Universitat Rovira i Virgili

Multiple-Criteria Decision Making Systems

Multi-Criteria Evaluation and Ranking of Large Language Models for Business Applications

Daniel Arias Cámara

UNIVERSITAT ROVIRA i VIRGILI

Tarragona, 2025

Contents

1 Abstract

The rapid evolution of Large Language Models (LLMs) has resulted in a diverse ecosystem of models, each offering distinct capabilities, pricing structures, deployment options, and performance characteristics. For businesses seeking to integrate AI solutions into their operations, whether for customer support, data analysis, marketing, or automation, selecting the most suitable LLM has become a complex and strategic decision.

Today, organizations can choose from a wide range of LLMs provided by leading technology companies and open-source communities, including GPT-4 by OpenAI, Claude by Anthropic, Gemini by Google, LLaMA by Meta, and Mixtral by Mistral, among others. These models differ significantly in key dimensions such as cost, reasoning capabilities, token context window, inference speed, customization capabilities, licensing flexibility, or multilingual support. Consequently, identifying the most appropriate model requires a thorough evaluation based on multiple criteria.

In this work, we propose a structured decision-making methodology to evaluate and rank LLMs using a set of business-relevant criteria. Applying Multi-Criteria Decision Analysis (MCDA) techniques, in this case ELECTRE-III, we define and assess performance metrics such as reasoning capabilities, cost per 1M tokens, context window size, multilingual support, and speed.

To demonstrate the applicability of our approach, we present two case studies. In the first case, we evaluate the selection of an LLM for a company aiming to integrate a chatbot into its customer service system. In the second case, we evaluate the selection process for a business that is developing an AI assistant for internal data analysis. These use cases illustrate how MCDA can support informed and tailored model selection for different business needs.

2 Problem Definition

The goal of this project is to support business decision-makers in selecting the most suitable Large Language Model (LLM) for enterprise applications using a Multi-Criteria Decision Analysis (MCDA) approach. This section outlines the key elements of the decision-making problem, including the decision-maker, performance criteria, alternatives, problem type, and the selected MCDA method.

2.1 Decision Maker and Objectives

The decision maker in this case is a **Chief Technology Officer (CTO)** in a medium or large business. The goal is to identify the LLM that best aligns with the company's operational needs and strategic goals, such as maximizing the return on investment, minimizing operational risks, and ensuring scalable and secure AI integration. Preferences may vary depending on business type, but typically include low-cost, high model accuracy, customizable deployment, and data governance compliance.

2.2 Criteria and Performance Variables

A family of criteria has been selected to evaluate the LLMs. Each criterion is independent, measurable, and represents a relevant dimension of performance for business applications.

2.2.1 Reasoning Capabilities

To measure intellectual capabilities in LLMs, Rein et al. [1] introduced the *Graduate-level Google-Proof Q&A* (GPQA) questionnaire in 2023. Designed to test advanced reasoning, GPQA goes beyond pattern matching and shallow retrieval by requiring deep understanding. It features challenging science questions at the Ph.D. level of qualifying exams across disciplines like physics, chemistry, biology, and computer science. Its "Google-Proof" design ensures that answers cannot be easily found online, making it a reliable tool to evaluate true comprehension and reasoning in LLMs.

Extracted from EPOCH AI[1], Figure 1 illustrates the performance of various state-of-the-art LLMs in the GPQA Diamond subset and a clear upward trend can be observed in their reasoning capabilities. Models released in early 2023, such as GPT-4 and Claude 2, showed moderate performance, typically between 30% and 40%. However, more recent models, including Claude 3.7 Sonnet, DeepSeek-R1, and Gemini 2.5 Pro Exp, have surpassed the *Expert human level*, reaching accuracies above 70%. This suggests that AI systems are not only improving in language fluency, but also making significant strides in scientific reasoning and problem-solving capabilities.

[1] https://epoch.ai/data/ai-benchmarking-dashboard

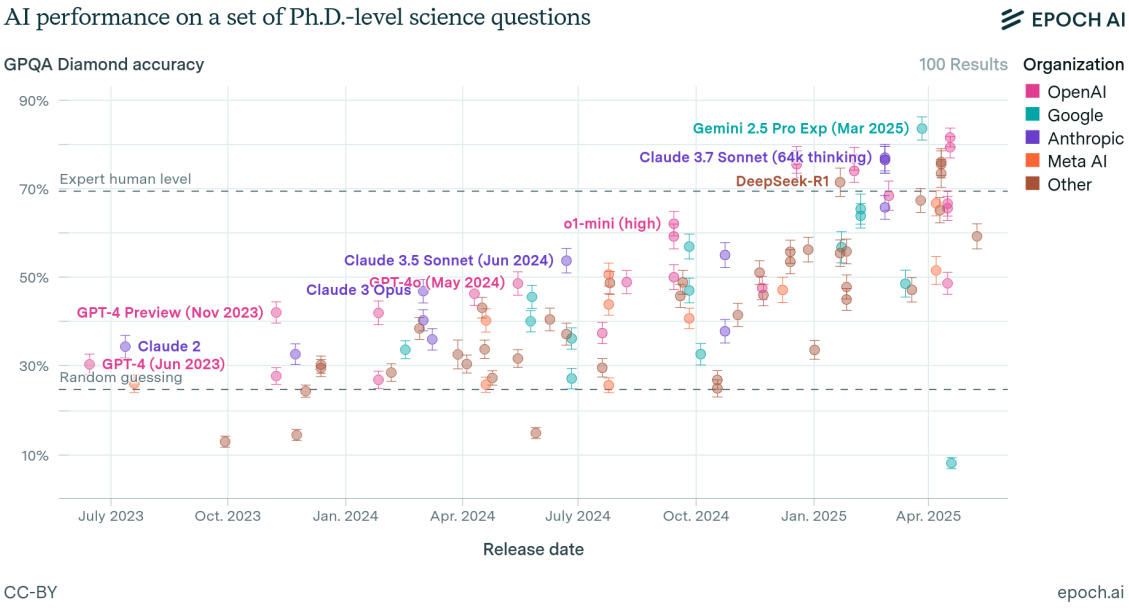

Figure 1: Performance of AI systems on GPQA Diamond in recent years

2.2.2 Cost per 1M Tokens

A token is a basic unit of text processed by LLMs. Depending on the language and context, a token may correspond to a single character, a sub-word fragment, or an entire word. For example, in English, the word "Artificial" could be broken into two tokens like "Art" and "ificial," whereas common words like "data" might be treated as a single token. Most LLMs tokenize both input and output text to calculate usage, and this token count directly impacts the cost of operating the model, especially in API-based access models [2].

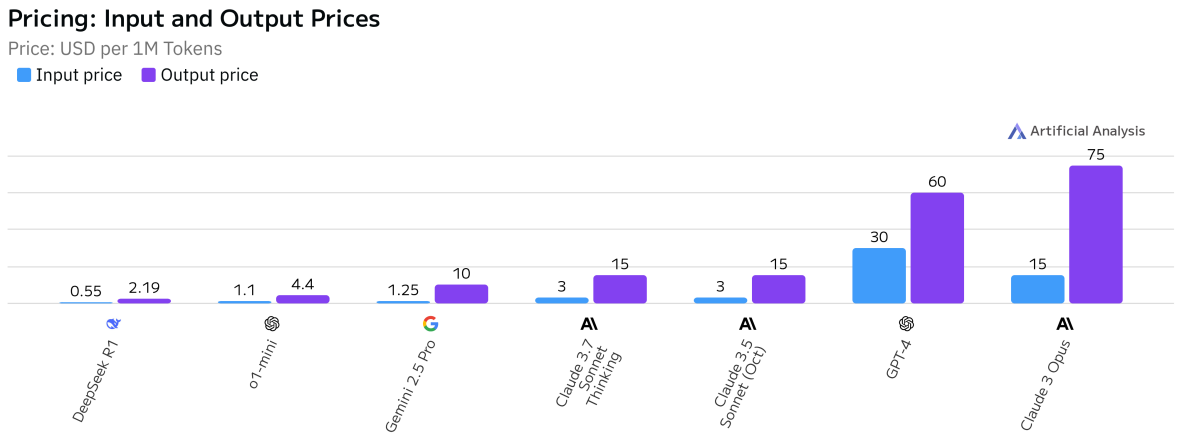

Figure 2: Input and output pricing per 1M tokens (USD)

Figure 2 presents the input and output pricing of various LLMs. The most expensive model for input tokens is GPT-4, with a cost of 30 USD per 1M tokens. In terms of output tokens, Claude 3 Opus is the most expensive, priced at 75 USD per 1M tokens. Interestingly, when comparing pricing with reasoning capabilities (as shown in Figure 1), the most expensive models are not always the most capable. For instance, Gemini 2.5 Pro and Claude 3.7 Sonnet demonstrate superior performance in the GPQA Diamond benchmark, yet they are more affordable than other less capable LLMs.

The pricing data used in this work are sourced from *OpenAI and Other LLM API Pricing Calculator*[2], a publicly available tool that aggregates and regularly updates the cost estimates of various LLM providers. This calculator enables transparent comparisons by displaying current API rates, token limits, and projected monthly costs based on usage. This criterion is essential to evaluate the economic feasibility of LLM deployment in real-world business environments.

2.2.3 Context Window Size

The context window, also known as the *context length*, refers to the maximum number of tokens that an LLM can process at once. This includes both input and output tokens. A larger context window allows the model to handle longer documents, maintain coherence over extended conversations, and incorporate more relevant information when generating responses [3].

For business applications, context window size is a critical factor in use cases such as document summarization, multi-turn customer support, and processing long legal or technical texts. Models with small context windows may lose track of previous information in lengthy interactions, which can negatively impact performance and user experience.

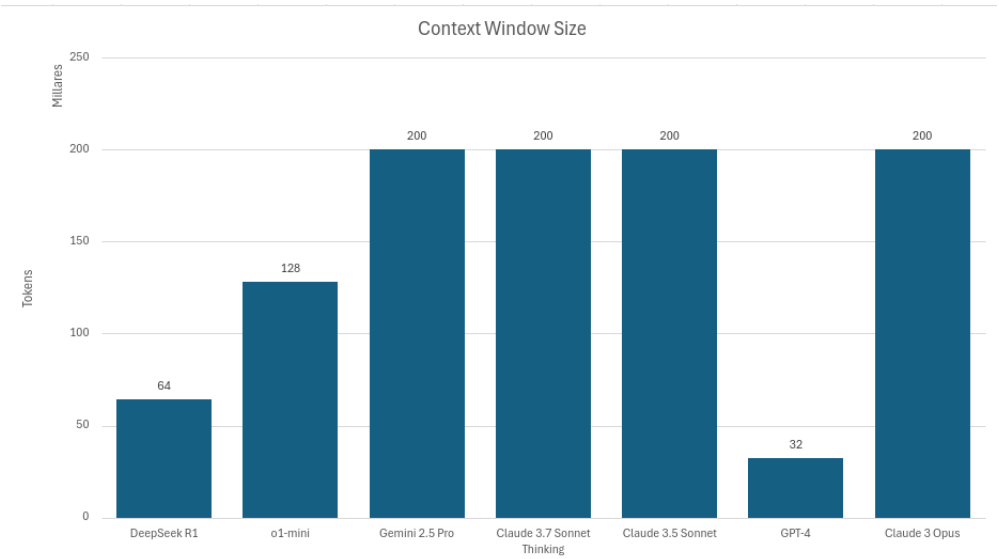

Figure 3: Context window size of selected LLMs (in thousands of tokens)

[2] https://yourgpt.ai/tools/openai-and-other-llm-api-pricing-calculator

Figure 3 shows the size of the context windows of several LLMs previosly shown in thousands of tokens. As illustrated, GPT-4 supports a maximum window size of 32K tokens, followed by DeepSeek R1 with 64K and o1-mini with 128K tokens. More advanced models such as Claude 3.7 Sonnet Thinking, Claude 3.5 Sonnet, Claude 3 Opus, and Gemini 2.5 Pro offer context windows of up to 200K tokens. Some of the newest models go even further: GPT-4.1 is reported to support up to 1 million tokens, while Gemini 1.5 Pro reaches up to 2 million tokens.

This criterion is treated as a quantitative and benefit-oriented variable, where larger context window sizes are preferred, as they enhance the model's capacity to reason over and integrate more extensive information.

2.2.4 Multilingual Support

Multilingual support refers to an LLM's ability to understand, generate, and translate text across multiple languages. This capability is essential for ensuring accessibility, inclusivity, and effective communication in linguistically diverse environments. Many state-of-the-art LLMs are multilingual by design, having been trained in large and diverse corpora that include dozens or even hundreds of languages [4].

One of the key challenges in evaluating multilingual support is the lack of standardized metrics among providers. Although many LLMs claim multilingual capabilities, their performance often varies significantly depending on the target language, especially for low-resource or morphologically rich languages. In addition, the model documentation may rely on different benchmarks, making it difficult to compare results consistently.

In this work, we use the Massive Multitask Language Understanding (MMLU) [5] benchmark to evaluate multilingual accuracy. Specifically, we rely on *zero-shot performance scores* derived from multilingual adaptations of the MMLU dataset, which assess the model's ability to correctly answer domain-specific questions in various languages without prior examples. MMLU provides a balanced and challenging benchmark across multiple academic and professional domains, making it a suitable proxy for real-world multilingual reasoning accuracy.

Although MMLU was originally designed in English, recent evaluations apply high-quality translations across many languages to ensure fair cross-lingual comparisons. This makes MMLU a reliable metric for estimating multilingual performance under consistent conditions. However, businesses may still adapt this evaluation using complementary benchmarks such as XTREME [6] or FLORES-200 [7] depending on their linguistic priorities and use cases.

Table 1 presents zero-shot multilingual performance on the MMLU benchmark for three OpenAI models: GPT-4o, o1-mini, and GPT-4.5. As shown, multilingual accuracy varies across languages and models. For example, all three models achieve over 88% accuracy in English, but performance drops in lower-resource languages like Yoruba and Swahili. The o1-mini model shows the most consistent and robust multilingual performance.

Language	GPT-4o	o1-mini	GPT-4.5
Arabic	0.8311	0.8900	0.8598
Bengali	0.8014	0.8734	0.8477
Chinese (Simplified)	0.8418	0.8892	0.8695
English (not translated)	0.8887	0.9230	0.8960
French	0.8461	0.8932	0.8782
German	0.8363	0.8904	0.8532
Hindi	0.8191	0.8833	0.8583
Indonesian	0.8397	0.8861	0.8722
Italian	0.8448	0.8970	0.8777
Japanese	0.8349	0.8887	0.8693
Korean	0.8289	0.8824	0.8603
Portuguese (Brazil)	0.8360	0.8952	0.8798
Spanish	0.8430	0.8992	0.8840
Swahili	0.7786	0.8540	0.8199
Yoruba	0.6208	0.7538	0.6818

Table 1: MMLU Language (0-shot) performance across OpenAI models

These results underscore the importance of evaluating multilingual accuracy at the language level, rather than relying solely on aggregate model claims. Organizations should consider their operational languages and consult empirical multilingual benchmarks like MMLU to make informed decisions about LLM integration.

2.2.5 Speed: Tokens per Second

Another key performance variable is the model speed, measured in tokens per second (TPS). This metric represents how quickly a language model can generate or process tokens during inference. Higher TPS values indicate faster response times, which is especially critical for real-time or high-volume applications.

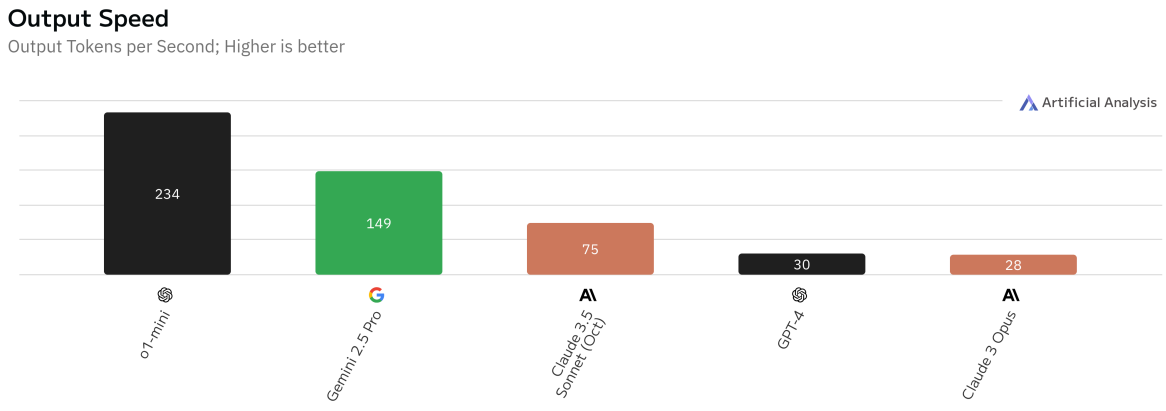

Figure 4: LLM speed measured in Tokens per Second (TPS)

Figure 4 illustrates the token generation speed of different LLMs. As can be observed, o1-mini shows 234 TPS, whereas other models show lower values, such as Claude 3 Opus, with a 28 TPS.

For businesses, the speed of the model directly affects both user experience and operational efficiency. In customer-facing applications, such as chatbots or virtual assistants, a fast response time is essential to maintain engagement and satisfaction. In back-end processes such as automated report generation, document summarization, or batch data analysis, faster models significantly reduce execution time and cost, especially when running at scale.

2.3 Alternatives

In this work, we consider a representative set of LLMs developed by the five most prominent organizations in the current AI landscape: OpenAI, Anthropic, Meta, Google DeepMind, and xAI. These models include both proprietary and open source solutions and have been selected on the basis of their industry relevance, technical performance, and suitability for business integration:

1. **GPT-4.1** (OpenAI)
2. **GPT o4-mini** (OpenAI)
3. **GPT o3** (OpenAI)
4. **Claude 3.5 Sonnet** (Anthropic)
5. **Claude 3 Opus** (Anthropic)
6. **Llama 3.1 405b** (Meta)
7. **Llama 3.1 70b** (Meta)
8. **Llama 3.1 8b** (Meta)
9. **Grok 2** (xAI)
10. **Grok 3** (xAI)
11. **Gemini 2.5 Pro** (Google DeepMind)
12. **Gemini 1.5 Pro** (Google DeepMind)
13. **Gemini 1.5 Flash** (Google DeepMind)

These models have been chosen to represent a wide spectrum of capabilities. Some prioritize reasoning accuracy and instruction-following behavior, while others emphasize multilingual support, extended context window sizes, high-speed inference, or cost-efficiency for large-scale deployment.

2.4 Type of Problem

The decision problem is a **ranking problem**. The objective is to rank the LLMs from best to worst according to the aggregated performance across the defined criteria. This ranking will support businesses in making informed and explainable AI technology adoption decisions.

2.5 Selected MCDA Method

For this project, the selected Multi-Criteria Decision Analysis (MCDA) method is **ELECTRE III**, implemented using Python. ELECTRE-III is well suited for *ranking problems* involving a mix of quantitative and qualitative criteria, which aligns with the complexity of evaluating Large Language Models (LLMs) for business applications [8].

This method is based on the concept of outranking, allowing for the modeling of *partial concordance and discordance* between alternatives. It also supports the definition of preference, indifference, and veto thresholds, which helps in dealing with imprecise or uncertain data—an important factor when comparing models with incomplete or approximate benchmark results. ELECTRE-III's robustness in handling conflicting criteria and its capacity to represent nuanced trade-offs make it particularly appropriate for our context, where no single model dominates across all performance dimensions.

2.5.1 Method Evaluation

To validate the suitability of the selected MCDA method, we used the *MCDA Method Selection Software*[3], an interactive tool that recommends MCDA methods based on a structured questionnaire. This questionnaire is divided into four sections:

1. **Problem Typology**: Here you can define how the problem is framed by (i) choosing the type of decision-making challenge under consideration and (ii) describing the criteria used to assess the alternatives.

    - What type of decision recommendation is requested? **Ranking**
    - What order of alternatives is requested? **Complete**
    - What scale leading the recommendation is requested? **Ordinal**
    - What is the nature of the problem in relation to the alternatives that constitute the set? **Stable**
    - What is the structure of the criteria used for the assessment? **Flat**
    - What is the type of performance of the criteria? **Deterministic**
    - What is the type of exact performances? **Per alternative**

[3] https://mcda.cs.put.poznan.pl/index.php

- What is the knowledge of the preference for the values of each criterion? **Known**
- What is the type of the known order of preference for the criteria? **Monotonic**
- What is the completeness status of the criteria set? **Complete**

2. **Preference Model**: Here you can define what type of model you would like to apply, accounting for (i) how the input data is used by the method, (ii) comparison of criteria performances, (iii) compensation between the criteria performances, (iv) aggregation of the criteria evaluations, and (v) the capacity of the MCDA methods to deal with inconsistent preference information.

   - How should the input information/performance data be used by the method(s)? **Quantitatively**
   - Should weights be used to differentiate the role of criteria in the aggregation procedure? **Yes**
   - Should interactions between criteria be considered to reflect a non-additive nature of preferences? **No**
   - Should criteria profiles not corresponding to the considered alternatives be used to derive a decision recommendation? **No**
   - Should the performances on multiple criteria be aggregated by the method(s) to provide the decision recommendation? **Yes**
   - Should the MCDA method(s) be capable to handle inconsistent preference information? **No**

3. **Elicitation of Preferences**: Here you can define what type of preferences information you can provide, how and with what frequency.

   - With what frequency would you like to provide the preference information? **One Time**
   - Would you like to include a level of confidence when providing the preferences **No**

4. **Exploitation of the Preference Relation Induced by the Preference Model**: Here you can decide how the preference relation induced by the preference model can be exploited to derive or enhance the decision recommendation.

   - What type of exploitation of the preference relation induced by the preference model would you like to be performed? **Univocal Relation**

Based on the responses provided for our decision problem, the tool recommended the following top five MCDA methods: (1) **ELECTRE III**, (2) ELECTRE III-H, (3) EVAMIX, (4) IDRA and (5) MAPPACC. As ELECTRE-III appears as the top-ranked method and satisfies all the requirements of our problem, it can be proved that it is the most appropriate technique for this project.

3 Problem Implementation

This section presents the implementation of the ELECTRE III method to evaluate and rank various LLMs based on multiple business-relevant criteria. Although this report focuses on explaining the analytical insights and results, the full implementation, including Python code in a notebook file, is publicly available at the following repository: https://github.com/Danie1Arias/LLM-BusinessApplications

3.1 Data Preparation

The evaluation is based on a performance table containing 13 LLM alternatives from leading AI companies such as OpenAI, Google DeepMind, Meta, Anthropic, and xAI. Each alternative is assessed across five quantitative criteria relevant to enterprise AI adoption:

- **Reasoning**: Performance on the GPQA benchmark (0–100 scale)
- **Cost**: Combined input-output API cost in USD per 1M tokens (USD $/1M Tokens)
- **Context Window**: Maximum token length supported (Tokens)
- **Multilingual Support**: Normalized multilingual benchmark score. In this case, English language as base line. (0–100 scale)
- **Speed**: Maximum generation speed in tokens per minute (TPM)

The table below summarizes the input data used for the ELECTRE III model:

Model	Reasoning	Cost	Context	Multilingual	Speed
GPT-4.1	66.3	10.0	1,047,576	89.6	30,000
GPT o4-mini	77.6	5.5	200,000	80.0	100,000
GPT o3	82.8	50.0	200,000	91.7	30,000
Claude 3.5 Sonnet	59.4	18.0	200,000	92.0	3,052
Claude 3 Opus	50.4	90.0	200,000	84.9	1,680
LLaMA 3.1 405b	50.7	3.25	128,000	91.6	2,215
LLaMA 3.1 70b	41.7	0.84	128,000	86.9	2,754
LLaMA 3.1 8b	30.4	0.10	128,000	68.9	12,546
Grok 2	56.0	12.0	8,192	86.2	1,200
Grok 3	75.4	18.0	131,072	91.2	4,320
Gemini 2.5 Pro	84.0	17.5	2,000,000	74.4	1,000,000
Gemini 1.5 Pro	46.2	12.5	2,000,000	75.3	4,000,000
Gemini 1.5 Flash	39.5	0.75	1,000,000	74.1	4,000,000

Table 2: Performance Table of LLM Alternatives

Relevant information for criteria such as Cost, Context Window Size, and Speed has been sourced primarily from publicly available tools including the Artificial Analysis Model Index and the LLM API Pricing Calculator. To ensure accuracy and consistency, these values were cross-validated with official model documentation and technical whitepapers published by the respective providers.

Among the selected criteria, Multilingual Support proved to be the most challenging to quantify. Not all providers offer transparent or standardized multilingual performance metrics, and the available benchmarks often differ in scope and methodology. Although some evaluations report *zero-shot performance* on multilingual tasks, others use custom evaluation suites or average cross-lingual metrics. This lack of consistency introduces variability in the interpretation of multilingual capabilities in models.

3.2 Case 1: Chatbot Application

In this first case, the CEO of a company is seeking to integrate a suitable LLM into their customer support chatbot to reduce the workload of the service team and improve responsiveness on the company website. More details in the code implementation[4].

3.2.1 Parameter Settings

Based on the performance characteristics shown in Table 2, the CEO defines the following criteria, weights, and ELECTRE III thresholds:

Criterion	Weight	Indifference (q)	Preference (p)	Veto (v)
Reasoning	0.05	5	10	–
Cost	**0.25**	2	5	7
Context Window	0.05	200,000	500,000	–
Multilingual	**0.3**	3	6	10
Speed	**0.35**	1,500	10,000	–

Table 3: Criteria weights and thresholds for chatbot application scenario.

The reasoning behind these choices is as follows:

- **Reasoning**: Although LLM needs to respond accurately, it will primarily rely on structured knowledge provided through prompts, making this a lower priority.
- **Cost**: This is a highly relevant criterion due to the high volume of customer interactions. Token pricing can significantly affect scalability and operational cost.
- **Context Window**: Since chatbot conversations tend to be short and focused, a large context window is not essential.

[4] https://github.com/Danie1Arias/LLM-BusinessApplications/blob/main/notebook_case1.ipynb

- **Multilingual Support**: This is crucial for an international company that serves a diverse user base. The model must handle queries in multiple languages effectively.
- **Speed**: The speed of the model is critical to ensure a seamless user experience (UX). Delays in response time can negatively affect customer satisfaction.

3.2.2 Result Analysis

Figure 6 illustrates the LLMs ranking obtained using the ELECTRE III method:

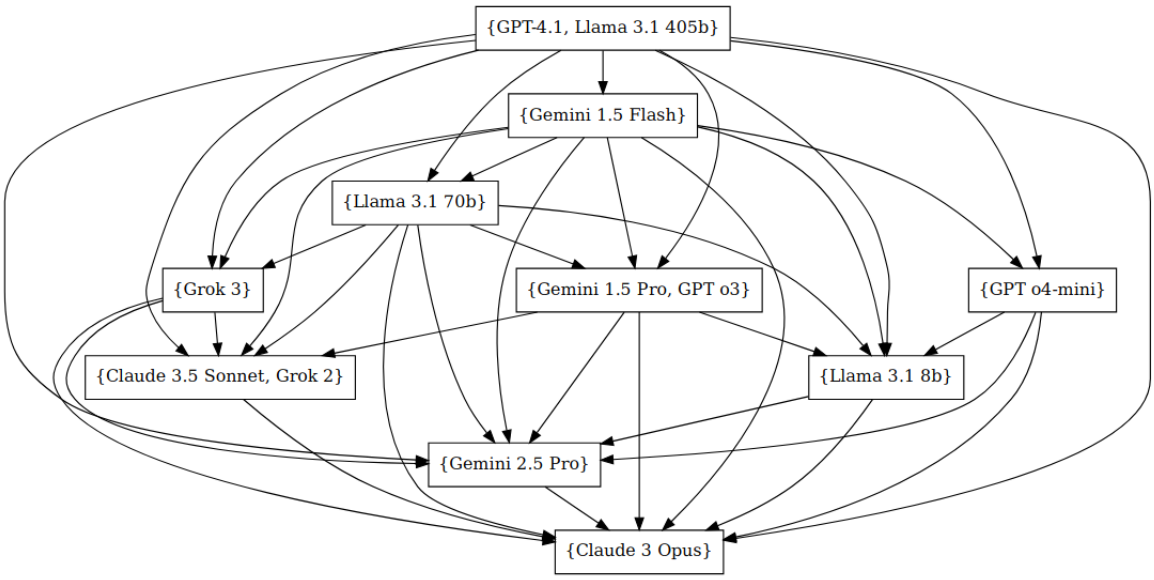

Figure 5: ELECTRE III Ranking for the first case (Chatbot integration).

We observe that **GPT-4.1** and **LLaMA 3.1 405b** occupy the top tier. Both models emerge as the most suitable candidates for chatbot deployment, primarily due to their strong performance in cost, multilingual capabilities, and speed. However, their performance profiles differ in certain aspects. For instance, GPT-4.1 offers significantly higher speed (30,000 TPM) compared to LLaMA 3.1 405b (2,215 TPM), but it also incurs a higher cost at 10 $/MToken versus 3.25 $/MToken for LLaMA. On the multilingual criterion, both models perform similarly, achieving values close to 90.

In the second tier, we find **Gemini 1.5 Flash**, which delivers strong performance in both speed and cost. However, it lags behind the top-tier models in multilingual support, with an accuracy score of 74.1. Closely following is **LLaMA 3.1 70b**, ranked fourth overall. This model stands out due to its exceptionally low cost (0.84 $/MToken) and solid multilingual performance (86.9), although its speed (2,754 TPM) is relatively modest.

The third tier includes **Grok 3**, **Gemini 1.5 Pro**, **GPT o3**, and **GPT o4-mini**. These models demonstrate strong performance in certain criteria but are penalized for weaknesses in others. For example, Grok 3 and GPT o3 both show excellent multilingual

scores (91.2 and 91.7, respectively), but their costs are considerably high at 18 $/MToken and 50 $/MToken, respectively.

In the fourth tier, we find **Claude 3.5 Sonnet**, **Grok 2**, and **LLaMA 3.1 8b**. As with the previous group, these models exhibit mixed results. Grok 2, for instance, performs well in multilingual support (86.2), but its high cost (12 $/MToken) and particularly low speed (1,200 TPM) limit its suitability for chatbot integration.

Finally, the bottom two tiers are occupied by **Gemini 2.5 Pro** and **Claude 3 Opus**, respectively. These models diverge significantly from the expected criteria for this use case. Claude 3 Opus, for example, has the highest cost among all alternatives (90 $/MToken), while Gemini 2.5 Pro shows acceptable results in some areas but underperforms in the most relevant dimensions, such as speed and multilingual support, which are critical for a chatbot-focused application.

3.3 Case 2: Business Analytics Application

In this case, the company intends to develop an internal system that provides strategic business recommendations to the management team based on available data. The development team must select the most appropriate LLM to support this decision-support tool. Further technical details can be found in the code implementation[5].

3.3.1 Parameter Settings

Based on the performance metrics shown in Table 2, the decision-maker has defined the following criteria, weights, and ELECTRE III thresholds:

Criterion	Weight	Indifference (q)	Preference (p)	Veto (v)
Reasoning	**0.50**	2	5	7
Cost	0.05	2	5	–
Context Window	**0.30**	200,000	500,000	750,000
Multilingual	0.05	3	6	–
Speed	0.10	1,500	10,000	–

Table 4: Criteria weights and thresholds for the business analytics scenario.

The rationale behind these parameter settings is outlined below:

- **Reasoning**: This is the most critical criterion, as the model must provide accurate, logical, and insightful suggestions to guide high-level business decisions.
- **Cost**: Cost is a relatively minor concern. The company is prepared to invest in a premium model if it ensures high-quality strategic recommendations.

[5] https://github.com/Danie1Arias/LLM-BusinessApplications/blob/main/notebook_case2.ipynb

- **Context Window**: This criterion holds significant weight, since the model will need to insulate and reason over extensive internal documents, past performance data, and strategic plans.
- **Multilingual Support**: As this is an internal system used exclusively in English, multilingual capabilities are of low importance.
- **Speed**: Response time is not a major concern, as the tool will be used occasionally in non-real-time settings. Therefore, user experience (UX) is not a primary driver in this context.

3.3.2 Result Analysis

Figure 6 illustrates the LLMs ranking obtained using the ELECTRE III method:

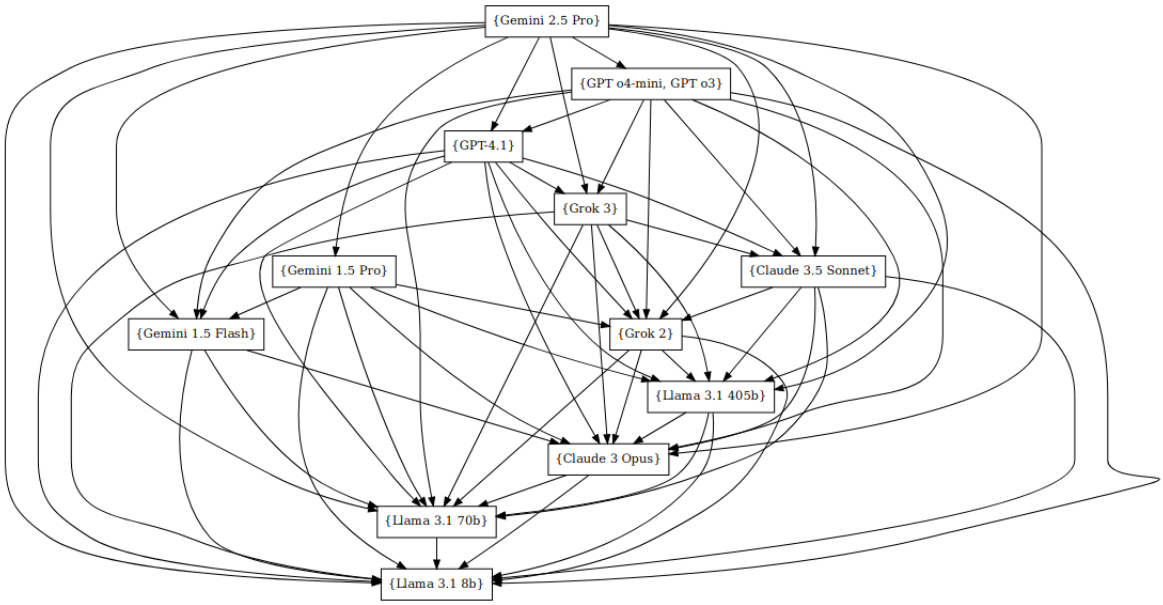

Figure 6: ELECTRE III Ranking for the second case (Business Analytics Application).

As shown in the ranking, **Gemini 2.5 Pro** emerges as the top-performing LLM. This model demonstrates outstanding results in the most critical criteria for the business analytics context. Specifically, it achieves the highest reasoning score among all evaluated models (84.0) and supports a maximum context window of 2,000,000 tokens, both essential for strategic decision-making and long-document processing.

In the second tier, we find **GPT o4-mini** and **GPT o3**. These models offer solid reasoning capabilities (77.6 and 82.8, respectively) but are limited to a smaller context window of 200,000 tokens. Despite this limitation, their strong reasoning performance keeps them competitive.

Just below, **GPT-4.1** is positioned in the third tier. While it benefits from a substantially larger context window (1,047,576 tokens), it scores lower in reasoning (66.3),

which slightly reduces its overall suitability for this particular use case.

The remaining models are distributed across lower tiers. In general, they exhibit weaker reasoning capabilities (the most heavily weighted criterion in this scenario) and offer only modest context lengths, typically ranging between 100,000 and 200,000 tokens. For example, while **Gemini 1.5 Pro** does provide a very large context window (2,000,000 tokens), its reasoning score of 46.2 significantly underperforms compared to higher-ranked models, limiting its overall appeal.

At the bottom of the ranking, we find **LLaMA 3.1 8b**. This model shows the lowest reasoning performance (30.4) and a limited context window (128,000 tokens), making it unsuitable for tasks requiring analytical depth and large memory. Although it offers high speed (12,546 TPM) and the lowest cost among all models evaluated (0.1 $/MToken), these advantages are not prioritized in this use case and therefore do not compensate for its deficiencies in the most relevant dimensions.

3.4 Comparative Analysis and Conclusion

As observed in the first case related to chatbot integration, the most important criteria were cost, speed, and multilingual capabilities. The ELECTRE III method identified **GPT-4.1** and **LLaMA 3.1 405b** as the most suitable models for this scenario, as they performed strongly in the aforementioned dimensions. In contrast, **Claude 3 Opus** ranked lowest due to its significantly higher cost, which negatively impacted its suitability for a high-volume, customer-facing application.

In the second case, the company aimed to develop an internal business analytics tool. This scenario had fundamentally different priorities, with reasoning capabilities and context window length being the dominant criteria. Here, **Gemini 2.5 Pro** was identified as the best alternative, as it demonstrated top-tier performance in both critical areas. This application, being less sensitive to cost or latency, favored models with advanced analytical potential and deep contextual understanding.

Overall, the ELECTRE III method proved to be highly effective for multi-criteria evaluation of LLMs. It not only identified the most appropriate models for each use case, but also produced a meaningful and interpretable ranking of all alternatives, from the most to least suitable.

One of the main challenges encountered during the project was the definition of suitable **indifference**, **preference**, and **veto thresholds**. Since LLMs can differ significantly across dimensions (e.g., cost, reasoning, speed), careful calibration was required to ensure that the threshold values led to realistic and actionable outcomes. This involved a preliminary sensitivity analysis to align the thresholds with real-world expectations and business priorities.

References

[1] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman, "Gpqa: A graduate-level google-proof qa benchmark," 2023. [Online]. Available: https://arxiv.org/abs/2311.12022

[2] I. Documentation, "Tokens and tokenization," 2024, accessed: 2025-05-13. [Online]. Available: https://www.ibm.com/docs/en/watsonx/saas?topic=solutions-tokens

[3] D. Bergman, "What is a context window?" 2024, accessed: 2025-05-13. [Online]. Available: https://www.ibm.com/think/topics/context-window

[4] L. Qin, Q. Chen, Y. Zhou, Z. Chen, Y. Li, L. Liao, M. Li, W. Che, and P. S. Yu, "A survey of multilingual large language models," *Patterns*, vol. 6, no. 1, p. 101118, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389924002903

[5] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," 2021. [Online]. Available: https://arxiv.org/abs/2009.03300

[6] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, "Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization," 2020. [Online]. Available: https://arxiv.org/abs/2003.11080

[7] NLLB Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang, "Scaling neural machine translation to 200 languages," *Nature*, vol. 630, no. 8018, pp. 841–846, 2024. [Online]. Available: https://doi.org/10.1038/s41586-024-07335-x

[8] J. R. Figueira, V. Mousseau, and B. Roy, *ELECTRE Methods*. New York, NY: Springer New York, 2016, pp. 155–185. [Online]. Available: https://doi.org/10.1007/978-1-4939-3094-4_5