-by-step reasoning affected the correctness of choices.

Benchmark 3

The third benchmark focused on *Numerical* questions and required models to return only the value of the answer, with no explanation. This format enabled testing of "pure" calculation accuracy without explicit reasoning support.

Benchmark 4

In continuity with B3, the fourth benchmark added implicit *CoT* via the instruction *"Let's think step by step"*. The aim was to observe whether encouraging progressive reasoning could yield benefits, especially for more complex numerical items.

Benchmark 5

The fifth benchmark differed from the others as it evaluated not only the correctness of the final numerical result but also the quality of the reasoning made explicit by the model. To this end, a dedicated scoring system was introduced, assigning each LLM a maximum score of 1, distributed across three dimensions:

- *Calculation* (0–0.2): ability to correctly perform calculation steps;

- *Reasoning* (0–0.4): coherence and completeness of the reasoning provided;
- *Correctness* (0 or 0.4): accuracy of the final answer.

In addition to quantitative measurement, a qualitative analysis of reasoning errors was carried out, classified into two non-mutually exclusive categories:

- *Interpretation error*: related to misunderstandings of the problem statement or the incorrect use of available data;
- *Planning error*: stemming from flawed logical sequences, improper formula application, or disorganized solution steps.

From these evaluations, so-called category accuracies were derived, calculated as the ratio between the actual scores obtained and the theoretical maximum scores for each dimension. The final metrics therefore considered both traditional aspects (calculation, correctness) and qualitative aspects (reasoning, error type), providing a more granular representation of model performance.

3.3 Benchmark Implementation & Testing

After defining the benchmarks, the next phase concerned the experimentation on LLMs, with the objective of systematically analyzing their performance. The following section presents the models selected for testing and describes the practical implementation, with reference to the software and tools employed.

3.3.1 LLMs selection

The first step was to select the LLM models to be evaluated among the numerous solutions currently available. The rapid diffusion of these models in recent years has fostered the entry of many companies into the sector, giving rise to a broad and continuously evolving market. As highlighted in the literature and confirmed by an exploratory analysis conducted online, the current offering includes multiple models, each designed to meet specific usage needs.

These solutions differ primarily with respect to three efficiency parameters: performance, latency, and cost. Each developer proposes variants of their model in an attempt to optimize the combination of these factors. However, achieving efficiency across all three dimensions simultaneously is not feasible: no model can deliver high performance with low latency while also maintaining low costs. As a result, different versions are developed that prioritize one characteristic at the expense of the others.

An example of this approach is represented by several next-generation language models, which offer different variants to balance performance, latency, and cost. A flagship version may be designed to guarantee high performance but with higher costs and greater latency due to the complexity of the computations involved. Conversely, a version optimized for speed and affordability may sacrifice performance on complex tasks. Finally, an intermediate variant may represent a compromise among these factors, offering a balanced solution for scenarios with variable requirements.

In practice, following an in-depth review, and after excluding certain models for geographical reasons (e.g., *Grok* by *xAI*, not yet available in the UK or EU) or due to issues with API acquisition and/or malfunction (such as *LLaMA* by *Meta* and *Qwen* by *Alibaba*), the following providers were selected:

Opena AI

OpenAI, a U.S.-based company founded in 2015, is one of the key players in the field of generative artificial intelligence and in the development of state-of-the-art language models. With the release of the *GPT-5* series, the company introduced three model variants, each designed to address different requirements in terms of performance, cost, and latency. *GPT-5* represents the company's flagship model, characterized by strong reasoning capabilities and suitable for complex applications, though with longer response times and significantly higher costs. At the opposite end is *GPT-5 nano*, conceived to maximize speed and cost efficiency while sacrificing the ability to handle complex, cognitively demanding tasks. Positioned in between is *GPT-5 mini*, which offers a compromise among accuracy, speed, and economic sustainability.

The availability of these three differentiated variants motivated their inclusion in the comparative analysis, in order to evaluate how different trade-offs among performance, la-

tency, and cost may affect practical managerial applications. Previous models (the *GPT-4* series and earlier versions) were not considered, as they are deemed obsolete and have been officially replaced by the *GPT-5* generation.

Anthropic

Anthropic is a U.S.-based company founded in 2021 by former *OpenAI* members, with a strong focus on the safety and reliability of artificial intelligence systems. The *Claude* family of models stands out for its emphasis on reasoning capabilities and suitability for supporting complex scenarios. Within this family, *Claude Opus 4.1* represents the most advanced model, capable of delivering high-level performance but characterized by significantly higher costs and greater latency. *Claude Sonnet 4*, by contrast, offers an intermediate solution, balancing accuracy and speed at a more sustainable cost level. Finally, *Claude Haiku 3.5* prioritizes response speed and efficiency, while partially sacrificing the ability to manage particularly complex tasks.

Given the economic constraints of the present research, the comparative analysis included *Claude Sonnet 4* and *Claude Haiku 3.5*, while excluding *Claude Opus 4.1*, which was considered excessively costly ($15.00 / 1M input tokens and $75.00 / 1M output tokens) relative to the study's objectives.

Google

Google is one of the leading global players in the field of artificial intelligence, supported by *DeepMind*'s contributions to the development of deep learning. In 2023, it launched the *Gemini* family of models, the successor to the *PaLM 2* line, designed to provide multimodal capabilities and native integration with the Google Cloud ecosystem. The most recent *Gemini 2.5* generation is characterized by a particularly large context length (up to 1M tokens), enabling a wide range of use cases. *Gemini 2.5 Pro* is the most powerful and versatile model but also the most costly, with expenses varying depending on prompt length. *Gemini 2.5 Flash* is designed to optimize the price-performance ratio, offering solid performance at lower cost, while *Gemini 2.5 Flash-Lite* represents the most lightweight version, suitable for scenarios requiring high speed and low cost.

The *Gemini 2.5 Pro* version was not included in the analysis because, during preliminary testing, access through the model's beta API produced errors attributable to Google server malfunctions, which prevented correct code execution and, consequently, reliable evaluation.

DeepSeek

DeepSeek is a more recent Chinese provider that has distinguished itself in the LLM market through a highly competitive approach in terms of cost and accessibility, while still maintaining satisfactory baseline performance. The latest version (*V3.1*) features a maximum context length of 128k tokens and prices significantly lower than those applied by major international competitors. These characteristics make the model particularly attractive in scenarios where budget constraints play a decisive role.

In conclusion, the Table 3.7 reports the selected versions.

Owner	**Version**	**Context Length**	**Input price ($/Mtok)**	**Ooutput price ($/Mtok)**
OpenAI	GPT-5	400k	1.25	10.00
	GPT-5 mini	400k	0.25	2.00
	GPT-5 nano	400k	0.05	0.40
Anthropic	Claude Sonnet 4	200k	3.00	15.00
	Claude Haiku 3.5	200k	0.80	4.00
Google	Gemini 2.5 Flash	1000k	0.30	2.50
	Gemini 2.5 Flash-Lite	1000k	0.10	0.40
DeepSeek	DeepSeek V3.1	128k	0.56	1.68

Table 3.7: Comparison of LLM providers, version, context length, and pricing

The table also reports additional characteristics:

- *Version*: identifies the specific variant of the model released by the provider;
- *Context length*: indicates the maximum number of tokens a model can process in

a single interaction, including both input (prompt, instructions, documents) and output;

- *Input price*: represents the cost, expressed in U.S. dollars, for processing 1 million input tokens (i.e., the text provided to the model as a prompt). (The cache miss price was considered);
- *Output price*: represents the cost, expressed in U.S. dollars, for processing 1 million output tokens.

3.3.2 Implementation

After defining the benchmarks and selecting the models, the practical implementation phase was initiated. The experiments were conducted in *Google Colab*, an environment that enabled straightforward management of both dataset loading and interaction with the APIs of the different LLMs. Each notebook followed the same logical sequence: importing libraries, loading the questions from file, defining execution parameters, setting the system prompt, calling the model, and finally recording the responses together with the corresponding token consumption and estimated costs.

Library import

The first step common to all benchmarks was the import of the libraries required for running the scripts. Some core libraries were included in every script to provide essential functions:

- *Time* was used to measure execution duration;
- *Pandas*: enabled the reading and management of datasets in Excel format;
- *Google.colab.files* allowed datasets to be uploaded directly into the Colab environment, simplifying the handling of input data.

```
Import time
From google import google.colab.files
Import panda as pd
```

In addition, each provider requires its own dedicated package, which makes it necessary to use different libraries for interacting with the models. Table 3.8

Provider	**Library**
Anthropic	`!pip install -q anthropic` `from anthropic import Anthropic`
OpenAI	`!pip install -q openai` `from openai import OpenAI`
Google	`!pip install -q google-generativeai` `import google.generativeai as genai`
DeepSeek (CoT)	`!pip install -q deepseek` `from deepseek import DeepSeek`

Table 3.8: Library installation and import examples by provider

Finally, the auxiliary *re* library was added in the numerical-answer benchmarks, as it was necessary to correctly extract the numerical values produced by the model.

Loading questions from file

After importing the libraries, the next step in each benchmark was loading the dataset containing the questions and their corresponding correct answers. This phase was essential both to ensure consistency across tests and to maintain flexibility with respect to different task types.

The file upload was performed using the *files.upload()* function, which allows an Excel file to be uploaded directly from the local computer. Subsequently, with *pandas.read_excel()*, the data were read into a DataFrame, and the columns were renamed consistently as *"question"* and *"correct_answer"*.

```
from google.colab import files
uploaded = files.upload()
file_name = next(iter(uploaded))
```

```
df = pd.read_excel(file_name, header=None)
df.columns = ["question", "correct_answer"]
```

This procedure, identical across all benchmarks and models (*Claude, ChatGPT, Gemini, DeepSeek*), ensured that the pipeline consistently received a standardized data format as input.

Definition of execution parameters

Each benchmark was configured by setting the key parameters governing the interaction with the models: temperature, maximum number of tokens, and input/output costs.

- *Maximum number of tokens*: In all benchmarks, this parameter was set to the maximum value allowed by the provider for the specific model (for example, 8,192 for *Claude Haiku 3.5*). This approach avoided the risk of truncation in open-ended tasks, while acknowledging that in closed tasks the actual consumption remained much lower;
- *Input and output costs*: Cost calculations were based on the official pricing declared by the providers, distinguishing between input tokens (prompts) and output tokens (generated responses). For example, for *Claude Haiku 3.5* the cost is $0.80 per million input tokens and $4.00 per million output tokens.;
- *Input price*: represents the cost, expressed in U.S. dollars, for processing 1 million input tokens (i.e., the text provided to the model as a prompt). (The cache miss price was considered).

```
MAX_TOKENS  = 8192
in_cost   = 0.80 / 1_000_000
out_cost = 4.00 / 1_000_000
```

Definition of the system prompt

The definition of the system prompt represented a central step in the implementation phase, as it made it possible to put into practice the prompting techniques previously

discussed. While the execution parameters (temperature, maximum number of tokens, costs) remained relatively standardized, prompt design required significant adjustments depending on the benchmark type and the chosen strategy.

In all benchmarks, role prompting was applied through the constant instruction "You are a Project Manager". This choice aimed to give the responses a managerial and technical character, consistent with the perspective guiding the research scenario.

Furthermore, zero-shot prompting was adopted in all cases, without including explicit input-output examples, in order to reproduce conditions closer to real-world scenarios: a manager is expected to receive answers to new questions without the need for predefined examples.

- *Benchmark 1*: Only role prompting was applied, constraining the model to return a single letter between A and D. No additional reasoning cues were used

```
system_prompt = (
    "You are a Project Manager."
    "Answer the question you are asked with the letter from
    A to D."
    "Do not say anything else except the letter.")
```

- *Benchmark 2*: In addition to role prompting, Chain of Thought (CoT) was introduced with "Let's think step by step." The goal was to test whether encouraging internal reasoning improved correctness, while keeping the final output to a single letter

```
system_prompt = (
    "You are a Project Manager. "
    "Let's think step by step. "
    "Answer the question you are asked with the letter from
       A to
    D. "
    "Do not say anything else except the letter.")
```

- *Benchmark 3*: The objective was to obtain a purely numerical output; role prompting was used

```
system_prompt = (
    "You are a Project Manager. "
    "Given a numerical problem, return ONLY the final number
        in
    digits, "
    "without text, symbols, or units. "
    "Use the dot as the decimal separator (e.g., 1234.56).")
```

– *Benchmark 4*: As in Benchmark 3, but augmented with implicit CoT to encourage step-by-step internal reasoning before producing the final number

```
system_prompt = (
    "You are a Project Manager. "
    "Let's think step by step. "
    "Given a numerical problem, return ONLY the final number
        in
    digits, "
    "without text, symbols, or units. "
    "Use the dot as the decimal separator (e.g., 1234.56).")
```

– *Benchmark 5*: A more structured prompt combined role prompting and explicit CoT. To facilitate parsing and evaluation, a strict output format was imposed.

```
system_prompt = (
    "You are a Project Manager. "
    "Think step by step. Solve numerical problems showing
       clear,
    numbered steps "
    "with formulas and substitutions. "
    "After the reasoning, print the final answer on a new
       last
    line EXACTLY as: "
    "ANSWER=<number>")
```

Model call

Once the parameters and the system prompt were defined, the next step was the actual interaction with the model through the respective APIs. This phase was common to all benchmarks: for each question in the dataset, a request was generated to the selected model, passing as arguments the system prompt, the question, and the configuration parameters (temperature, max_tokens).

The structure of the call was almost identical for all providers (*Anthropic, OpenAI, Google, DeepSeek*):

```
response = client.messages.create(
    model=model,
    system=system_prompt,
    max_tokens=MAX_TOKENS,
    messages=[{"role": "user", "content": str(question)}])
```

The *response* object contained both the text generated by the model and the metadata related to token consumption and execution time.

The differences concerned the type of output expected and, consequently, the logic used to process the model's response:

- *Benchmark 1 and 2*: The model's output was reduced to a single letter (A–D). The script extracted the first valid occurrence contained in the response.

```
risposta = response.content[0].text.strip().lower()
prima_lettera = next((c for c in risposta if c in 'abcd'), '
    ')
print(prima_lettera)
```

- *Benchmark 3 and 4*: A function with regular expressions was used to isolate the final number from the generated text, discarding any unwanted characters.

```
import re
num_pat = re.compile(r'[-+]?\d+(?:[\.,]\d+)?(?:[eE][-+]?\d+)
    ?')

def only_number(s: str) -> str:
```

```
    m = num_pat.search(str(s))
    return m.group(0).replace(",", ".") if m else ""

raw = response.content[0].text.strip()
num = only_number(raw)
print(num)
```

– *Benchmark 5*: In this case, the model produced a detailed reasoning followed by a final line in the format ANSWER=<number>.

```
answer_pat = re.compile(r'ANSWER\s*=\s*([-+]?\d+(?:[
.,]\d+)?(?:[eE][-+]?\d+)?)')
num_pat_all = re.compile(r'[-+]?\d+(?:[.,]\d+)?(?:[eE][-+]?\
   d+)?
')

def extract_final_number(text: str) -> str:
    m = answer_pat.search(text)
    if m:
        return m.group(1).replace(",", ".")
    nums = num_pat_all.findall(text)
    return nums[-1].replace(",", ".") if nums else ""

raw = response.content[0].text.strip()
final_number = extract_final_number(raw)
print(final_number)
```

Recording responses, usage, and estimated costs

The final phase of each benchmark concerned the recording of the responses generated by the models, together with usage data (tokens, costs, and execution time). This step enabled the transformation of the model's output into a structured dataset, useful both for evaluating accuracy and for analyzing economic and computational efficiency.

In all scripts, the following values were computed:

- input tokens (prompt provided to the model);
- output tokens (generated response);
- estimated cost (calculated by multiplying tokens by the official rates);
- total execution time.

```
usage = response.usage
prompt_tokens = usage.input_tokens
completion_tokens = usage.output_tokens

total_prompt_tokens     += prompt_tokens
total_completion_tokens += completion_tokens
total_cost += prompt_tokens * in_cost + completion_tokens *
   out_cost

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Estimated total cost: {total_cost:.4f} $")
print(f"TOTAL EXECUTION TIME: {elapsed_time:.2f} seconds")
print(f"TOTAL INPUT TOKENS: {total_prompt_tokens}")
print(f"TOTAL OUTPUT TOKENS: {total_completion_tokens}")
```

For the specific benchmarks:

- *Benchmark 1 and 2*: Only the letter corresponding to the answer was recorded. Token and cost information was printed at the end of execution, without additional intermediate details.

```
print(first_letter)
```

- *Benchmark 3 and 4*: The extracted numerical value was saved, ignoring any accessory characters. Token usage and costs were also recorded, with only the final number printed.

```
print(num)
```

– *Benchmark 5*:In addition to the final number, the complete explanation generated by the model was stored, allowing for a qualitative analysis of the reasoning process. In this benchmark, the aim was not only to verify the correctness of the result but also to assess the quality of the reasoning.

```
print("QUESTION:", question)
print("REASONING:", raw)
print("FINAL RESULT:", final_number)
```

3.4 Statistical Significance Testing

In addition to the evaluation metrics already discussed, it was necessary to verify the statistical significance of the differences observed between the benchmarks. Simple variations in accuracy do not guarantee that such differences are due to the introduction of a prompting technique or to the task itself, rather than to random fluctuations.

For this purpose, the ***McNemar*** test was employed, a widely used non-parametric method for comparing the performance of two classifiers on the same data. The test does not focus on overall accuracy but rather on the discordant cases: that is, the instances where one model provides the correct answer while the other fails, and vice versa. The idea is to assess whether these discrepancies are evenly distributed or whether one situation clearly prevails. In the first case, no significant differences between the models can be detected, whereas in the second it is possible to conclude that one of the classifiers exhibits a real advantage.

In this research, the *McNemar* test was used mainly to compare benchmarks based on the same type of questions but differing in the use of Chain-of-Thought. It was also applied to scenarios with *implicit* and *explicit* CoT. Analyses were conducted both on the overall samples of 300 questions per benchmark and on sub-samples by question type (theoretical and numerical) and difficulty level (easy, medium, hard).

The tests were run using the software *Stata*, which provides the χ^2 statistic and the related p-values. Two approaches were considered: the **asymptotic p-value** based on the χ^2 approximation; the **exact p-value**, based on combinatorial calculations, more reliable with small samples or few discordances.

Since in this study the maximum sample size was 300 questions and the number of discordances was often limited, the exact p-value was used as the main reference. Results were interpreted according to the conventional threshold of 95% significance ($\alpha = 0.05$): a difference was considered significant only when the p-value was below this level.

The use of the McNemar test strengthens the methodological validity of the analysis. It reduces the risk of over-interpreting marginal differences and provides a more reliable picture of LLM performance. The outcomes of the tests are presented in the next chapter, alongside the descriptive metrics, to give a complete evaluation in both descriptive and inferential terms.

Chapter 4

Results

4.1 Introduction

This chapter presents the results of the experimental activity, organized on several levels of analysis. It opens with the outcomes of the survey, which collected evaluators preferences and defined the weights used in the Analytic Hierarchy Process (AHP).

The section **Benchmark-level Results** reports the findings of the individual benchmarks, describing model performance in terms of overall accuracy, question type and difficulty, as well as operational parameters such as cost and latency. Each block of results is then summarized through the AHP, which makes it possible to combine different dimensions into a single comparative index.

The section **Cross-benchmark Comparison** adopts a transversal perspective, comparing the results obtained across the various benchmarks. In this context, statistical significance tests (McNemar) are also considered, in order to verify whether the observed differences should be interpreted as real effects or as random fluctuations.

The structure of the chapter thus makes it possible to move from the detailed analysis of individual benchmarks to a comparative and statistically validated reading, laying the groundwork for the critical discussion developed in the following chapter.

4.2 Survey

The survey is used to determine the relative importance of the evaluation criteria to be integrated into the Analytic Hierarchy Process (AHP). Participants, project managers at Amazon Luxembourg, were asked to assign each criterion, accuracy, cost, and latency, an importance score on a scale from 1 to 7, following the standard AHP approach.

A total of 30 evaluators were interviewed, and the reported mean values therefore reflect the average of the preferences expressed by this sample. The aggregation of the responses made it possible to derive the mean values, which were subsequently normalized and used as weights within the multi-criteria process. The results highlight a clear priority assigned to accuracy (mean 6.37), followed by latency (4.93) and, to a lesser extent, cost (4.33).

The results of the survey are reported below, respectively for the criteria of **Accuracy** (Figure 4.1), **Cost** (Figure 4.2), and **Latency** (Figure 4.3).

Table 4.1 reports the final normalized weights:

Criterion	Mean Value	Normalized weight
Accuracy	6.37	0.407
Cost	4.33	0.277
Latency	4.93	0.315

Table 4.1: Survey results

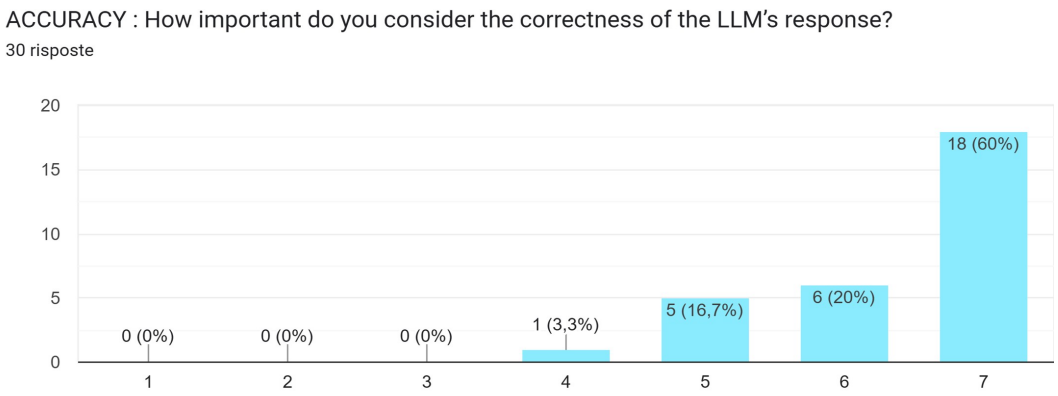


Figure 4.1: Survey results for Accuracy

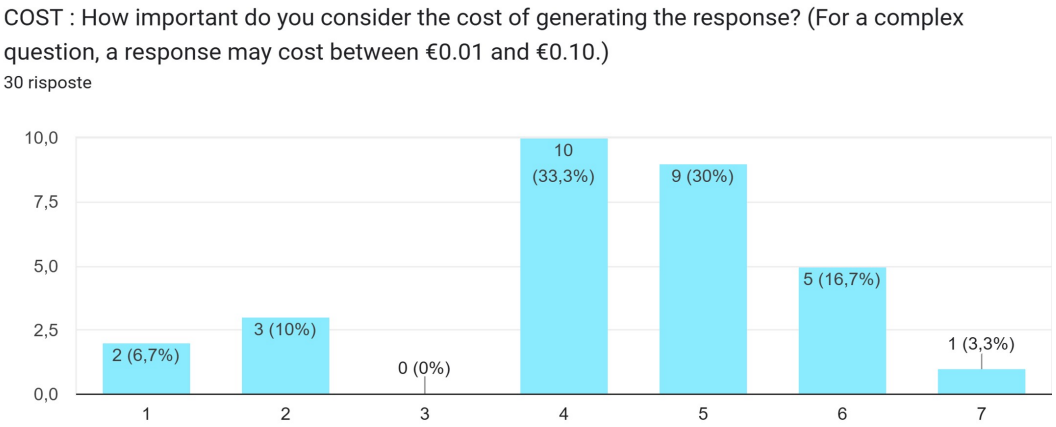


Figure 4.2: Survey results for Cost

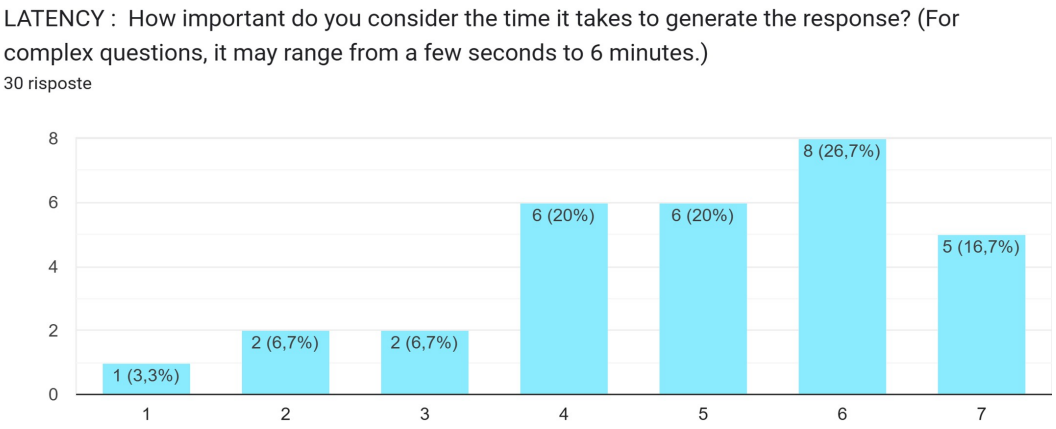


Figure 4.3: Survey results for Latency

4.3 Benchmark-level Results

This section reports the results obtained from the different benchmarks designed. The analysis follows a progressive structure: for each benchmark, the overall and disaggregated accuracy values are presented (distinguishing between theoretical and numerical questions, as well as by difficulty level), followed by the comparison with the human evaluation threshold and the description of the operational parameters (cost and latency). Finally, the results are synthesized through the application of the Analytic Hierarchy Process (AHP), which makes it possible to integrate the different criteria into a single

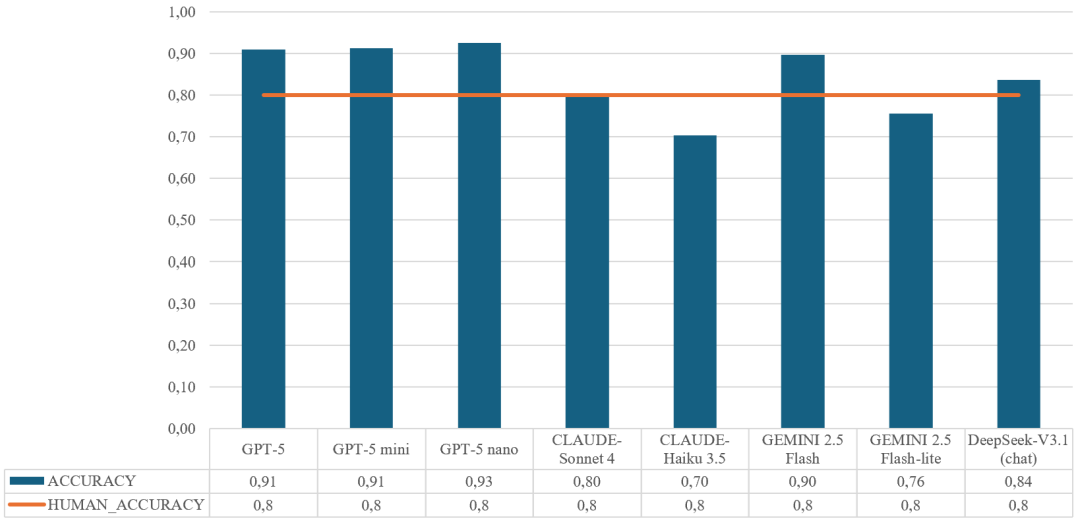
composite index and to produce a final ranking of the models. The weights used in the AHP were derived from the survey.

The following paragraphs present in detail the performance of the models across the five benchmarks.

4.3.1 Benchmark 1 Results

The first benchmark, based on single-choice questions, provided a baseline for evaluating model performance. Overall accuracy ranges from a minimum of 0.70 (Claude Haiku 3.5) to a maximum of 0.93 (GPT-5 nano), with intermediate values for GPT-5 (0.91), GPT-5 mini (0.91), Gemini 2.5 Flash (0.90), DeepSeek-V3.1 (0.84), Claude Sonnet 4 (0.80), and Gemini 2.5 Flash-Lite (0.76).

The comparison with human accuracy (set at 0.8) positions model results against a human baseline, the average performance expected from a trained individual with domain-specific knowledge.On this basis, GPT-5 (0.91), GPT-5 mini (0.91), GPT-5 nano (0.93), Gemini 2.5 Flash (0.90), and DeepSeek-V3.1 (0.84) outperform the human average. Claude Sonnet 4 matches the human benchmark, whereas all remaining models perform below this threshold.(Figure 4.4)

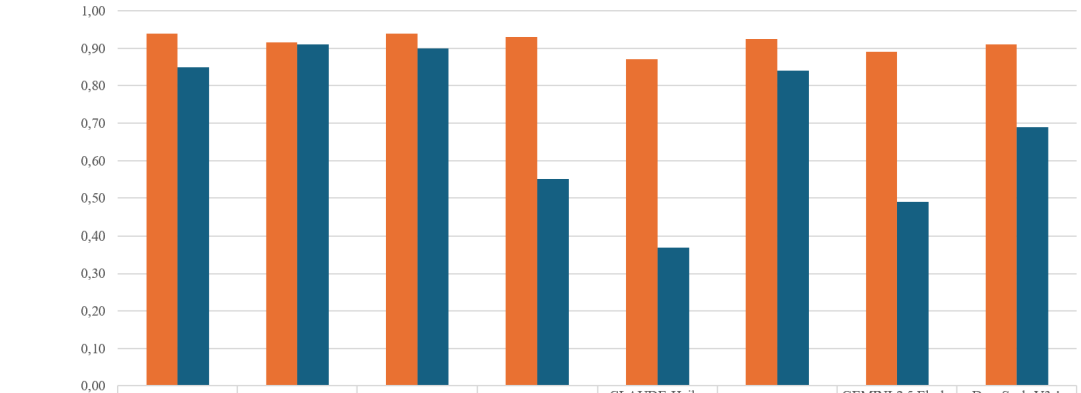


	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY	0,91	0,91	0,93	0,80	0,70	0,90	0,76	0,84
HUMAN_ACCURACY	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Figure 4.4: Accuracy of LLMs Compared to Human Baseline (Benchmark 1)

A more detailed analysis, distinguishing between theoretical questions (Accuracy_T) and numerical questions (Accuracy_N), highlights several relevant discrepancies. Some

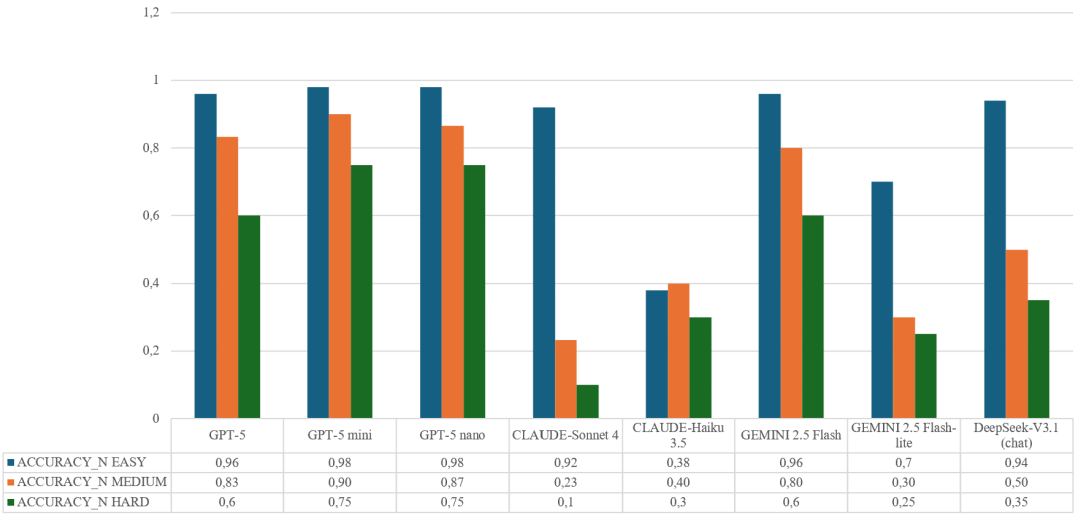
models present balanced values (GPT-5, GPT-5 mini, GPT-5 nano, Gemini 2.5 Flash), while others show greater heterogeneity, such as Claude Haiku 3.5 (0.87 T vs. 0.37 N), Claude Sonnet 4 (0.93 T vs. 0.55 N), DeepSeek-V3.1 (0.91 T vs. 0.69 N), and Gemini Flash-Lite (0.89 T vs. 0.49 N). (Figure 4.5)



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY_T	0,94	0,92	0,94	0,93	0,87	0,93	0,89	0,91
ACCURACY_N	0,85	0,91	0,9	0,55	0,37	0,84	0,49	0,69

Figure 4.5: Comparison of Theoretical vs. Numerical Accuracy (Benchmark 1)

Within the numerical subset, an additional analysis was performed by difficulty level (easy, medium, hard) (Figure 4.6). For the *easy* questions, almost all LLMs achieved high accuracy scores, including GPT-5 (0.96), GPT-5 mini (0.98), GPT-5 nano (0.98), Claude Sonnet 4 (0.92), Gemini 2.5 Flash (0.96), and DeepSeek-V3.1 (0.94). In contrast, substantially lower performance was observed for Claude Haiku 3.5 (0.38) and Gemini Flash-Lite (0.70). On *medium* questions, GPT-5 (0.83), GPT-5 mini (0.90), GPT-5 nano (0.87), with Gemini 2.5 Flash at 0.80. In the same category, several models showed significant difficulties, including Claude Sonnet 4 (0.23), Claude Haiku 3.5 (0.40), Gemini Flash-Lite (0.30), and DeepSeek-V3.1 (0.50). Finally, on *hard* questions, the highest accuracies were again achieved by GPT-5 mini and GPT-5 nano (0.75), followed by GPT-5 and Gemini 2.5 Flash (0.60). The lowest values were reported by Claude Haiku 3.5 (0.30), Gemini Flash-Lite (0.25), and Claude Sonnet 4 (0.10).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY_N EASY	0,96	0,98	0,98	0,92	0,38	0,96	0,7	0,94
ACCURACY_N MEDIUM	0,83	0,90	0,87	0,23	0,40	0,80	0,30	0,50
ACCURACY_N HARD	0,6	0,75	0,75	0,1	0,3	0,6	0,25	0,35

Figure 4.6: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 1)

From the perspective of operational parameters, the results highlight notable differences. GPT-5 recorded the highest cost ($2.21) and latency (5113 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.31 and $0.085) with similar latencies (2467 s and 2538 s). Gemini 2.5 Flash presented a cost of $0.014 and a latency of 1889 s, while Gemini 2.5 Flash-Lite stood out for its minimum cost ($0.0046) and latency of 286 s. Claude Sonnet 4 and Claude Haiku 3.5 recorded $0.3921 and $0.048, with latencies of 957 s and 353 s, respectively. Finally, DeepSeek-V3.1 reported a cost of $0.0264 and a latency of 450 s. Table 4.2

LLM	**Accuracy**	**Cost ($)**	**Latency (s)**
GPT-5	0.91	2.2078	5112.63
GPT-5 mini	0.91	0.3085	2466.60
GPT-5 nano	0.93	0.0619	2538.44
Claude-Sonnet 4	0.80	0.3921	956.91
Claude-Haiku 3.5	0.70	0.0480	353.47
Gemini 2.5 Flash	0.90	0.0141	1888.55
Gemini 2.5 Flash-lite	0.76	0.0046	285.51
DeepSeek-V3.1 (chat)	0.84	0.0264	450.26

Table 4.2: Performance comparison of LLMs in terms of accuracy, cost, and latency

To integrate the different evaluation criteria, the Analytic Hierarchy Process (AHP) was applied, combining accuracy, cost, and latency into a single synthetic index. The final

ranking resulting from the AHP is reported in Table 4.3. The results place GPT-5 nano in first position, followed by DeepSeek-V3.1 and Gemini 2.5 Flash. In the subsequent positions are Gemini 2.5 Flash-Lite, GPT-5 mini, Claude Haiku 3.5, GPT-5 and finally Claude Sonnet 4,.

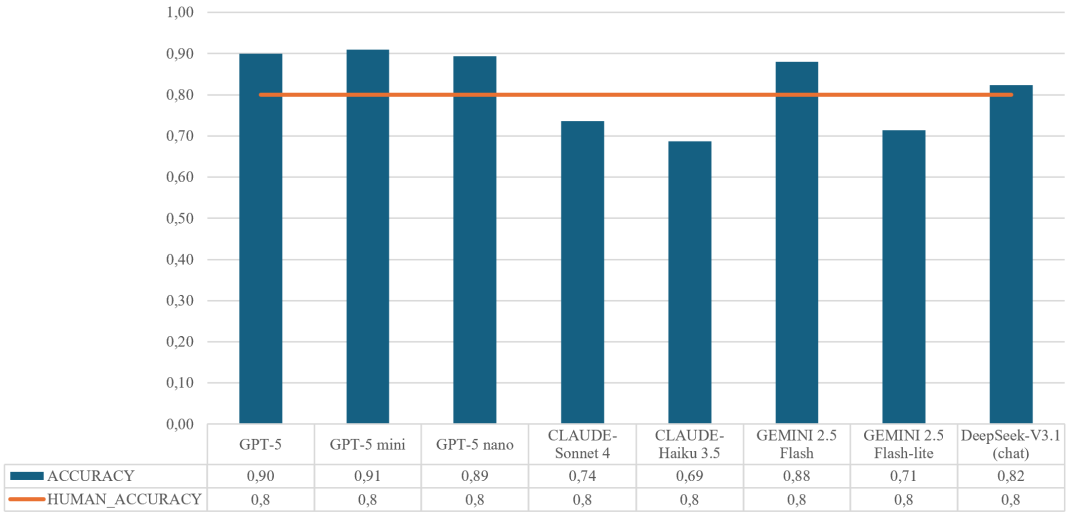
Rank	Model
1	GPT-5 nano
2	DeepSeek-V3.1 (chat)
3	Gemini 2.5 Flash
4	Gemini 2.5 Flash-Lite
5	GPT-5 mini
6	Claude-Haiku 3.5
7	GPT-5
8	Claude-Sonnet 4

Table 4.3: Final ranking of models according to the AHP index (Benchmark 1).

4.3.2 Benchmark 2 Results

The second benchmark assessed model performance on single-choice questions with the addition of the Chain-of-Thought (CoT) technique, in order to examine the effect of reasoning on answer quality.

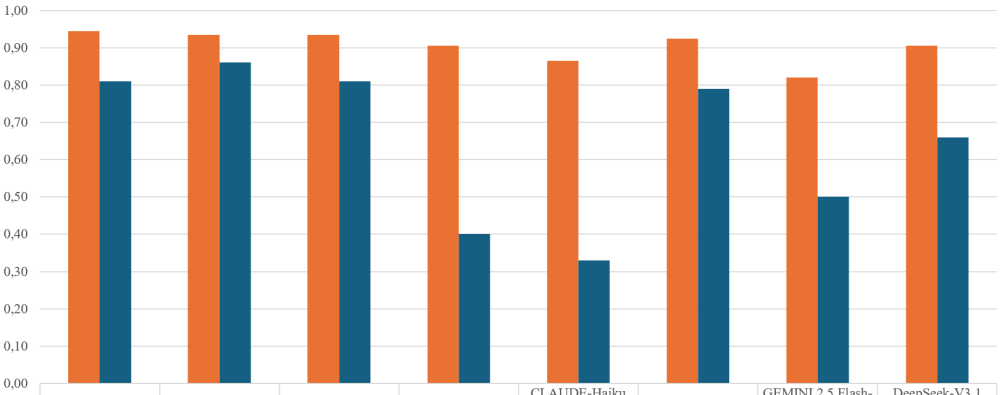
The comparison between model accuracy and human accuracy (set at 0.8, taken as the reference corresponding to the average performance expected from a trained individual with domain-specific knowledge) highlights notable differences. GPT-5 (0.90), GPT-5 mini (0.91), GPT-5 nano (0.89), Gemini 2.5 Flash (0.88) and DeepSeek-V3.1 (0.82) exceed the human baseline. Claude Sonnet 4 (0.74) is positioned relatively close to the human level, while Claude Haiku 3.5 (0.69) and Gemini 2.5 Flash-Lite (0.71) remain clearly below it. (Figure 4.7)



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY	0,90	0,91	0,89	0,74	0,69	0,88	0,71	0,82
HUMAN_ACCURACY	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Figure 4.7: Accuracy of LLMs Compared to Human Baseline (Benchmark 2)

The distinction between theoretical questions (Accuracy_T) and numerical questions (Accuracy_N) (Figure 4.8) highlights a certain variability. GPT-5 mini and GPT-5 nano show balanced performance (0.94 T – 0.86 N and 0.94 T – 0.81 N, respectively), while other models display marked discrepancies: Claude Sonnet 4 records a theoretical value of 0.91 but a numerical value of 0.40, Claude Haiku 3.5 scores 0.87 T and 0.33 N, Gemini Flash-Lite 0.82 T and 0.50 N, and DeepSeek-V3.1 0.91 T and 0.66 N.

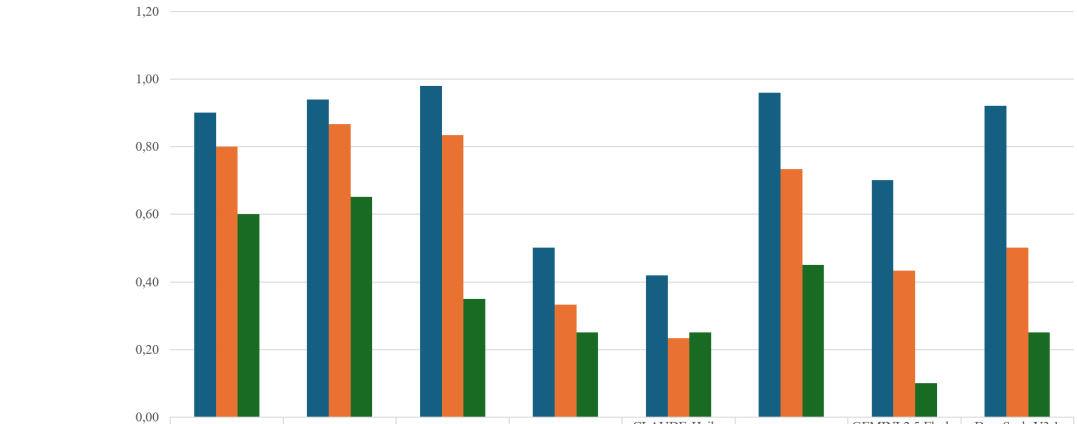


	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY_T	0,95	0,94	0,94	0,91	0,87	0,93	0,82	0,91
ACCURACY_N	0,81	0,86	0,81	0,40	0,33	0,79	0,50	0,66

Figure 4.8: Comparison of Theoretical vs. Numerical Accuracy (Benchmark 2)

For numerical questions divided by difficulty level (easy, medium, hard), the data reported in Figure 4.9 show clear differences. In *easy* questions, the highest values are achieved by Gemini 2.5 Flash (0.96), GPT-5 mini (0.94), and GPT-5 nano (0.98). On *medium* questions, scores remain high for GPT-5 (0.80), GPT-5 mini (0.87), and GPT-5

nano (0.83), while models such as Claude Sonnet 4 (0.33) and Claude Haiku 3.5 (0.23) show notable difficulties. For the *hard* questions, the performance of all LLMs decreases markedly. Within this setting, GPT-5 (0.60) and GPT-5 mini (0.65) still exhibit the highest scores, whereas the lowest values are observed for Gemini 2.5 Flash-Lite (0.10) and for Claude Sonnet 4, Claude Haiku 3.5, and DeepSeek-V3.1 (all at 0.25).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY_N EASY	0,90	0,94	0,98	0,50	0,42	0,96	0,70	0,92
ACCURACY_N MEDIUM	0,80	0,87	0,83	0,33	0,23	0,73	0,43	0,50
ACCURACY_N HARD	0,60	0,65	0,35	0,25	0,25	0,45	0,10	0,25

Figure 4.9: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 2)

From the perspective of operational parameters (Table 4.4), GPT-5 registered the highest cost ($2.30) and the longest latency (4831 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.36 and $0.13) and latencies of 2790 s and 2681 s, respectively. Gemini 2.5 Flash achieved a cost of $0.024 with a latency of 1975 s, while Gemini Flash-Lite recorded the lowest cost overall ($0.0223). Claude Sonnet 4 and Claude Haiku 3.5 incurred costs of $0.58 and $0.051, with latencies of 829 s and 314 s, respectively, the latter being the lowest latency observed. Finally, DeepSeek-V3.1 reported a cost of $0.033 and a latency of 590 s.

LLM	Accuracy	Cost ($)	Latency (s)
GPT-5	0.90	2.2962	4830.74
GPT-5 mini	0.91	0.3583	2789.82
GPT-5 nano	0.89	0.1282	2681.42
Claude-Sonnet 4	0.74	0.5819	1182.77
Claude-Haiku 3.5	0.69	0.0514	322.40
Gemini 2.5 Flash	0.88	0.0238	1975.48
Gemini 2.5 Flash-lite	0.71	0.0223	455.84
DeepSeek-V3.1 (chat)	0.82	0.0326	589.79

Table 4.4: Performance comparison of LLMs in terms of accuracy, cost, and latency

Finally, the application of the Analytic Hierarchy Process (AHP) made it possible to synthesize accuracy, cost, and latency into an overall ranking (Table 4.5). The results place DeepSeek-V3.1 in first position, followed by GPT-5 nano and Gemini 2.5 Flash. Subsequent positions are occupied by GPT-5 mini, Gemini 2.5 Flash-Lite, Claude Haiku 3.5, GPT-5 and Claude Sonnet 4.

Rank	Model
1	DeepSeek-V3.1 (chat)
2	GPT-5 nano
3	Gemini 2.5 Flash
4	GPT-5 mini
5	Gemini 2.5 Flash-Lite
6	Claude-Haiku 3.5
7	GPT-5
8	Claude-Sonnet 4

Table 4.5: Final ranking of models according to the AHP index (Benchmark 2).

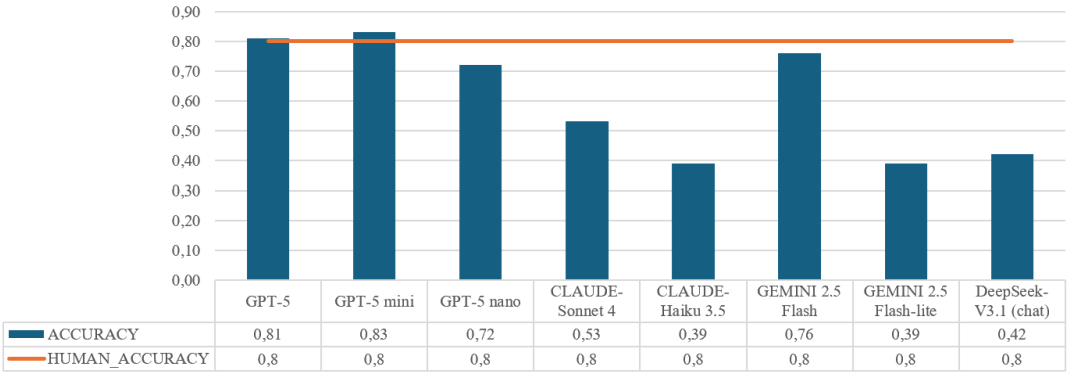
4.3.3 Benchmark 3 Results

The third benchmark evaluated model performance on numerical response questions, where no textual explanation was required and only the provision of a value was expected.

This format allows for a direct assessment of the ability to perform calculations and return the correct numerical result.

Overall accuracy (Figure 4.10) ranges from 0.81 (GPT-5 mini) to 0.39 (Claude Haiku 3.5 and Gemini 2.5 Flash-Lite). High results were achieved by GPT-5 (0.81), GPT-5 nano (0,72) and Gemini 2.5 Flash (both 0.76). Considerably lower values were observed for Claude Sonnet 4 (0.53) and DeepSeek-V3.1 (0.42).

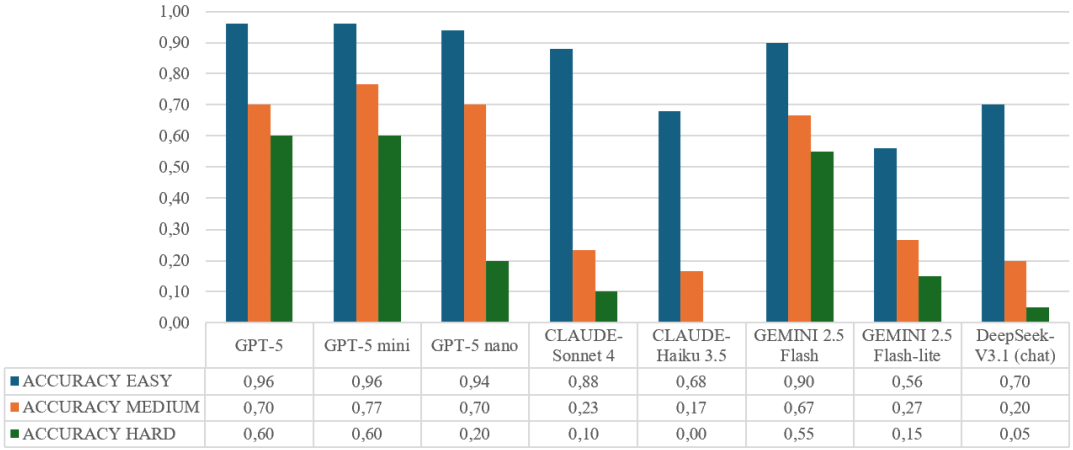
The comparison with human evaluation (threshold 0.8) shows that only GPT-5 (0.81) and GPT-5 mini(0.83) exceed the reference level. All other models fall below the threshold: GPT-5 nano (0.72), Gemini 2.5 Flash (0.76), Claude Sonnet 4 (0.53), Claude Haiku 3.5 (0.39), Gemini Flash-Lite (0.39), and DeepSeek-V3.1 (0.42).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY	0,81	0,83	0,72	0,53	0,39	0,76	0,39	0,42
HUMAN_ACCURACY	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Figure 4.10: Accuracy of LLMs Compared to Human Baseline (Benchmark 3)

The breakdown by difficulty level (easy, medium, hard) highlights substantial differences (Figure 4.11). On *easy* questions, very high accuracies are reported for GPT-5 (0.96), GPT-5 mini (0.96), GPT-5 nano (0.94) and Gemini Flash (0.90). On *medium* questions, values remain strong for GPT-5 mini (0.77), GPT-5 and GPT-5 nano (both 0.70) and Gemini Flash (0.67), while models such as Claude Sonnet 4 (0.23), Claude Haiku 3.5 (0.17), Gemini Flash-Lite (0.27), and DeepSeek-V3.1 (0.20) show significant difficulties. On *hard* questions, GPT-5, GPT-5 mini (both 0.60) and Gemini Flash (0.55) again confirm superior performance compared to the other models, which remain at very low values.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY EASY	0,96	0,96	0,94	0,88	0,68	0,90	0,56	0,70
ACCURACY MEDIUM	0,70	0,77	0,70	0,23	0,17	0,67	0,27	0,20
ACCURACY HARD	0,60	0,60	0,20	0,10	0,00	0,55	0,15	0,05

Figure 4.11: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 3)

The operational parameters (Table 4.6) highlight substantial differences. GPT-5 shows the highest cost ($2.10) and the longest latency (4200 s). The GPT-5 mini and nano versions report lower costs ($0.21 and $0.10) with latencies of 1644 s and 2186 s. Gemini 2.5 Flash records a cost of $0.0072 and a latency of 849.32 s, while Gemini Flash-Lite reports the lowest cost ($0.0022) and the lowest latency (86.66 s) . The Claude models present intermediate values: Sonnet 4 ($0.32 and 469 s) and Haiku 3.5 ($0.020 and 103 s). Finally, DeepSeek-V3.1 registers $0.026 and 402 s.

LLM	**Accuracy**	**Cost ($)**	**Latency (s)**
GPT-5	0.81	2.103	4200.29
GPT-5 mini	0.83	0.2056	1644.28
GPT-5 nano	0.72	0.1011	2186.05
Claude-Sonnet 4	0.53	0.3248	469.37
Claude-Haiku 3.5	0.39	0.0197	103.14
Gemini 2.5 Flash	0.76	0.0072	849.32
Gemini 2.5 Flash-lite	0.39	0.0022	86,66
DeepSeek V3.1 (chat)	0.42	0.0261	402.47

Table 4.6: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to synthesize the three evaluation criteria (accuracy, cost, and latency) into an overall ranking (Table 4.7). The results place GPT-5 mini in first position, followed by Gemini 2.5 Flash

and GPT-5. The subsequent positions are occupied by GPT-5 nano, Claude Haiku 3.5, Gemini 2.5 Flash-Lite, DeepSeek-V3.1 and Claude Sonnet 4.

Rank	Model
1	GPT-5 mini
2	Gemini 2.5 Flash
3	GPT-5
4	GPT-5 nano
5	Claude-Haiku 3.5
6	Gemini 2.5 Flash-Lite
7	DeepSeek-V3.1 (chat)
8	Claude-Sonnet 4

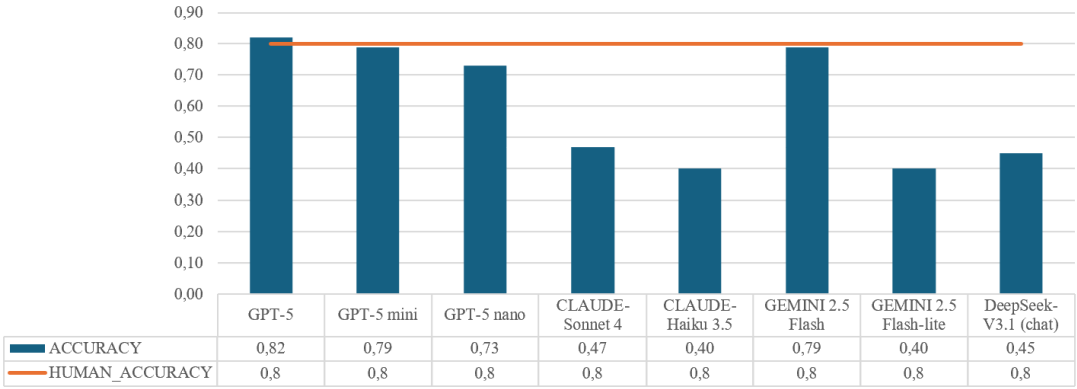
Table 4.7: Final ranking of models according to the AHP index (Benchmark 3).

4.3.4 Benchmark 4 Results

The fourth benchmark evaluated the performance of LLMs on numerical response questions using the chain-of-thought (CoT) technique. The goal was to assess whether the addition of implicit reasoning could improve accuracy.

Overall accuracy (Figure 4.12) ranges from a maximum of 0.82 (GPT-5) to a minimum of 0.40 (Claude Haiku 3.5 and Gemini Flash-Lite). GPT-5 mini and Gemini 2.5 Flash reach 0.79, while GPT-5 nano achieves 0.73 and DeepSeek-V3.1 remains at 0.45. Claude Sonnet 4 shows an intermediate value of 0.47.

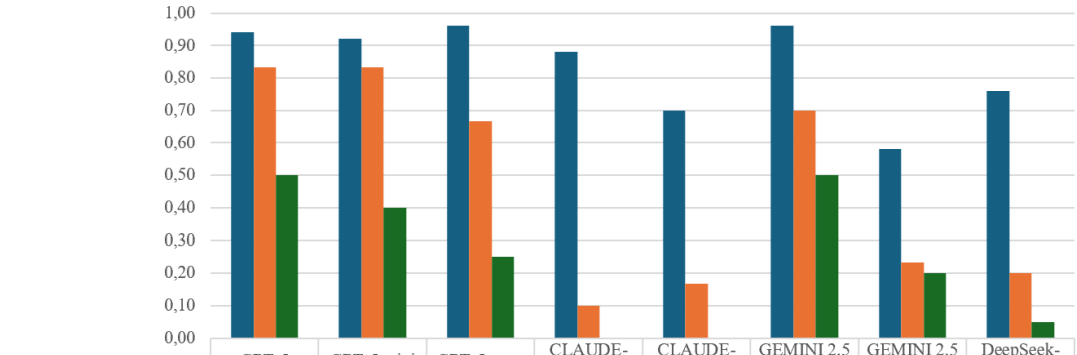
Comparison with human evaluation (threshold 0.8) highlights that only GPT-5 (0.82) exceeds the reference level. All other models remain below: GPT-5 mini (0.79), GPT-5 nano (0.73), Gemini 2.5 Flash (0.79), Claude Sonnet 4 (0.47), Claude Haiku 3.5 and Gemini Flash-Lite (both 0.40), and DeepSeek-V3.1 (0.45).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY	0,82	0,79	0,73	0,47	0,40	0,79	0,40	0,45
HUMAN_ACCURACY	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Figure 4.12: Accuracy of LLMs Compared to Human Baseline (Benchmark 4)

The breakdown by difficulty level (easy, medium, hard) shows differentiated patterns (Figure 4.13). In easy questions, high values are observed for GPT-5 (0.94), GPT-5 mini (0.92), GPT-5 nano and Gemini 2.5 Flash (both 0.96). In medium questions, the accuracy remains high for GPT-5 and GPT-5 mini (0.83), but is lower for GPT-5 nano (0.67) and Gemini Flash (0.70). In hard questions, GPT-5 and Gemini Flash maintain relatively high values (0.50), followed by GPT-5 mini (0.40), while the other models show very low performance, in some cases close to zero (Claude Sonnet 4 and Claude Haiku 3.5).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
ACCURACY EASY	0,94	0,92	0,96	0,88	0,70	0,96	0,58	0,76
ACCURACY MEDIUM	0,83	0,83	0,67	0,10	0,17	0,70	0,23	0,20
ACCURACY HARD	0,50	0,40	0,25	0,00	0,00	0,50	0,20	0,05

Figure 4.13: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 4)

The operational parameters (Table 4.8) show significant differences. GPT-5 recorded the highest cost ($2.04) and latency (3203 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.27 and $0.10) with latencies of 2719 s and 2153 s. Gemini 2.5 Flash showed a low cost ($0.0095) and latency of 1035 s, while Gemini Flash-Lite registered the lowest cost ($0.0031) and Claude Haiku 3.5 reported the lowest latency of 105 s. DeepSeek-

V3.1 was placed at $0.025 and 422 s while Claude Sonnet 4 recorded a cost of $0.38 and a latency of 522 s.

LLM	Accuracy	Cost ($)	Latency (s)
GPT-5	0.82	2.037	3203,36
GPT-5 mini	0.79	0.2698	2718,98
GPT-5 nano	0.73	0.1031	2153,39
Claude-Sonnet 4	0.47	0.3769	521,69
Claude-Haiku 3.5	0.40	0.0234	104,55
Gemini 2.5 Flash	0.79	0.0095	1034,58
Gemini 2.5 Flash-lite	0.40	0.0031	244.39
DeepSeek-V3.1 (chat)	0.45	0.0252	422,1

Table 4.8: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to integrate accuracy, cost, and latency into a synthetic ranking (Table **??**). The results place Gemini 2.5 Flash in the first position, followed by Claude Haiku 3.5 and GPT-5 mini. Subsequent positions are occupied by Gemini 2.5 Flash-Lite, GPT-5 nano, GPT-5, DeepSeek-V3.1 and Claude Sonnet 4.

Rank	Model
1	Gemini 2.5 Flash
2	Claude-Haiku 3.5
3	GPT-5 mini
4	Gemini 2.5 Flash-Lite
5	GPT-5 nano
6	GPT-5
7	DeepSeek-V3.1 (chat)
8	Claude-Sonnet 4

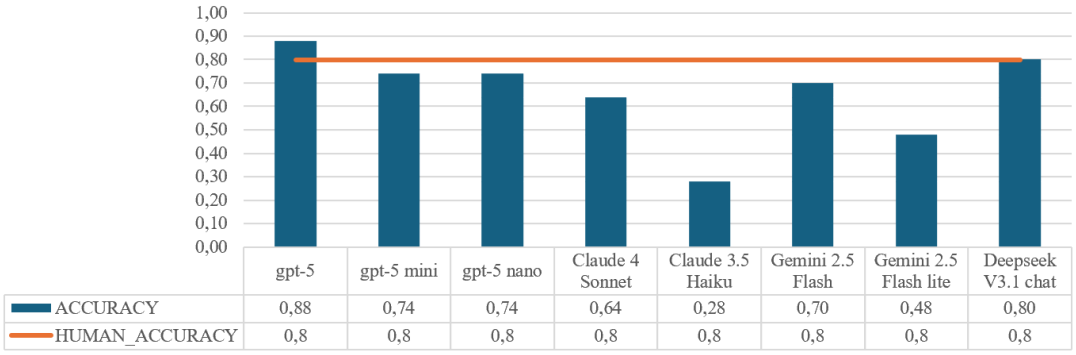
Table 4.9: Final ranking of models according to the AHP index (Benchmark 4).

4.3.5 Benchmark 5 Results

The fifth benchmark analyzed the performance of LLMs on numerical response questions requiring explicit reasoning. In addition to providing the numerical result, the models also produced detailed explanations of the process, which were evaluated in terms of accuracy (considering correctness only), calculation ability, reasoning coherence, and type of reasoning error.

Taking into account only the correctness of the numerical value, the results shown in Figure 4.14 range from a maximum of 0.88 (GPT-5) to a minimum of 0.28 (Claude Haiku 3.5). GPT-5 mini and GPT-5 nano achieved identical performance levels of 0.74, followed by Gemini 2.5 Flash with a score of 0.70. Claude Sonnet 4 reached a lower value of 0.60. DeepSeek-V3.1 recorded a score of 0.80, whereas Gemini Flash-Lite exhibited performance at 0.48.

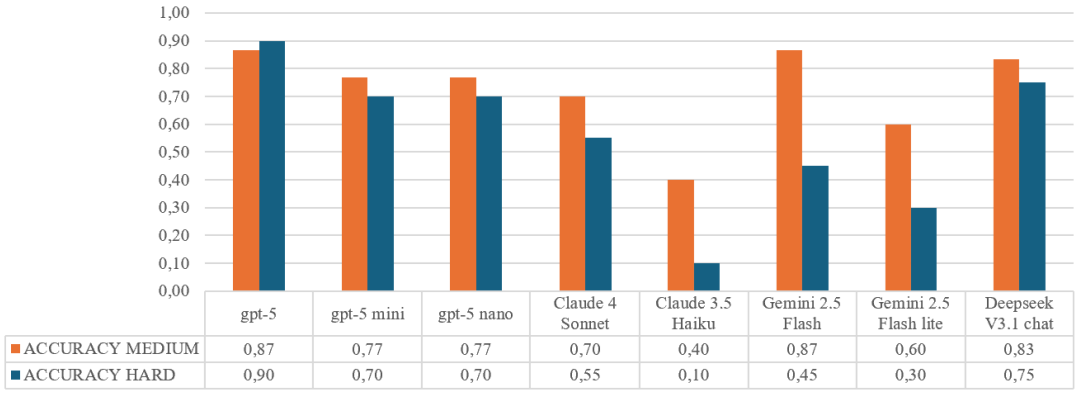
The comparison against the human evaluation benchmark, set at a threshold of 0.8, showed that only GPT-5 surpassed this level, achieving a score of 0.88. DeepSeek-V3.1 matched the benchmark exactly, reaching the threshold without exceeding it. All remaining models recorded performance values below the human reference level.



	gpt-5	gpt-5 mini	gpt-5 nano	Claude 4 Sonnet	Claude 3.5 Haiku	Gemini 2.5 Flash	Gemini 2.5 Flash lite	Deepseek V3.1 chat
ACCURACY	0,88	0,74	0,74	0,64	0,28	0,70	0,48	0,80
HUMAN_ACCURACY	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8

Figure 4.14: Accuracy of LLMs Compared to Human Baseline (Benchmark 5)

The breakdown by difficulty level (medium and hard) shows different patterns (Figure 4.15). In *medium* questions, GPT-5 and Gemini 2.5 Flash achieve the highest value (0.87), followed by DeepSeek-V3.1 (0.83). In *hard* questions, the highest values are observed for GPT-5 (0.90), DeepSeek-V3.1 (0.75), GPT-5 mini and GPT-5 nano (both 0.70), while the lowest scores are recorded for Claude Haiku 3.5 (0.10) and Gemini 2.5 Flash-Lite (0.30).



	gpt-5	gpt-5 mini	gpt-5 nano	Claude 4 Sonnet	Claude 3.5 Haiku	Gemini 2.5 Flash	Gemini 2.5 Flash lite	Deepseek V3.1 chat
ACCURACY MEDIUM	0,87	0,77	0,77	0,70	0,40	0,87	0,60	0,83
ACCURACY HARD	0,90	0,70	0,70	0,55	0,10	0,45	0,30	0,75

Figure 4.15: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 5)

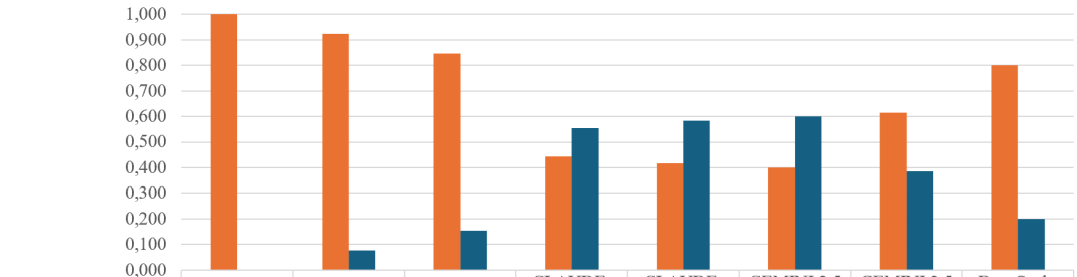
The analysis of the calculation and reasoning components (Table 4.10) indicates that calculation performance is generally strong across the evaluated models, with the notable exception of Claude Haiku 3.5, which achieved a lower score of 0.53. For the remaining models, calculation scores ranged from 0.75 for Gemini Flash-Lite to a perfect score of 1.00 for GPT-5. Reasoning performance exhibited a wider dispersion across models, with scores ranging from 0.913 for GPT-5 to 0.535 for Claude Haiku 3.5. GPT-5 mini and GPT-5 nano achieved comparable reasoning scores of 0.805 and 0.815, respectively. DeepSeek-V3.1 also demonstrated strong reasoning capabilities, reaching a high value of 0.895.

LLM	**Calculation**	**Reasoning**
GPT-5	1.00	0.913
GPT-5 mini	0.93	0.805
GPT-5 nano	0.95	0.815
Claude-Sonnet 4	0.88	0.790
Claude-Haiku 3.5	0.53	0.535
Gemini-2.5 Flash	0.91	0.815
Gemini-2.5 Flash-Lite	0.75	0.685
DeepSeek-v3.1	0.95	0.895

Table 4.10: Calculation and reasoning scores for LLMs (Benchmark 5).

A qualitative analysis of error types was conducted to better understand the reasoning failures observed in lower-performing models, distinguishing between interpretation and planning errors (Figure 4.16)). The values show that interpretation errors tend to be more

frequent than planning errors. For example, GPT-5 mini records 0.923 for interpretation and 0.077 for pianification, while GPT-5 reaches 1 and 0.00, respectively.Claude models and the Gemini Flash variants are the only models for which planning errors exceed interpretation errors. In particular, Gemini Flash exhibits a clear imbalance, with interpretation errors accounting for 0.40 and planning errors reaching 0.60.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
INTERPRETATION	1,000	0,923	0,846	0,444	0,417	0,400	0,615	0,800
PIANIFICATION	0,000	0,077	0,154	0,556	0,583	0,600	0,385	0,200

Figure 4.16: Comparison of Reasoning Errors: Interpretation vs. Pianification (Benchmark 5)

Finally, operational parameters (Table 4.11) reveal relevant differences among models. GPT-5 exhibits the highest cost ($1.90) and the longest latency (2843 s). Among the GPT-5 variants, GPT-5 mini reduces the cost to $0.30 with a latency of 2343 s, while GPT-5 nano further lowers the cost to $0.10 but maintains a relatively high latency of 2597 s. Gemini 2.5 Flash reports a cost of $0.98 and a latency of 1156 s, whereas Gemini Flash-Lite achieves the lowest cost overall ($0.067). Claude Haiku 3.5 delivers the lowest latency (290 s) with a cost of $0.083. DeepSeek-V3.1 records an intermediate profile, with a cost of $0.133 and a latency of 2296 s.

LLM	**Accuracy**	**Cost ($)**	**Latency (s)**
GPT-5	0.88	1.9007	2843.10
GPT-5 mini	0.74	0.3036	2343.16
GPT-5 nano	0.74	0.1099	2597.39
Claude-Sonnet 4	0.64	0.5987	645.26
Claude-Haiku 3.5	0.28	0.0827	290.30
Gemini 2.5 Flash	0.70	0.982	1155.86
Gemini 2.5 Flash-lite	0.48	0.0673	997.28
DeepSeek-V3.1 (chat)	0.80	0.1328	2295.68

Table 4.11: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to synthesize model results into a single ranking by integrating accuracy, cost, and latency (Table 4.12). The final ranking places Claude-Haiku 3.5 in first position, followed by Gemini 2.5 Flash and GPT-5. The intermediate positions are occupied by DeepSeek-V3.1 (chat), Claude Sonnet 4 while Gemini 2.5 Flash-Lite ranks sixth. GPT-5 nano and GPT-5 mini close the ranking.

Rank	**Model**
1	Claude-Haiku 3.5
2	Gemini 2.5 Flash
3	GPT-5
4	DeepSeek-V3.1 (chat)
5	Claude-Sonnet 4
6	Gemini 2.5 Flash-Lite
7	GPT-5 nano
8	GPT-5 mini

Table 4.12: Final ranking of models according to the AHP index (Benchmark 5).

4.4 Cross-benchmark Comparison

After the analysis of the individual benchmarks, this section adopts a comparative perspective, relating the results obtained in the different experimental scenarios. The comparisons make it possible to assess to what extent changes in question format or the introduction of prompting techniques have influenced LLM performance.

The analysis proceeds through direct comparisons between pairs of benchmarks, in order to isolate the effect of specific experimental variables, such as the use of Chain-of-Thought or the shift from single-choice to numerical response questions. For each comparison, the observed variations in accuracy, cost, and latency are reported, with the aim of highlighting recurring patterns or systematic differences between configurations.

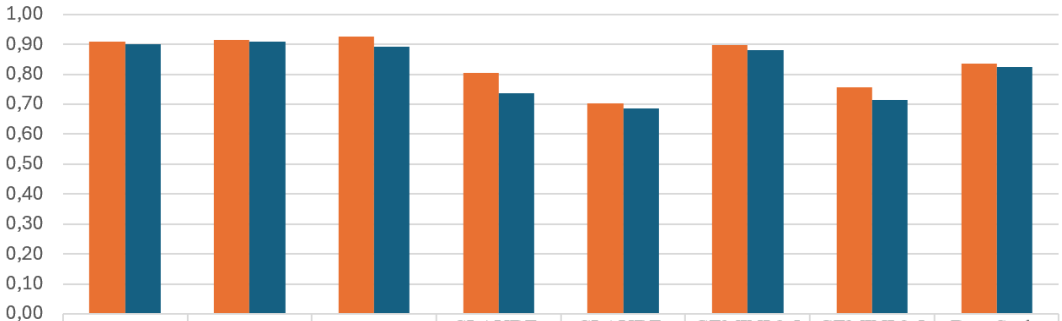
Beyond the descriptive level, the comparisons were subjected to a statistical significance analysis using the McNemar test, applied to verify whether the differences between two benchmarks can be considered statistically relevant at the 95% confidence level. In this

way, the conclusions are based not only on numerical variations but also on their inferential robustness, reducing the risk of overinterpreting marginal differences.

4.4.1 Benchmark 1 vs Benchmark 2

The comparison between Benchmark 1 (without Chain-of-Thought) and Benchmark 2 (with Chain-of-Thought) makes it possible to assess the impact of introducing implicit reasoning on single-choice questions.

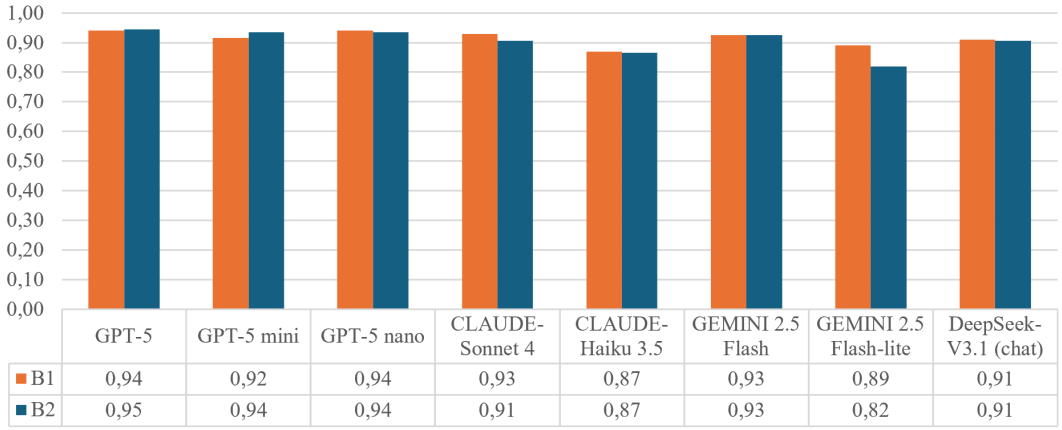
In terms of overall accuracy, the introduction of implicit Chain-of-Thought in Benchmark 2 led to a performance decline for seven out of eight models (Figure 4.17). The largest decreases were observed for Claude Sonnet 4 (–6 pp), followed by Gemini 2.5 Flash-Lite (–5 pp) and GPT-5 nano (–4 pp). No model exhibited an improvement in accuracy, while GPT-5 mini was the only model whose performance remained unchanged (0 pp).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1	0,91	0,91	0,93	0,80	0,70	0,90	0,76	0,84
B2	0,90	0,91	0,89	0,74	0,69	0,88	0,71	0,82

Figure 4.17: Overall Accuracy Comparison between Benchmark 1 and Benchmark 2

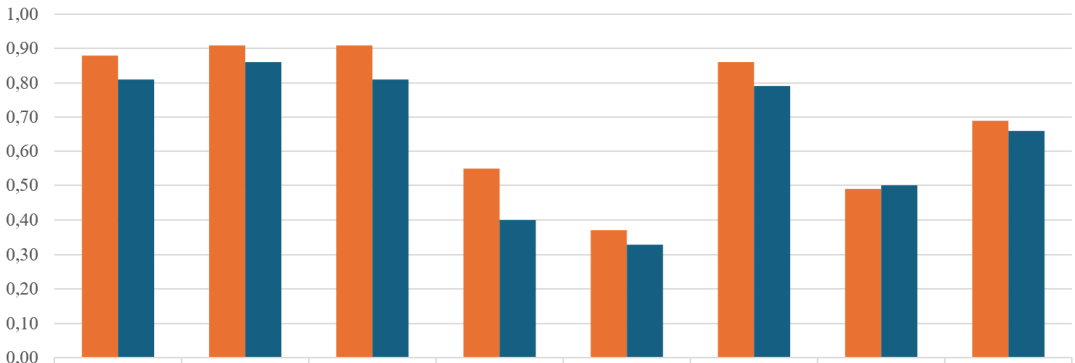
When single-choice questions are split into theoretical and numerical (Figure 4.18), heterogeneous trends emerge. For theoretical questions, several models exhibit no change in accuracy, including GPT-5 nano, Claude Haiku 3.5, Gemini 2.5 Flash, and DeepSeek-V3.1. Only GPT-5 (+1 pp) and GPT-5 mini (+2 pp) show slight improvements, whereas Claude Sonnet 4 (–2 pp) and Gemini Flash-Lite (–7 pp) experience noticeable performance declines. Overall, the impact of implicit Chain-of-Thought on this subset of questions appears marginal.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1	0,94	0,92	0,94	0,93	0,87	0,93	0,89	0,91
B2	0,95	0,94	0,94	0,91	0,87	0,93	0,82	0,91

Figure 4.18: Theoretical Accuracy Comparison between Benchmark 1 and Benchmark 2

A more heterogeneous pattern emerges for numerical questions (Figure 4.19). None of the models remains stable, and all models experience a performance decline except for Gemini 2.5 Flash-Lite, which shows a marginal improvement (+1 pp). The most pronounced degradations are observed for Claude Sonnet 4, which records a sharp decrease of –15 pp, and GPT-5 nano, which declines by –10 pp.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1	0,88	0,91	0,91	0,55	0,37	0,86	0,49	0,69
B2	0,81	0,86	0,81	0,4	0,33	0,79	0,5	0,66

Figure 4.19: Numerical Accuracy Comparison between Benchmark 1 and Benchmark 2

From the perspective of statistical significance, the comparison between Benchmark 1 and Benchmark 2 was verified using the ***McNemar*** test, applied both to the overall sample and to the subgroups by type of question (theoretical and numerical). As reported in Table 4.13 (Overall Accuracy), the performance differences between the two benchmarks do not reach the conventional 95% confidence level for almost all models ($p > 0.05$). The only exceptions are Claude Sonnet 4 and GPT-5 nano, which exhibit a statistically significant reduction in accuracy ($p = 0.0066$ and $p = 0.0192$, respectively). A similar pattern

is observed for theoretical questions (Table 4.14), where only Gemini Flash-Lite shows a statistically significant decline in accuracy (p = 0.0026). In contrast, for numerical questions (Table 4.15), statistically significant differences between the two benchmarks are observed for the majority of models. Among these, GPT-5 nano and Claude Sonnet 4 stand out as exhibiting the most pronounced and robust accuracy reductions, as reflected by their lower p-values (p = 0.0063 and p = 0.0275, respectively).

LLM	**B1**	**B2**	**Sample**	**Chi-square**	**p-value**	**Exact p-value**	**Significance (95%)**
GPT-5	0.910	0.900	300	1.45	0.2278	0.2266	NO
GPT-5 mini	0.913	0.910	300	0.00	1.0000	1.0000	NO
GPT-5 nano	0.927	0.893	300	5.26	0.0218	0.0192	YES
Claude-Sonnet 4	0.803	0.737	300	7.22	0.0072	0.0066	YES
Claude-Haiku 3.5	0.703	0.687	300	0.55	0.4576	0.4583	NO
Gemini 2.5 Flash	0.897	0.880	300	2.77	0.0960	0.0923	NO
Gemini 2.5 Flash-lite	0.757	0.713	300	3.35	0.0673	0.0660	NO
DeepSeek-V3.1 (chat)	0.837	0.823	300	0.90	0.3428	0.3438	NO

Table 4.13: Overall Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

LLM	**B1**	**B2**	**Sample**	**Chi-square**	**p-value**	**Exact p-value**	**Significance (95%)**
GPT-5	0.940	0.945	200	0.00	1.0000	1.0000	NO
GPT-5 mini	0.915	0.935	200	1.50	0.2207	0.2188	NO
GPT-5 nano	0.940	0.935	200	0.00	1.0000	1.0000	NO
Claude-Sonnet 4	0.930	0.905	200	1.78	0.1824	0.1797	NO
Claude-Haiku 3.5	0.870	0.865	200	0.00	1.0000	1.0000	NO
Gemini 2.5 Flash	0.925	0.925	200	0.25	0.6170	1.0000	NO
Gemini 2.5 Flash-lite	0.890	0.820	200	8.45	0.0037	0.0026	YES
DeepSeek-V3.1 (chat)	0.910	0.905	200	0.00	1.0000	1.0000	NO

Table 4.14: Theoretical Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

LLM	B1	B2	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0.880	0.810	100	4.00	0.0455	0.0391	YES
GPT-5 mini	0.910	0.860	100	2.29	0.1306	0.0156	YES
GPT-5 nano	0.910	0.810	100	6.75	0.0094	0.0063	YES
Claude-Sonnet 4	0.550	0.400	100	4.78	0.0288	0.0275	YES
Claude-Haiku 3.5	0.370	0.330	100	0.41	0.5224	0.5235	NO
Gemini 2.5 Flash	0.860	0.790	100	4.00	0.0455	0.0391	YES
Gemini 2.5 Flash-lite	0.490	0.500	100	0.00	1.0000	1.0000	NO
DeepSeek-V3.1 (chat)	0.690	0.660	100	0.57	0.4497	0.4531	NO

Table 4.15: Accuracy Numerical: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

With respect to costs (Table 4.16), the introduction of Chain-of-Thought prompting leads to higher expenditures across all evaluated models. The increase remains limited for GPT-5 (+4%) and moderate for GPT-5 mini (+16.14%), while GPT-5 nano experiences a substantial rise, with costs more than doubling (+107.11%). A pronounced escalation is also observed for the Gemini family, with Gemini 2.5 Flash reporting a +68.79% increase and Gemini 2.5 Flash-Lite exhibiting the largest relative growth (+384.78%). By contrast, Claude Haiku 3.5 (+7.08%), Claude Sonnet 4 (+48.41%), and DeepSeek-V3.1 (+23.48%) show comparatively smaller cost increments.

LLM	B1 ($)	B2 ($)	Variation (%)
GPT-5	2.208	2.296	4.00%
GPT-5 mini	0.309	0.358	16.14%
GPT-5 nano	0.062	0.128	107.11%
Claude-Sonnet 4	0.392	0.582	48.41%
Claude-Haiku 3.5	0.048	0.051	7.08%
Gemini 2.5 Flash	0.014	0.024	68.79%
Gemini 2.5 Flash-lite	0.005	0.022	384.78%
DeepSeek-V3.1 (chat)	0.026	0.033	23.48%

Table 4.16: Cost comparison between Benchmark 1 and Benchmark 2 with percentage variation

Latency (Table 4.17) exhibits more moderate variations compared to costs following the introduction of CoT. A slight reduction in latency is observed for GPT-5 (–5.51%) and

Claude Haiku 3.5 (–8.79%), indicating marginal efficiency gains. Conversely, GPT-5 mini (+13.10%), GPT-5 nano (+5.63%), and Gemini 2.5 Flash (+4.60%) experience limited latency increases. More pronounced degradations are recorded for Claude Sonnet 4 (+23.60%), Gemini Flash-Lite (+59.66%), and DeepSeek-V3.1 (+30.99%).

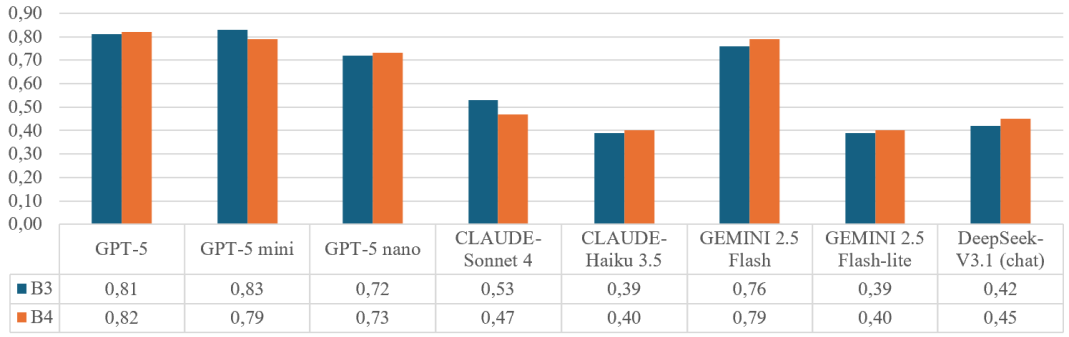
LLM	**B1 (s)**	**B2 (s)**	**Variation (%)**
GPT-5	5112.63	4830.74	-5.51%
GPT-5 mini	2466.60	2789.82	13.10%
GPT-5 nano	2538.44	2681.42	5.63%
Claude-Sonnet 4	956.91	1182.77	23.60%
Claude-Haiku 3.5	353.47	322.40	-8.79%
Gemini 2.5 Flash	1888.55	1975.48	4.60%
Gemini 2.5 Flash-lite	285.51	455.84	59.66%
DeepSeek-V3.1 (chat)	450.26	589.79	30.99%

Table 4.17: Latency comparison between Benchmark 1 and Benchmark 2 with percentage variation

4.4.2 Benchmark 3 vs Benchmark 4

The comparison between Benchmark 3 (numerical answer) and Benchmark 4 (numerical answer with implicit Chain-of-Thought) makes it possible to assess the impact of introducing implicit reasoning in numerical response tasks.

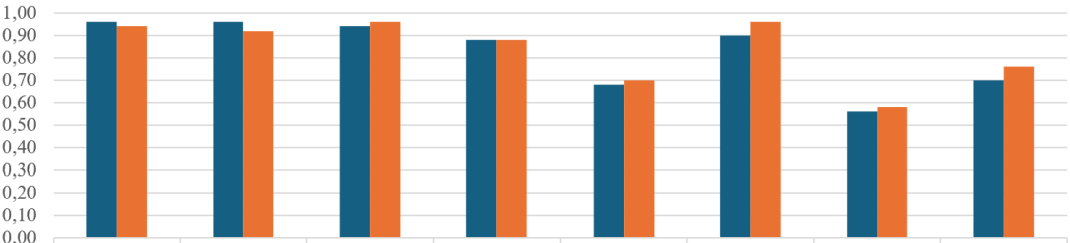
In terms of overall accuracy, the observed changes remain relatively limited, with variations ranging from –6 to +3 percentage points. GPT-5, GPT-5 nano, Claude Haiku 3.5, and Gemini 2.5 Flash-Lite each register a marginal improvement of +1 percentage point. Gemini 2.5 Flash and DeepSeek-V3.1 exhibit slightly larger gains, with accuracy increasing by +3 percentage points. By contrast, Claude Sonnet 4 and GPT-5 mini experience moderate declines in performance, with decreases of –6 and –4 percentage points, respectively. (Figure 4.20)



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B3	0,81	0,83	0,72	0,53	0,39	0,76	0,39	0,42
B4	0,82	0,79	0,73	0,47	0,40	0,79	0,40	0,45

Figure 4.20: Overall Accuracy Comparison between Benchmark 3 and Benchmark 4

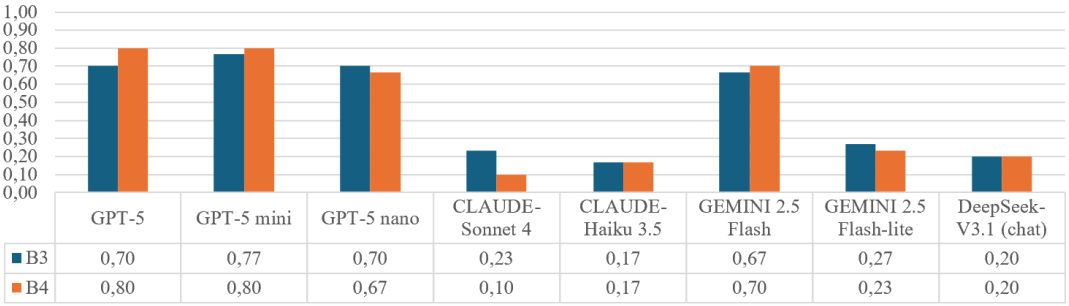
For easy questions (Figure 4.21), performance changes remain generally modest across models. GPT-5 nano, Claude Haiku 3.5, and Gemini 2.5 Flash-Lite each record a slight improvement of +2 percentage points. In contrast, GPT-5 and GPT-5 mini exhibit small decreases in accuracy (–2 and –4 pp, respectively), while Gemini 2.5 Flash and DeepSeek-V3.1 show more pronounced gains of +6 pp. Claude Sonnet 4 displays no change in performance. Overall, the impact of implicit CoT on easy questions appears limited and does not follow a systematic pattern across models.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B3	0,96	0,96	0,94	0,88	0,68	0,90	0,56	0,70
B4	0,94	0,92	0,96	0,88	0,70	0,96	0,58	0,76

Figure 4.21: Easy-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

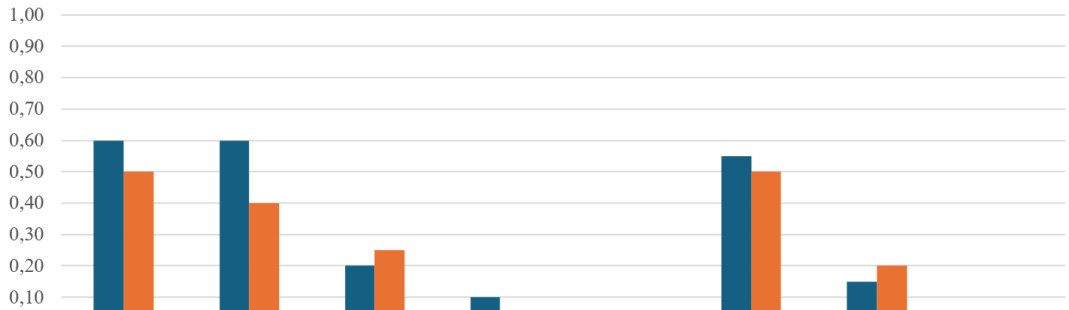
For medium-difficulty questions (Figure 4.22), the picture is more heterogeneous. Claude Haiku 3.5 and DeepSeek-V3.1 show no variation in accuracy, whereas GPT-5 mini and Gemini 2.5 Flash each achieve a modest improvement of +3 pp. By contrast, GPT-5 nano and Gemini 2.5 Flash-Lite experience slight performance declines of –3 and –4 pp, respectively. The largest effects are observed for GPT-5, which records a substantial improvement of +10 pp, and for Claude Sonnet 4, which exhibits a marked drop in accuracy of –13 pp.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B3	0,70	0,77	0,70	0,23	0,17	0,67	0,27	0,20
B4	0,80	0,80	0,67	0,10	0,17	0,70	0,23	0,20

Figure 4.22: Medium-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

For the most difficult questions (Figure 4.23), implicit CoT is associated with performance reductions for approximately half of the evaluated models. Claude Haiku 3.5 and DeepSeek-V3.1 display stable accuracy, while only GPT-5 nano and Gemini 2.5 Flash-Lite exhibit slight improvements of +5 pp. In contrast, GPT-5 and Claude Sonnet 4 both experience notable declines (-10 pp), and GPT-5 mini shows the largest drop in performance (–20 pp). Overall, these results indicate that, for harder tasks, the additional instruction does not provide systematic benefits and may instead hinder model performance or fail to yield meaningful improvements.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B3	0,6	0,6	0,2	0,1	0	0,55	0,15	0,05
B4	0,5	0,4	0,25	0	0	0,5	0,2	0,05

Figure 4.23: Hard-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

The statistical significance analysis, conducted both in the general sample and in the subgroups by difficulty level, is reported in Tables Y.1 to Y.4. As shown in Table 4.18 (Overall Accuracy), in the general sample only Claude Sonnet 4 reaches statistical significance at the conventional 95% confidence level ($p = 0.041$). In the easy, medium, and hard subsets (Tables 4.19, 4.20, and 4.21), no statistically significant differences are observed between Benchmark 3 and Benchmark 4 for any of the evaluated models.

LLM	B3	B4	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,81	0,82	100	0,00	1,000	1,000	NO
GPT-5 mini	0,83	0,79	100	0,90	0,343	0,344	NO
GPT-5 nano	0,72	0,73	100	0,00	1,000	1,000	NO
Claude-Sonnet 4	0,53	0,47	100	4,17	0,041	0,031	YES
Claude-Haiku 3.5	0,39	0,40	100	0,00	1,000	1,000	NO
Gemini 2.5 Flash	0,76	0,79	100	1,23	0,267	0,267	NO
Gemini 2.5 Flash-lite	0,39	0,40	100	0,27	0,606	0,607	NO
DeepSeek-V3.1 (chat)	0,42	0,45	100	0,44	0,505	0,508	NO

Table 4.18: Overall Accuracy: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

LLM	B3	B4	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,96	0,94	50	0,00	1,000	1,000	NO
GPT-5 mini	0,96	0,92	50	0,50	0,480	0,500	NO
GPT-5 nano	0,94	0,96	50	0,00	1,000	1,000	NO
Claude-Sonnet 4	0,88	0,88	50	inf	0,000	1,000	NO
Claude-Haiku 3.5	0,68	0,70	50	0,00	1,000	1,000	NO
Gemini 2.5 Flash	0,90	0,96	50	2,25	0,134	0,125	NO
Gemini 2.5 Flash-lite	0,56	0,58	50	0,00	1,000	1,000	NO
DeepSeek-V3.1 (chat)	0,70	0,76	50	0,80	0,371	0,375	NO

Table 4.19: Accuracy Easy: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

LLM	B3	B4	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,70	0,80	30	1,50	0,221	0,219	NO
GPT-5 mini	0,77	0,80	30	0,25	0,617	0,625	NO
GPT-5 nano	0,70	0,67	30	0,00	1,000	1,000	NO
Claude-Sonnet 4	0,23	0,10	30	2,25	0,134	0,125	NO
Claude-Haiku 3.5	0,17	0,17	30	0,50	0,480	1,000	NO
Gemini 2.5 Flash	0,67	0,70	30	0,00	1,000	1,000	NO
Gemini 2.5 Flash-lite	0,27	0,23	30	0,25	0,617	0,625	NO
DeepSeek-V3.1 (chat)	0,20	0,20	30	0,25	0,617	1,000	NO

Table 4.20: Accuracy Medium: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

LLM	B3	B4	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,60	0,50	20	0,50	0,480	0,500	NO
GPT-5 mini	0,60	0,40	20	2,25	0,134	0,125	NO
GPT-5 nano	0,20	0,25	20	0,00	1,000	1,000	NO
Claude-Sonnet 4	0,10	0,00	20	0,50	0,480	0,500	NO
Claude-Haiku 3.5	0,00	0,00	20	inf	0,000	1,000	NO
Gemini 2.5 Flash	0,55	0,50	20	0,17	0,683	1,000	NO
Gemini 2.5 Flash-lite	0,15	0,20	20	0,50	0,480	1,000	NO
DeepSeek-V3.1 (chat)	0,05	0,05	20	0,00	1,000	1,000	NO

Table 4.21: Accuracy Hard: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

With respect to costs (Table 4.22), the comparison between Benchmark 3 and Benchmark 4 indicates that the introduction of CoT leads to moderate cost variations for most models. GPT-5 and DeepSeek-V3.1 exhibit slight cost reductions (–3.14% and –3.57%, respectively), whereas GPT-5 mini, Gemini 2.5 Flash, and Gemini 2.5 Flash-Lite show relatively larger increases of +31.23%, +30.95%, and +37.51%, respectively. The Claude models also experience moderate cost growth, with Claude Sonnet 4 increasing by +16.04% and Claude Haiku 3.5 by +18.78%. Overall, the adoption of implicit CoT results in moderate cost increases for the majority of the evaluated models, with only limited cases of cost reduction.

LLM	B3 ($)	B4 ($)	Variation (%)
GPT-5	2,1030	2,0370	-3,14%
GPT-5 mini	0,2056	0,2698	31,23%
GPT-5 nano	0,1011	0,1031	1,98%
Claude-Sonnet 4	0,3248	0,3769	16,04%
Claude-Haiku 3.5	0,0197	0,0234	18,78%
Gemini 2.5 Flash	0,0072	0,0095	30,95%
Gemini 2.5 Flash-lite	0,0022	0,0031	37,51%
DeepSeek-V3.1 (chat)	0,0261	0,0252	-3,57%

Table 4.22: Cost comparison between Benchmark 3 and Benchmark 4 with percentage variation

As for latency (Table 4.23), the introduction of implicit CoT is associated with longer response times for most models. GPT-5 exhibits a clear reduction in latency (–27.73%),

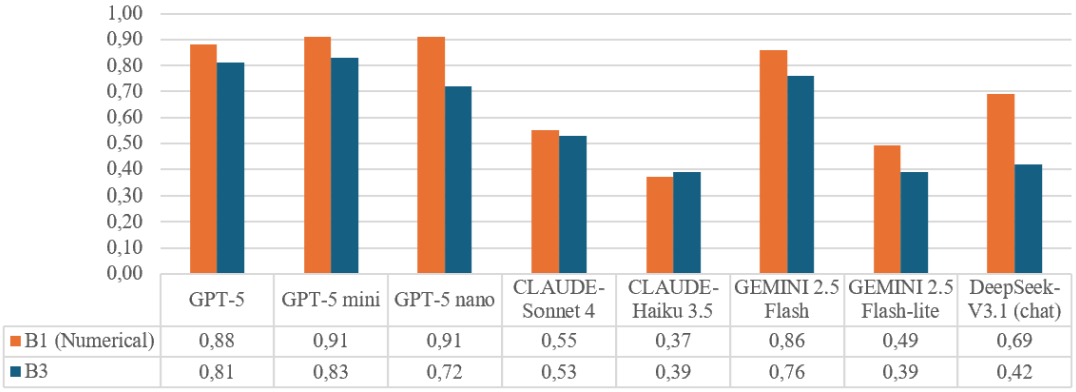
whereas GPT-5 nano shows a negligible decrease (–1.49%). More pronounced latency growth is observed for Gemini Flash-Lite (+182.01%) and GPT-5 mini (+65.36%). By contrast, Claude Haiku 3.5 (+1.37%), Claude Sonnet 4 (+11.15%), and DeepSeek-V3.1 (+4.88%) experience only limited increases in response time.

LLM	**B3 (s)**	**B4 (s)**	**Variation (%)**
GPT-5	4200,29	3203,36	-23,73%
GPT-5 mini	1644,28	2718,98	65,36%
GPT-5 nano	2186,05	2153,39	-1,49%
Claude-Sonnet 4	469,37	521,69	11,15%
Claude-Haiku 3.5	103,14	104,55	1,37%
Gemini 2.5 Flash	849,32	1034,58	21,81%
Gemini 2.5 Flash-lite	86,66	244,39	182,01%
DeepSeek-V3.1 (chat)	402,47	422,10	4,88%

Table 4.23: Latency comparison between Benchmark 3 and Benchmark 4 with percentage variation

4.4.3 Benchamrk 1 vs Benchmark 3

The comparison between Benchmark 1 (numerical single-choice) and Benchmark 3 (numerical open-answer) reveals a general reduction in overall accuracy when moving from constrained to open numerical responses (Figure 4.24). Claude Haiku 3.5 constitutes the only exception, exhibiting a modest improvement of +2 pp. All other models experience accuracy declines, with GPT-5, GPT-5 mini, and GPT-5 nano decreasing by –7, –8, and –19 pp, respectively. The Gemini models are particularly affected, as both Gemini 2.5 Flash and Gemini 2.5 Flash-Lite record reductions of –10 pp. Claude Sonnet 4 shows a limited decrease (–2 pp), whereas DeepSeek-V3.1 exhibits the largest performance drop (-27 pp). Overall, the numerical answer format proves more challenging, with substantial negative shifts for nearly all models.

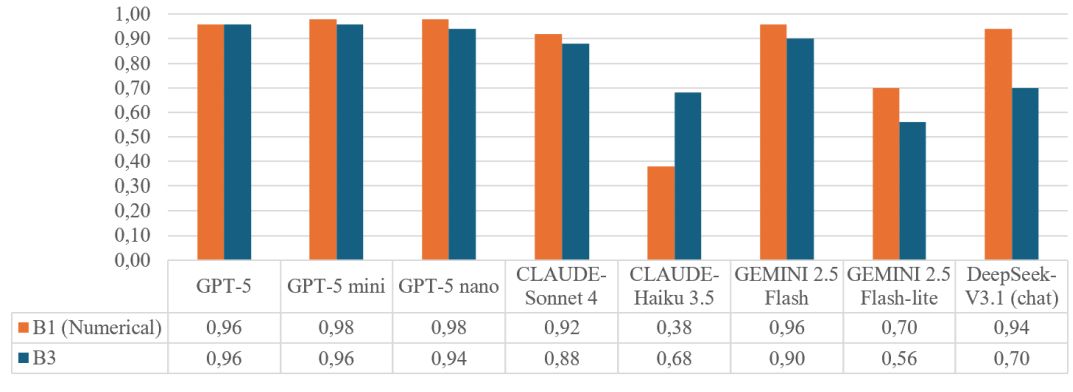


	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1 (Numerical)	0,88	0,91	0,91	0,55	0,37	0,86	0,49	0,69
B3	0,81	0,83	0,72	0,53	0,39	0,76	0,39	0,42

Figure 4.24: Overall Accuracy Comparison between Benchmark 1 and Benchmark 3

Breaking down overall accuracy by difficulty levels makes it possible to better understand how the change from numerical single choice (B1) to numerical answer (B3) affects performance, revealing dynamics not visible in the overall mean.

For easy questions (Figure 4.25), the impact of moving from single-choice to open numerical responses remains comparatively limited, although most models exhibit a decline in performance. Claude Haiku 3.5 represents a notable exception, recording a substantial improvement of +30 pp, while GPT-5 maintains stable accuracy. The Gemini models experience moderate decreases, with Gemini 2.5 Flash and Gemini 2.5 Flash-Lite declining by –6 and –14 pp, respectively. Similar but smaller reductions are observed for GPT-5 mini (–2 pp), GPT-5 nano, and Claude Sonnet 4 (–4 pp each). DeepSeek-V3.1 shows the most pronounced deterioration in performance, with a drop of –24 pp.

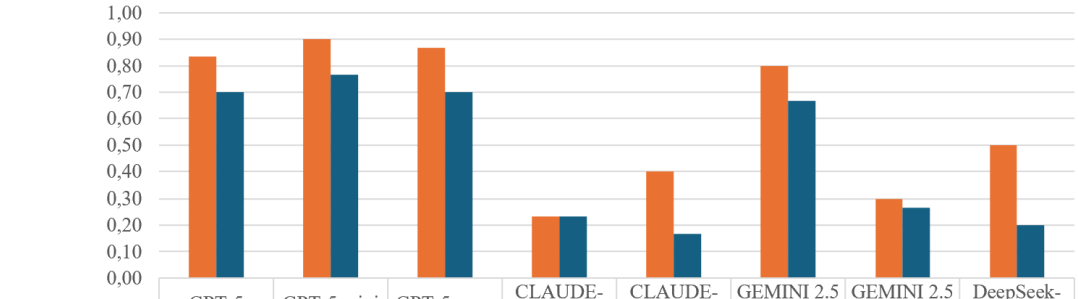


	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1 (Numerical)	0,96	0,98	0,98	0,92	0,38	0,96	0,70	0,94
B3	0,96	0,96	0,94	0,88	0,68	0,90	0,56	0,70

Figure 4.25: Easy-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

For medium-difficulty questions (Figure 4.26), the transition to Benchmark 3 results in widespread performance declines across models. Claude Sonnet 4 is the only model

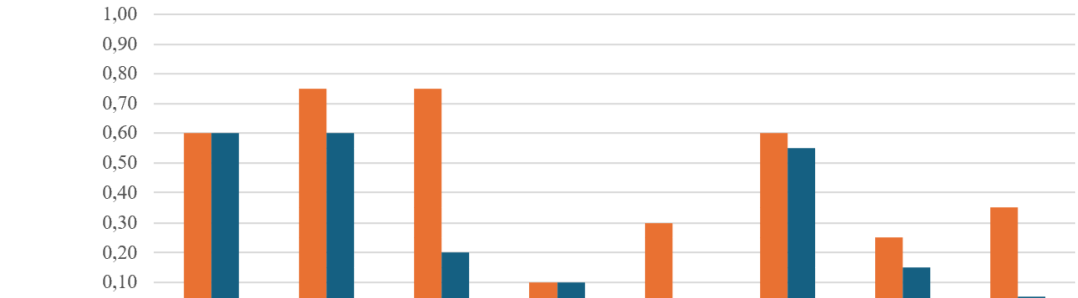
whose accuracy remains unchanged, whereas Claude Haiku 3.5 experiences a substantial reduction (-23 pp).DeepSeek-V3.1 shows the most pronounced deterioration, with accuracy decreasing by –30 pp.GPT-5, GPT-5 mini, and Gemini 2.5 Flash each record moderate losses (-13 pp), while GPT-5 nano declines by –17 pp. By contrast, Gemini 2.5 Flash-Lite exhibits a negligible decrease (-3 pp), indicating an almost stable performance in this setting.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1 (Numerical)	0,83	0,90	0,87	0,23	0,40	0,80	0,30	0,50
B3	0,70	0,77	0,70	0,23	0,17	0,67	0,27	0,20

Figure 4.26: Medium-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

GPT-5 mini and GPT-5 nano exhibit substantial accuracy losses of –15 and –55 pp, respectively.Claude Haiku 3.5 and DeepSeek-V3.1 also experience severe declines, each decreasing by –30 pp. The Gemini models show comparatively smaller reductions, with accuracy losses ranging between –5 and –10 pp. By contrast, GPT-5 and Claude Sonnet 4 are the only models whose performance remains stable under this setting. In this category, no model shows improvements: the transition to the numerical answer format systematically penalizes performance.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B1 (Numerical)	0,60	0,75	0,75	0,10	0,30	0,60	0,25	0,35
B3	0,60	0,60	0,20	0,10	0,00	0,55	0,15	0,05

Figure 4.27: Hard-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

From the perspective of statistical significance, the comparison between Benchmark 1 and Benchmark 3 reveals few relevant differences. On the overall sample of 100 questions (Table 4.24), differences are statistically significant at the 95% level for GPT-5 mini (exact p = 0.039) and Gemini 2.5 Flash (exact p = 0.004).

Looking at the difficulty subgroups, in the Easy subset (Table 4.25) significance is observed only for DeepSeek-V3.1 (exact p = 0.002) and Claude Haiku 3.5 (exact p = 0.004)

In the Medium subset (Table 4.26), differences are significant for only DeepSeek-V3.1 (exact p = 0.022).

In the Hard subset (Table 4.27), statistically significant differences emerge for GPT-5 nano (exact p = 0.003), Claude Haiku 3.5 (exact p = 0.031) and DeepSeek-V3.1 (exact p = 0.031).

LLM	**B1**	**B3**	**Sample**	**Chi-square**	**p-value**	**Exact p-value**	**Significance (95%)**
GPT-5	0,88	0,81	100	2,769	0,096	0,092	NO
GPT-5 mini	0,91	0,83	100	4,083	0,043	0,039	YES
GPT-5 nano	0,91	0,72	100	14,087	0,0002	6,604	NO
Claude-Sonnet 4	0,55	0,53	100	0,045	0,831	0,832	NO
Claude-Haiku 3.5	0,37	0,39	100	0,023	0,880	0,880	NO
Gemini 2.5 Flash	0,86	0,76	100	7,681	0,006	0,004	YES
Gemini 2.5 Flash-lite	0,49	0,39	100	3,030	0,082	0,080	NO
DeepSeek-V3.1 (chat)	0,69	0,42	100	20,485	6,011	1,401	NO

Table 4.24: Overall Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

LLM	**B1**	**B3**	**Sample**	**Chi-square**	**p-value**	**Exact p-value**	**Significance (95%)**
GPT-5	0,96	0,96	50	0,500	0,480	1,000	NO
GPT-5 mini	0,98	0,96	50	0,000	1,000	1,000	NO
GPT-5 nano	0,98	0,94	50	0,500	0,480	0,500	NO
Claude-Sonnet 4	0,92	0,88	50	0,125	0,724	0,727	NO
Claude-Haiku 3.5	0,38	0,68	50	7,840	0,005	0,004	YES
Gemini 2.5 Flash	0,96	0,90	50	1,500	0,221	0,219	NO
Gemini 2.5 Flash-lite	0,70	0,56	50	0,643	0,423	0,424	NO
DeepSeek-V3.1 (chat)	0,94	0,70	50	8,643	0,003	0,002	YES

Table 4.25: Accuracy Easy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

LLM	B1	B3	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,83	0,70	30	3,125	0,077	0,070	NO
GPT-5 mini	0,90	0,77	30	1,500	0,221	0,219	NO
GPT-5 nano	0,87	0,70	30	3,125	0,077	0,070	NO
Claude-Sonnet 4	0,23	0,23	30	0,100	0,752	1,000	NO
Claude-Haiku 3.5	0,40	0,17	30	2,769	0,096	0,092	NO
Gemini 2.5 Flash	0,80	0,67	30	2,500	0,114	0,109	NO
Gemini 2.5 Flash-lite	0,30	0,27	30	0,750	0,386	0,388	NO
DeepSeek-V3.1 (chat)	0,50	0,20	30	4,923	0,027	0,022	YES

Table 4.26: Accuracy Medium: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

LLM	B1	B3	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,60	0,60	20	0,500	0,480	1,000	NO
GPT-5 mini	0,75	0,60	20	0,800	0,371	0,375	NO
GPT-5 nano	0,75	0,20	20	7,692	0,006	0,003	YES
Claude-Sonnet 4	0,10	0,10	20	0,250	0,617	1,000	NO
Claude-Haiku 3.5	0,30	0,00	20	4,167	0,041	0,031	YES
Gemini 2.5 Flash	0,60	0,55	20	1,500	0,221	0,219	NO
Gemini 2.5 Flash-lite	0,25	0,15	20	0,571	0,450	0,453	NO
DeepSeek-V3.1 (chat)	0,35	0,05	20	4,167	0,041	0,031	YES

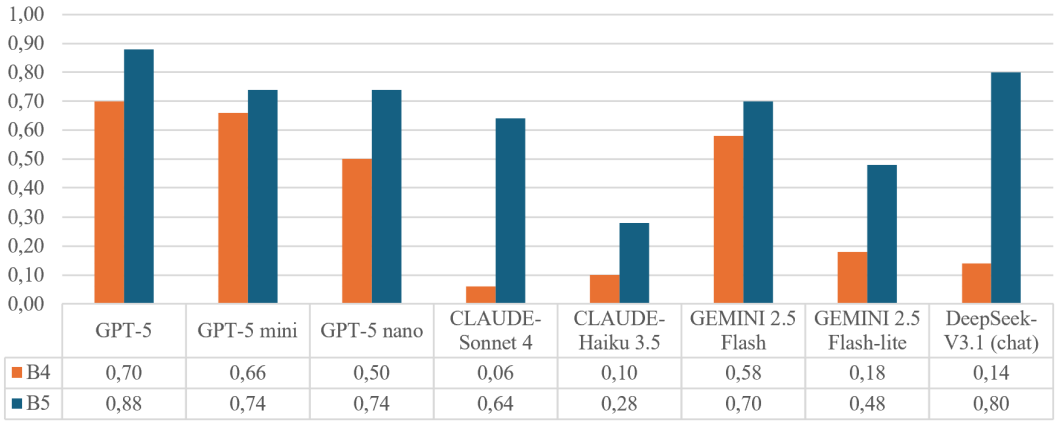
Table 4.27: Accuracy Hard: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

4.4.4 Benchamrk 4 vs Benchamrk 5

The comparison between Benchmark 4 (numerical answer with implicit Chain-of-Thought, limited to medium and hard questions) and Benchmark 5 (numerical answer with explicit Chain-of-Thought, requiring written reasoning) represents a crucial step in understanding whether implicit prompting to reason is sufficient, or whether the explicit articulation of logical steps is a necessary condition to improve performance.

In terms of overall accuracy (Figure 4.28), the transition from implicit CoT (B4) to explicit CoT (B5) produces a substantial improvement for all models. GPT-5 (+18 pp), GPT-5 mini (+8 pp) and GPT-5 nano (+24 pp) gain accuracy. The largest improvements are observed in Claude Sonnet 4 (+58 pp) and DeepSeek-V3.1 (+66 pp). The Gemini

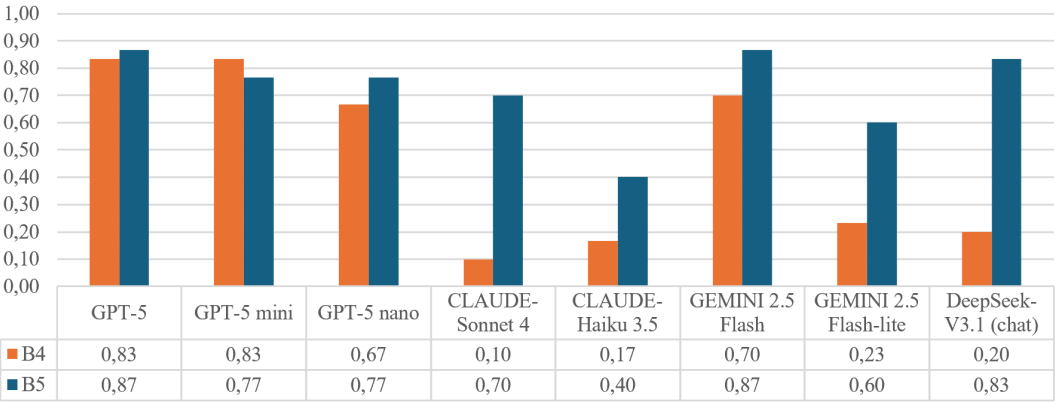
models also show marked gains (Flash +12 pp, Flash-Lite +30 pp), as does Claude Haiku 3.5 (+18 pp).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B4	0,70	0,66	0,50	0,06	0,10	0,58	0,18	0,14
B5	0,88	0,74	0,74	0,64	0,28	0,70	0,48	0,80

Figure 4.28: Overall Accuracy Comparison between Benchmark 4 and Benchmark 5

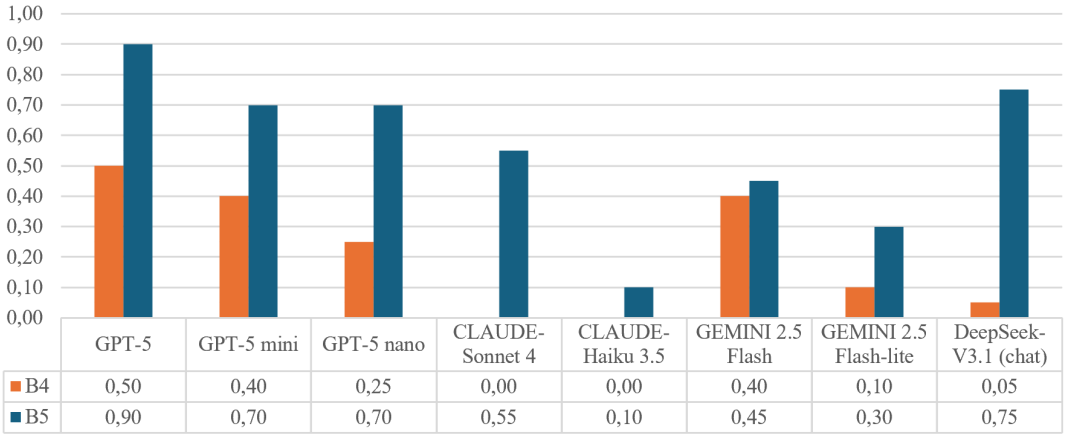
When focusing on medium-difficulty questions (Figure 4.29), accuracy improvements are observed for almost all evaluated models. Claude Sonnet 4, Gemini 2.5 Flash-Lite, and DeepSeek-V3.1 exhibit particularly large gains, with increases of +60, +37, and +63 pp, respectively. More moderate but still positive improvements are recorded for Claude Haiku 3.5 (+23 pp), GPT-5 nano (+10 pp), and Gemini 2.5 Flash (+17 pp). GPT-5 mini represents the only exception, showing a slight decrease in performance (-6 pp).



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B4	0,83	0,83	0,67	0,10	0,17	0,70	0,23	0,20
B5	0,87	0,77	0,77	0,70	0,40	0,87	0,60	0,83

Figure 4.29: Medium-Level Accuracy Comparison between Benchmark 4 and Benchmark 5

For hard questions, explicit CoT again brings widespread benefits (Figure 4.30). GPT-5 (+40 pp), GPT-5 mini (+30 pp), GPT-5 nano (+45 pp), Claude Sonnet 4 (+55 pp), Gemini Flash-Lite (+20 pp), and DeepSeek-V3.1 (+70 pp) all register significant improvements.



	GPT-5	GPT-5 mini	GPT-5 nano	CLAUDE-Sonnet 4	CLAUDE-Haiku 3.5	GEMINI 2.5 Flash	GEMINI 2.5 Flash-lite	DeepSeek-V3.1 (chat)
B4	0,50	0,40	0,25	0,00	0,00	0,40	0,10	0,05
B5	0,90	0,70	0,70	0,55	0,10	0,45	0,30	0,75

Figure 4.30: Hard-Level Accuracy Comparison between Benchmark 4 and Benchmark 5

From the perspective of statistical significance, the comparison between Benchmark 4 and Benchmark 5 highlights a substantial number of relevant differences. On the overall sample of 50 questions (Table 4.28), variations are significant for GPT-5 nano (exact $p = 0.004$), Claude Sonnet 4 (exact $p < 0.001$), Claude Haiku 3.5 (exact $p = 0.022$), Gemini 2.5 Flash (exact $p < 0.001$), Gemini 2.5 Flash-Lite (exact $p = 0.003$), and DeepSeek-V3.1 (exact $p < 0.001$).

Looking at the subgroups, in the Medium set (Table 4.29) significant differences emerge for Claude Sonnet 4 (exact $p < 0.001$), Gemini 2.5 Flash-Lite (exact $p = 0.007$), and DeepSeek-V3.1 (exact $p < 0.001$).

In the Hard set (Table 4.30), significance is confirmed for Claude Sonnet 4 (exact $p < 0.001$) and DeepSeek-V3.1 (exact $p = 0.001$). Significant differences are also observed for GPT-5 (exact $p = 0.008$), GPT-5 mini (exact $p = 0.031$), and GPT-5 nano (exact $p = 0.004$), indicating that performance changes between Benchmark 4 and Benchmark 5 are non-random for these models.

LLM	B4	B5	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,70	0,88	50	2,29	0,131	0,125	NO
GPT-5 mini	0,66	0,74	50	0,90	0,343	0,344	NO
GPT-5 nano	0,50	0,74	50	7,56	0,006	0,004	YES
Claude-Sonnet 4	0,06	0,64	50	27,03	0,000	0,000	YES
Claude-Haiku 3.5	0,10	0,28	50	4,92	0,027	0,022	YES
Gemini 2.5 Flash	0,58	0,70	50	26,28	0,000	0,000	YES
Gemini 2.5 Flash-lite	0,18	0,48	50	8,52	0,004	0,003	YES
DeepSeek-V3.1 (chat)	0,14	0,80	50	29,26	0,000	0,000	YES

Table 4.28: OverallAccuracy: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

LLM	B4	B5	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,83	0,87	30	0,00	1,000	1,000	NO
GPT-5 mini	0,83	0,77	30	0,25	0,617	0,625	NO
GPT-5 nano	0,67	0,77	30	0,57	0,450	0,453	NO
Claude-Sonnet 4	0,10	0,70	30	16,06	0,000	0,000	YES
Claude-Haiku 3.5	0,17	0,40	30	3,27	0,070	0,065	NO
Gemini 2.5 Flash	0,70	0,87	30	3,20	0,074	0,063	NO
Gemini 2.5 Flash-lite	0,23	0,60	30	6,67	0,010	0,007	YES
DeepSeek-V3.1 (chat)	0,20	0,83	30	17,05	0,000	0,000	YES

Table 4.29: Accuracy Medium: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

LLM	B4	B5	Sample	Chi-square	p-value	Exact p-value	Significance (95%)
GPT-5	0,50	0,90	20	6,13	0,013	0,008	YES
GPT-5 mini	0,40	0,70	20	4,17	0,041	0,031	YES
GPT-5 nano	0,25	0,70	20	7,11	0,008	0,004	YES
Claude-Sonnet 4	0,00	0,55	20	10,08	0,001	0,000	YES
Claude-Haiku 3.5	0,00	0,10	20	1,33	0,248	0,250	NO
Gemini 2.5 Flash	0,40	0,45	20	0,13	0,724	0,727	NO
Gemini 2.5 Flash-lite	0,10	0,30	20	1,78	0,182	0,180	NO
DeepSeek-V3.1 (chat)	0,05	0,75	20	10,56	0,001	0,001	YES

Table 4.30: Accuracy Hard: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

4.5 Summary Results

The preliminary survey defined the weights to be applied in the Analytic Hierarchy Process (AHP), giving priority to **Accuracy** (normalized weight 0.407), followed by **Latency** (0.315) and **Cost** (0.277).

At the benchmark level, results show different patterns. In Benchmark 1 (Single Choice), accuracy ranged from 0.70 to 0.93, with five models above the human reference threshold (0.80). The AHP ranking placed GPT-5 nano first, followed by DeepSeek-V3.1 and Gemini 2.5 Flash. With the introduction of implicit Chain-of-Thought in Benchmark 2, accuracies ranged from 0.69 to 0.91, again with five models above the baseline; AHP ranked DeepSeek-V3.1 first, followed by GPT-5 nano and Gemini 2.5 Flash .

Benchmark 3 (Numerical Answer) showed more polarized results, between 0.39 and 0.83, with only GPT-5 and GPT-5 mini exceeding the human threshold. AHP placed GPT-5 mini first, followed by Gemini 2.5 Flash and GPT-5.

In Benchmark 4 (Numerical Answer with implicit CoT), accuracies ranged from 0.40 to 0.82, with only GPT-5 above the threshold; AHP ranked Gemini 2.5 Flash first, followed by Claude Haiku 3.5 and GPT-5 mini. Finally, Benchmark 5 (Numerical Answer with explicit reasoning) reported accuracies from 0.28 to 0.88, with again only GPT-5 above the threshold. AHP ranked Claude Haiku 3.5 first, followed by Gemini 2.5 Flash and GPT-5.

Cross-benchmark comparisons helped isolate the effect of question format and prompting strategies. From Benchmark 1 to Benchmark 2, the addition of implicit CoT did not lead to systematic improvements: accuracy remains stable only for GPT-5 mini and slightly declines for all other models. Statistical testing reveals the presence of significant differences, particularly within the numerical questions, for most models, including GPT-5, GPT-5 mini, GPT-5 nano, Claude Sonnet 4, and Gemini 2.5 Flash.

The comparison between Benchmark 1 and Benchmark 3, moving from single-choice numerical questions to open numerical answers, showed a decline in accuracy for almost all models, except Claude Haiku 3.5, exhibiting a slight improvement in performance. McNemar's tests confirmed significant differences for Claude Haiku 3.5, Gemini Flash, GPT-5 nano, GPT-5 mini and DeepSeek-V3.1, particularly in hard-difficulty questions.

The comparison between Benchmark 3 and Benchmark 4, which evaluates the effect of implicit Chain-of-Thought prompting in numerical open answers, does not reveal robust performance changes. Accuracy variations are generally small, with slight increases or decreases across models and modest improvements observed primarily on easier questions.With the exception of Claude Sonnet 4, no statistically significant differences emerge, indicating that the observed variations are descriptive rather than systematic.

A clearer effect appeared in the comparison between Benchmark 4 and Benchmark 5, where explicit CoT improved the performance of most models, particularly on hard questions. Statistical tests confirmed significant differences in this setting, especially within the hard subset, for the GPT-5 family, Claude Sonnet 4, Gemini models, and DeepSeek-V3.1.These results indicate that explicit reasoning instructions produce robust effects.

Overall, the results provide a structured and replicable picture of LLM performance, highlight the trade-offs between accuracy, cost, and latency, and document in a systematic way the role of question format and prompting strategies. These findings form the basis for the critical discussion developed in the next chapter.

Chapter 5

Discussion

5.1 Main Results

The section dedicated to the main results aims to critically interpret the empirical evidence emerging from the benchmarks, relating it to the research objectives of the thesis. As recalled in the Introduction, the study is structured around two main research questions: (RQ1) which combinations of datasets, evaluation metrics, and prompting techniques enable the construction of meaningful benchmarks for project management contexts; (RQ2) which language model currently achieves the best overall performance, providing a comparative framework useful for supporting managerial choices.

To systematically address these questions, the results were discussed through four operational questions derived from the experimental benchmarks. These concern: the effectiveness of Chain of Thought, analyzed in both its implicit and explicit variants; the impact of question format on model performance; the insights provided by the rankings obtained through the Analytic Hierarchy Process (AHP); and finally, the extent to which LLMs are able to perform better than human evaluators in project management decision-making tasks.

This structure makes it possible to translate the two research questions into a more detailed analysis, capable of highlighting not only performance trends but also the theoretical and practical implications that follow.

The four questions discussed here therefore represent an operational declination of the research questions, aimed at highlighting strengths, weaknesses, and differences among

models in scenarios of varying complexity. In this way, the results are interpreted not only in technical terms but also in light of the real needs of project management, in line with the overall objective of the thesis.

5.1.1 Question 1 – How does Chain of Thought (CoT), in its implicit and explicit forms, affect the performance of LLMs?

Recent literature attributes a central role to Chain of Thought in enhancing the performance of Large Language Models, suggesting that step-by-step reasoning can lead to higher accuracy, especially in complex tasks. However, most empirical evidence has focused on explicit CoT, in which the model is required to produce a sequence of logical steps before providing the final answer. In this research, in addition to that configuration, an implicit CoT was also introduced and tested, in which the instruction *"Let's think step by step"* was added to the prompt without requiring the reasoning to be made explicit: the expected output was only the final answer. This methodological choice allowed for a broader evaluation of CoT's impact, distinguishing between the two approaches and comparing their effects.

Implicit CoT: B1 vs B2 and B3 vs B4

The first level of analysis focused on single-choice questions (Benchmark 1 vs Benchmark 2). From a descriptive perspective, the results present a relatively consistent pattern: accuracy decreases for almost all models, with only a few cases exhibiting near-stable performance. These declines are not uniform across question types. While theoretical questions show only limited variations, the most pronounced reductions in accuracy systematically emerge in numerical questions.

At first glance, this pattern suggests a mild negative effect of implicit Chain-of-Thought on single-choice performance, particularly in numerical items. Statistical significance testing, however, refines this interpretation. Differences in overall and theoretical accuracy are rarely statistically significant, indicating that most observed variations are not robust. By contrast, numerical questions display significant differences for several models, pointing to a greater sensitivity of numerical single-choice tasks to implicit CoT.

Overall, implicit Chain-of-Thought does not exert a systematic effect on single-choice accuracy, but its impact becomes more visible and often detrimental when numerical reasoning is required.

A more heterogeneous pattern emerges from the comparison between Benchmark 3 and Benchmark 4, which focuses on numerical open-answer questions, a format that requires models to autonomously generate the correct response rather than select from predefined options. Given the procedural nature of this task, implicit Chain-of-Thought prompting might be expected to provide some benefit. However, the results indicate that the effect of adding the instruction *"Let's think step by step"* varies substantially with question difficulty. For easy questions, only occasional and limited improvements are observed. In medium-difficulty questions, the effect is ambiguous and inconsistent, with some models improving, others remaining stable, and others exhibiting performance declines. For hard questions, implicit CoT generally fails to produce benefits and often leads to reduced accuracy, plausibly due to overthinking that introduces additional planning or calculation errors.

Statistical significance testing supports this interpretation. No robust differences emerge in the overall sample or within the medium- and hard-difficulty subsets, indicating that the observed variations are not systematic. The only borderline signal appears for Claude Sonnet 4 in the overall group; however, its p-values lie very close to the conventional threshold, suggesting that this effect is likely contingent rather than indicative of a stable underlying pattern.

Overall, the two comparisons show that implicit CoT is not a reliable strategy: the observed effects are weak, non-significant in most cases, and sometimes even negative. While fluctuations in single-choice tasks tend to be negligible, implicit CoT becomes increasingly counterproductive in numerical questions as task difficulty increases.

Explicit CoT: B4 vs B5

A markedly different picture emerges from the comparison B4 vs B5, dedicated to explicit CoT. In this case, models were required not only to *"think step by step"* but also to make the reasoning sequence explicit. The results highlight a clear and consistent improvement: almost all models significantly increased their accuracy in medium and hard

numerical answers.

The benefits are particularly substantial for models that had shown the weakest performance with implicit CoT: Claude-Sonnet 4, Gemini 2.5 Flash-Lite, and DeepSeek-v3.1 report improvements ranging from +30 to +66 percentage points, recovering much of their performance gap. GPT-5, GPT-5 nano and Claude Haiku 3.5 also show relevant improvements, while GPT-5 mini and Gemini 2.5 Flash record appreciable gains that strengthen their positioning, though without reaching the absolute levels of GPT-5. Unlike implicit CoT, these improvements are corroborated by significance tests, which confirm the robustness of the effects for most models.

From an interpretative perspective, these results demonstrate that the key difference lies not in the mere prompting of reasoning but in its explicit articulation. The requirement to display logical steps forces the model to structure its solution path, reducing the risk of disordered reasoning and uncontrolled overthinking typical of implicit CoT. In this sense, explicit CoT functions as an internal regularization mechanism, improving the coherence of responses and, especially in complex tasks, enhancing their reliability.

General Synthesis

Combining the three comparisons (B1 vs B2, B3 vs B4, and B4 vs B5), a clear conclusion emerges:

- Implicit CoT does not systematically improve LLM performance. The observed variations are weak, non-significant in most cases, and in complex scenarios tend even to reduce accuracy.
- Explicit CoT, by contrast, produces consistent and statistically robust improvements.

From a theoretical perspective, these results temper the notion, sometimes generalized in the literature, that CoT is inherently beneficial. Only the explicit articulation of reasoning steps leads to tangible improvements. From a practical perspective, this implies that implicit CoT should not be adopted as a default in managerial applications of project management, while explicit CoT can represent an effective strategy for tackling complex

tasks, provided that the higher costs and latency associated with generating extended reasoning are taken into account.

5.1.2 Question 2 – How does the performance of an LLM vary when addressing numerical questions in single-choice format compared to numerical answer format?

To address this question, a comparison was carried out between Benchmark 1, consisting exclusively of numerical questions in single-choice format, and Benchmark 3, composed of numerical answer questions. The aim was to verify whether the presence of predefined options implicitly guides the models toward the correct answer, as opposed to a format that requires autonomous generation of the numerical value.

The results show that the question format has a substantial impact on the performance of LLMs. In Benchmark 1, multiple-choice options act as an anchor, reducing the effort of autonomous generation and providing the model with useful references to narrow the space of possible answers. This mechanism translates into generally higher levels of accuracy. In Benchmark 3, by contrast, LLMs must produce the numerical value without any external support, highlighting increasing difficulties in maintaining precision, particularly in medium- and high-difficulty questions. The limited improvements observed for Claude Haiku 3.5 on easy questions remain marginal and do not alter the overall pattern of declining performance.

The integration with statistical significance tests reinforces these conclusions: the differences observed between the two formats are not only visible at a descriptive level but, in many cases, also statistically robust, particularly for models such as Claude Haiku 3.5, GPT-5 mini, GPT-5 nano, and DeepSeek-v3.1.

This confirms that the advantage of single-choice questions over numerical answers is not a random phenomenon or limited to contingent variations, but rather a systematic effect linked to the structure of the question. Notably, statistically significant differences emerge primarily in the hardest questions, indicating that as problem complexity increases, the presence of predefined options plays a critical role in constraining the reasoning process and limiting error propagation. In highly complex numerical tasks, the

absence of such constraints exposes models to planning and calculation errors, leading to a pronounced decline in accuracy.

In summary, the comparison demonstrates that multiple-choice options provide a "positive bias" that guides LLMs toward the correct answer and supports their accuracy, whereas the absence of alternatives in the numerical answer format exposes their vulnerability in autonomous calculations. Statistical evidence consolidates this interpretation, showing that question format constitutes a decisive, rather than marginal, factor in model performance.

5.1.3 Question 3 – What insights emerge from the AHP rankings? Which LLM performs best in each benchmark, and why?

To integrate the different evaluation criteria emerging from the benchmarks, an Analytic Hierarchy Process (AHP) model was applied. This allowed the synthesis of three heterogeneous dimensions (accuracy, cost, and latency) into a single comparative index. The weights were derived from a survey conducted among project management professionals at Amazon based in Luxembourg, to capture the priorities and evaluation criteria of practitioners who routinely deal with planning, scheduling, and resource allocation in complex managerial contexts. It is important to emphasize, however, that these parameters are inherently subjective, as they reflect stakeholder judgments. Consequently, variations in the assigned weights may lead to changes in the final rankings.

The resulting rankings nevertheless allow for several relevant considerations. In the single-choice benchmarks (B1 and B2), the top-performing models are GPT-5 nano, DeepSeek-V3.1, and Gemini-2.5 Flash. GPT-5 nano ranks first in B1 but drops to second place in B2, reflecting its sensitivity to implicit Chain-of-Thought prompting. By contrast, DeepSeek-V3.1 moves from second place in B1 to the top position in B2, indicating greater robustness under the same prompting condition. Gemini-2.5 Flash consistently secures third place in both benchmarks, supported by a favorable trade-off between accuracy and cost. However, its comparatively high latency negatively affects its overall ranking, preventing it from challenging the leading positions.

In the numerical benchmarks without CoT (B3) and with implicit CoT (B4), both GPT-5 mini and Gemini 2.5 Flash consistently appear among the top three ranked models.

GPT-5 mini's strong positioning is primarily driven by its comparatively higher accuracy, which compensates for its higher computational costs and latency. Gemini 2.5 Flash also performs well, benefiting from a combination of solid accuracy and relatively low cost.

In B3, GPT-5 ranks third; however, in B4 it drops to sixth place, largely due to a marked increase in latency following the introduction of implicit CoT. In the same benchmark, Claude Haiku 3.5 attains second place, supported by its very low response time and low cost.

By contrast, Claude Sonnet 4 and DeepSeek-V3.1 consistently occupy the lowest positions in the rankings. Claude Sonnet 4 is penalized by limited accuracy combined with high costs, while DeepSeek-V3.1 is disadvantaged by lower accuracy and comparatively high response times.

In the explicit CoT benchmark (B5), a clear reordering of the rankings emerges. Claude Haiku 3.5 attains the top position, overtaking both Gemini 2.5 Flash and GPT-5. This result suggests that, when explicit reasoning is required, lighter models can achieve a more favorable balance between performance and resource consumption. GPT-5, while not dominant, consistently remains among the top performers, holding the third position as already observed in B3. Notably, DeepSeek-V3.1 and Claude Sonnet 4 exhibit a marked improvement compared to previous benchmarks, climbing several positions in the ranking. This shift suggests that explicit reasoning steps help mitigate their earlier errors.

In summary, the AHP analysis reveals that there is no single "absolute winner" across all benchmarks. Rather, each model excels in specific configurations depending on the trade-off between accuracy, cost, and latency. Several cross-cutting insights are worth highlighting:

- Gemini-2.5 Flash consistently secures a position among the top three models across all benchmarks, confirming its robustness and versatility. Its ranking slightly improves in numerical answer benchmarks compared to single-choice settings, indicating a stronger relative performance when numerical reasoning is required.
- DeepSeek-V3.1 exhibits strong performance in single-choice tasks, both without CoT and with implicit CoT prompting, driven by a favorable combination of solid accuracy, moderate cost, and acceptable latency. By contrast, its ranking drops

substantially in numerical open-answer tasks, primarily due to a decline in accuracy. This pattern indicates that while DeepSeek-V3.1 is well-suited to constrained decision settings, it is less reliable when required to perform fully autonomous numerical reasoning.

- GPT-5 mini shows a particularly noteworthy trajectory, dropping from first place in B3 to the lowest rank in B5. This pattern indicates weak adaptability to explicit reasoning, as explicit Chain-of-Thought reduces its accuracy compared to both implicit and no-CoT settings.
- Overall, the models most frequently on the podium, apart from single-choice tasks, are Gemini-2.5 Flash, GPT-5, and GPT-5 mini, which together emerge as the most reliable across benchmarks.

Finally, it is essential to underscore the importance of the trade-off between accuracy, cost, and latency. No model excels simultaneously across all three dimensions. Some, like GPT-5, deliver very high accuracy at the expense of resource efficiency, while others, such as Gemini Flash or GPT-5 mini, offer more balanced solutions. This represents a crucial point both theoretically, as it challenges the notion of a universally "best" model, and practically, as it guides managerial decision-making according to operational priorities and available resources.

5.1.4 Question 4 – Do LLMs perform better than humans in project management tasks?

To address this question, model performance was systematically compared against a human baseline, defined as the average accuracy achieved by a trained individual with domain- specific knowledge. This threshold provides a realistic reference point for managerial decision-making capabilities, against which the outcomes of the experimental analysis can be critically interpreted.

The evidence reveals a heterogeneous picture. On the one hand, high-end models such as GPT-5, GPT-5 mini, and Gemini-2.5 Flash frequently exceed the human baseline, demonstrating the ability to provide answers that are not only comparable but, in several cases, superior to those of a human decision-maker.

The results reveal a heterogeneous pattern. In single-choice tasks, several models exceed the human baseline, indicating strong performance in structured decision settings. By contrast, in numerical questions, surpassing the human reference level is less frequent, with most models struggling particularly on medium- and high-complexity questions.

From an interpretative perspective, this finding highlights the selective nature of performance: not all LLMs can be regarded as reliable substitutes or complements to human reasoning, but the most advanced models clearly demonstrate the capacity to compete with and in some cases outperform human evaluators. Consequently, the suitability of LLMs for decision-making in project management should not be assessed in absolute terms, but rather in relation to the specific model adopted and the decision context in which it is applied.

In summary, comparison with the human baseline confirms that the technology has reached a sufficient degree of maturity to represent a concrete support for managerial decision making, provided that the variability across models is carefully considered. The ability of the best-performing LLMs to surpass human accuracy underscores their potential, while the weaker performance of other models calls for a cautious and selective adoption.

5.2 Secondary Results

The section dedicated to the secondary results complements the analysis of the main benchmarks by examining aspects that, while not directly linked to the core research questions, contribute to a more comprehensive understanding of LLM performance in project management decision-making.

Three areas are particularly relevant: the trade-offs between accuracy, cost, and latency, which reflect the different design strategies adopted by providers; the perceptions collected through the survey, which shed light on the priorities of potential users; and finally, the analysis of error patterns in explicit reasoning tasks, which distinguishes between calculation, interpretation, and planning limitations.

These results do not alter the overall conclusions of the thesis but add depth and detail to the interpretation of the benchmarks, offering a more nuanced perspective that is closely aligned with the practical needs of project management.

5.2.1 Performance Trade-offs in LLMs

A noteworthy secondary finding emerging from the benchmarking exercise concerns the heterogeneity of performance across models and providers in the project management domain. The observed differences are not incidental; rather, they reflect the underlying design choices and market positioning strategies explicitly outlined by the respective developers. Therefore, the analysis carried out in this study confirms that the general features ascribed to various LLMs, particularly in terms of accuracy, cost, and latency, also hold when these models are applied to specific operational contexts such as project management.

For OpenAI, the GPT-5 family illustrates most clearly a strategy oriented towards maximizing accuracy. The flagship version delivered consistently high performance even in demanding numerical tasks, frequently standing out as the only model capable of exceeding the human-level accuracy threshold. However, this robustness comes at the expense of significantly higher costs and slower response times compared to other providers. The lighter variants, GPT-5 mini and nano, behaved as expected: they ensured faster outputs and lower costs, but with a progressive decline in accuracy as task complexity increased.

A different pattern emerged for Anthropic. The Claude models (Sonnet 4 and Haiku 3.5) performed well in multiple-choice theoretical questions, where textual coherence plays a central role, yet struggled with numerical answer tasks. This polarization reflects a design philosophy prioritizing safety and discursive consistency over autonomous calculation capabilities. It is therefore unsurprising that, particularly under implicit chain-of-thought conditions, some of these models exhibited sharp drops in accuracy.

The Gemini-2.5 family by Google demonstrated a more balanced approach. Flash proved to be one of the strongest compromises, approaching GPT family's performance levels while maintaining considerably lower costs and latency. In contrast, Flash-Lite pushed efficiency to its limit: extremely fast and inexpensive, but with unstable accuracy and sensitivity to task complexity. This internal differentiation confirms Google's strategic positioning, which emphasizes flexible deployment options tailored to varying operational requirements.

Finally, DeepSeek-V3.1 reaffirmed its orientation toward cost efficiency. Its very low costs and reduced latency make the model attractive in scenarios where efficiency consid-

erations outweigh accuracy requirements. However, its performance proved highly sensitive to task format: accuracy dropped sharply when moving from single-choice questions to numerical open-answer tasks of medium and high difficulty, while a strong recovery is observed under explicit CoT. This pattern underscores a pronounced dependence on both question structure and prompting strategy, suggesting that DeepSeek-V3.1 is best suited to carefully selected use cases rather than general-purpose deployment.

Taken together, these secondary results show that the characteristics described by providers in their official documentation are consistently borne out within the project management domain. OpenAI stands out for accuracy, Anthropic for textual coherence, Google for balancing efficiency with performance, and DeepSeek for economic accessibility. In this respect, architectural and strategic choices do not remain abstract claims; they translate into concrete outcomes when LLMs are applied to complex and realistic settings such as project management.

5.2.2 Impact of Implicit CoT on Costs and Latency

An interesting secondary result from the comparisons between B1 vs B2 and B3 vs B4 concerns the impact of introducing implicit Chain-of-Thought (CoT) on costs and latency. While implicit CoT did not consistently improve performance, its effect on response times and computational resources deserves further consideration.

Costs

The introduction of implicit CoT resulted in a significant increase in operational costs for several models. For example, models like GPT-5 nano and Gemini 2.5 Flash-Lite showed a notable rise in costs, suggesting that adding CoT introduces computational complexity that does not always lead to better performance. This is especially relevant in business settings where economic efficiency is crucial. In these cases, the extra computational effort required for CoT could outweigh its cognitive benefits.

Latency

The results show that, although a few models such as GPT-5 experience a reduction in latency following the introduction of implicit CoT, the majority of models exhibit an increase in response time, with particularly pronounced effects observed for models such as Gemini 2.5 Flash-Lite. This indicates that the impact on latency is largely unfavorable and depends on how individual models process the additional reasoning steps introduced by CoT , as well as on the computational resources required to support this processing. Increased latency may limit the use of implicit CoT in business applications that require high computational performance, where response times are critical.

These findings emphasize the importance of considering both costs and latency when evaluating the potential of CoT in real-world applications. While implicit CoT may offer some benefits in certain scenarios, its impact on operational efficiency could restrict its use in environments where speed and cost-effectiveness are essential.

5.2.3 Survey Results and Evaluators' Perceptions

An additional secondary result derives from the survey conducted to capture stakeholder preferences regarding evaluation criteria. Although initially conceived as a methodological tool to support the AHP process, the survey provided valuable insights into how LLMs are perceived in the context of project management.

The responses revealed a very clear hierarchy of priorities: accuracy was regarded as the dominant criterion, while latency and cost were considered considerably less important. This orientation reflects the central role of correctness in project management decision-making, where errors in evaluation can lead to significant operational and economic consequences.

The relatively lower weights assigned to cost and latency indicate that respondents are willing to tolerate higher expenditures and longer response times, provided that output quality is preserved. This result partially diverges from the benchmarks, where models revealed the need to manage systematic trade-offs between performance, efficiency, and economic sustainability. An interesting contrast therefore emerges: while the experimental data highlight the inevitability of trade-offs, the perceptions collected through

the survey tend to minimize them, placing accuracy as the dominant and non-negotiable criterion.

It is important to note, however, that these results retain an inherently subjective component, as they derive from the evaluations of a limited sample of 30 Amazon Project Managers based in Luxembourg. While the respondents routinely operate in complex project management environments, the resulting weights cannot be regarded as fully representative of all potential stakeholders or organizational contexts. Rather, they should be interpreted as reflecting the priorities and sensitivities of a specific group of practitioners, providing an informed yet circumscribed perspective on decision-making criteria.

Overall, the survey highlights an aspect that complements the benchmarks: while the latter measure the actual performance of the models, the survey reflects the subjective priorities of practitioners who are likely to adopt these systems in real project management contexts. The integration of these two perspectives allows for a broader understanding, in which accuracy emerges as a non-negotiable criterion, whereas latency and cost play a secondary and instrumental role.

5.2.4 Performance Across Theoretical vs. Numerical Questions

The results presented in Figure 4.5 reveal clear differences in the ability of the models to answer theoretical versus numerical questions. Some models, such as GPT-5 mini and GPT-5 nano, demonstrate relatively balanced performance across both types of question, whereas others, including Claude-Sonnet 4 and Claude Haiku 3.5, perform considerably better on theoretical questions than on numerical ones. In contrast, lightweight variants, such as Gemini Flash-Lite, exhibit significant limitations in numerical calculations while performing adequately on conceptual tasks.

These discrepancies suggest that models are not universally suitable for all types of tasks: some excel when linguistic understanding and abstract reasoning are required, whereas others are better suited for numerical computation.

From an applied perspective, this underscores the importance of context-specific model selection in project management: depending on the task requirements, whether conceptual reasoning or numerical accuracy, the choice of the most appropriate model may vary.

5.2.5 Understanding Error Patterns in Explicit Reasoning

A further result of the analysis relates to the decomposition of performance in numerical questions with explicit reasoning. Errors were classified into calculation and reasoning errors, with the latter divided into interpretation and pianification errors.

The data show that the ability to perform calculations is consistently strong across all models, with the exception of Claude Haiku 3.5, with scores ranging from 0.75 for Gemini Flash-Lite up to 1.00 for GPT-5.

By contrast, performance in the reasoning dimension is systematically lower across all models, with scores spanning from 0.913 for GPT-5 down to 0.535 for Claude Haiku 3.5, indicating that reasoning constitutes a more challenging capability than calculation for the entire set of evaluated models.

A closer inspection of reasoning errors reveals a differentiated pattern across models. For GPT-based models and DeepSeek-V3.1, interpretation errors dominate, indicating that difficulties often arise already at the stage of correctly understanding or framing the problem. By contrast, the Claude models and Gemini 2.5 Flash exhibit a higher incidence of planning errors than interpretation errors, suggesting that, although the task is generally understood, these models struggle to structure multi-step reasoning or to consistently execute the required computational logic. This disparity underscores that reasoning failures arise from multiple sources, reflecting model-specific deficiencies in issue interpretation or reasoning orchestration.

These findings suggest that numerical computation is a relatively stable capability among LLMs, while reasoning remains a more fragile area, particularly when tasks are difficult to interpret and involve the integration of multiple data points. Introducing this distinction allows for a more detailed interpretation of model performance and brings the evaluation closer to cognitive perspectives, making it possible to identify not only how often models fail, but also why they fail.

5.3 Theoretical Implications

The analysis conducted in this study goes beyond the presentation of empirical findings, offering theoretical reflections that enrich the broader debate on LLMs. Previous litera-

ture has largely assessed models using generic benchmarks, often focused on linguistic or abstract reasoning tasks. However, the results of this research highlight the need to revise and extend such approaches, incorporating perspectives more closely aligned with the challenges of professional domains such as project management.

This section discusses the main theoretical implications that emerged, organized according to the methodological and analytical dimensions that guided the study.

5.3.1 Domain-specific benchmarks

A first theoretical contribution lies in the development of a methodology for constructing benchmarks tailored to professional domains, such as project management. The adoption of targeted datasets, combined with dedicated prompting techniques and domain-calibrated evaluation tools, demonstrates the limitations of generic benchmarks which, although widely used in the literature, fail to capture the complexity of real-world LLM applications. This points to the need for domain-specific benchmarking frameworks that are able to provide more meaningful measures of performance and of actual utility of the models in concrete scenarios.

5.3.2 Task design and the difficulty pyramid

The research also highlights the crucial role of task design in evaluating model performance. The use of different formats (single choice, numerical and numerical with reasoning) showed that, even with identical content, the results of the model can vary significantly. This indicates that performance depends not only on the intrinsic capabilities of LLMs, but also on the way in which tasks are structured.

To systematize this complexity, the study drew on Bloom's taxonomy, which classifies cognitive activities along a progression from basic to advanced levels. Building on this framework, a difficulty pyramid was developed to organize the questions gradually, from simple tasks related to recognition or recall to more complex ones that require articulated reasoning and the explicit presentation of logical steps. This progression is not only of methodological value but also reflects managerial practice: while it is unlikely that decision-makers deal with basic multiple-choice questions, they frequently face open-

ended numerical problems requiring structured reasoning. The theoretical implication is that task design, when organized through a Bloom-inspired hierarchy of difficulty, becomes a critical variable for benchmarking LLMs. Only through this approach can one meaningfully assess their ability to support complex decision-making processes and address real managerial needs.

5.3.3 Error taxonomy

Another theoretical contribution stems from the introduction of a new taxonomy of errors. In numerical questions with explicit reasoning, evaluation went beyond the binary distinction between correct and incorrect answers, encompassing different error types: calculation errors (numerical mistakes) and reasoning errors. The latter were further divided into interpretation errors (misunderstanding of the task or prompt) and planning errors (mistakes in structuring the reasoning or applying formulas). This framework enriches the theoretical literature by aligning the evaluation of LLMs more closely with cognitive models, as it enables assessment not only of how often models fail, but also of how they fail.

5.3.4 The role of CoT

A key point concerns the theoretical reflection on Chain of Thought (CoT). The literature has predominantly focused on explicit CoT, where models are required to make their reasoning transparent, and it is often portrayed as universally beneficial. This study examined both explicit CoT and implicit CoT, the latter involving only the final answer without requiring logical steps to be displayed.

The results show that implicit CoT, in most cases, does not have a statistically significant impact on performance and, in some situations, may even reduce accuracy. Explicit CoT, by contrast, improved output quality for the majority of models tested by encouraging a more structured reasoning process. The theoretical implication is that the benefits of CoT cannot be assumed to apply universally, but must be interpreted in relation to both the model and the application context. This calls for a more critical and contextualized perspective, challenging theories that frame CoT as an inherently advantageous strategy.

5.3.5 Survey and AHP

The research also integrated a survey with the Analytic Hierarchy Process (AHP), introducing a socio-technical dimension into the evaluation framework. This approach made it possible to consider three criteria simultaneously, accuracy, cost, and latency, and derive relative weights based on the preferences of the participants. The theoretical implication is twofold. First, it broadens the perspective of benchmarking models, shifting from a purely technical analysis to a multi-criteria evaluation. Second, it recognizes that stakeholder perceptions play a crucial role in adoption processes. The convergence between empirical results and subjective perceptions reinforces the robustness of the framework, while divergences highlight potential areas of tension between what models deliver and what users expect. In this way, the evaluation of LLMs is reframed not only as a technical exercise, but also as a social and contextual one.

5.3.6 Statistical validation of results

Finally, the use of statistical tools such as the McNemar test allowed verification of whether the observed differences between models and conditions were statistically significant. This methodological step, still relatively uncommon in the LLM benchmarking literature, enabled a distinction between robust effects and random fluctuations. The theoretical implication is that evaluation models should rest on a solid inferential basis, moving beyond simple percentage comparisons and promoting a more rigorous and reliable approach to studying performance.

5.4 Practical Implications

Beyond their theoretical significance, the findings of this study also provide a set of practical insights that can assist organizations and managers in the adoption of LLMs within project management operations. Whereas earlier research has often described the potential of such models in broad or generic terms, the results presented here underline the need to turn empirical evidence into concrete guidance that can inform managerial decisions and support the selection of models suited to real operational settings.

This section outlines the principal practical implications emerging from the analysis and considers how they may influence both the strategic choices of firms and the development directions pursued by technology providers.

5.4.1 Defining priorities and making informed model choices

The results of this study indicate that there is no single "best" model in absolute terms. Each provider follows a distinct strategy and offers specific trade-offs between accuracy, cost, and latency. For companies, this means that selection cannot rely on generic rankings but must instead be guided by internal priorities and the operational context. Within this perspective, tools such as surveys combined with the Analytic Hierarchy Process (AHP) take on strategic value. When applied inside an organization, they make it possible to capture stakeholder preferences and translate them into concrete evaluation criteria, producing tailored rankings that reflect the firm's actual needs. For instance, a company that places the highest emphasis on accuracy is likely to opt for models such as GPT-5, while those facing tighter constraints on cost or latency may find Gemini or DeepSeek more suitable.

Surveys are not only useful for firms but also for providers. They offer a means of better understanding market needs and of steering the development of solutions that align with stakeholder priorities. The practical implication is that the selection and evolution of LLMs should follow a fit for purpose logic, built on a clear identification of priorities emerging both from the demand side (companies and users) and the supply side (providers).

5.4.2 Formulating queries for LLMs

A second practical implication concerns the way managers interact with the models. The study demonstrates that the phrasing of queries has a decisive impact on the quality of responses. If a manager prefers to receive only the final answer without an explanation of the reasoning, at present GPT-5 is the only model that maintains a good level of accuracy even without an explicit Chain of Thought. For the other models, however, the results of Benchmark 5 suggest that explicit CoT should be used, as it makes the reasoning

process transparent and reduces the risk of errors. This means that firms should not limit themselves to selecting a model but also need to develop skills and internal guidelines for prompting, so as to identify the most effective interaction style for their operational and decision-making needs.

5.4.3 Summary

In conclusion, the practical implications of this research revolve around two key aspects. First, the definition of priorities, supported by tools such as surveys and AHP, which enables companies to select models that truly match their requirements while also providing providers with valuable indications for future development. Second, the formulation of queries, which acts as a concrete lever for improving the reliability and usefulness of outputs, thereby ensuring that LLMs can be employed more effectively in supporting decision-making processes in project management contexts.

Chapter 6

Conclusions

This study started from a clear observation: as projects and supply chains become increasingly complex, managers need effective tools to support decision-making. Large Language Models (LLMs) have emerged rapidly and show significant potential. However, two central questions remained: can these models be trusted in real managerial contexts, and how can they be evaluated systematically and rigorously?

Two main challenges characterize the current debate. First, managers often adopt new technologies without sufficient evidence of their effectiveness, exposing organizations to costly or suboptimal decisions. Second, academic research has not yet developed benchmarks specific to professional domains, which are necessary for systematic and comparable evaluations, particularly in project management.

This research addressed two primary questions: (RQ1) which combinations of datasets, evaluation metrics, and prompting techniques are best suited for constructing meaningful and replicable benchmarks; and (RQ2) which models provide the most reliable performance for practical managerial use.

To address RQ1, the study proposed a methodological framework that combines project management–specific datasets, calibrated prompting strategies, and a wide set of evaluation metrics. It integrates multiple question formats, a hierarchy of difficulty inspired by Bloom's taxonomy, a detailed error taxonomy, and statistical validation. The resulting benchmarks are systematic, replicable, and closely aligned with real-world decision-making requirements, providing both a foundation for future research and practical guidance for organizations.

The role of Chain of Thought (CoT) was carefully examined. Implicit CoT, where reasoning occurs internally without being displayed, does not consistently improve accuracy and, in some cases, reduces it. Moreover, implicit CoT increases operational costs and latency, affecting efficiency and limiting its practical use in time- and resource-sensitive environments. By contrast, explicit Chain-of-Thought prompting, which requires models to articulate their reasoning, improves response quality for nearly all models, although the magnitude of its impact varies across tasks and models.

Regarding RQ2, no single model outperformed all others across all benchmarks. GPT-5 achieved the highest accuracy in most of the benchmarks but incurred higher costs and slower response times. Gemini Flash offered a balanced trade-off between performance, cost, and latency. GPT-5 mini exhibited consistently high accuracy in both theoretical and numerical questions, showing limited sensitivity to the introduction of CoT prompting. Claude performed well in theoretical questions tasks but struggled with numerical problems, while DeepSeek prioritized cost-efficiency at the expense of performance on complex tasks. The error analysis indicates that many failures stem from difficulties in interpreting complex questions and integrating multiple pieces of information, rather than from limitations in basic numerical computation alone.

The survey integrated with the Analytic Hierarchy Process (AHP) added an additional socio-technical perspective. While the benchmarks showed trade-offs among accuracy, cost, and latency, respondents consistently emphasized accuracy as the top priority, confirming its central importance in managerial decision-making.

In conclusion, this study contributes both theoretically and practically. Theoretically, it provides replicable, domain-specific, and multidimensional benchmarks that account for cognitive processes and operational contexts. Practically, it offers managers and providers tools to make informed model selections, optimize prompting strategies, and align model performance with organizational priorities.

Ultimately, the research answers the key question: can LLMs be trusted to support decision-making in project management contexts? The answer is cautiously positive. Leading models, such as GPT-5, GPT-5 mini, and in specific cases Gemini Flash, not only match but sometimes surpass human performance, showing potential as decision-support partners. Nevertheless, variability across models and persistent challenges in

reasoning indicate that adoption should remain selective, context-aware, and guided by robust benchmarks, with a clear understanding of the associated trade-offs.

6.1 Delimitations

The study was intentionally framed within specific boundaries to maintain alignment with its objectives.

First, the focus was placed on project management, a domain in which systematic benchmarks for evaluating LLM performance are still limited, despite the fact that managerial decisions in this area have a direct impact on planning efficiency, coordination, and overall project outcomes.

The benchmark itself was based on pre-defined question types (single choice, numerical, and numerical with reasoning), organized in a hierarchy of difficulty inspired by Bloom's taxonomy. Broader assessment formats, such as extended case studies, were deliberately excluded. While such formats might mirror real-world practices more closely, they would have introduced methodological complexity inconsistent with the need for replicability and systematic comparison.

Regarding model selection, the analysis was restricted to a set of current commercial LLMs, excluding open-source solutions and earlier versions. This decision allowed the study to concentrate on technologies most relevant to present-day business applications.

Finally, prompting techniques such as Zero-Shot e Role prompting were chosen to reflect practical usage scenarios. More advanced approaches, such as Tree-of-Thought or ReAct, were not considered, as they require significantly greater computational resources and were deemed inconsistent with the pragmatic orientation of the study.

6.2 Limitations

Alongside the deliberate choices made in this study, a few limitations need to be acknowledged, as they affect how the results can be interpreted and applied.

First, the AHP survey was conducted with a sample of 30 Amazon Project Managers based in Luxembourg. While this respondent profile ensures a strong alignment with real

organizational decision-making, the results may nonetheless reflect priorities that are specific to Amazon's internal processes, culture, and operating environment. Consequently, the identified priorities may not be fully generalizable to other organizations or industries characterized by different project management practices and operational constraints.

Second, practical and financial constraints limited the number of times each model could be tested. Running additional trials would have made it possible to check the consistency of the results and reduce variability. The study also did not include high-cost models like Claude Opus or open-source solutions such as Llama, which naturally narrowed the comparison.

Third, LLMs themselves can be unpredictable. Factors such as temperature settings or updates from the provider can affect their responses in ways that are difficult to control, adding a degree of uncertainty to the results and limiting how broadly they can be applied.

The benchmark, although carefully designed, was built using a relatively small set of questions and scenarios from two academic institutions (Politecnico di Torino, Eindhoven University of Technology TU/e), and internationally recognized professional certifications. While these sources are of high quality and encompass both academic and professional domains, the resulting benchmark entirely may not fully reflect the diversity and complexity of decision-making scenarios faced in various project management contexts.

In addition, the study did not directly address some intrinsic limitations of LLMs, including the risk of biased outputs, the possibility of generating hallucinations, and the lack of transparency and explainability that often characterizes these systems. These issues have important implications for fairness, reliability, and accountability.

Finally, the statistical analysis relied on McNemar's test, which is appropriate for comparing classifiers on the same set of cases. However, its reliability depends on the number of discordant results, which was relatively small in this study. This means that the high p-values do not show that the models with and without Chain-of-Thought are equivalent, they simply indicate that, with the available data, no significant differences could be detected. Stronger conclusions would require a larger sample or more varied cases.

6.3 Future Research Streams

Several avenues for further research emerge from this study.

A first line of research concerns cross-domain applications. Extending the methodology to contexts beyond project management, such as supply chain management or finance, would make it possible to verify the replicability of the benchmark and to assess the adaptability of the results to heterogeneous professional domains.At the same time, within project management itself, the use of domain-specific datasets in key areas such as project scheduling, cost estimation, resource allocation, risk management, quality management, and stakeholder communication would allow for more targeted testing of model performance, highlighting strengths and weaknesses across different managerial contexts.

A second area of exploration concerns the integration of additional question types and evaluation metrics, as outlined in Table 3.3, to broaden the scope of the benchmark.

A third stream combines statistical robustness with the study of model variability. Increasing the total number of benchmark questions would strengthen the reliability of significance tests, while targeted analyses of generation parameters (such as temperature) and advanced prompting strategies (e.g., self-consistency) would provide deeper insight into the stability of LLM outputs. Together, these steps would help distinguish random fluctuations from structural variability in model behavior.

A fourth development lies in the design of dynamic benchmarks, where interaction between user and model plays a central role. Incorporating *Multi-turns* could bring evaluations closer to real-world usage scenarios, where iterative and adaptive exchanges are common.

A promising direction for future research is to expand the stakeholder base beyond Amazon managers by involving practitioners from a broader range of companies and industries in the AHP survey. This would enable a comparative analysis of whether the priorities identified in this study are generalizable across organizational contexts or instead reflect company-specific dynamics.

References

Balloccu, S. et al. (Feb. 2024). *Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs*.

Banh, L. & Strobel, G. (2023). "Generative artificial intelligence". *Electronic Markets*, 33, 63. 10.1007/s12525-023-00680-1.

Bartz-Beielstein, T. et al. (Dec. 2020). *Benchmarking in Optimization: Best Practice and Open Issues*.

Bengesi, S. et al. (Nov. 2023). "Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers". *IEEE Access*, 12, 69812–69837. ISSN: 21693536. 10.1109/ACCESS.2024.3397775.

Brand, J. et al. (2023). *Using LLMs for Market Research*.

Brown, T. B. et al. (May 2020). "Language Models are Few-Shot Learners". *Advances in Neural Information Processing Systems*, 2020-December. ISSN: 10495258.

Brynjolfsson, E. & Mitchell, T. (Dec. 2017). "What can machine learning do? Workforce implications: Profound change is coming, but roles for humans remain". *Science*, 358, 1530–1534. ISSN: 10959203. 10.1126/SCIENCE.AAP8062/SUPPL_FILE/AAP8062-BRYNJOLFSSON-SM.PDF.

Busch, K. & Leopold, H. (Oct. 2024). *Towards a Benchmark for Large Language Models for Business Process Management Tasks*.

Chang, Y. et al. (Mar. 2024). *A Survey on Evaluation of Large Language Models*.

Chen, P. et al. (2024). *CoMM: Collaborative Multi-Agent, Multi-Reasoning-Path Prompting for Complex Problem Solving*.

Christiano, P. F. et al. (June 2017). "Deep reinforcement learning from human preferences". *Advances in Neural Information Processing Systems*, 2017-December, 4300–4308. ISSN: 10495258.

Cinkusz, K. et al. (Jan. 2024). “Cognitive Agents Powered by Large Language Models for Agile Software Project Management”. *Electronics (Switzerland)*, 14. ISSN: 20799292. 10.3390/ELECTRONICS14010087.

Clavié, B. et al. (2023). “Large Language Models in the Workplace: A Case Study on Prompt Engineering for Job Type Classification”. *Lecture Notes in Computer Science (LNCS)*. Vol. 13913. Springer, 3–17. 10.1007/978-3-031-35320-8_1.

Cunningham, P. et al. (2008). “Supervised Learning”. *Cognitive Technologies*, 21–49. ISSN: 16112482. 10.1007/978-3-540-75171-7_2.

Dam, S. K. et al. (Nov. 2024). *A Complete Survey on LLM-based AI Chatbots*.

Dol, M. & Geetha, A. (Aug. 2021). “A Learning Transition from Machine Learning to Deep Learning: A Survey”. *Proceedings of the 2021 International Conference on Emerging Techniques in Computational Intelligence, ICETCI 2021*, 89–94.

Dziri, N. et al. (2022). “On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models?” *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022)*. Association for Computational Linguistics, 5271–5285. 10.18653/v1/2022.naacl-main.387.

Einola, K. & Khoreva, V. (Jan. 2023). “Best friend or broken tool? Exploring the co-existence of humans and artificial intelligence in the workplace ecosystem”. *Human Resource Management*, 62, 117–135. ISSN: 1099050X.

Elkins, S. et al. (2024). “How Teachers Can Use Large Language Models and Bloom’s Taxonomy to Create Educational Quizzes”. *arXiv preprint arXiv:2401.05914*.

Eriksson, M. et al. (May 2025). *Can We Trust AI Benchmarks? An Interdisciplinary Review of Current Issues in AI Evaluation*.

Ferrara, E. (Nov. 2023). “Should ChatGPT be Biased? Challenges and Risks of Bias in Large Language Models”. *First Monday*, 28. 10.5210/fm.v28i11.13346.

Franke, S. et al. (2025). “Can ChatGPT Solve Undergraduate Exams from Warehousing Studies? An Investigation”. *Computers*, 14:2, 52. ISSN: 2073-431X. 10.3390/computers14020052.

George, S. et al. (Feb. 2023). “A Review of ChatGPT AI’s Impact on Several Business Sectors”. *Partners Universal International Innovation Journal*, 1, 9–23. ISSN: 2583-9675. 10.5281/ZENODO.7644359.

Gignac, G. E. & Szodorai, E. T. (May 2024). “Defining intelligence: Bridging the gap between human and artificial perspectives”. *Intelligence*, 104, 101832. ISSN: 0160-2896. 10.1016/J.INTELL.2024.101832.

Goodfellow, I., Bengio, Y., et al. (Oct. 2017). “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning”. *Genetic Programming and Evolvable Machines 2017 19:1*, 19, 305–307. ISSN: 1573-7632. 10.1007/S10710-017-9314-Z.

Goodfellow, I., Pouget-Abadie, J., et al. (Oct. 2020). “Generative adversarial networks”. *Communications of the ACM*, 63, 139–144. ISSN: 15577317. 10.1145/3422622.

Gu, J. et al. (Mar. 2025). *A Survey on LLM-as-a-Judge*.

Guo, X. et al. (2025). “FinEval: A Chinese Financial Domain Knowledge Evaluation Benchmark for Large Language Models”. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. Long Papers, 6258–6292.

Haleem, A. et al. (Oct. 2022). “An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges”. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 2, 100089. ISSN: 2772-4859.

Handley, L. (June 2023). *Supply chains: How AI could ‘remove all human touchpoints’*.

Hendrycks, D. et al. (Jan. 2021). *Measuring Massive Multitask Language Understanding*.

Ho, X. et al. (Nov. 2020). *Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps*.

Hoek, R. V. et al. (Nov. 2022). *How Walmart Automated Supplier Negotiations*.

Horzyk, A. et al. (2023). “Construction and Training of Multi-Associative Graph Networks”. *Lecture Notes in Computer Science*. Vol. 14171 LNAI. Springer, Cham, 277–292. ISBN: 978-3-031-43418-1. 10.1007/978-3-031-43418-1_17.

Jackson, I. et al. (2024). “Generative artificial intelligence in supply chain and operations management: a capability-based framework for analysis and implementation”. *International Journal of Production Research*, 62, 6120–6145. ISSN: 1366588X.

Janiesch, C. et al. (Sept. 2021). “Machine learning and deep learning”. *Electronic Markets*, 31, 685–695. ISSN: 14228890. 10.1007/S12525-021-00475-2/TABLES/2.

Ji, Z. et al. (Dec. 2023). “Survey of Hallucination in Natural Language Generation”. *ACM Computing Surveys*, 55. ISSN: 15577341. 10.1145/3571730/ASSET/CC5D3792-8BC0-4675-8584-B507476E20EC/ASSETS/IMAGES/LARGE/CSUR-2022-0173-F01.JPG.

Jiang, Y. et al. (Mar. 2022). “Quo vadis artificial intelligence?” *Discover Artificial Intelligence 2022 2:1*, 2, 1–19. ISSN: 2731-0809. 10.1007/S44163-022-00022-8.

Kiela, D. et al. (Apr. 2021). *Dynabench: Rethinking Benchmarking in NLP*.

Kojima, T. et al. (2022). “Large Language Models are Zero-Shot Reasoners”. *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*.

Kong, A. et al. (2023). “Better Zero-Shot Reasoning with Role-Play Prompting”. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2024)*. Vol. 1. Association for Computational Linguistics, 4099–4113. 10.18653/v1/2024.naacl-long.228.

Kshetri, N. et al. (Apr. 2024). “Generative artificial intelligence in marketing: Applications, opportunities, challenges, and research agenda”. *International Journal of Information Management*, 75, 102716. ISSN: 0268-4012.

Kumar, S. et al. (Nov. 2023). *A Comprehensive Review on Sentiment Analysis: Tasks, Approaches and Applications*.

Lecun, Y. et al. (May 2015). “Deep learning”. *Nature*, 521, 436–444. ISSN: 14764687.

Lewis, B. & Crews, A. (1985). “The evolution of benchmarking as a computer performance evaluation technique. MIS Quarterly , 7–16.” *MIS Quarterly , 7–16.*

Lewis, B. C. & Crews, A. E. (1985). “The Evolution of Benchmarking as a Computer Performance Evaluation Technique”. *MIS Quarterly*.

Li, Y. (Sept. 2023). “A Practical Survey on Zero-shot Prompt Design for In-context Learning”. *International Conference Recent Advances in Natural Language Processing, RANLP*. Incoma Ltd, 641–647. 10.26615/978-954-452-092-2_069.

Li, Z. et al. (2024). “Optimizing Inventory Management using a Multi-Agent LLM System”. *Proceedings of The International Conference on Electronic Business*, 12–13.

Liang, P. et al. (2023). “Holistic Evaluation of Language Models”.

Lin, Y.-T. & Chen, Y.-N. (2023). “LLM-EVAL: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models”. *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, 47–58.

Liu, S. et al. (July 2023). "Using AI-generated suggestions from ChatGPT to optimize clinical decision support". *Journal of the American Medical Informatics Association*, 30, 1237–1245. ISSN: 1527974X. 10.1093/JAMIA/OCAD072,

Liu, Y., Cao, J., et al. (Feb. 2024). *Datasets for Large Language Models: A Comprehensive Survey*.

Liu, Y., Khandagale, S., et al. (Nov. 2021). *Synthetic Benchmarks for Scientific Research in Explainable Machine Learning*.

Liu, Y. L. et al. (June 2024). *ECBD: Evidence-Centered Benchmark Design for NLP*.

Lunardi, R. et al. (2025). "On Robustness and Reliability of Benchmark-Based Evaluation of LLMs". *arXiv preprint arXiv:2509.04013*.

Lv, Z. (Jan. 2023). "Generative artificial intelligence in the metaverse era". *Cognitive Robotics*, 3, 208–217. ISSN: 2667-2413. 10.1016/J.COGR.2023.06.001.

McIntosh, T. R. et al. (Oct. 2024). *Inadequacies of Large Language Model Benchmarks in the Era of Generative Artificial Intelligence*. 10.1109/TAI.2025.3569516.

Mehri, S. & Eskenazi, M. (2020). *USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation*.

Meske, C. et al. (Dec. 2022). "Explainable and responsible artificial intelligence". *Electronic Markets*, 32, 2103–2106. ISSN: 14228890.

Miao, X. et al. (June 2024). "Demystifying Data Management for Large Language Models". *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, 547–555. ISBN: 9798400704222. 10.1145/3626246.3654683.

Miller, J. K. & Tang, W. (May 2025). *Evaluating LLM Metrics Through Real-World Capabilities*.

Miralles-González, P. et al. (May 2025). *Pushing the boundary on Natural Language Inference*.

Mishra, S. & Arunkumar, A. (2021). "How Robust are Model Rankings: A Leaderboard Customization Approach for Equitable Evaluation". *arXiv preprint arXiv:2106.05532*.

Mohri, M. et al. (2012). *Foundations of machine learning*. MIT Press. ISBN: 9780262018258.

Moller, P. (Mar. 2023). *ChatGPT and the Like: AI in Logistics | DHL Freight*.

Mushtaq, A. et al. (2025). "WorldView-Bench: A Benchmark for Evaluating Global Cultural Perspectives in Large Language Models". *arXiv preprint arXiv:2505.09595*.

Pacchiardi, L. et al. (June 2025). “PredictaBoard: Benchmarking LLM Score Predictability”. *Findings of the Association for Computational Linguistics: ACL 2025*. Association for Computational Linguistics, 15245–15266.

Pahuja, S. et al. (Jan. 2025). “Comprehensive Review of Generative artificial Intelligence: Mechanisms, Models, and Applications”. *Procedia Computer Science*, 258, 3731–3740. ISSN: 1877-0509. 10.1016/J.PROCS.2025.04.628.

Parrish, A. et al. (Mar. 2022). *BBQ: A Hand-Built Bias Benchmark for Question Answering*.

Powers, D. M. W. (2015). *What the F-measure doesn’t measure... Features, Flaws, Fallacies and Fixes*.

Prieto, S. A. et al. (Jan. 2023). “Investigating the use of ChatGPT for the scheduling of construction projects”. *Buildings*, 13. 10.3390/buildings13040857.

Quan, Y. & Liu, Z. (May 2024). *EconLogicQA: A Question-Answering Benchmark for Evaluating Large Language Models in Economic Sequential Reasoning*.

Reuel, A. et al. (2024). “BetterBench: Assessing AI Benchmarks, Uncovering Issues, and Establishing Best Practices”. *Proceedings of the 38th International Conference on Neural Information Processing Systems*.

Reynolds, L. & McDonell, K. (Feb. 2021). “Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm”. *Conference on Human Factors in Computing Systems*. Association for Computing Machinery. ISBN: 9781450380959.

Russell, S. J. & Norvig, P. (2016). *Artificial intelligence : a modern approach*. Pearson, 1132. ISBN: 9781292153964.

Savage, N. (May 2023). “Drug discovery companies are customizing ChatGPT: here’s how”. *Nature biotechnology*, 41, 585–586. ISSN: 15461696. 10.1038/S41587-023-01788-7;KWRD=LIFE+SCIENCES.

Schramowski, P. et al. (Mar. 2022). “Large pre-trained language models contain human-like biases of what is right and wrong to do”. *Nature Machine Intelligence*, 4, 258–268. ISSN: 25225839.

Shen, Y. et al. (Apr. 2023). “ChatGPT and Other Large Language Models Are Double-edged Swords”. *Radiology*, 307, 2023. ISSN: 15271315.

Shi, J. et al. (Dec. 2025). “Optimization-based Prompt Injection Attack to LLM-as-a-Judge”. *CCS 2024 - Proceedings of the 2024 ACM SIGSAC Conference on Computer*

and Communications Security. Association for Computing Machinery, Inc, 660–674. ISBN: 9798400706363. 10.1145/3658644.3690291.

Sivarajkumar, S. et al. (2024). "An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing: Algorithm Development and Validation Study". *JMIR Medical Informatics*, 12. ISSN: 22919694. 10.2196/55318,

Skórnóg, D. & Kmiecik, M. (Oct. 2023). "Supporting the inventory management in the manufacturing company by ChatGPT". *LogForum*, Vol. 19, 535–554. ISSN: 1734-459X. 10.17270/J.LOG.2023.917.

Sokol, A. et al. (June 2025). *BenchmarkCards: Standardized Documentation for Large Language Model Benchmarks*.

Stanovich, K. E. & West, R. F. (2000). "Individual differences in reasoning: Implications for the rationality debate?" *Behavioral and Brain Sciences*, 23, 645–726. ISSN: 0140525X. 10.1017/S0140525X00003435,

Susarla, A. et al. (June 2023). "The Janus Effect of Generative AI: Charting the Path for Responsible Conduct of Scholarly Activities in Information Systems". *Information Systems Research*, 34, 399–408. ISSN: 15265536.

Talby, D. (June 2025). *Why leaderboards fall short in measuring AI model value*.

Tyagi, K. et al. (Jan. 2022). "Unsupervised learning". *Artificial Intelligence and Machine Learning for EDGE Computing*, 33–52. 10.1016/B978-0-12-824054-0.00012-5.

Vaswani, A. et al. (2017). "Attention Is All You Need". *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Wang, A. et al. (Feb. 2019). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*.

Wang, M. et al. (Sept. 2024). *Minstrel: Structural Prompt Generation with Multi-Agents Coordination for Non-AI Experts*.

Wang, X. et al. (2022). "Self-Consistency Improves Chain of Thought Reasoning in Language Models". *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. International Conference on Learning Representations.

Wang, Y. et al. (2023). "Self-Instruct: Aligning Language Models with Self-Generated Instructions". *arXiv preprint arXiv:2212.10560*.

Wei, J. et al. (2022). “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models Chain-of-Thought Prompting”. *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

White, J. et al. (2023). “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT”. *Proceedings of the 30th Conference on Pattern Languages of Programs*. 10.5555/3721041.3721046.

Winston, P. H. (1993). *Artificial intelligence*. Addison-Wesley Pub. Co. ISBN: 0201533774.

Yang, Z. et al. (Sept. 2018). *HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering*.

Yao, S., Yu, D., et al. (2023). “Tree of Thoughts: Deliberate Problem Solving with Large Language Models”. *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 36. Neural Information Processing Systems Foundation.

Yao, S., Zhao, J., et al. (2022). “ReAct: Synergizing Reasoning and Acting in Language Models”. *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*.

Zairi, M. & Leonardo, P. (1996). *Practical Benchmarking: The Complete Guide. Springer, pp. 22–27.*

Zhan, J. (Apr. 2022). *Call for establishing benchmark science and engineering*.

Zhang, B. et al. (2023). *ZhuJiu: A Multi-dimensional, Multi-faceted Chinese Benchmark for Large Language Models*. 10.18653/V1/2023.EMNLP-DEMO.44.

Zhao, J. et al. (2025). *Role-Play Paradox in Large Language Models: Reasoning Performance Gains and Ethical Dilemmas*.

Zhen, Y. et al. (2024). “LLM-Project: Automated Engineering Task Planning via Generative AI and WBS Integration”. *Proceeding of the 14th IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER 2024*. Institute of Electrical and Electronics Engineers Inc., 605–610. ISBN: 9798331506056. 10.1109/CYBER63482.2024.10749328.

Zheng, L. et al. (2023). “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena”. *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 36. Neural Information Processing Systems Foundation.

Zhong, W. et al. (2023). “AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models”. *arXiv preprint arXiv:2304.06364*.

Zhu, A. et al. (2024). “FanOutQA: A Multi-Hop, Multi-Document Question Answering Benchmark for Large Language Models”. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Vol. 2, 18–37.

Zhuang, J. (Apr. 2023). *Introducing the Instacart Plugin for ChatGPT*.

Ringraziamenti

Giunta alla conclusione di questo importante percorso, desidero ringraziare tutti coloro che hanno reso possibile questo traguardo.

Un ringraziamento speciale va al Prof. Giovanni Zenezini,relatore, e al Prof. Filippo Maria Ottaviani, correlatore, per la fiducia dimostratami nell'affidarmi un tema di tesi altamente stimolante, per l'entusiasmo trasmesso e per il costante supporto.

Ringrazio mia mamma, da sempre il mio punto di riferimento, anche quando la distanza ci separa. La tua presenza è costante: se alle elementari mi aiutavi a scrivere i temini, oggi mi accompagni all'aeroporto per partire a lavorare all'estero. In ogni fase della mia vita il tuo sostegno è insostituibile, e ogni mio traguardo sarà sempre anche tuo.

Grazie a papà per tutti i sacrifici compiuti nel corso degli anni per permettermi di inseguire e realizzare i miei sogni. Il tuo impegno, spesso silenzioso ma ininterrotto, è stato un supporto fondamentale in ogni passo del mio percorso.

Un pensiero speciale va alle mie nonne, che spero abbiano potuto vedere, attraverso i miei occhi, tutti i luoghi in cui ho vissuto durante questi anni di università. Spero di avervi rese orgogliose e di aver saputo custodire i valori che mi avete trasmesso. A Nonna Maria vorrei dire che non deve aver paura quando volo sull'"apparecchio", come lo chiamava lei; a Nonna Loretta, che non ho mai smesso di sognare, come le avevo promesso.

Infine, desidero ringraziare me stessa, per aver avuto il coraggio di affrontare e superare paure e sfide difficili. Per aver mantenuto, lungo tutto il percorso, l'umiltà di imparare dagli altri e di costruire legami autentici con persone provenienti da ogni parte del mondo.

Grazie Giulia.
Avanti così.