Building a Tool for Human-in-the-Loop Strategic Problem-Solving with LLMs

Ann Abeysekera*, Altin Azizi*, Bilgehan Emiral*
Technical University of Munich
Department of Informatics
Garching, Germany
annabey@kth.se, altin.azizi@tum.de, bilgehan.emiral@tum.de

Abstract

Logic trees can be used in various fields to solve real-world problems. This project is an attempt at enhancing the use of logic trees by combining them with Large Language Models (LLMs). In this project, we explore how these two components could potentially be combined by creating a working prototype and then evaluating its performance on a selection of 32 known real-world problems. Results show that the performance of the prototype is decent and that it potentially can be optimized further with additional user testing. In that regard, we have started to explore which areas this type of application can solve independently and what role the human interference plays in achieving effective usage with the product.

1 Introduction

This report is part of collaborative research between the Department of Natural Language Processing and the Department of Social Computing at the Technical University of Munich. The project aimed to develop an interactive, tree-based problem-solving tool powered by LLMs to enhance logical reasoning and decision-making. The overarching objective was to create an advanced prototype capable of simulating the structured thought process of strategy consultants when solving business cases. Given the inherent limitations of LLMs, it was not expected of the model to perform without flaws. Therefore, a key aspect of the project was to investigate to which extent the human intervention is required to guide and refine the model's output across different types of business cases.

Logic trees are a fundamental tool in strategy consulting, widely used for structuring problem-solving approaches and identifying solutions in real-world scenarios. Their versatility makes them useful for tasks such as pinpointing root causes (B. Garrette, 2018). There are different types of logic trees for different types of problems. *Quantity trees* decompose problems algebraically using mathematical operators. *Issue trees* break down "why" or "how" questions systematically. *Hypothesis trees* test yes/no statements to validate potential solutions.

Real-world business problems are often highly complex and ambiguous, leading to large tree structures (B. Garrette, 2018). This underscores the need for a tool that can assist in constructing such trees efficiently. While LLMs have the potential to generate logic trees, they require proper guidance to ensure accuracy and relevance. This project sought to bridge that gap by developing an interactive system that allows users to collaboratively build and refine logic trees with the support of LLMs. [1]

In the context of business case analysis, recent studies have advanced our understanding of LLM capabilities. Chain-of-thought prompting enables systematic breakdown of complex tasks (Wei et al., 2022a), while zero-shot reasoning capabilities enhance model performance without task-specific training (Kojima et al., 2022). Integrating these findings into logic tree structures ensures a more robust approach to problem-solving, especially when dealing with multi-layered issues commonly encountered in strategic decision-making. These methodological advances help bridge the gap between theoretical innovation and pragmatic solutions in business analysis. [1]

2 Related Work

Recent advancements in NLP and specifically LLMs have improved the ability of models to

[1] This paragraph was rephrased using generative AI.

perform complex reasoning tasks. One of the key approaches to enhance LLM performance is **Chain-of-Thought (CoT) reasoning**, which enables models to break down multi-step problems into intermediate steps (Wei et al., 2022b). By explicitly reasoning through a problem step-by-step, CoT allows additional computation to be allocated to problems requiring deeper reasoning, ultimately improving interpretability. This structured approach also provides insights into how an answer is derived, making it easier to debug errors in the reasoning process. CoT reasoning is applicable to a wide range of tasks that humans typically solve through language-based reasoning (Wei et al., 2022b).

Another relevant framework for structured reasoning is the **Pyramid Principle**, which is particularly useful for hypothesis trees. The Pyramid Principle structures arguments in a hierarchical manner, it starts with the key message at the top of the pyramid, followed by supporting arguments to this message, which is supported by data and facts at the bottom of the structure. This ensures that messages are quickly understood, both in written and verbal communication (Minto, 2009).

Another advancement for further enhancement in the reasoning process is **Probabilistic Tree-of-Thought Reasoning (Probtree)** (Cao et al., 2023). While LLMs using CoT reasoning can answer knowledge-intensive and complex questions, they still suffer from limitations such as incorrect or unnecessary retrieval and a lack of forward- or backward-looking reasoning (Cao et al., 2023). Probtree mitigates these issues by incorporating probabilistic reasoning, working from leaf nodes to the root while considering confidence levels for both answering and decomposing questions. LLMs assign higher confidence to answers at the leaf node level by leveraging Closed-Book QA (which relies on parametric knowledge) and Open-Book QA (which incorporates external retrieved knowledge). This reduces the impact of incorrect retrieval. Additionally, for non-leaf nodes, LLMs employ a broader scope of reasoning, integrating information from child nodes to enable global reasoning. This approach enhances the model's ability to recover from local errors and produce more reliable outputs (Cao et al., 2023). [1]

A widely recognized principle in structured problem-solving is the **Mutually Exclusive and Collectively Exhaustive (MECE) principle**, which is a desired principle for logic trees. The mutually exclusive aspect ensures that no elements in the tree overlap which avoids redundancy and the collectively exhaustive aspect ensures that all possible explanations or issues are covered, preventing gaps in the analysis (B. Garrette, 2018).

3 Methodology

This project aimed to develop an interactive, tree-based problem-solving tool that leverages LLMs to assist users in structuring and analyzing complex business cases. The core approach was to integrate LLMs into a human-in-the-loop system, allowing users to guide and refine the logic trees dynamically.

3.1 System Architecture

The application was designed as a web-based tool where users could build, modify, and analyze logic trees in real time. As can be viewed in Figure 1 the system architecture consisted of three main components:

1. **Logic Tree Management** - A **TreeManager** module was responsible for handling the creation, editing, and structuring of logic trees. Users could select different tree types, including issue trees, hypothesis trees, and quantity trees, depending on the nature of the problem.

2. **LLM Integration** – A dedicated **LLMService** enabled interaction with multiple LLMs, optimizing prompts to generate tree structures, suggest refinements, and retrieve relevant contextual knowledge..

3. **External Knowledge Retrieval** - An **ExternalKnowledgeService** allowed the system to fetch additional information from sources such as Google and Wikipedia to enrich decision-making.

3.2 Implementation Process

The implementation followed a structured pipeline to ensure the relevance and accuracy of the logic tree generation. The steps involved were:

1. **User Query Processing** - The system first receives a question from the user that outlines the problem to be analyzed.

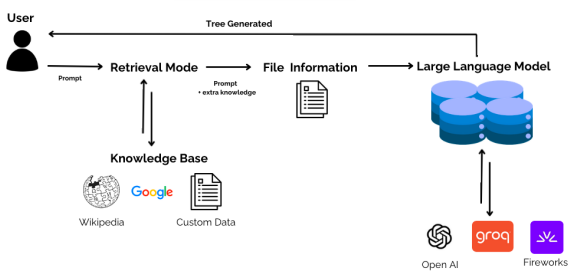


Figure 1: System Architecture

2. **Information Retrieval** - Based on the user's question, the system retrieves information from three potential sources:
    - Google search results
    - Wikipedia content
    - Custom user-uploaded files
3. **Content Filtering and Chunking** - Once the relevant pages are retrieved, the system assesses their length. If the content exceeds 10,000 tokens, it is chunked into smaller segments.
4. **Relevance Scoring** - The system ranks the chunks using a combination of sparse and embedding similarity search to determine the most relevant sections of the text.
5. **Prompt Construction** - The system constructs the LLM prompt by incorporating:
    - The user's original query
    - The most relevant information chunks
    - Few-shot learning examples of well-structured tree generations
6. **Tree Generation and Refinement** - The LLM generates an initial logic tree based on the prompt. Users can then interact with the tool to modify and refine the tree dynamically.

3.3 Frontend Technologies

The frontend architecture of the project is built upon the React library and the TypeScript programming language. The primary reason for selecting React is its component-based architecture, which allows the decomposition of complex user interfaces into modular parts. This approach facilitates the isolated management of UI components, which is particularly advantageous in applications involving intricate interactions, such as the Problem-Solving tool. TypeScript enhances this architecture by introducing type safety, enabling early error detection during development and ensuring code consistency.

For UI design, Tailwind CSS has been employed due to its utility-first approach, which facilitates rapid UI development. By leveraging predefined CSS classes for constructing customizable components, it eliminates the need for repetitive code writing. This approach ensures a consistent design language throughout the project and aligns seamlessly with component-based development.

For state management across the application, React's Context API has been adopted. Instead of more complex state management solutions such as Redux, Context API has been preferred due to its sufficiency for a medium-scale application and its requirement for less boilerplate code. This choice centralizes the management of complex data structures, such as tree-based data models, ensuring a consistent data flow across different components of the application. [1]

3.4 User Interface Functionalities

The user experience begins with components that allow users to clearly define the problem they wish to solve. This interface provides users with structured input fields and formatting options to articulate their problems in detail. Additionally, users can specify different perspectives and evaluation criteria when constructing problem definitions.

A key feature of the application is its ability to break down problems into sub-components and present them within a visual hierarchy. Users can view the structural representation of a problem through interactive nodes and connections, and expand or prune the tree. This visualization facilitates the comprehension of complex problem structures and provides users with a holistic perspective.

The core of user interaction is formed by features that allow the dynamic structuring of the problem tree, including adding and modifying nodes. Users can append sub-nodes to existing nodes, edit their contents, and delete them when necessary. The system ensures logical coherence among nodes through arrow-based visual connec-

tions. This feature aligns with the evolving nature of problem-solving processes.

The integration of a LLM for AI-driven suggestions and analysis constitutes the application's intelligent assistant feature. This functionality analyzes the user's defined problem structures and provides suggestions for improvement, new sub-problem branches, or alternative solution approaches. Users can accept, modify, or delete these recommendations, maintaining human oversight at all stages. This "human-in-the-loop" approach balances AI support with user control.

Additional contextual information and insights can be incorporated by modifying the title and description of nodes, as well as by applying color coding. Users can annotate each node with notes, references, or explanatory comments. This feature facilitates the documentation of the thought process underlying problem-solving and allows for later review. It also enhances logical structuring and simplifies collaboration among multiple users.

Tools for optimizing the visual organization of the tree structure significantly enhance the user experience. These tools provide functionalities such as drag-and-drop node positioning, visual alignment, and arrow-based relationship mapping, allowing users to arrange the problem tree according to their preferences. These visual editing capabilities support the improved comprehension and presentation of complex problem structures.

Finally, for nodes that require more detailed analysis, the system enables the creation of new canvases derived from selected nodes. This functionality allows specific problem segments to be transformed into independent problems. Users can seamlessly navigate between the newly created canvases and the source canvas, ensuring continuity in problem analysis within a frontend-supported user interface. [1]

3.5 Case Studies

As mentioned earlier, the conceptual approach of this study heavily draws inspiration from case studies conducted by top management consulting firms.

To illustrate the application of our approach, we reference two example cases that demonstrate the use of issue trees in business problem-solving:

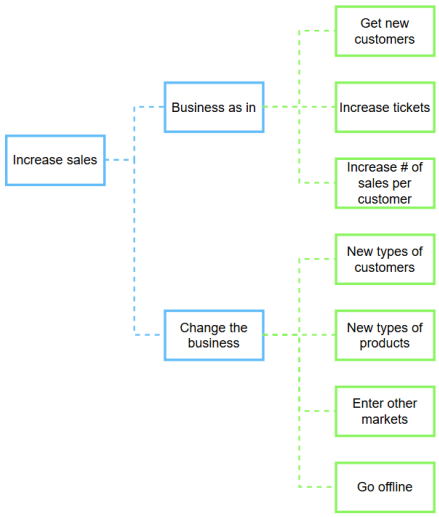


Figure 2: Issue tree for question*"How can a company increase sales?"* (Source)

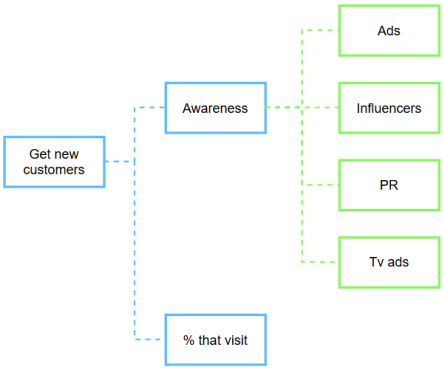


Figure 3: Issue tree for the follow up question*"How else can a company increase sales?"* (Source).

- Figure 2: An issue tree answering the question, "How can a company increase sales?"
- Figure 3: A refined issue tree responding to the follow-up question, "How else can a company increase sales?"

4 Evaluation

The quality metrics involved in this study are based on recognized concepts in the field of strategy consulting along with research in the field of explainable artificial intelligence. The evaluation was designed to generate qualitative results reflecting the quality of the outputs. The token consumption data of the all cases were also collected for review.

4.1 Visual quality of product

User experience (UX) plays an important role in the usability of the tool. However, due to the scope

of this project, direct evaluation with real users is not included. Instead, we assess the product's visual quality by analyzing how well it aligns with theoretical UX-principles for the logic trees.

A method proposed to evaluate explainability in AI-driven tools is based on the framework proposed by Hoffman et al. (2019). This framework outlines five key factors for measuring explainability:

1. **Goodness** - How well the explanations align with correct reasoning.
2. **Satisfaction** - How understandable and useful the explanations are.
3. **Users' Mental Models** - How well users can predict and trust the system's behavior.
4. **Curiosity About the Model** – How much the explanations encourage further exploration.
5. **Trust & Performance** – How well the system performs compared to expected results.

For this project, three of these factors —Goodness, Satisfaction, and Performance— are particularly relevant. These will be assessed based on predefined criteria to determine:

- Ease of understanding how the tool generates logic trees.
- Trust in the model's outputs, based on logical consistency.
- Performance compared to manually constructed tree examples.

By evaluating these factors, we can gain insights into the product's explainability and usability without requiring direct user testing.

4.2 First Case Study Evaluation

Another evaluation has been made with the use of available material from the organisation *The Duke MBA Consulting Club* (DMCC). The cases are cases that have been used in industry interviews and they cover a variety of fields including the fields of pharmaceutics, financial services, oil and gas, telecommunications, retail, non-profit, transportation, insurance, steel, hospitality and the public sector. This material was used since these particular set of interview cases had sufficient supplementary information to support evaluations of the interviews on top of having quality case prompts and suggestive answers.

The evaluation parameters were chosen in accordance with an interview feedback form from their Casebook (see Appendix 16). The general layout of this interview feedback form has been used for the evaluations of the application, with a few alterations introduced to adapt their suggested procedure for this type of application. The overall categories presented in the interview feedback form are *Execution*, *Communication* and *Behavioral*, see Figure 4 to view a visual representation of the categories in the feedback form. The category *Communication* was left out since it measures aspects of human communication such as professionalism and the ability to create suitable writing during an interview process.

However, content for the category *Execution* was considered interesting to measure in this study. It in turn consisted of the subcategories *Structure*, *Quantitative Ability* and *Business Intuition* with the underlying components of evaluation presented in Figure 4.

Execution	
Structure	**Logical Approach**
	MECE
	Appropriate drive to solution
Quantitative Ability	**Speed**
	Accuracy
	Reaction to mistakes
Business Intuition	**Practical**
	Insightful
	Breadth & depth across multiple functions
	Creativity
Communication	
Professionalism	**Clarity of writing and page layout** **Ability to refer back** **Comfort, reaction to mistakes**
Business Intuition	**Poise** **Confident-Persuasive** **Articulate-concise** **Client ready**
Optional	
Behavioural	**Quality of star stories** **Length** **Clarity** **Relevance**

Figure 4: Content of original case interview feedback form (See Appendix 16).

The subcategory of *Quantitative Ability* was also left out because some of its underlying components are trivial while other components are difficult

Execution	
Structure	Logical Approach
	MECE
	Appropriate drive to solution
Business Intuition	Practical
	Insightful
	Creativity
Optional	
Behavioural	Length

Figure 5: Modified version of form adapted for evaluation of application.

to measure with the available resources and considering the nature of the task.

Moreover, for the subcategory of *Business Intuition* all underlying components were included except the component named *Breadth & depth across multiple functions*. In addition, the feature *Length* in the *Optional* section is similar to the category *Speed* therefore both can be addressed simultaneously. In this particular application, this category has been evaluated by measuring token usage for a given test-case.

A new feature has been introduced, which is the factor of uncertainty of the task, which can arise from subtle differences in certain formulations of the given cases. This feature will be referred to as *Uncertainty*. It will be based on the comment/comments given to each case that is evaluated.

The evaluation categories in use for this study are therefore *Structure* and *Business Intuition* as well as *Uncertainty*. These final evaluation categories are collected to be seen in Figure 5.

Close attention was payed particularly to the MECE principle in the subcategory *Structure*. This is performed manually by verifying whether:

- The components of the tree are mutually exclusive, meaning no overlap exists between different branches.
- The tree is collectively exhaustive, ensuring that all relevant aspects of the problem are covered without gaps.

Before carrying out the evaluations some of the case prompt were rewritten to contain only one question since many of the prompts included several questions or lacked a clear formulation for a request of the solution even though the prompt presented the issue in the context of searching for advice, see Appendix 14 for further details. The evaluation was carried out by one evaluator. The process involved the generation of 2 levels of each logic tree and assessing to which extent the generated output solves the given prompt. Note that, optimal output trees are thus not expected. The token consumption information as well as the generated trees were then saved for potential future efforts of cross-validation. Moreover, it can be noted that for the first case study as opposed to the second case study, section 4.3, the most confident answer for each node was saved in the second layer since the outputs of the prototype could otherwise risk generating cluttered outputs that are difficult to examine.

4.3 Second Case Study Evaluation

Another evaluation has been made with the use of a collection of business cases with already known solution trees. The cases are all collected in Table 1. The evaluation procedure used the same parameters and procedure as the first case study, section 4.3.

Case number	Logic tree	Title
1	See Appendix 6	Leather goods
2	See Appendix 7	Airlines decrease revenue
3	See Appendix 8	Nespresso market share drop
5	See Appendix 10	Profitability
6	See Appendix 11	Telco case
7	See Appendix 12	TV ad revenue
8	See Appendix 13	Airline fuel cost

Table 1: Cases with logic trees.

This evaluation differs from the evaluations in section by excluding the *Uncertainty* parameter since the case prompts were not rewritten. Instead, the parameter *Likeness* was introduced in order to track similarities between the produced logic tree output and the expected output of the solution logic trees.

5 Results

For Table 3 the process was straightforward. The prompts were inserted as they were and the output was compared to existing solution logic trees (see

Number	Case Name	Structure 1-4	Intuition 1-4	Average Score	Uncertainty 1-4
1	Purple Pill Company	3	2	2.5	2
2	Buy Low, Sell High	3	3	3	3
3	Oklahoma Gas Company	3	2	2.5	3
4	Heavy Things Fitness	3	3	3	1
5	Orange Yoga Studio	3	2	2.5	3
6	Surfboard wax in Hawaii	3	3	3	2
7	Pacific Northwest Telco	3	2	3	3
8	The Everything Retailer	3	3	3	2
9	Gee & Gee´s House of Brands	3	3	3	2
10	Going Green	3	3	3	4
11	AllHealthy CRM	3	3	3	3
12	Coyotes	3	2	2.5	3
13	Ferry Follies	3	3	3	2
14	AutoDrivers	3	2	2.5	3
15	Steel Works	2	1	1.5	4
16	WOEM	3	3	3	2
17	Money in Michigan	3	2	2.5	3
18	Pharma Co.	-	-	-	-
19	Quality Control	-	-	-	-
20	WorkIT	3	1	3	4
21	So Fresh and So Clean	3	2	2.5	2
22	Kid Country	3	2	2.5	3
23	Texas Oil	3	1	2	4
24	Consumer Products Strategy	3	2	2.5	3
25	Pharmaceutical Growth	3	2	2.5	3
26	Insurance Co. Restructuring	1	2	2.5	3

Table 2: Qualitative results for 26 typical interview cases. Note that cases 18 and 19 are missing, due to difficulties with prompt reformulation. The subcategories *Structure* and *Business Intuition* follow a scale from 1-4 representing ascending performance. *Uncertainty* is an additional qualitative measure with a scale from 1-4 indicating the quality of the prompt itself. Since the cases are rewritten and have different conditions due to formulations, this parameter reflects difficulties encountered in adjusting the prompt to be suitable for the application to process, see Appendix 14 for further details. The higher the value, the higher the uncertainty of that particular case result.

Number	Case Name	Structure 1-4	Business Intuition 1-4	Average Score	Likeness
1	Leather goods	2	2	2	Not similar
2	Airlines decreased revenue	2	3	2.5	Not similar
3	Nespresso market share drop	4	3	3.5	Similar
4	Elevator case	3	3	3	Similar
5	Profitability	4	3	3.5	Similar
6	Telco case	2	2	2	Not similar
7	TV ad revenue	2	3	2.5	Not similar
8	Airline fuel cost	3	4	3.5	Not similar

Table 3: Qualitative results for 8 typical why-issue tree cases. The subcategories *Structure* and *Business Intuition* follow a scale from 1-4 representing ascending performance. *Likeness* is an additional qualitative measure indicating if there exists similarities in produced logic trees compared to solution logic trees (See Appendix 7).

Appendix 7).

For Table 2 the case scenarios and their respective prompts have been collected (see Appendix 14). All of the prompts from the case study were reviewed and some of them have then been rewritten in order to be used in the application. This was necessary since there existed various formulations, such as for instance, cases with several questions in one prompt. This is obviously difficult for the application to solve without creating confusing output.

Furthermore, the token consumption for the various cases are summarized in Table 4.

6 Discussion

The results from the evaluations as well as observations made during the creation of the prototype are collected in the following section.

6.1 Results analysis

The cases presented in Table 3 present average scores that are in the range of 2 - 3.5. This is an indicator that the application performed rather well considering the use of a scale from 1-4 showing ascending performance. The *Likeness* parameter also shows interesting results. As indicated in the last column of Table 3 the logic trees are usually rather different from the familiar solution trees, however, there are some cases when these solution trees are rather similar or exactly the same.

The cases presented in Table 2 present average scores that are in the range of 2-3. This is an indicator that the application does give helpful outputs for prompts from various fields.

The *Uncertainty* field in Table 2 gives an overview of the troublesome cases. The exact weaknesses can be further analysed in Appendix 15.

The main difference between the cases in Table 3 and Table 2 is that the prompt content in Table 2 is considerably longer and therefore more complex. Although this difference, both types of cases give similar qualitative scores with better performance among the shorter and more clearly formulated prompts in Table 3.

Token Consumption 1

Completion	Prompt	Total
2233	18794	21027
866	9457	10323
8135	16248	24383
833	8581	9414
4895	13656	18551
2063	9918	11981
659	3124	3783
1370	11385	12755
1035	8551	9586
860	7189	8049
846	7279	8125
755	5905	6660
1077	8615	9692
890	7191	8081
655	5868	6523
897	7211	8108
984	7222	8206
-	-	-
-	-	-
987	7214	8201
951	7280	8231
990	7254	8244
895	7231	8126
714	5846	6560
1010	7299	8309
1309	7400	8709

Token Consumption 2

Completion	Prompt	Total
575	3113	3688
710	3145	3855
645	3131	3776
739	3900	4639
393	2319	2712
648	3110	3758
608	3126	3734
659	3124	3783

Table 4: Token consumption details for Table 2 followed by token consumption details for Table 3. *Completion* and *Prompt* are the amount of tokens in the output and input respectively and the *Total* is the combination of both.

The clear strengths of the application in comparison to human performance are naturally the enhanced access of information and the speed in which it generates responses. Additionally, it was observed that the application with its current setup generates decent MECE solutions as well

as a clear logical structure, in almost all of its responses. On the other hand, the application was also flawed in its ability to focus and prioritize solution paths. In general it was also not that specific or practical in its suggestions.

Table 3 and Table 2 in combination with the results from Table 4 show that the application is in general efficient at presenting helpful solutions to the prompts. The token consumption of the test-cases is shorter with a token size of approximately 3000 while the cases presented in Table 2 have long prompts ranging between 3000 to 18000. The completions for Table 3 range between 500 - 700 tokens while the completions for the longer prompts generally required longer completions as well. The token usage varied between 700 - 8000 tokens given the evaluation procedures in this particular study. All in all, these token counts indicate the amount of tokens that are generally needed for these types of problems and also show that the most efficient output trees do not require that many tokens.

6.2 Features

It is a craft to create good strategy trees and therefore any feature that is used or introduced into the application needs to be evaluated to what extent it enhances or hinders that work.

Certain features can clarify and streamline the process of interpreting the trees, but the effects of those features vary greatly from individual to individual. Still, the adoption of universally understandable designs for the implementation may be of importance to encourage increased usage. However, pinpointing those features will likely require a robust strategy for further testing.

It can be noted that the prototype is merely a work in progress. It demonstrates a selection of key features, but the effect of the separate features has not yet been researched enough to be finalized at the state they are currently in. It may also be the case that certain features will be considered as not necessary and that certain features may need to be introduced. A good alternative would be to create a new version with only the vital features and evaluate this as well to measure if that alone can improve the performance of the application. Suggestively, create separate versions of the minimal required features to pinpoint a particular combination of 1-2 essential features for the application.

However, a selection of features to investigate and research/test further could be selected features such as node design, including evaluations of colors, readability and general ease of usage. It could also be non-design related features as the maintenance of the application itself. Ensuring that the application aligns with up-to-date technology services and effective security standards.

6.3 Alignment with theory

An initially observed issue with the application was that it was often not possible for the LLM to generate the exact logic-tree-structures of known solved cases. However, this can be seen rather as an advantage since the LLM showed to generate other types of trees with more creative outputs. On the other hand, there is the other extreme where the LLM hallucinates. This can, for instance, occur when a formulation is unclear or a word is ambiguous.

Since the application is based on certain well-known theories in the field of strategy consulting, it is also interesting to explore the effects of them in the application. Introducing more theoretical components can be beneficial, but preferably only the most important such features should be introduced to avoid confusion among users.

Then there are also certain differences between types of users. For professionals that are actively engaged with strategy consulting content, there may be a different level of understanding and therefore also need for theoretical support frameworks compared to a user that does not have any prior knowledge of the field. For instance, it is possible to include more specific decomposition models like introducing the option for users to use common decomposition types such as *Segmentation*, *Opposite Words* or *Process Structure*.

The paper addressed in 4.1 is revisited as it brings up the use of certain parameters to measure the usefulness of among others LLM based applications. An initial estimation for this application would give reasonable scores on most of the parameters. It does well in ensuring trust in the

model's outputs by being logically consistent and its performance compared to manually constructed tree examples is sufficient. However, as suggested above its ease of understanding can most likely be improved with a better selection of features but this would require more elaborate user testing to determine more certainly.

7 Conclusion

In conclusion it can be gathered that the application performs well. The generated outputs are observed to be insightful, shown to follow the MECE principle and performs very well for clearly formulated prompts. Although, in most real-world cases, it is rather the exception than the norm to find well defined prompts and this creates a unique set of issues of examining the various approaches in which human assistance can be incorporated into the tool. It is however interesting to investigate since optimally such an application would cut down on the time it takes to solve various real-world cases. Efficient use of the application is also a concern since token consumption directly affects the running costs of the product.

There are many approaches to improving the efficiency of the application, but in order to adjust it for better performance, there is a need to further investigate different user groups and their specific needs and obstacles when using the application. Furthermore, implementation of theoretical concepts can be useful as well as re-assessing the value of already implemented features.

Distribution of workload

Ann - Responsible for the Evaluations.
Altin - Responsible for the Backend-code.
Bilgehan - Responsible for the Frontend-code.

References

O. Sibony B. Garrette, C. Phelps. 2018. *Cracked it!:How to solve big problems and sell solutions like top strategy consultants*, volume 1. Palgrave Macmillan Cham.

Shulin Cao, Jiajie Zhang, Jiaxin Shi, Xin Lv, Zijun Yao, Qi Tian, Juanzi Li, and Lei Hou. 2023. Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions. *Preprint*, arXiv:2311.13982.

Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. 2019. Metrics for explainable ai: Challenges and prospects. *Preprint*, arXiv:1812.04608.

Takeshi Kojima, Shixiang Gu, Miles Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners.

Barbara Minto. 2009. *The Pyramid Principle: Logic in Writing and Thinking*. Financial Times/Prentice Hall.

Jason Wei, Yi Tay, Paul Barham, He He, William Zhou, Tal Schuster, Quoc Le, Azalia Mirhoseini, et al. 2022a. Chain-of-thought prompting elicits reasoning in large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Appendix A: Logic trees

Case 1: Why did our leather goods company make less money in 2020 compared to 2019?

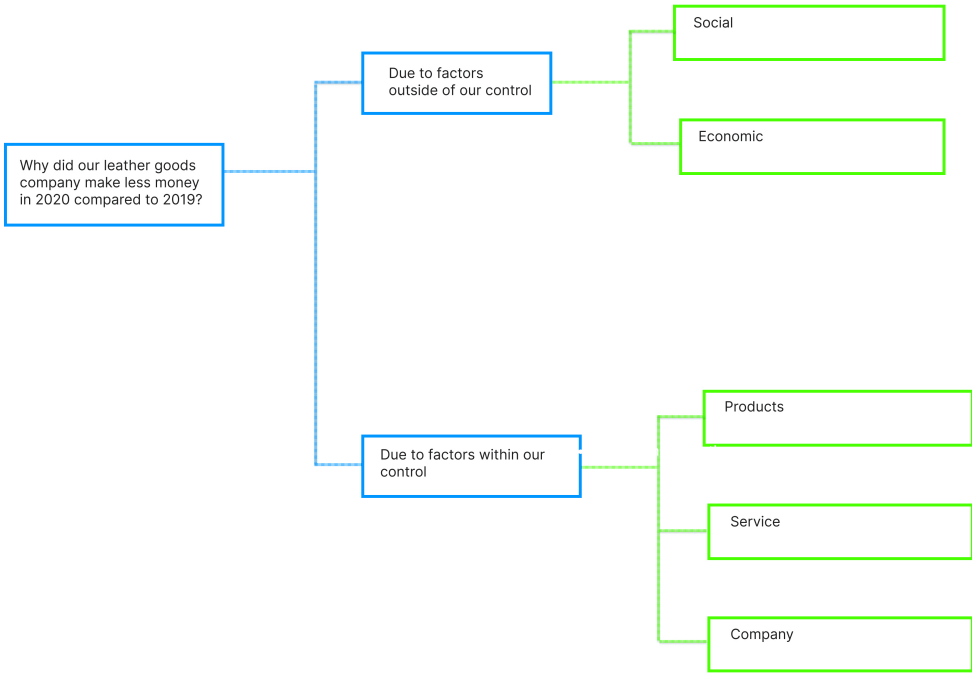


Figure 6: Solution logic tree for Case 1: *Leather goods* (Source)

Case 2: Why the airline company revenues decreased by $50m over the past 2 years?

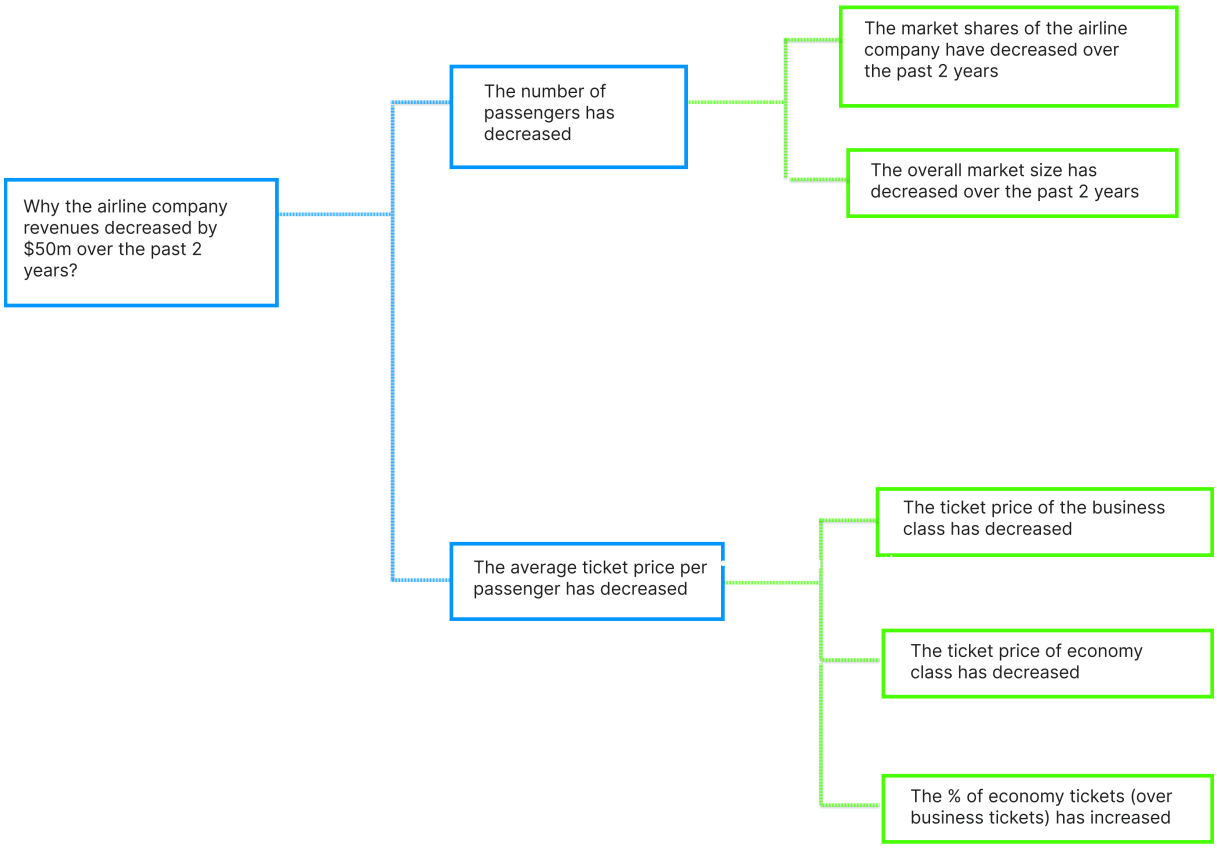


Figure 7: Solution logic tree for Case 2: *Airlines decreased revenue* (Source)

Case 3: What are possible reasons for Nespresso´s market share drop in London´s coffee capsules market?

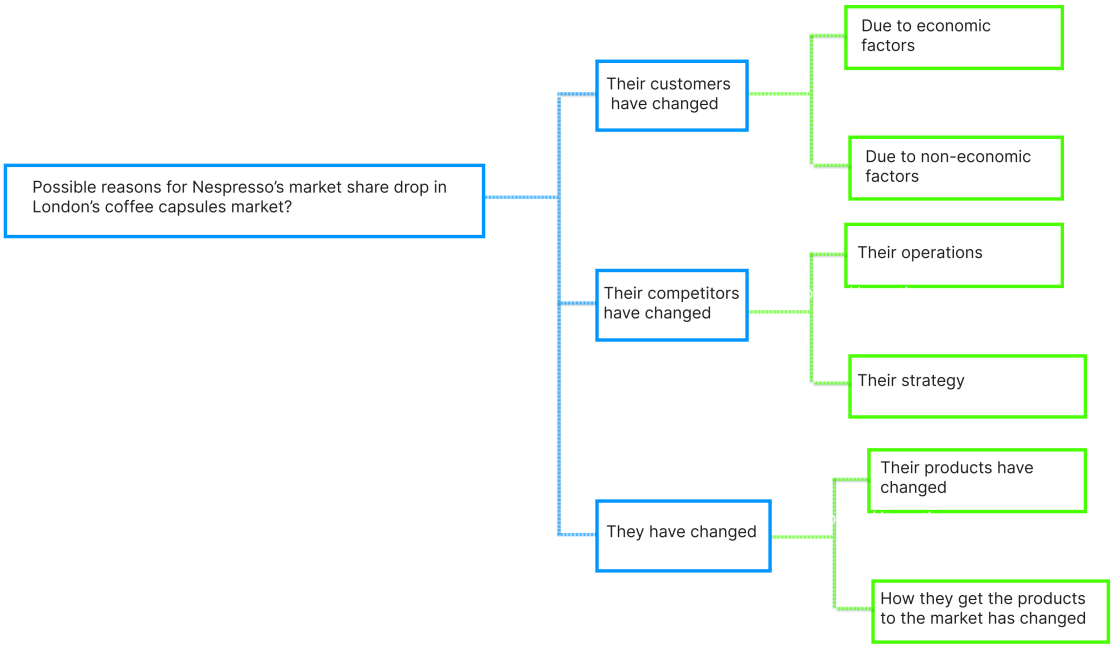


Figure 8: Solution logic tree for Case 3: *Nespresso market share* (Source)

Case 4: Why are tenants waiting too long tor the elevator?

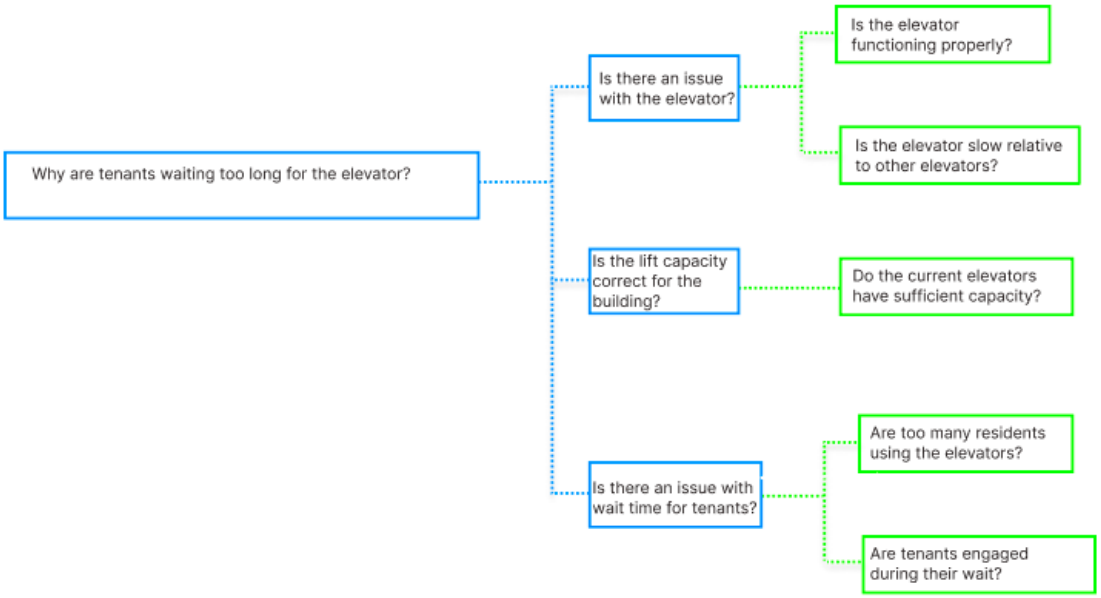


Figure 9: Solution logic tree for Case 4: *Elevator case* (Source)

Case 5: Why is our company not profitable?

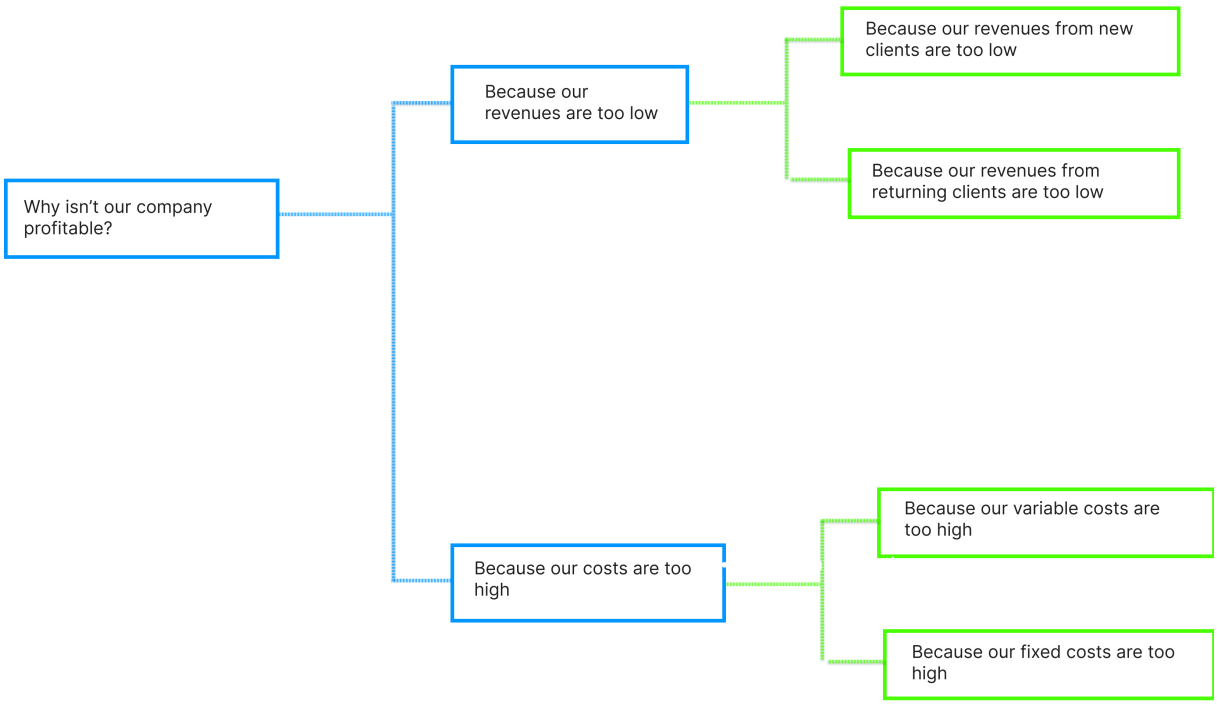


Figure 10: Solution logic tree for Case 5: *Profitability* (Source)

Case 6: Why are more clients of a Telco unsubscribing from their mobile services?

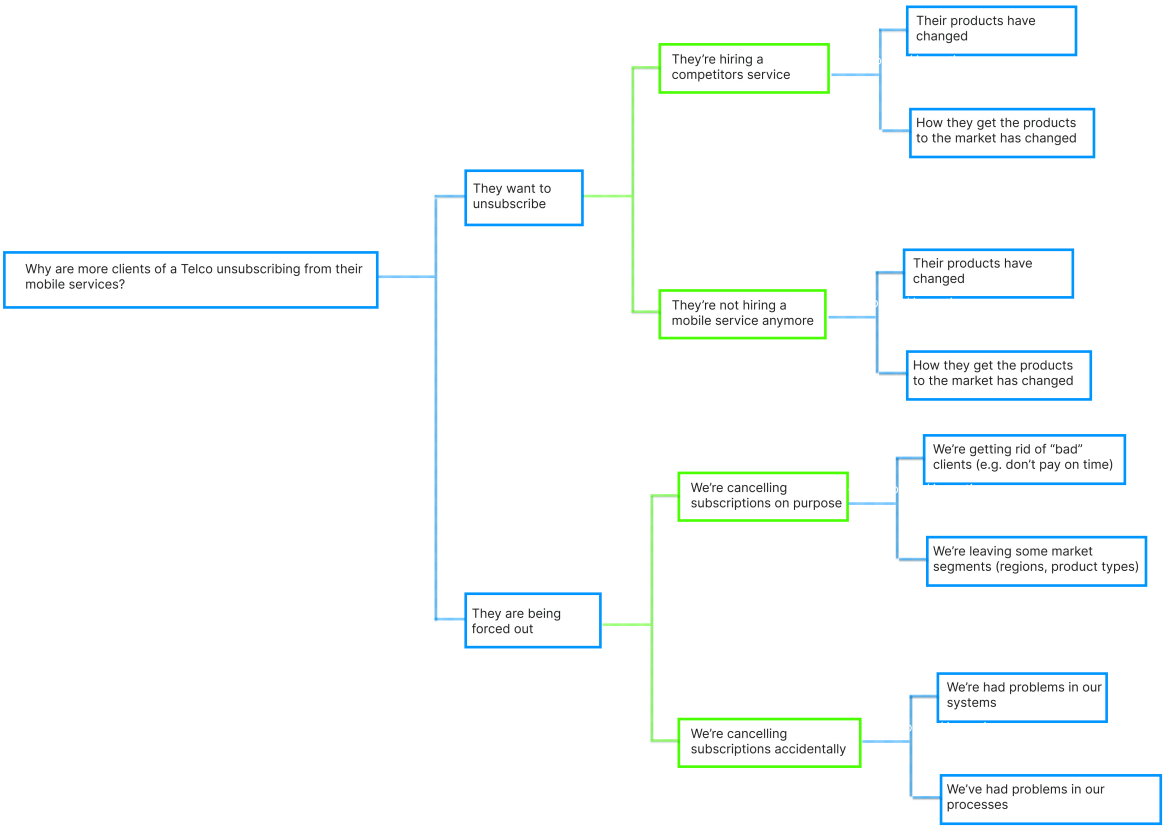


Figure 11: Solution logic tree for Case 6: *Telco case* (Source)

Case 7: Why have TV-ad-revenues dropped in the last decade?

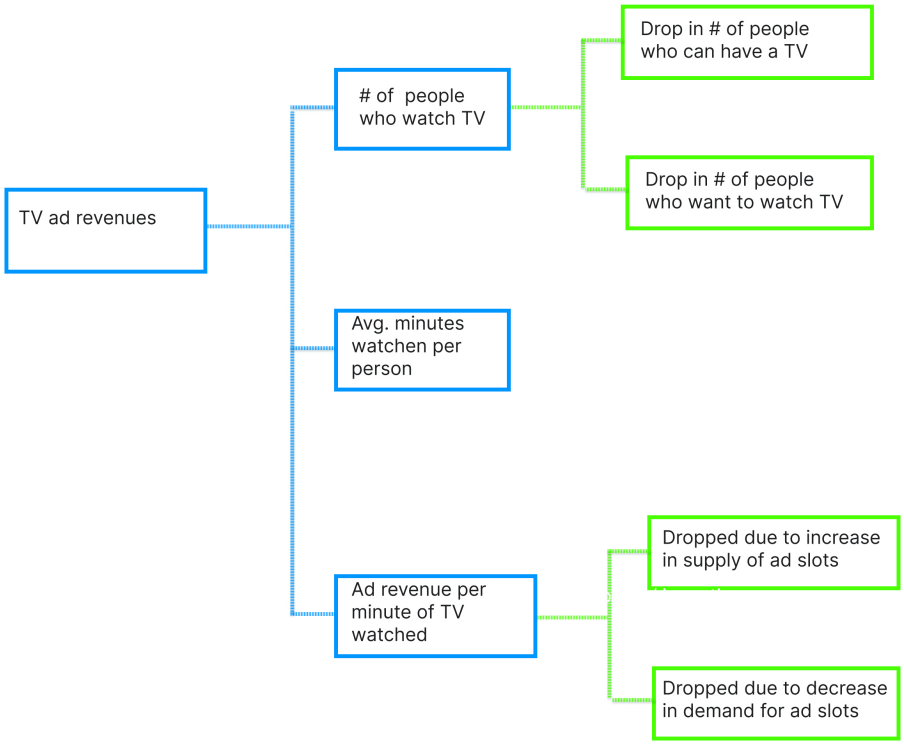


Figure 12: Solution logic tree for Case 7: *TV-ad-revenue* (Source)

Case 8: Why have airline´s fuel cost risen?

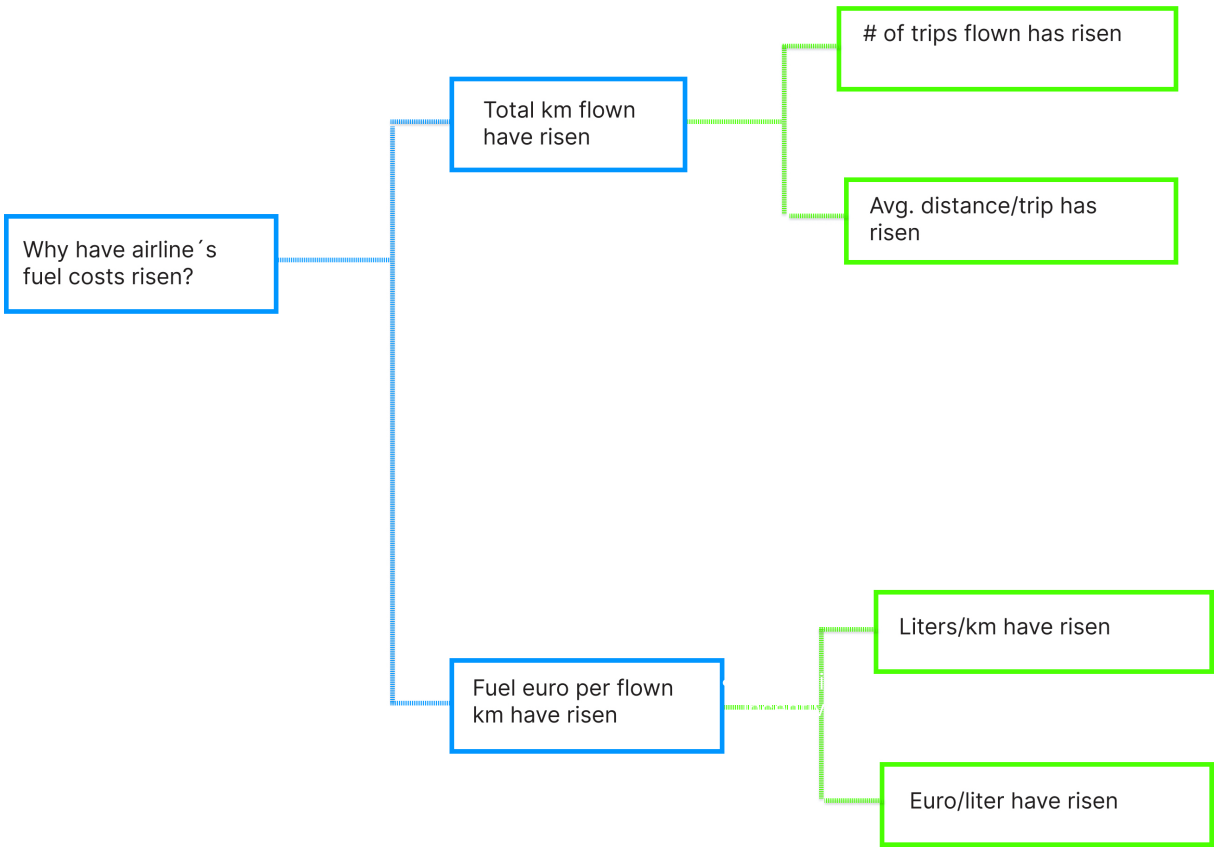


Figure 13: Solution logic tree for Case 8: *Airline fuel cost* (Source)

Appendix B: Tables

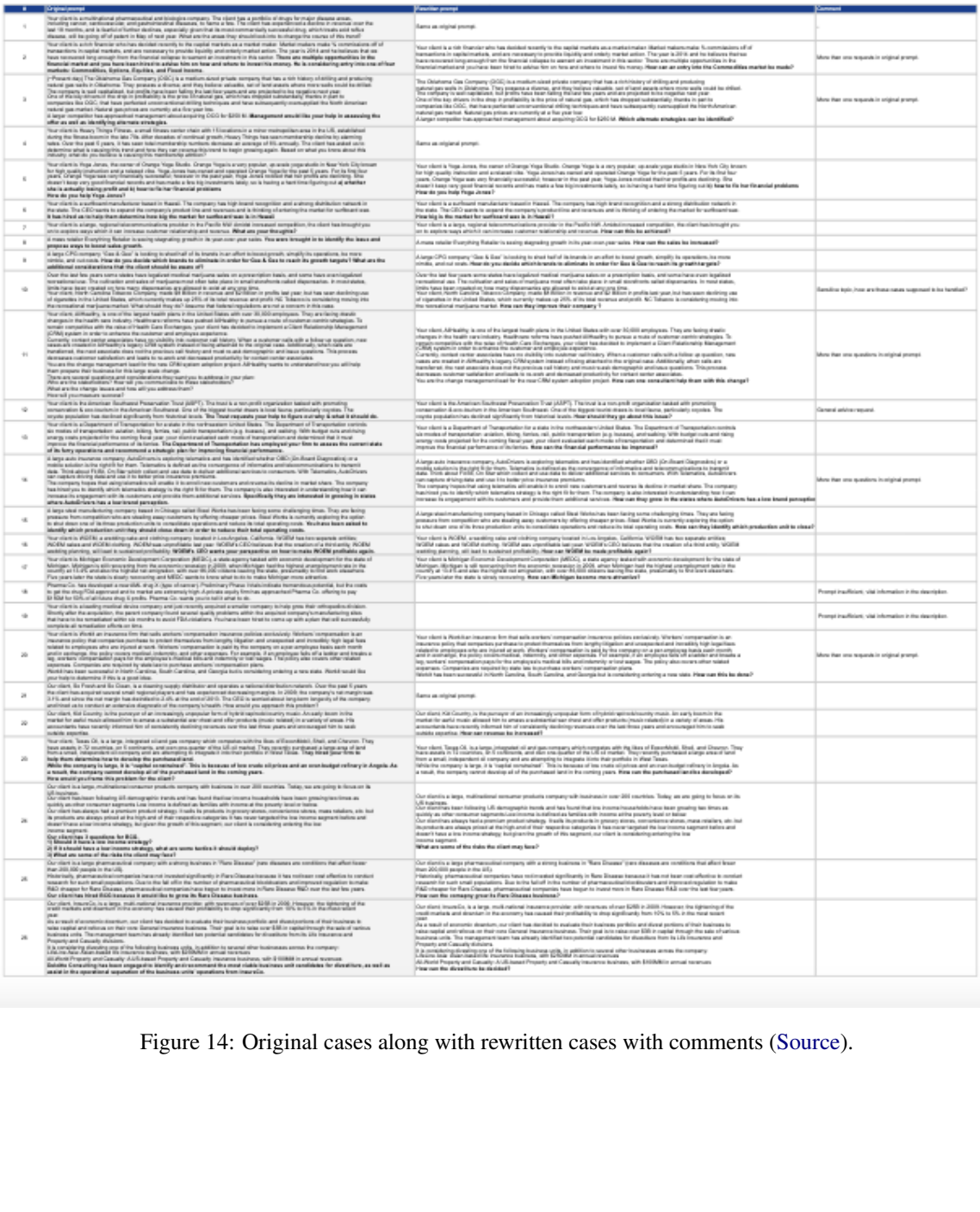
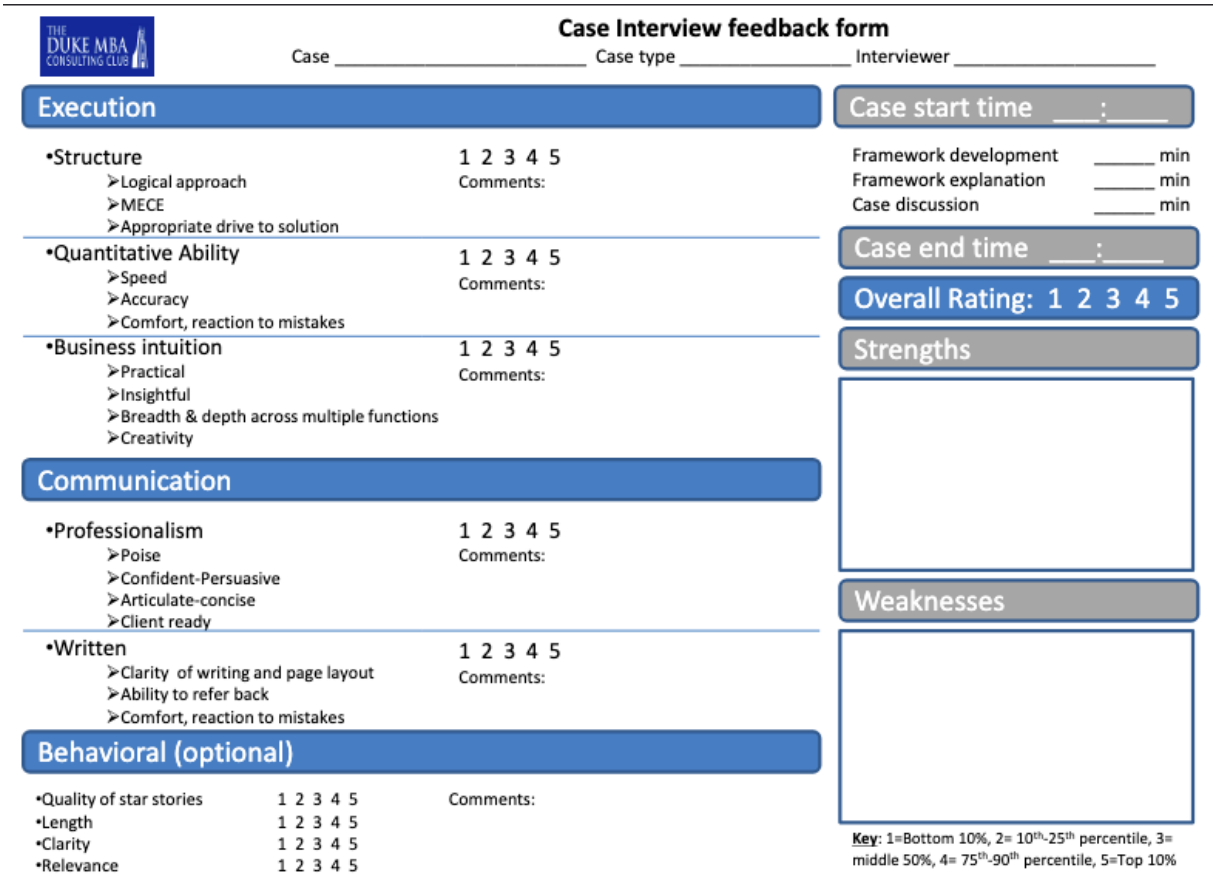


Figure 14: Original cases along with rewritten cases with comments (Source).

		Performance score				
#	Case Name	Structure (1-4)	Business Intuition (1-4)	Comment - Uncertainty(1-4)		Final Score Average
1	Purple Pill Company	3	2	Lack of (1.3) (2.1) (2.2)	2	2.5
2	Buy Low, Sell High	3	3	Lack of (1.3) (2.1) - Case reformulated	3	3
3	Oklahoma Gas Company	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	2.5
4	Heavy Things Fitness	3	3	Lack of (1.3)	1	3
5	Orange Yoga Studio	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	2.5
6	Surfboard Wax in Hawaii	3	3	Lack of (1.3) - Case reformulated	2	3
7	Pacific Northwest Telco	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	3
8	The Everything Retailer	3	3	Lack of (1.3) - Case reformulated	2	3
9	Gee & Gee's House of Brands	3	3	Lack of (1.3) - Case reformulated	2	3
10	Going Green	3	3	Lack of (1.3) - Case reformulated - Sensitive topic.	4	3
11	AllHealthy CRM	3	3	Lack of (1.3) - Case reformulated - Serious topic	3	3
12	Coyotes	3	2	Lack of (1.3) (2.2) - Case reformulated	3	2.5
13	Ferry Follies	3	3	Lack of (1.3) - Case reformulated	2	3
14	AutoDrivers	3	2	Lack of (1.3) (2.1) - Case reformulated	3	2.5
15	Steel Works	2	1	Lack of (1.2) (1.3) (2.1) (2.2) - Case reformulated	4	1.5
16	WOEM	3	3	Lack of (1.3) - Case reformulated	2	3
17	Money in Michigan	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	2.5
18	Pharma Co.	-	-	Prompt insufficient and not even possible to reformulate.	-	-
19	Quality Control	-	-	Prompt insufficient and not even possible to reformulate.	-	-
20	WorkIT	3	1	Lack of (1.3) (2.1) (2.2) (2.3) - Case reformulated	4	3
21	So Fresh and So Clean	3	2	Lack of (1.3) (2.1)	2	2.5
22	Kid Country	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	2.5
23	Texas Oil	3	1	Lack of (1.3) (2.1) (2.2) (2.3) - Case reformulated	4	2
24	Consumer Products Strategy	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	2.5
25	Pharmaceutical Business Growth	3	2	Lack of (1.3) (2.2) - Case reformulated - Serious topic	3	2.5
26	Insurance Co. – Business Restru	3	2	Lack of (1.3) (2.1) (2.2) - Case reformulated	3	.2.5

Figure 15: Complete results of the 26 evaluated cases with comments indicating exact weaknesses of each case. The category *Structure* has 3 subcomponents; (1.1) Logical approach, (1.2) MECE, (1.3) Appropriate drive to solution. The category *Business intuition* also has 3 subcomponents; (2.1) Practical, (2.2) Insightful, (2.3) Creativity, as indicated in 5. (Original source of cases)



Case Interview feedback form

Case _____ Case type _____ Interviewer _____

Execution

- Structure — 1 2 3 4 5 — Comments:
  - Logical approach
  - MECE
  - Appropriate drive to solution
- Quantitative Ability — 1 2 3 4 5 — Comments:
  - Speed
  - Accuracy
  - Comfort, reaction to mistakes
- Business intuition — 1 2 3 4 5 — Comments:
  - Practical
  - Insightful
  - Breadth & depth across multiple functions
  - Creativity

Communication

- Professionalism — 1 2 3 4 5 — Comments:
  - Poise
  - Confident-Persuasive
  - Articulate-concise
  - Client ready
- Written — 1 2 3 4 5 — Comments:
  - Clarity of writing and page layout
  - Ability to refer back
  - Comfort, reaction to mistakes

Behavioral (optional)

- Quality of star stories — 1 2 3 4 5 — Comments:
- Length — 1 2 3 4 5
- Clarity — 1 2 3 4 5
- Relevance — 1 2 3 4 5

Case start time ___:___

Framework development _____ min
Framework explanation _____ min
Case discussion _____ min

Case end time ___:___

Overall Rating: 1 2 3 4 5

Strengths

Weaknesses

Key: 1=Bottom 10%, 2= 10th-25th percentile, 3= middle 50%, 4= 75th-90th percentile, 5=Top 10%

Figure 16: Original interview feedback form (Source).