Using Large Language Models for Creating, Modifying and Interpreting Decision Tables in Multi-Criteria Method DEX

Marko Bohanec
Jožef Stefan Institute
Department of Knowledge Technologies
Jamova cesta 39, SI-1000 Ljubljana, Slovenia
marko.bohanec@ijs.si

***Abstract**. The potential of large language models (LLMs) was examined in the context of decision tables as used in the qualitative multi-criteria decision modelling method DEX (Decision EXpert). Interactive dialogues were conducted with two open-source LLM chatbots, DeepSeek and Llama, with their outputs evaluated from the perspective of an expert decision analyst. The interaction focused on the construction and modification of a single decision table, as well as its interpretation in terms of table properties, decision rules, attribute weights, and visualizations. Findings suggest that LLMs offer a convenient means for learning and executing simple tasks. However, at their current stage of development, they remain inconsistent and prone to errors, rendering them unsuitable for serious applications.*

Keywords. Large language models, multi-criteria decision making, method DEX (Decision EXpert), decision tables, decision rules

1 Introduction

Generative artificial intelligence (AI), particularly large language models (LLMs) (Ozdemir, 2023; Atkinson-Abutridy, 2024), has emerged as a transformative technology across various areas, revolutionizing tasks such as content creation, problem-solving, and decision-making. LLMs, trained on vast datasets, exhibit unparalleled capabilities in generating human-like text, answering complex queries, and even creating novel content.

Multi-Criteria Decision Making (MCDM) (Ishizaka & Nemery, 2013; Kulkarni, 2022) is a discipline focused on supporting decision-makers who face problems involving multiple, often conflicting criteria. It aims to identify the best or most preferred alternative(s) by explicitly considering these criteria and decision-maker's preferences. MCDM methods are widely applied across fields such as engineering, finance, and public policy, where decisions must balance trade-offs among diverse objectives. Among many MCDM approaches, DEX (Decision EXpert) (Bohanec, 2022) stands out as a qualitative, rule-based method that uses decision tables to represent and aggregate criteria hierarchically, facilitating transparent and interpretable decision models. Decision tables in DEX encode decision-maker's or expert's knowledge using if-then rules that map combinations of attribute values to evaluation outcomes.

This study explores the potential of LLMs for the creation, modification, and interpretation of DEX decision tables. The research was motivated by an event that surprised the author of this paper. In May 2025, he experimented with the DeepSeek chatbot (https://www.deepseek.com/) to test its file-upload capabilities. He uploaded a decision table (Table 1), formatted in a simple text-based layout, to see if the system would accept the data. After typing the prompt "*Uploaded is a DEX decision table"*, he accidentally pressed *Enter* before providing further instructions. To his surprise, the chatbot autonomously processed the file, correctly interpreting its structure and generating a concise set of decision rules (Response 1).

This one-time experience clearly indicated that LLMs are capable of representing and interpreting decision tables. This raised the follow-up research questions: what else can contemporary LLMs offer to a decision analyst working with DEX decision tables? To what extent and with what level of quality can they assist in the three core tasks: creation, modification, and interpretation of such tables? What functionality is provided out of the box, what requires user-supplied methodological guidance, and what types of errors or distractions may arise during LLM use?

The research method was deliberately kept light-weight and simple: we conducted several sessions with open-source chatbots running the models DeepSeek-r1:14b (14.8 billion parameters) and Llama3:3 (70.6 billion), both installed locally at the author's organization. We asked the same questions in the same order, except when responses of one or both LLMs

were unsatisfactory and required additional instructions. We conducted several sessions using decision tables of various sizes: two to three input attributes, up to 24 elementary rules. We observed the responses and assessed them from the viewpoint of an expert decision analyst.

Table 1. A DEX decision table in text format.

Injury History	Health/Nutrition Habits	Health & Injury Profile
frequent	poor	very poor
frequent	average	poor
frequent	good	poor
frequent	excellent	average
occasional	poor	poor
occasional	average	average
occasional	good	good
occasional	excellent	good
rare	poor	average
rare	average	good
rare	good	good
rare	excellent	excellent
none	poor	average
none	average	good
none	good	excellent
none	excellent	excellent

Response 1. DeepSeek's interpretation of Table 1.

Analysis of Health & Injury Profile Decision Table
This is a decision table that maps combinations of "Injury History" and "Health/Nutrition Habits" to a "Health & Injury Profile" assessment.
Key Observations:
1. **Injury History Impact**: The frequency of injuries has a clear correlation with the health profile:
   - Frequent injuries consistently lead to poorer health profiles
   - No injury history generally leads to better assessments
2. **Nutrition Habits Influence**: Better nutrition habits can mitigate some effects of injury history:
   - Even with frequent injuries, excellent nutrition can result in an "average" profile
   - Poor nutrition with no injury history only reaches "average"
3. **Combined Effects**:
   - The best profile ("excellent") requires either:
     - No injury history + good/excellent nutrition, OR
     - Rare injuries + excellent nutrition
   - The worst profile ("very poor") only occurs with frequent injuries + poor nutrition

Decision Rules Summary:
- **Very Poor**: Only when injury history is frequent AND nutrition is poor
- **Poor**: Frequent injuries with average/good nutrition OR occasional injuries with poor nutrition
- **Average**: Several combinations including frequent+excellent, occasional+average, rare+poor, or none+poor
- **Good**: Occasional+good/excellent or rare+average/good
- **Excellent**: Rare+excellent or none+good/excellent

In the next section, we briefly review the related work. In sections 3–5 we explore the performance of LLMs for the creation, modification and interpretation of decision tables, respectively. Basic concepts of DEX decision tables related to each task are explained along the way. To keep the text reasonably short and consistent, we illustrate the process using a small decision table with two input attributes and 12 elementary rules. Section 6 concludes the paper.

2 Related Work

While LLMs are receiving a lot of attention in recent scientific literature, their coverage in the contexts of MCDM and/or decision tables is still scarce. Several authors have suggested incorporating LLMs in the MCDM process. Wang et al. (2025) proposed a framework using LLMs in the model preparation and evaluation stages, comparing it with the MCDM methods AHP (Analytic Hierarchy Process) and FCE (Fuzzy Comprehensive Evaluation). Similarly, Svoboda & Lande (2024) proposed a decision analysis framework for cybersecurity that combines AHP with the GPT-4 LLM. The same LLM is used in software 1000minds (https://www.1000minds.com/), which incorporates an AI assistant to enhance user interaction by suggesting decision criteria and alternatives according to the MCDM method PAPRIKA (Potentially All Pairwise RanKings of all possible Alternatives). Radovanović et al. (2024) used LLMs to learn the structure and some components of DEX models from data.

While decision tables are an important component of information systems, they are still seldom addressed in connection with LLMs. In their study, Lu et al. (2024) provide a comprehensive overview of table-related tasks, covering tasks like table question-answering, spreadsheet manipulation and table data analysis, using both traditional and LLM-supported techniques. Goossens et al. (2023) experimentally evaluated an automated approach to generating decision tables from natural language, using GPT-3. Their findings indicate that GPT-3 can understand the decision context, identify the input and output variables, and provide template decision tables for problem-solving, but in general it cannot create complete and correct decision tables. The latter finding is important because it relates to completeness and consistency, two essential characteristics of DEX decision tables, as explained in the next section.

3 Creating DEX Decision Tables

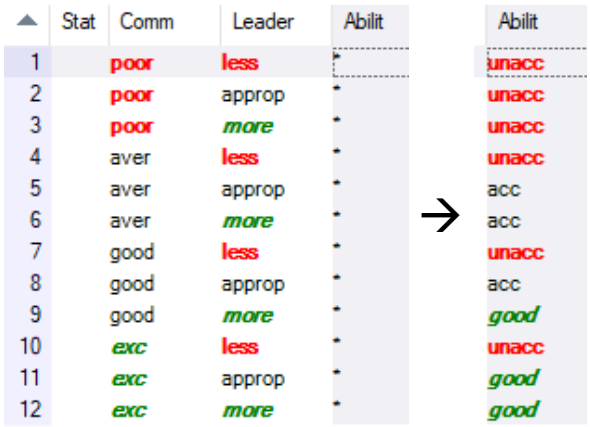
A DEX decision table is always bound to the context of one or more input attributes and one output attribute (outcome). *Attributes* are qualitative variables that can take their values from a discrete set of values, which are usually represented by words. Usually, value sets (called *scales*) are small (two to five values) and preferentially ordered from "bad" to "good" values. For example, Table 1 consists of two input attributes: (1) *Injury History*, having the scale <frequent, occasional, rare, none>, and (2) *Health/Nutrition Habits*: <poor, average, good, excellent>. The output attribute is *Health & Injury Profile*, using the scale <very poor, poor, average, good, excellent>. Each row in the table is called an *elementary decision rule* and maps some combination of input values to some outcome value. Generally, a DEX decision table is expected to have two important properties:

- *Completeness*: Contains all possible combinations of input attributes' values, without duplicates.

- *Preferential consistency*: When comparing two rules and the second one has all input values better than or equal to the first one, then the second outcome should be better or equal, too. We say that the second rule *dominates* the first. If this holds for all pairs of comparable rules, the table is *monotone* and is said to "obey the principle of dominance".

Traditionally, DEX decision tables are created interactively using software such as DEXiWin (Bohanec, 2024a; 2025). After defining input and output attributes together with their scales, DEXiWin constructs the decision table containing all possible combinations of input values, but with undefined outcomes (denoted by '*').

Fig. 1 shows an example decision table, which combines the input attributes *Comm* (communication skills of an employee candidate) and *Leader* (leadership skills) to *Ability* (aggregated assessment of the candidate's abilities). The example is extracted from the employee-selection model called *EmploySmallSuite*, which is distributed with DEXiWin. This example table is used hereafter.



	Stat	Comm	Leader	Abilit
1		poor	less	*
2		poor	approp	*
3		poor	more	*
4		aver	less	*
5		aver	approp	*
6		aver	more	*
7		good	less	*
8		good	approp	*
9		good	more	*
10		exc	less	*
11		exc	approp	*
12		exc	more	*

	Abilit
1	unacc
2	unacc
3	unacc
4	unacc
5	acc
6	acc
7	unacc
8	acc
9	good
10	unacc
11	good
12	good

Figure 1. Decision table creation in DEXiWin.

The right-hand side of Fig. 1 shows the *Abilit* column after it has been completely defined by the user. The process is usually quite dynamic, as it provides several techniques for defining outcomes (direct assignment, considering dominance of rules, considering weights of attributes, and considering attributes' symmetries), while continuously assessing the completeness and consistency of the table. A detailed description of those is beyond the scope of this paper; see Bohanec (2024a) for more information.

Can we construct (or reconstruct) such a decision table using LLMs? The dynamic DEXiWin-type interaction seems unfeasible in the text-oriented communication employed by chatbots. As an alternative, let us first define the bounding attributes and let the LLM to construct a decision table without further instructions:

P1: *Consider three qualitative attributes according to method DEX:*
(1) Comm (communication skills of an employee candidate), possible values: poor, average, good, excellent;
(2) Leader (leadership skills): less, appropriate, more;
(3) Ability (aggregated assessment of the candidate's abilities): unacceptable, acceptable, good.
Create a DEX decision table that maps Comm and Leader to the values of Ability. Call this table T0.

The two LLMs suggested different decision tables and presented them in the format similar to Fig. 1. The outcomes were not equal to Fig. 1, though. Importantly, however, both tables were complete (each contained 12 distinct input value combinations) and monotone, even if no instructions were given to this end. In order to save space in this paper and compactify responses, we formulated an additional request:

P2: *Display T0 using Comm as rows and Leader as columns.*

The response to P2, which was the same for both LLMs, is shown in Response 2. Notice that the required format would be unsuitable for decision tables with three or more input attributes.

Response 2. Decision table T0. Column headings correspond to the values of *Leader*, cells contain values of *Ability*.

Comm	Less	Appropriate	More
Poor	Unacceptable	Unacceptable	Acceptable
Average	Unacceptable	Acceptable	Acceptable
Good	Acceptable	Acceptable	Good
Excellent	Acceptable	Good	Good

While this result is satisfactory, we need to make a few remarks. In our previous experience with LLMs, we noticed that they tend to "forget" their suggestions from one question to another. This is why we explicitly requested to name the table T0. Such need of keeping track of results is annoying, distracts the user from the primary task and is easy to forget. Furthermore, the naming technique actually turned out ineffective with Llama, which proposed different table contents after P1 and P2. This is considered a serious flaw of LLMs, which seems to be improving with newer versions, but has still not been satisfactory resolved.

4 Modifying DEX Decision Tables

Even though the syntax of T0 is correct, we are not entirely satisfied with some of the rules. Thus, we requested the change:

P3: *Modify T0 so that whenever Comm=poor or Leader=less, Ability is unacceptable. Call the result TR.*

Both LLMs responded correctly (Response 3). In this way, we successfully reconstructed the decision table from Fig. 1.

Response 3. Decision table TR.

Comm	Less	Appropriate	More
Poor	Unacceptable	Unacceptable	Unacceptable
Average	Unacceptable	Acceptable	Acceptable
Good	Unacceptable	Acceptable	Good
Excellent	Unacceptable	Good	Good

While we successfully constructed TR in just two steps, this was typically not the case with larger decision tables (containing up to 24 rows). Usually, the initial decision tables suggested by LLMs (such as T0) were quite good and did make sense, however they only reflected "general" opinions, known to the LLM after consulting a vast amount of data. Human decisions are subjective, so decision tables should reflect decision maker's subjective preferences and should be formulated accordingly. For larger decision tables, this turned out a challenging task, which we tried to accomplish by formulating rules, similar to those in request P3. The process typically required several steps and was often distracted by LLM's forgetting previously formulated rules, which the user assumed they were already fixed. At the current level of LLM development, it seems that it is still more effective to create decision tables using specialized tools and importing them to LLMs, as illustrated in Table 1.

5 Interpreting DEX Decision Tables

Traditional tools for building decision tables, such as DEXiWin, guide the decision maker to focus on one elementary decision rule at a time. As the user defines individual outcomes, built-in algorithms monitor and ensure the overall consistency and completeness of the table. However, as the number of rules increases, particularly beyond 20, the resulting tables may become difficult to interpret as a whole. Therefore, more compact and readable representations are essential to support users in reviewing the content during model construction. Once complete, such representations also enhance the model's transparency and trustworthiness. To this end, DEXiWin offers multiple methods for interpreting decision tables in terms of aggregate (compact) decision rules, decision trees, attribute weights, and 3D visualizations (Bohanec, 2024a).

5.1 Properties of Decision Tables

To assess LLMs' abilities to interpret decision tables, let us continue with the TR table and first ask some questions about its properties.

P4: *Is TR complete?*

Both LLMs answered correctly that TR was complete and provided valid explanations referring to 12 unique non-empty combinations of input attributes' levels.

P5: *Is TR preferentially consistent?*

DeepSeek answered positively with a convincing explanation, summarizing that "*TR is preferentially consistent because the rankings and transitions between communication, leadership, and ability align logically without any contradictions*". Llama answered negatively, correctly quoting the definition of preferential consistency, but failing to detect that the whole table indeed adhered to the principle *"[when] one attribute improves (e.g., moving from "poor" to "average" Comm, or from "less" to "appropriate" Leader), the output (Ability) either stays the same or improves, but does not worsen.*"

P6: *Is TR symmetric?*

TR is clearly non-symmetric, because its input attributes use value scales of different cardinalities. Both LLM answered correctly and convincingly, justifying their decision:

- DeepSeek: "*TR lacks symmetry because the interactions between communication and leadership skills do not produce equivalent outcomes when the roles of these attributes are switched or varied independently*."
- Llama: *"[...] it's clear that Comm and Leader have distinct effects on Ability, and their influence is not symmetric.*"

5.2 Representations Using Decision Rules

The following requests were aimed to check the LLMs abilities to represent TR using different and possibly more compact representations.

P7: *Represent table TR using aggregate decision rules.*

When formulating this request, we did not define the concept of aggregate decision rules, assuming that the LLMs already knew it from DEX-related literature. This was not the case, and the answers reflected the LLMs own understanding of the concept. Nevertheless, the answers were logically correct. Response 4 shows the rules proposed by Llama. Notably, none of the LLMs identified the highly compact rule "Ability is unacceptable (unacc) if: Leader is less (regardless of Comm)." DEXiWin normally discovers such rules by allowing them to have mutually overlapping conditions.

Decision trees (Rokach & Maimon, 2014) are widely used in machine learning for representing decision logic. We asked:

P8: *Represent table TR using a decision tree.*

As expected, the LLMs had no issues with this representation. Among the two, DeepSeek's solution (Response 5) was considered slightly more readable.

Response 4. TR represented according to Llama's understanding of "aggregate rules".

- **Ability is unacceptable (unacc) if**:
  - Comm is poor (regardless of Leader),
  - or Comm is average and Leader is less,
  - or Comm is good/excellent and Leader is less.
- **Ability is acceptable (acc) if:**
  - Comm is average and Leader is appropriate or more,
  - or Comm is good and Leader is appropriate.
- **Ability is good if:**
  - Comm is good and Leader is more,
  - or Comm is excellent and Leader is appropriate or more.

Response 5. Decision tree representation of TR by DeepSeek.

- If Comm = poor: Directly conclude Ability = unacceptable (unacc).
- If Comm = average:
  - Next test: Leadership (Leader)
    - If Leader = less › Ability = unacc
    - If Leader = appropriate or more › Ability = acceptable (acc)
- If Comm = good:
  - Next test: Leader
    - If Leader = less › Ability = unacc
    - If Leader = appropriate › Ability = acc
    - If Leader = more › Ability = good
- If Comm = excellent:
  - Next test: Leader
    - If Leader = less › Ability = unacc
    - If Leader = appropriate or more › Ability = good

Response 6. "Ripple-down rules" representation of TR as proposed by Llama (incorrect).

Rule 1: If Comm is poor, then Ability is unacc.
Rule 2: If Comm is average and Leader is less, then Ability is unacc.
Rule 3: If Comm is average and Leader is appropriate or more, then Ability is acc.
Rule 4: If Comm is good and Leader is less, then Ability is unacc.
Rule 5: If Comm is good and Leader is appropriate, then Ability is acc.
Rule 6: If Comm is good and Leader is more, then Ability is good.
Rule 7: If Comm is excellent and Leader is less, then Ability is unacc.
Rule 8: If Comm is excellent and Leader is appropriate or more, then Ability is good.

Ripple-down rules (Compton & Jansen, 1990) are a somewhat less known representation of decision logic. After identifying the most common outcome (e.g., *Ability*=unacceptable) at the top level, the next levels consist of a hierarchy of context-dependent rules, where each new rule refines a parent rule by adding exceptions. Ripple-down rules are not implemented in DEXiWin and we wanted to know how are they handled by the LLMs:

P9: *Represent table TR using ripple-down rules.*

Both LLMs failed. Instead of making a general-to-detailed hierarchy of conditions, they both suggested *"[...] a set of rules that are applied in a specific order*", as shown in Response 6.

Observing the results of P7–9, we noticed that the LLMs did not take advantage of the fact that all attribute scales were preferentially ordered. To see if this affects LLMs in any way, we formulated the sequence of follow-up requests:

P10: *Consider that all attributes in table TR are preferentially ordered. This means that you can use the relations '>=' (better than or equal) and '<=' (worse than or equal) to compare values on these scales. Using these relations on input attributes Comm and Leader, represent table TR in terms of aggregate decision rules.*

P11: *Using the relations, represent table TR in terms of a decision tree.*

P12: *Using the relations, represent table TR in terms of ripple-down rules.*

Unfortunately, this made more harm than good. While the new interpretations indeed used the suggested relations, making more compact representations in some cases, many of them contained subtle and hard-to-detect errors. For instance, Llama's aggregate rules (Response 7) and ripple-down rules (Response 8) contain logical errors, which can be identified by following the respective logic and comparing the results with TR presented in Response 3. For instance, the rules for *Comm*=good and *Leader*=appropriate should result in *Ability*=acceptable, but do not. In contrast, DeepSeek's solutions remained largely unchanged: the aggregate and ripple-down rules were logically sound, but the decision tree grew more complex and introduced a minor logical flaw (not shown here).

Response 7. Representation of TR using aggregate rules with relational operators (Llama, incorrect).

1. If Comm <= average and Leader is any value, then Ability is unacc.
2. If Comm >= good and Leader >= appropriate, then Ability is good.
3. If Comm >= average and Leader >= appropriate, but not (Comm >= good and Leader >= appropriate), then Ability is acc.
4. In all other cases, Ability is unacc.

Response 8. Representation of TR using ripple down rules with relational operators (Llama, incorrect).

1. Rule 1: If Comm <= poor, then Ability is unacc.
2. Rule 2: If Comm <= average and Leader < appropriate, then Ability is unacc.
3. Rule 3: If Comm >= average and Leader >= appropriate, but Comm < good, then Ability is acc.
4. Rule 4: If Comm >= good and Leader >= appropriate, then Ability is good.

5.3 Representations Using Weights

In the next set of questions, we wanted to test LLMs' understanding of attribute weights (importances). This

is an overarching concept in MCDM (Kulkarni, 2022), where weights represent the relative importance assigned to each attribute and are fundamental in aggregating criteria to evaluate and rank alternatives. In DEX, weights are derived from decision rules, employing the least square regression.

We first checked the assessment of weights using information gain, a method commonly used in machine learning (Rokach & Maimon, 2014) from tabular data.

P13: *Assess the weights (importances) of input attributes in table TR using information gain. Explain the method and carry out the calculation. At the end, normalize the weights so that their sum equals 100.*

Both LLMs correctly explained the method and its main steps, but executed it inappropriately, giving incorrect results. The DeepSeek's explanation was insufficient to determine the cause, while Llama apparently made a mistake determining the table's size, using the number 16 instead of 12.

P14: *Now use ordinal attribute values. Assess attribute weights (importances) in table TR employing least squares regression. Explain the method and carry out the calculation. At the end, normalize the weights so that their sum equals 100.*

Again, both LLMs correctly explained the least squares method, which consists of approximating TR with

$$Ability = \beta_0 + \beta_{Comm} Comm + \beta_{Leader} Leader + \varepsilon$$

where β_0 is the intercept, β_{Comm} and β_{Leader} are the coefficients representing the effect of *Comm* and *Leader* on *Ability*, and ε is the error term. Both LLMs incorrectly assumed that $\beta_0 = 0$, and suggested to "*use a least squares regression calculator*" for the rest. Afterwards, they assumed their own values for β_{Comm} and β_{Leader} (not explaining how) and then carried out the normalization of weights correctly. Consequently, weights, calculated by the LLMs, were different from each other, and wrong: 56.67% and 43.33% by DeepSeek, and 55.56% and 44.44% by Llama. The correct answer, according to DEXiWin, is 50.98% and 49.02%.

All these results indicate that LLMs are indeed weak in mathematical computation and logical inference: they tend to "hallucinate" (Banerjee, 2024) and make up their answers. We knew this before and it was expected. However, the real problem is that LLMs' explanations of applied methods, calculations and inference procedures are exceptionally convincing to the level that can easily mislead the user. Errors are subtle and hard to detect, requesting the users to know and understand the applied methods really well, to meticulously check each and every response, and possibly even use external tools to find out the correct outcomes.

5.4 Visual Representations

In the last group of questions, we tested the LLMs' abilities to visualize data tables. Both DeepSeek and Llama do this by producing Python code, which, when run externally, draws the requested charts. Our first attempts were not really satisfactory, but after adding additional requests of how charts should look like, the results were becoming better and better. The following request resulted in computer code that, after some minor editing, gave a nice 3D chart of TR (Fig. 2), similar to those produced by DEXiWin.

P15: *Display table TR in a 3D chart, or provide code in one file. Use different colors for different levels of Ability. Use red and green color, respectively, for bad and good values of Ability. Connect displayed data points with dashed lines in each attribute direction.*

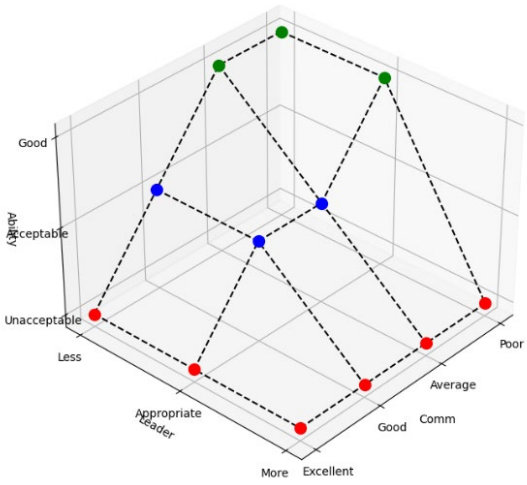


Figure 2. 3D chart of TR (DeepSeek + code edit).

2D charts, after several trial-and-error attempts and minor code editing, were also satisfactory (two variations produced by Llama are shown in Fig. 3 and Fig. 4).

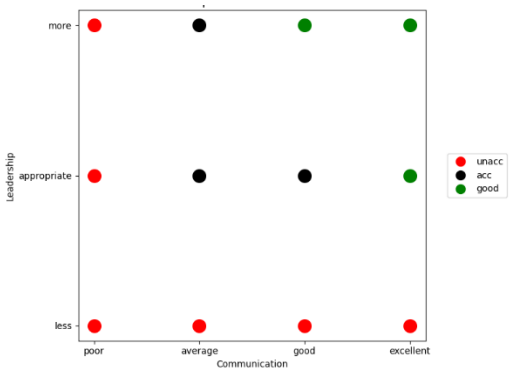


Figure 3. 2D chart of TR, using dots (Llama + code edit).

Figure 4. 2D chart of TR, filling rectangles (Llama + code edit).

6 Conclusion

We explored the abilities of large language models to operate with DEX decision tables that map two or more qualitative attributes to a qualitative outcome. The research was carried out by running two recent and popular open-source LLMs (DeepSeek and Llama) in parallel, asking them questions that addressed the most basic functionality necessary to handle an individual decision table: construction and modification, and interpretation in terms of table properties, decision rules, weights, and visualizations.

Specifically, we wanted to sense the operation of LLMs from the position of an ordinary user, who might want to apply LLMs on decision tables. However, the results themselves were assessed from the position of a skilled decision analyst.

Our findings yielded mixed results. On the one hand, LLM chatbots demonstrate an impressive ability to interpret user inputs and provide a substantial amount of knowledge "out of the box". LLM's responses are highly persuasive, and its engaging interface captures the user's attention, making the interaction both effective and enjoyable. Although the DEX method is not among the most widely known MCDM approaches, LLMs demonstrate a solid understanding of its core principles and key characteristics.

On the other hand, working with LLMs can often be uneven and, at times, frustrating. As highlighted in sections 5.2 (decision rules) and 5.3 (weights), the LLMs frequently provide convincing explanations that may conceal subtle errors that are difficult to detect and may mislead the user. The context (e.g., the current decision table), which the user may reasonably assume to be fixed, can change implicitly without notice or explanation (this behavior was particularly evident with Llama). Additionally, the programming code generated by LLMs occasionally contains errors that are nontrivial to diagnose and correct. After successfully resolving some issue, it may reappear several steps later. Consequently, rather than focusing on solving the problem at hand, the user needs to adopt the role of a meticulous supervisor, critically evaluating each step. This is both cognitively demanding and requires a high level of expertise in the applied methods, which is normally unnecessary while using specialized MCDM software.

In summary, at their current stage of development, LLMs are not yet suitable for reliably working with DEX decision tables. They are highly effective for learning the underlying concepts and performing basic tasks, such as summarizing decision tables (section 1) and analyzing their properties (section 5.1). LLMs may also prove useful for exploring less accessible algorithms, such as ripple-down rules (section 5.2), or generating customized visualizations (section 5.4). However, for more advanced operations, the limitations of LLMs remain too significant.

This study focused on working with individual decision tables. However, a typical DEX model comprises multiple decision tables, each associated with an internal node in a hierarchical attribute structure. Given the challenges we encountered in constructing and maintaining the consistency of even a single decision table within a session, it is reasonable to conclude that managing complete DEX models exceeds the capabilities of current LLM technology.

Work to date has been restricted to only two LLMs and small decision tables with only two or three input attributes, one output attribute, and at most 24 elementary decision rules. In practice, however, decision tables often contain 100 or more rules. Also, we did not investigate the ability of LLMs to restructure decision tables, for instance when adding or deleting an attribute or its value, or changing attribute weights; this typically requires quite advanced algorithms (Bohanec, 2024b). Moreover, the generalizability and replicability of our findings could have been enhanced through a more systematic experimental design, using APIs (Application Programming Interfaces). Future research may extend in these directions, particularly as LLMs continue to evolve. Given their rapid development, significant improvements in their applicability to decision table analysis can reasonably be expected.

Acknowledgments

The author acknowledges the financial support from the Slovenian Research and Innovation Agency for the programme Knowledge Technologies (research core funding No. P2-0103).

References

Atkinson-Abutridy, J. (2024). *Large Language Models: Concepts, Techniques and Applications.* Boca Raton: CRC Press. https://doi.org/10.1201/9781003517245.

Banerjee, S., Agarwal, A., & Singla, S. (2024). LLMs will always hallucinate, and we need to live with this. 10.48550/arXiv.2409.05746.

Bohanec, M. (2022). DEX (Decision EXpert): A qualitative hierarchical multi-criteria method. In: *Multiple Criteria Decision Making* (ed. Kulkarni, A.J.), Studies in Systems, Decision and Control 407, Singapore: Springer, doi: 10.1007/978-981-16-7414-3_3, 39–78.

Bohanec, M. (2024a). *DEXiWin: DEX Decision Modeling Software, User's Manual, Version 1.2*. Ljubljana: Institut Jožef Stefan, Delovno poročilo IJS DP-14747. Accessible from https://dex.ijs.si/dexisuite/dexiwin.html.

Bohanec, M. (2024b): *Transformations of decision tables in qualitative multi-criteria decision modeling method DEX*. Institut Jožef Stefan, Delovno poročilo IJS DP-14761, 2024. https://kt.ijs.si/MarkoBohanec/pub/2024_DP14761_Transformations.pdf

Bohanec, M. (2025): DEXi Suite: DEXi decision modelling software. *SoftwareX* 31, 102240. https://doi.org/10.1016/j.softx.2025.102240 .

Compton, P., & Jansen, R. (1990). A philosophical basis for knowledge acquisition. *Knowledge Acquisition* 2(3), 241–257. doi:10.1016/S1042-8143(05)80017-2.

Goossens, A., Vandevelde, S., Vanthienen, J., & Vannekens, J. (2023). GPT-3 for Decision Logic Modeling. *RuleML+RR'23: 17th International Rule Challenge and 7th Doctoral Consortium*, Oslo, Norway.

Ishizaka, A., & Nemery, P. (2013). *Multi-criteria decision analysis: Methods and software*. Chichester: Wiley. doi:10.1002/9781118644898.

Kulkarni, A.J. (Ed.) (2022). *Multiple Criteria Decision Making*. Studies in Systems, Decision and Control 407, Singapore: Springer, doi: 10.1007/978-981-16-7414-3_3.

Lu, W., Zhang, J., Fan, J., Fu, Z., Chen, Y., & Du, X. (2025). Large language model for table processing: A survey. *Frontiers of Computer Science*, *19*(2), 192350. https://doi.org/10.1007/s11704-024-40763-6.

Ozdemir, S. (2024*). Quick Start Guide to Large Language Models: Strategies and Best Practices for ChatGPT, Embeddings, Fine-Tuning, and Multimodal AI*. 2nd Edition. Addison-Wesley Professional. ISBN: 978-0135346563.

Radovanović, S., Delibašić, B., & Vukanović, S. (2024). Combining LLM and DIDEX method to predict Internal Migrations in Serbia. *Proc. 24th International Conference on Group Decision and Negotiation & 10th International Conference on Decision Support System Technology (GDN ICDSST 2024), Vol. 1: Technology as a support tool* (Eds. S.P. Duarte, P. Zaraté, A. Lobo, B. Delibašić, T. Wachowicz, M.C. Ferreira), University of Porto.

Rokach, L., & Maimon, O. (2014). *Data Mining with Decision Trees: Theory and Applications* (2nd. ed.). River Edge, USA: World Scientific Publishing Co., Inc. ISBN:978-981-4590-07-5.

Svoboda, I., & Lande, D. (2024). Enhancing multi-criteria decision analysis with AI: Integrating analytic hierarchy process and GPT-4 for automated decision support. https://arxiv.org/abs/2402.07404.

Wang, H., Zhang, F., Mu, C. (2025). One for all: A general framework of LLMs-based multi-criteria decision making on human expert level. arXiv e-prints, 2502.15778. https://arxiv.org/abs/2502.15778.