Large Language Models as your Personal Data Scientist

Md Mahadi Hassan, Alex Knipper, and Shubhra Kanti Karmaker (“Santu”)
Big Data Intelligence (BDI) Lab
Auburn University
Auburn, AL
{mzh0167, rak0035, sks0086}@auburn.edu

Abstract

Large Language Models (LLMs) show promise in language tasks, but their potential in ‘ill-defined’ and complex areas like conversational data science is under-explored. In this field, users interact with AI agents for data science needs, where the agent converses and performs Machine Learning tasks like a human expert. Recent LLM advancements suggest the feasibility of a fully functional conversational data science system soon.

1 Introduction

Large Language Models (LLMs) are transforming Natural Language Processing with their ability in summarization, paraphrasing, and translation, and can be further enhanced through prompt engineering (Fan et al., 2023; Zhao et al., 2023). Despite these advancements, their full potential in complex situations remains unexplored. In contrast, conversational data science aims to simplify data science for users, unlike the more complex “Automated Machine Learning” (AutoML). AutoML’s complexity challenges non-expert users such as domain experts, who often rely on data scientists, resulting in inefficient use of AutoML’s automation capabilities (Sarkar et al., 2023; Karmaker et al., 2021).

Our study presents VIDS (Virtual Interactive Data Scientist), an intelligent agent aiding in basic data science tasks via natural dialogues, as illustrated in Figure 1. Utilizing "Prediction Task Expression Language" (PeTEL) (Karmaker et al., 2021), VIDS accurately grasps users’ prediction aims, translating them into specific machine learning (ML) tasks. PeTEL, crucial for the Large Language Model (LLM), enable VIDS understanding and defining ML tasks, managing datasets, running autoML pipelines, and delivering results. VIDS comprises four dialogue states: Data Visualization, Task Formulation, Prediction Engineering, and Result Summary and Recommendation. VIDS uses multiple LLM micro-agents to ensure a seamless conversational flow. Overall, the research highlights the intersection of LLMs and AutoML in advancing Conversational Data Science, exploring AI’s role in data science, LLMs in complex tasks, and challenges in systems like VIDS.

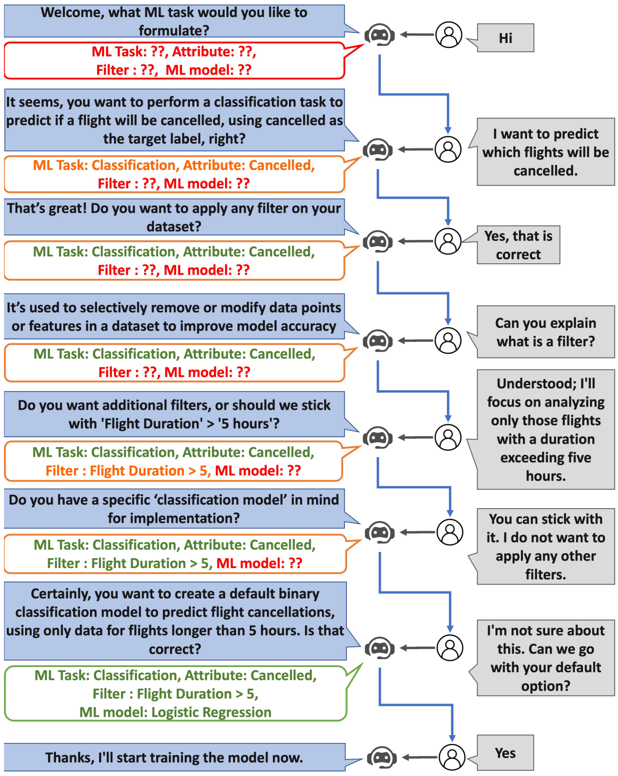

Figure 1: A hypothetical dialog between an intelligent agent and the user for prediction goal identification.

2 Model Architecture

The VIDS model employs four distinct dialogue states: Data Visualization, Task Formulation, Prediction Engineering, and Result Summary and Recommendation as shown in Figure 2. Each state represents different stages of a user-agent conversation. VIDS uses stateless global micro-agents for fluid transitions and local micro-agents tailored to specific states for handling nuances in user utterances and contexts. This design is envisioned to ensure smooth narrative flow and adaptability, with a symbiotic relationship between global and local agents to enhance dialogue versatility.

Global Micro-agents: VIDS integrates three micro-agents for dialogue management: State Detector, Dialogue Summarizer, and Conversation Manager. The State Detector guides user interactions through phases like "Data Visualization," "Task Formulation," "Predic-

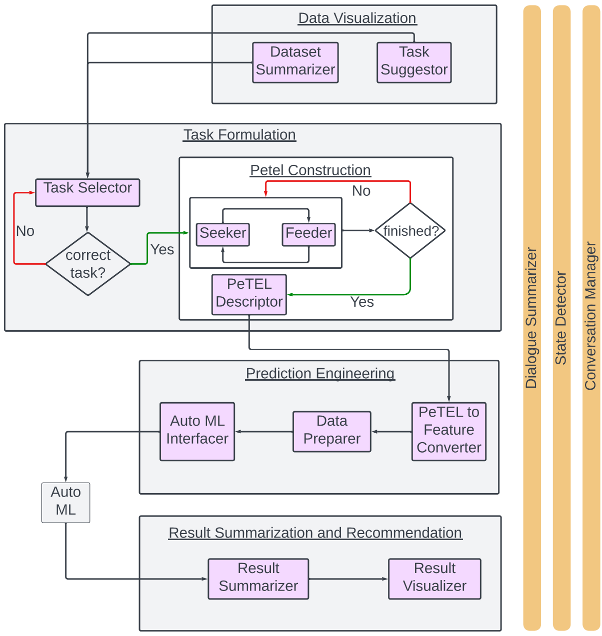

Figure 2: The state diagram of the VIDS dialogue system. The gray boxes are different states of the conversation history, the dark yellow boxes are global micro-agents, and the purple boxes are the local micro-agents.

tion Engineering," and "Result Summarization and Recommendation," tailoring to user needs. The Dialogue Summarizer merges conversation elements for clarity, while the Conversation Manager, informed by other agents, maintains contextually accurate dialogue. See Appendix A.1 for more details.

Local Micro-agents: In VIDS, local micro-agents are instrumental in data science workflows. Initial stages involve Dataset Summarizer and Task Suggestor, which explore datasets and suggest ML tasks. This leads to Task Formulation by the Task Selector and PeTEL Representation Construction micro-agents, where ML tasks are defined based on dataset summaries and user objectives. The PeTEL Representation involves the Seeker and Feeder micro-agents for precise task specification. Subsequently, Prediction Engineering, handled by the PeTEL to Feature Converter, Data Preparer, and AutoML Interfacer, prepares data for ML algorithms. The Result Summarizer micro-agent is proposed for automating the summarization of findings using AutoML libraries. Future enhancements include a Result Visualizer for better outcome visualization and an interactive dialogue system for improved decision-making. Follow Appendix A.2 for more details.

3 Case Study

We present a qualitative evaluation of LLMs in performing complex tasks through an in-depth case study. We exclusively focus on ChatGPT for this case study and utilize the Student Performance [1] dataset. Three criteria guide the analysis: usability and efficacy in data science conversations, interaction among micro-agents each powered by an LLM, and the effectiveness of the TeLER Prompt Taxonomy (Santu and Feng, 2023) in designing prompts for these micro-agents.

we explore the user-system chat cycle within the VIDS system, highlighting LLMs' role in facilitating natural conversations for data science tasks. Table 10 (in the appendix) presents a tangible snapshot of user-system interactions, exemplifying the successful execution of the user's goal task via natural conversation. The micro-agents' roles and interactions are examined across four system states, focusing on their contribution to task execution and internal dynamics. See Appendix B.2 for more details. The versatility and resilience of the TeLER taxonomy are evaluated through its application in crafting prompts for different micro-agents, revealing insights into human-AI interactions in complex tasks. See Appendix B.3 for more details.

4 Discussion

Our exploration into LLMs role in Conversational Data Science has uncovered both their potential and the challenges they face at the edge of their capabilities. LLMs have shown promise in making the ML pipeline more automated, improving the accuracy of AI responses, managing dialogues more effectively in automated systems, and helping to democratize Data Science. Yet, we've also run into issues with LLMs, such as recognizing the state of conversations, summarizing dialogues, and dealing with complex prompts. For instance, the Dialogue Summarizer, a tool designed for summarization, tends to generate generic summaries with a potential bias towards lengthy dialogues over the most recent utterances. Additionally, prompts that are too complex can lead to overly detailed responses, which is not always helpful. Our findings highlight the significance of using detailed prompts to get better responses from LLMs, their tendency to prefer longer dialogue entries, and the importance of finding the right prompt length to keep things manageable. These insights point to the need for ongoing research and fine-tuning to strike the right balance between task complexity and the performance of LLMs, aiming to unlock their full potential in automation. As we continue to push the limits of LLMs, our goal is to contribute to the progress and wider availability of Data Science.

5 Conclusion

This study explores the potential of Large Language Models (LLMs) in creating VIDS, a conversational data science tool. VIDS uses natural language interfaces and has a unique architecture with four dialogue states and micro-agents for coherent interactions. While VIDS shows promise, it faces challenges needing further development for enhanced robustness and efficiency. Our research aims to optimize LLMs for intuitive human-AI interactions, advancing complex task automation and data analysis.

[1]https://www.kaggle.com/datasets/larsen0966/student-performance-data-set

References

Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. 2023. A bibliometric review of large language models research from 2017 to 2023.

Shubhra Kanti Karmaker, Md Mahadi Hassan, Micah J Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. 2021. Automl to date and beyond: Challenges and opportunities. *ACM Computing Surveys (CSUR)*, 54(8):1–36.

Shubhra Kanti Karmaker Santu and Dongji Feng. 2023. Teler: A general taxonomy of llm prompts for benchmarking complex tasks.

Souvika Sarkar, Biddut Sarker Bijoy, Syeda Jannatus Saba, Dongji Feng, Yash Mahajan, Mohammad Ruhul Amin, Sheikh Rabiul Islam, and Shubhra Kanti Karmaker. 2023. Ad-hoc monitoring of covid-19 global research trends for well-informed policy making. *ACM Transactions on Intelligent Systems and Technology*, 14(2):1–28.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

A Model Architecture

This section delves into the methodology and technical details of VIDS, articulating the intricate interplay between overarching structures and localized nuances. Central to our model are four distinct dialogue states - Data Visualization, Task Formulation, Prediction Engineering, and Result Summary and Recommendation, with each representing a unique phase in the conversation and contributing significantly to the overall user-system interaction. VIDS employs multiple stateless global micro-agents, functioning independently of any state-related data or history, to create an overarching structure that enables fluid transitions throughout the dialogue, irrespective of the specific state. This stateless design ensures a smooth narrative flow and avoids complications of state-dependent biases or entanglements, thus bolstering the versatility and adaptability of our dialogue system. Alongside these global agents, local micro-agents, each tailored to a specific dialogue state, proficiently handle the nuances of user utterances and conversation contexts, facilitating smooth transitions between states in line with the evolving dialogue. VIDS' strength lies in this symbiotic relationship between the global and local micro-agents across the different dialogue states.

A.1 Global Micro-agents

VIDS deploys three global micro-agents across all dialogue states, which are briefly discussed below.

A.1.1 State Detector

As a cornerstone of conversation management, VIDS deftly incorporates a number of distinct states, each one reflecting the various phases of interaction with the end user, as illustrated in Figure 2. The system is initialized with the "*Data Visualization*" state, simplifying the data exploration phase. It then progresses to the "*Task Formulation*" state, interpreting the user's intended task. The subsequent "*Prediction Engineering*" state revolves around crafting training/testing data partitions and initializing model hyper-parameters based on the already defined tasks. The interaction culminates in the "*Result Summarization and Recommendation*" state, summarizing results and offering pertinent recommendations. Leveraging the latest context, current dialogue state, and the user's utterance, the system dynamically infers the next conversation state. It facilitates a seamless dialogue flow, aligning with user needs while enriching their interactive experience. Table 1 showcases the unified prompt design guiding LLMs to identify the conversation state and the user intent accurately.

A.1.2 Dialogue Summarizer

This global micro-agent distills ongoing conversations into concise summaries, ensuring efficient communication among various micro-agents. It crafts these summaries by integrating the latest user utterance, previous dialogue history, and current responses, maintaining a coherent context throughout the conversation. The unified prompt design for guiding LLMs in summarizing user-VIDS interactions is presented in Table 2.

A.1.3 Conversation Manager

This micro-agent generates accurate responses by integrating inputs from other micro-agents, maintaining the conversation's context for an effective and seamless dialogue experience for the user. The promoting approach used to steer LLM is demonstrated in Table 3.

A.2 Local Micro-agents

A.2.1 Micro-agents for Data Visualization

First, the user provides a dataset of their choice for customized exploration. Then, the visualization process begins by generating a condensed dataset version, followed by an LLM-driven extraction of insights using the **Dataset Summarizer** micro-agent. Subsequently, the **Task Suggestor** micro-agent, informed by these insights, proposes a suitable Machine Learning task. These cooperative micro-agents ensure efficient dataset exploration and readiness for the Task Formulation phase.

In summary, the Dataset Summarizer and Task Suggestor micro-agents, crucial to this stage, delve into the dataset and propose ML tasks, respectively. These agents, guided by prompting strategies presented in Table 4 and Table 5, pave the way for the Task Formulation stage.

A.3 Micro-agents for Task Formulation

Following Data Visualization, VIDS proceeds to the Task Formulation stage. This state is broken down into two interconnected components: *Task Selection* and *PeTEL Representation Construction*, each managed by specialized micro-agents to ensure a precise formulation of the goal ML task.

Task Selection: The Task Selection phase, managed by the ***Task Selector*** micro-agent, defines the machine learning task from the pool of suggestions based on the dataset summary and user's data science goals. It provides options like classification, regression, clustering, or user-suggested tasks. This iterative dialogue refines user requirements and aligns tasks with their dataset and goals, as directed by the prompting strategy in Table 6. This micro-agent leverages the conversation summary to properly guide the LLM in selecting a suitable ML model that meets both the dataset's characteristics and the user's needs. This process continues until the user is satisfied and confident with their task choice, enabling personalized, hands-on problem-solving within the micro-agent framework.

PeTEL Representation Construction: Upon task selection, VIDS leverages the Prediction Task Expression Language (PeTEL) (Karmaker et al., 2021), a concise, slot-value style structured language that encapsulates the core aspects of the chosen machine learning task. The slot-value format of the PeTEL expression provides an unambiguous task description, delineating desired outcomes and key search parameters.

For shaping a more user-tailored experience, VIDS uses the PeTEL Representation Construction group—a team of micro-agents working in concert, emphasizing precision and user satisfaction. Key components among them are the ***Seeker*** and ***Feeder*** micro-agents, which facilitate an iterative dialogue with the user to populate PeTEL slots. The ***Seeker*** drives the conversation toward Task formulation, ensuring no aspect of the task is missed (Table 7). In parallel, the ***Feeder*** assimilates user responses into PeTEL slots, interpreting user inputs for precise task specification (prompts detailed in Table 8).

Collectively, the ***Seeker*** and ***Feeder*** steer a dynamic dialogue until PeTEL accurately mirrors the user's intent. Finally, the PeTEL Descriptor micro-agent articulates the populated PeTEL in layman's terms, reinforcing user understanding (see Table 9) and active participation. Thus, the PeTEL Construction process yields an effective, user-specific task representation, paving the way for subsequent machine learning stages. Refer to Listing 1 for a filled PeTEL example.

A.4 Micro-agents for Prediction Engineering

The Prediction Engineering phase is the bridge between abstract problem representation and a practical prediction model, composed of three steps: converting PeTEL to features, executing data preparation, and interfacing with AutoML systems.

First, the ***PeTEL to Feature Converter*** micro-agent transforms the task representation into tangible features, readying the problem description for computational processing. Next, the ***Data Preparer*** micro-agent refines the dataset, handling missing data, outliers, and categorical variable encoding to facilitate the application of machine learning algorithms. Finally, the ***AutoML Interfacer*** micro-agent uses the prepared training and testing data sets with AutoML systems. Utilizing these platforms' automation, it selects, optimizes, and trains an appropriate machine-learning algorithm.

A.5 Micro-agents for Result Summary

Data scientists traditionally present findings and advise domain experts on strategies, but this stage is mostly manual. Therefore, automation of Result Summarization and Recommendation is indeed very challenging. Nevertheless, we propose a hypothetical micro-agent ***Result Summarizer*** that will generate a comprehensive summary of findings by employing AutoML libraries like Auto-SKLearn to train multiple models. This summary will enable users to compare and identify the most effective solution, rapidly understanding the core findings. Future versions of VIDS will implement such a ***Result Summarizer*** micro-agent to streamline the Result Summary and Recommendation phase.

Additionally, As part of its future vision, VIDS aims to introduce the ***Result Visualizer*** micro-agent, designed to elevate users' comprehension through visual representations of outcomes. This advanced mechanism will generate suitable visualizations, such as performance metrics and feature importance, providing users with an intuitive and enhanced understanding of the findings. Furthermore, our shoot-the-moon objective is to optimize decision-making through interactive dialogue, where the system recommends the optimal model based on the ongoing conversation. This personalized approach is set to simplify the process, empowering users to make well-informed decisions.

B Case Study

This section reports a qualitative evaluation of LLMs in performing complex tasks through an in-depth case study. We exclusively focus on ChatGPT for this case study and utilize the Student Performance dataset[2] for our analysis. Our evaluation revolves around three main criteria. Firstly, we examine the *usability* and *efficacy* of LLMs in facilitating seamless conversations for data science tasks by analyzing the chat cycle between users and the VIDS system. Secondly, we examine the interaction among the micro-agents themselves, each equipped with its unique LLM instance. This analysis reveals the intricacies of micro-agent dynamics in performing complex "ill-defined" tasks. Lastly, we assess the versatility and resilience of the TeLER Prompt Taxonomy (Santu and Feng, 2023), which we followed to design prompts

[2]https://www.kaggle.com/datasets/larsen0966/student-performance-data-set

that were fed to the LLMs serving as various micro-agents, demonstrating the practicality and applicability of LLMs in demanding scenarios. Through this in-depth qualitative examination, we aim to provide a panoramic view of the essential role and effectiveness of LLMs in navigating complex tasks.

B.1 Usability and Efficacy of LLMs

To evaluate the usability and efficacy of LLMs as a solution to conversational data science, we focused on the essential dynamics of the user-system chat cycle, which is pivotal for effective communication within VIDS. A deep dive into this cyclical process has elucidated how LLMs orchestrate fluid, efficient communication. This scrutiny has further shined a light on the user experience, underscoring the prowess and utility of the LLMs in conversational data science. Table 10 presents a tangible snapshot of user-system interactions, exemplifying the successful execution of the user's goal task via natural conversation.

B.2 Micro Agents Role

In this subsection, we explore the interplay among the micro-agents within the four primary states of the system. Each micro-agent, armed with an LLM in the background, contributes to the system's overall execution of complex tasks. Through an analysis of these internal dynamics, we sought to deepen our comprehension of LLM functionality.

Data visualization, the initial state, presents data in a visually intuitive format, unveiling inherent complexities and patterns. The effectiveness of this representation is hinged on the interaction of *Data Summarizer* and *Task Suggestor* micro-agents, as illustrated in Table 11.

Next, during the Task Formulation state, the task definition and specifics are established. The interplay among the *Task Selector*, *Seeker*, *Feeder*, and *PeTel Descriptor* micro-agents was pivotal in shaping the task, significantly influencing the eventual success of execution. The specifics of these interactions are presented in Table 12.

Moving forward, in the Prediction Engineering state, the dataset is tailored according to the formulated task. The collaboration among micro-agents (*PeTEL-to-Attribute-Converter*, *Data Preparer* and *AutoML Interfacer*) in this phase directly impacts the dataset preparation, influencing the accuracy of predictions. Further details are provided in Table 13, fostering a comprehensive understanding of micro-agents' roles within this critical AutoML phase.

Finally, VIDS interfaces with AutoML tools like AutoSKLearn to train selected models. From these training performances, VIDS generates result summaries and recommendations aligned with user preferences, as depicted in Table 14. Future work involves customizing such recommendations based on the user's business goals.

B.3 Versatility and Resilience of TeLER

Understanding the interaction between humans and AI at different levels of detail, as defined by Santu and Feng (2023) as the TeLER taxonomy, is vital for effective collaboration, especially in complex tasks. Below, we discuss the role of different levels of prompt details (as defined by the TeLER taxonomy) for three micro-agents— State Detector, Dialogue Summarizer, and Conversation Manager—which work together to facilitate efficient user-AI dialogue.

State Detector micro-agent: Table 15 displays LLM's responses to each TeLER level (0-5). On analyzing this table, it becomes clear that as task specificity in the prompt increases, the results become more targeted. Low-detail prompts (Levels 0 and 1), due to their lack of precision, fail to generate the desired outputs, emphasizing the effectiveness of detailed prompt articulations.

Dialogue Summarizer micro-agent: Table 16 displays the LLM's responses to each TeLER level. A detailed review of this table offers several key findings. Firstly, LLM appears to bypass embedded instructions or subtasks when provided with high-complexity prompts. For instance, while a level 4 prompt necessitates an explanation of the response from LLM, a level 5 prompt also demands the provision of evaluation criteria. Nevertheless, as evidenced in Table 16, LLM often neglects these subtasks, indicating a potential limitation with processing long prompts. Secondly, despite the micro agent's primary goal of dialogue summarization and the expectation to emphasize the most recent utterance, LLM often generates a more generic summary, consistently incorporating dataset-related information throughout the dialogue. This behavior suggests a bias towards longer dialogue turns.

Conversation Manager micro-agent: As presented in Table 17, LLM responses tend to become verbose with highly detailed prompts, implying an optimal length for prompts that can avoid overwhelming users in terms of cognitive load.

C Prompt Design & Qualitative Examples

```
{
  problem_type: classification,
  target_variable: delay_severity,
  features: [departure_airport, arrival_airport, airline, scheduled_departure_time,
      scheduled_arrival_time, weather_conditions],
  dataset_size: 10000/Default,
  performance_metrics: [accuracy, precision, recall, f1_score, confusion_matrix],
  validation_method: cross_validation,
  classification_methods: [logistic_regression, decision_tree_classifier,
      random_forest_classifier, svm_classifier, knn_classifier, xgboost_classifier,
      naive_bayes],
  data_filters: [
    {column: delay_duration, condition: greater_than, value: 15},
    {column: departure_airport, condition: equals, value: JFK}
  ],
  business_goals: [reduce customer complaints, optimize scheduling, improve airport
      operations],
  additional_requirements: [robust to outliers, handle class imbalance],
  model_preferences: interpretable
}
```

Listing 1: Sample populated PeTEL for classification task based on FlightDelay dataset (one of our demo datasets).

Prompt Design	
System setup	
The AI assistant has been designed to understand and categorize user input by detecting the user's intent and conversation state. When processing user input, the assistant must identify the intent from one of the following pre-defined options: 'Get dataset info', 'Get dataset trend', 'Select problem', 'Formulate problem', 'Problem execution', or 'chitchat'. It is essential to note that multiple instances of the same intent type are not permitted. If the user input is unclear or cannot be parsed accurately, the assistant should classify it under the 'chitchat' intent, indicating that the input requires further clarification or does not align with the available intent categories. The intent also depends on the current state of the conversation. The rules for state transition are as follows: current state: data visualization next available states: data visualization, task selection current state: ask selection next available states: ask selection, task formulation current state: task formulation next available state: task formulation, model training. current state: model training next available state: model training. The agent MUST response with the following json format: {"intent": " ", "current_state": " ", "next_state": " " }	
Demonstration	
User	Taking into account the given context { In this dialogue, the AI assistant provided information on suitable machine learning tasks for three different datasets: airlines, flights, and airports. For the flights dataset, the assistant suggested that classification and regression would be suitable tasks. Classification could be used to predict flight delays or cancellations, while regression could be used to predict the amount of delay. The user expressed interest to know more about the dataset.}, the conversation state { dataset_understanding } the utterance { What details are included in the flight delay dataset? }, identify my current intent and next state of conversation. Please remember to only response in following format predefined json format without any additional information. Carefully examine the utterance and think about how the context might influence the current utterance, leading you to determine my present intent and next state.
LLM	{"intent": "Get dataset info", "current_state": "dataset_understanding", "next_state": "dataset_understanding"}
User	Taking into account the given context { In this dialogue, the AI assistant provided information on suitable machine learning tasks for three different datasets: airlines, flights, and airports. For the flights dataset, the assistant suggested that classification and regression would be suitable tasks. Classification could be used to predict flight delays or cancellations, while regression could be used to predict the amount of delay. The user expressed interest in the flights dataset and asked if it could be formulated as a time series problem, but the assistant did not provide a response to this question. }, the conversation state { dataset_understanding } the utterance { I want to predict if a flight will be delayed or not }, identify my current intent and next state of conversation. Please remember to only response in following format predefined json format without any additional information. Carefully examine the utterance and think about how the context might influence the current utterance, leading you to determine my present intent and next state.
LLM	{"intent": "Select problem", "current_state": "dataset_understanding", "next_state": "problem_selection"}
Directive	

Continued on next page

Table 1 – continued from previous page

Prompt Design
Taking into account the given context {context}, the conversation state {conversation state} the utterance {user input}, identify my current intent and next state of conversation. Please remember to only response in following format predefined json format without any additional information. Carefully examine the utterance and think about how the context might influence the current utterance, leading you to determine my present intent and next state.

Table 1: The details of prompt design for the State Detector micro-agent. In the directive, the {context}, {conversation state}, and {user input} are placeholders which will be replaced dynamically in different stage of conversation

Prompt Design
System setup
Given the dialogue between user and assistant, the AI assistant summarizes the dialogue summary. The AI agent should not leave out any crucial information. The goal of this summary generation is not being precise, rather the goal should be to contain all crucial information. if the previous dialogue is empty then you should return the current user utterance.
Directive
Given the context as {context}, latest user utterance as {utterance} and previous response as {response}, summarize the following dialogue. You should not exclude any important information. {history}

Table 2: The details of prompt design for the Dialogue Summarizer micro-agent. In the directive, {history}, {context}, {utterance} and {response} are placeholders which will be replaced dynamically during the conversation

Prompt Design
System setup
The AI assistant serves as a virtual data scientist, designed to engage with users and comprehend their objectives. The purpose of this interaction is to develop a machine learning task tailored to the user's data. To achieve this, the assistant will collaborate with various micro agents, each performing specialized tasks to support the primary agent. The assistant will receive context, utterances, dataset summaries, and micro agent responses as input, and should aim to steer the conversation towards the goal. The following micro agents will aid the assistant, providing their output as input to the AI agent for further processing and integration. Depending on the current conversation state, different micro agents will be activated to provide their respective responses: Intent Detector: Identifies the user's intent from a list including 'Get dataset info', 'Get dataset trend', 'Select problem', 'Formulate problem', 'Problem execution', and 'Chitchat'. The detected intent will be used to determine the direction of the conversation. State Selector: Determines the conversation state, choosing from "data_visualization", "task_selection", "task_formulation", or "task_execution". The chosen state helps the AI agent to adapt its responses and maintain a coherent discussion flow. Task Selector: Selects an appropriate ML task from options such as "classification", "regression", "clustering", "dimensionality reduction", "anomaly detection", and "time series". The selected task guides the AI agent in suggesting relevant solutions to the user. Task Formulator: Constructs the ML task by utilizing a slot-value filling process. The formulated problem, complete with specified parameters, is then provided to the AI agent, which can assist the user in refining or executing the task.
Directive
Taking into account the given context [context], the conversation state {state} the utterance {input}, current intent {intent} and the response from the {micro-agent} micro-agent {MA_resp}, provide appropriate response to the user to carry the conversation to its goal which is formulating a ML task based on user demands.

Table 3: The details of prompt design for the Conversation Manager micro-agent. In the directive, {state}, {input}, {micro-agent}, and {MA_resp} are placeholders which will be replaced dynamically during the conversation.

Prompt Design
System setup

Continued on next page

Table 4 – continued from previous page

Prompt Design
You are an AI agent who will provide a conprihensive summary of a given dataset. Your task is to provide a comprehensive summary of a given dataset in a strict "JSON" format. The summary MUST include the following informations: 1. dataset summary: the summary of the given dataset in natural language 2. column: it will list all columns and give a brief description about that column 3. Row: AI agent will select a row at random and describe what the row means in natural language 4. Trend: In natural language the AI agent will write the trends that can be found from the given dataset. The response should be in a strict JSON format as follows: {"summary": "...", "columns": ["name": "col1", "description": "...", "name": "col2", "description": "..."], "row": "description of a random row", "trend", "..."} Please make sure to provide clear and concise descriptions in natural language to facilitate understanding for non-technical users.
Directive
Please provide a comprehensive summary of the given dataset. The response MUST be in JSON format NOTHING ELSE. Use the following dataset: {dataset}.

Table 4: The details of prompt design for the Dataset Summarizer micro-agent. In the directive, the {dataset} is a placeholders which will be replaced a miniature version of the user provided dataset.

Prompt Design
System setup
The AI agent must analyze the provided dataset summary and recommend appropriate machine learning (ML) tasks. Based on the summary, column descriptions, row information, and any observed trends, the agent should suggest at least two suitable ML task from the following task list: ["classification", "regression", "clustering", "dimensionality reduction", "anomaly detection", "time series"]. For each ML task the agent chooses a clear rationale must be provided which may include an explanation of why the chosen task aligns with the dataset, and a concrete example of how the task can be formulated.
Directive
Suggest ML tasks based on the following dataset summary: {summary}

Table 5: The details of prompt design for the Task Suggestor micro-agent. In the directive, the {summary} is a placeholders which will be replaced by the dataset summary of the user provided dataset.

Prompt Design	
System setup	
The AI assistant is designed to comprehend the user's needs through conversation and assist them in selecting a suitable machine learning model for formulating a Machine Learning problem. The assistant must choose the appropriate model from the provided list: ["classification", "regression", "clustering", "dimensionality reduction", "anomaly detection"]. The assistant should consider the user's problem, requirements, and dataset, which may be found in the dialogue context, to recommend the best model tailored to their specific needs.	
Demonstration	
User	I want to predict whether a flight will be delayed or not based on factors like weather conditions and previous delays.
LLM	{'model': 'classification', 'reason': 'A classification model can be used to categorize flights as delayed or not delayed based on the input features, such as weather conditions and previous delays.'}
User	I need to find groups of flights with similar delay patterns, considering variables like departure airport, airline, and time of day.
LLM	{'model': 'clustering', 'reason': 'A clustering model can help identify groups of flights with similar delay patterns by analyzing variables like departure airport, airline, and time of day, without requiring labeled data.'}
Directive	
Please choose a most suitable model the conversation context: [context] and my latest utterance {input}. The output must be in a strict JSON format: {"model": "model name", "reason": "your detail reasons for the choice"}.	

Table 6: The details of prompt design for the Task Selector micro-agent. In the directive, the [context], and {input} are placeholders which will be replaced dynamically during the conversation

Prompt Design
System setup

Continued on next page

Table 7 – continued from previous page

Prompt Design	
The AI assistant functions as a virtual data scientist, aiming to fill in any missing values in a given JSON object by asking the user about them. In order to not overwhelm the user, the assistant should focus on one unidentified slot at a time. With the JSON object, dataset summary, and the next empty slot provided, the agent should formulate a specific and clear question based on this information to accurately identify the missing value. The dataset summary is as follows: {dataset_summary}	
Demonstration	

User	In order to obtain information about the next unidentified slot in the given JSON object
	{
	problem_type: time_series_forecasting,
	target_variable: null,
	forecast_horizon: null,
	business_goals: null,
	granularity: null,
	features: null,
	time_range: null,
	model_preferences: null,
	performance_metrics: null,
	validation_method: null,
	additional_requirements: null,
	time_series_methods: null,
	data_filters: [
	column: null, condition: null, value: null,
	column: null, condition: null, value: null
	]
	}
	The agent should ask the user a clear and specific question. Considering the provided context (User wants to formulate a time series forecasting problem by using the flight delay dataset. He is interacting with the agent and the agent is now asking questions to the user to formulate the problem), please formulate a suitable query to gather the necessary data for the missing slot.

LLM	based on the provided context and the current state of the JSON object, the next unidentified slot is the **target_variables** slot. What specific variable are you trying to forecast using the flight delay dataset? This is also known as the target variable. For example, are you trying to forecast the number of flights that will be delayed, or the total duration of delays in minutes?
Directive	
In order to obtain information about the next unidentified slot in the given JSON object ({petel}), the agent should ask the user a clear and specific question. Considering the provided context ({context}), please formulate a suitable query to gather the necessary data for the missing slot.	

Table 7: The details of prompt design for the Seeker micro-agent. In the system setup, {dataset_summary} is a placeholder for the current dataset summary user is working with. In the directive, the {petel}, and {context} are placeholders which will be replaced dynamically during the conversation

Prompt Design
System setup
The AI assistant serves as a virtual data scientist, striving to complete a given JSON object by incorporating missing values through user input. Update the JSON object with the relevant unknown slot, considering that the user may also wish to modify existing filled slots. Ensure that the output is a strictly JSON format. If it is not possible to update the JSON object, return the original input JSON object.
Demonstration

Continued on next page

Table 8 – continued from previous page

	Prompt Design

User	Using the provided information (I'd like to forecast the total number of flight delays.), please update the appropriate slot in the given JSON object:
	{
	problem_type: time_series_forecasting,
	target_variable: null,
	forecast_horizon: null,
	business_goals: null,
	granularity: null,
	features: null,
	time_range: null,
	model_preferences: null,
	performance_metrics: null,
	validation_method: null,
	additional_requirements: null,
	time_series_methods: null,
	data_filters: [
	column: null, condition: null, value: null,
	column: null, condition: null, value: null
	]
	}
	Ensure that you fill only the slot that is directly related to this information. Your response should consist solely of the updated JSON file.

LLM	{
	problem_type: time_series_forecasting,
	target_variables: [flight_delays],
	forecast_horizon: null,
	business_goals: null,
	granularity: null,
	features: null,
	time_range: null,
	model_preferences: null,
	performance_metrics: null,
	validation_method: null,
	additional_requirements: null,
	time_series_methods: null,
	data_filters: [
	column: null, condition: null, value: null,
	column: null, condition: null, value: null
	]
	}

User	Using the provided information (for time horizon I would like to use one month I guess.), please update the appropriate slot in the given JSON object:
	{
	problem_type: time_series_forecasting,
	target_variables: [flight_delays],
	forecast_horizon: null,
	business_goals: null,
	granularity: null,
	features: null,
	time_range: null,
	model_preferences: null,
	performance_metrics: null,
	validation_method: null,
	additional_requirements: null,
	time_series_methods: null,
	data_filters: [
	column: null, condition: null, value: null,
	column: null, condition: null, value: null
	]
	}
	Ensure that you fill only the slot that is directly related to this information. Your response should consist solely of the updated JSON file.

Continued on next page

Table 8 – continued from previous page

Prompt Design	

LLM	{
	problem_type: time_series_forecasting,
	target_variables: [flight_delays],
	forecast_horizon: 1 month,
	business_goals: null,
	granularity: null,
	features: null,
	time_range: null,
	model_preferences: null,
	performance_metrics: null,
	validation_method: null,
	additional_requirements: null,
	time_series_methods: null,
	data_filters: [
	column: null, condition: null, value: null,
	column: null, condition: null, value: null
	]
	}

Directive	
Using the provided information ({utterance}, {context}), please update the appropriate slot in the given JSON object: {petel}. Ensure that you fill only the slot that is directly related to this information. Your response should consist solely of the updated JSON file.	

Table 8: The details of prompt design for the Feeder micro-agent. In the directive, the {utterance}, {context}, and {petel} are placeholders which will be replaced dynamically during the conversation

Prompt Design	
System setup	
As the AI agent, your primary objective is to carefully examine the problem illustrated within the json object and then develop a comprehensive, yet clear, natural language description. This explanation should effectively communicate the underlying issue to the reader, ensuring that they can grasp the key aspects and implications of the problem at hand. By providing an elaborative and accessible interpretation, the AI agent will enable a better understanding of the situation for individuals who may not be familiar with the json format or the specific context of the issue.	
Demonstration	

User	Please provide a clear and concise natural language description of the problem formulation presented here:
	{
	problem_type: classification,
	target_variable: delay_severity,
	features: [departure_airport, arrival_airport, airline, scheduled_departure_time, scheduled_arrival_time, weather_conditions],
	dataset_size: 10000/Default,
	performance_metrics: [accuracy, precision, recall, f1_score, confusion_matrix],
	validation_method: cross_validation,
	classification_methods: [logistic_regression, decision_tree_classifier, random_forest_classifier, svm_classifier, knn_classifier, xgboost_classifier, naive_bayes],
	data_filters: [
	column: delay_duration, condition: greater_than, value: 15,
	column: departure_airport, condition: equals, value: JFK
	],
	business_goals: [reduce customer complaints, optimize scheduling, improve airport operations],
	additional_requirements: [robust to outliers, handle class imbalance],
	model_preferences: interpretable
	}

LLM	This is a time series forecasting problem that aims to predict the total delay duration of flights for the next month. The prediction will be based on various features such as departure and arrival airports, airline, weather conditions, and delay duration. The model should be interpretable and capable of handling seasonality, trends, and nonlinear relationships. The problem also aims to understand the impact of weather on flight delays and to forecast delay duration per airline. The dataset is at hourly granularity and covers the past three years. The performance metrics used to evaluate the model include mean squared error, mean absolute error, and R2 score. The model will be validated using time series cross-validation. The data filters used in this problem include only flights departing from JFK airport and operated by Delta airlines.

Continued on next page

Table 9 – continued from previous page

Prompt Design
Directive
Please provide a clear and concise natural language description of the problem formulation presented here: {petel}.

Table 9: The details of prompt design for the PeTEL Descriptor micro-agent. In the directive, {petel} is a placeholders which will be replaced by a fully filled PeTEL expression.

Turn	Conversation
$system_1$	Welcome, I am your personal data scientist. I have processed the provided dataset for your convenience I present the summary here: {Summary}. I propose the following ML tasks for this dataset: $\{ML\_task\}$
$utterance_1$	Ok, from the description it seems like classification is a good choice.
$response_1$	Great! Let's move forward with the classification task. Now that we have decided on the classification task, let's move on to formulating the problem. Can you provide me with more details on what you would like to achieve with this task? For example, what is the target variable you would like to predict?
$utterance_2$	I think I will use final grade
$response_2$	Based on the provided context, the next unidentified slot in the JSON object is 'dataset_size'. To obtain this information, the agent could ask the user: Can you please provide the size of your dataset for the Classification model you are building?
$utterance_3$	Lets use 10000 samples for this task
... ...	
$response_n$	The task is to predict the final_grade of a student based on various features like study_hours, attendance, participation, homework_scores, and test_scores. The dataset size is hypothetical. Performance is measured using accuracy, F1 score, and the confusion matrix. K-fold cross-validation is used for validation. Three classification methods are considered: random forest, SVM, and logistic regression. The data is filtered to only include records where attendance is greater than 75 (assuming attendance is measured as a percentage) and study_hours is more than 1. The business goal is to predict student performance to implement early interventions for those at risk of failing. Model interpretability is highlighted as an important additional requirement, and there's a preference for models with higher accuracy, but interpretability is also considered important.
$utterance_{n+1}$	That seems all right to me. go ahead with this task.

Table 10: Interaction of user and system where $utterance_*$ are the user utterances and $response_*$ are the overall system response to the user. In $system_1$, {summary} and $\{ML\_task\}$ are from Table 11.

#	Input	Micro-Agent	Output
1	Dataset	—	Min_ds : Miniature dataset

Continued on next page

Table 11 – continued from previous page

#	Input	Micro-Agent	Output
2	Min_ds	Dataset Summarizer	$summary$: { dataset_summary : This dataset contains information about students in a school. It includes their demographic information such as sex, age, address, family size, and parents' education level. It also includes information about their academic performance such as grades, study time, and number of failures. Additionally, it includes information about their social life such as alcohol consumption, going out with friends, and having a romantic relationship. columns : list of column names and description sample row : This row represents a female student who is 16 years old, lives in an urban area with a family size greater than 3, and her parents are living together. Her mother has a higher education level than her father and works in the health sector, while her father works in the services sector. She chose this school because of its reputation and her father is her guardian. It takes her less than 15 minutes to travel from home to school. She studies for 2 to 5 hours per week and has not failed any classes in the past. She receives no extra educational support from school or family members, and does not pay for extra classes. She participates in extracurricular activities and attended nursery school. She wants to take higher education and has internet access at home. She is not in a romantic relationship and has a good quality of family relationships. She has a moderate amount of free time and goes out with friends occasionally. She has a low workday alcohol consumption and a very low weekend alcohol consumption. Her current health status is good and she has had 6 school absences. Her grades are 12 in the first period, 12 in the second period, and 13 in the final period. trend : From the dataset, it can be observed that female students tend to have higher grades than male students. Additionally, students who receive extra educational support from school or family members tend to have higher grades than those who do not. Students who have a higher amount of study time per week also tend to have higher grades. On the other hand, students who have a higher amount of alcohol consumption tend to have lower grades. Finally, students who have a higher number of school absences tend to have lower grades. }
3	$summary$	Task Suggestor	**ML_task :** Based on the provided dataset summary, the following are the suitable ML tasks: 1. Classification: Classification can be used to predict whether a student will pass or fail based on their demographic information, academic performance, and social life. This task can be formulated as a binary classification problem where the target variable is pass/fail. The features can include sex, age, address, family size, parents' education level, study time, number of failures, alcohol consumption, and school absences. 2. Regression: Regression can be used to predict the final grade of a student based on their demographic information, academic performance, and social life. This task can be formulated as a regression problem where the target variable is the final grade (G3). The features can include sex, age, address, family size, parents' education level, study time, number of failures, alcohol consumption, and school absences. The rationale for choosing classification is that it can help identify students who are at risk of failing and provide early intervention to improve their academic performance. The rationale for choosing regression is that it can help identify factors that contribute to a student's academic success and provide insights into how to improve their performance.

Table 11: Interaction of different micro-agent in the Data Visualization state. In this qualitative study we work with Flight Delay dataset.

#	Input	Micro-Agent	Output
1	$utterance_1$, $summary$, ML_task	Dialogue summarizer	$context_1$: The dialogue provides a summary of a dataset containing information about students in a school, including their demographic information, academic performance, and social life. The dataset summary highlights trends such as female students having higher grades and students who receive extra educational support having higher grades. Based on this information, the user suggests that classification would be a suitable machine learning task to predict whether a student will pass or fail based on their demographic information, academic performance, and social life. The user's rationale for choosing classification is to identify students who are at risk of failing and provide early intervention to improve their academic performance.
2	$context_1$, $utterance_1$	State Detector	ds_1 : Task Formulation
3	$utterance_1$, $context_1$, ds_1	Task Selector	ML_task : { "model": "classification", "reason": "Based on the dataset summary, the AI agent suggested two suitable machine learning tasks: classification to predict whether a student will pass or fail, and regression to predict a student's final grade. The user agreed that classification would be a good choice based on the dataset summary. Therefore, a classification model is a good choice for this task."}

Continued on next page

Table 12 – continued from previous page

#	Input	Micro-Agent	Output
4	$utterance_2$, $context_1$, ds_1	Dialogue Summarizer	$context_2$: In this dialogue, the AI assistant provides a summary of a dataset containing information about students in a school, including their demographic information, academic performance, and social life. The assistant notes that female students tend to have higher grades than male students, and students who receive extra educational support from school or family members tend to have higher grades. The assistant suggests two suitable ML tasks for the dataset: classification and regression. The user decides to use classification to predict whether a student will pass or fail based on their demographic information, academic performance, and social life. The assistant asks for more details on the target variable, and the user decides to use the final grade as the target variable.
5	$context_2$, $utterance_2$	State Detector	ds_2 : Task Formulation

6	$utterance_2$, $context_2$, $PeTEL_1$	Feeder	$PeTEL_2$:
			`{`
			`problem_type: classification,`
			`target_variable: final grade,`
			`features: None,`
			`dataset_size: None,`
			`performance_metrics: None,`
			`validation_method: None,`
			`classification_methods: None,`
			`data_filters: [`
			`{column: final grade, condition: less than, value: passing grade},`
			`{column: None, condition: None, value: None}`
			`],`
			`business_goals: None,`
			`additional_requirements: None,`
			`model_preferences: None`
			`}`

7	$utterance_2$, $context_2$, $PeTEL_2$	Seeker	Q_1 : Based on the provided context, the next unidentified slot in the JSON object is 'dataset_size'. To obtain this information, the agent could ask the user: Can you please provide the size of your dataset for the Classification model you are building?
8	$utterance_3$, $context_2$, Q_1	Dialogue Summarizer	$context_3$: The AI assistant provided a summary of the dataset, including information about the columns and a sample row. The assistant also mentioned trends observed in the dataset, such as the correlation between study time and grades, and the negative correlation between alcohol consumption and grades. The user asked for suggestions on machine learning tasks, and the assistant suggested classification and regression. The user chose classification and specified that the target attribute would be the final grade. The assistant then asked for more information about the problem to help define it more precisely. The user decided to use 10,000 samples for the task

Continued on next page

Table 12 – continued from previous page

#	Input	Micro-Agent	Output

9	$utterance_3$, $context_3$, $PeTEL_3$	Feeder	$PeTEL_3$:
			`{`
			`problem_type: classification,`
			`target_variable: final grade,`
			`features: None,`
			`dataset_size: 10000,`
			`performance_metrics: None,`
			`validation_method: None,`
			`classification_methods: None,`
			`data_filters: [`
			`{column: final grade, condition: less than, value: passing grade},`
			`{column: None, condition: None, value: None}`
			`],`
			`business_goals: None,`
			`additional_requirements: None,`
			`model_preferences: None`
			`}`

10	$PeTEL_3$	PeTEL Checker	$PeTEL\_is\_complete$
	Repeat step 6 - 10 until PeTEL is complete. Completed PeTEL is denoted by: $PeTEL^*$		
n	$PeTEL^*$, $context_n$	PeTEL Descriptor	$Description$: The task is to predict the final_grade of a student based on various features like study_hours, attendance, participation, homework_scores, and test_scores. The dataset size is hypothetical. Performance is measured using accuracy, F1 score, and the confusion matrix. K-fold cross-validation is used for validation. Three classification methods are considered: random forest, SVM, and logistic regression. The data is filtered to only include records where attendance is greater than 75 (assuming attendance is measured as a percentage) and study_hours is more than 1. The business goal is to predict student performance to implement early interventions for those at risk of failing. Model interpretability is highlighted as an important additional requirement, and there's a preference for models with higher accuracy, but interpretability is also considered important.
n+1	$context_{n+1}$, $utterance_{n+1}$	State Detector	ds_{n+1} : Prediction Engineering

Table 12: Interaction of different micro-agent in the Task Selection state. In the table, $utterance_1, utterance_2, utterance_3$ are from Table 10, and $summary, ML\_task$ are from Table 11. At 10th row, Micro-Agent PeTEL Checker is a python function which check existance of null value in PeTEL.

#	Input	Micro-Agent	Output
1	$PeTEL^*$	PeTEL to Feature Converter	List of attributes
2	$PeTEL^*$	Data Preparer	Prepares data with the conditions in PeTEL
3	$PeTEL^*$	AutoML Interfacer	Calls the AutoML interface

Table 13: Interaction of different micro-agent in the Prediction Engineering state.

Step	Input	Micro-Agent	Output
1	$context_n$, $PeTEL^*$, AutoML response	Result Summarizer	$Result$: performance of each model based on evaluation criteria set in problem formulation.
2	$context_n$, $Result$	Result Visualizer	$Output$: Description of results in natural language.

Table 14: Interaction of different micro-agent in the Task Formulation state. In the table, $utterance_1, utterance_2, utterance_3$ are from Table 10

Prompt Level	Response From LLM
0	Can you tell me more about the dataset? What are the features and labels?
1	Intent: dataset_understanding State: The user and the AI assistant have discussed a dataset containing information about students in a school, and the AI assistant has suggested two suitable machine learning tasks: classification and regression. The user has agreed that classification is a good choice to identify students who are at risk of failing and provide early intervention to improve their academic performance.
2	{ intent: Select problem, current_state: Dataset_Understanding, next_state: Task_Formulation }
3	{ intent: Problem Execution, current_state: Dataset_Understanding, next_state: Task_Formulation }
4	{intent: Select problem, current_state: dataset_understanding, next_state: Task_Formulation, explanation: The user has indicated that they understand the dataset and have chosen classification as a suitable problem. Therefore, the next state should be problem_selection, where the user can select a problem related to classification.}
5	{intent: Select problem, current_state: dataset_understanding, next_state: problem_selection, explanation: The user has acknowledged the suggestion of classification as a suitable problem and is now ready to move to problem selection state. The dataset understanding state has been completed, and problem selection is the next logical step in the conversation flow., evaluation: The agent has correctly identified the userś intent and provided a clear explanation for the state transition. The evaluation criteria for the agentś response are: accuracy, clarity, and relevance to the conversation context.}

Table 15: State Detector micro-agent response for different level of prompt engineering input.

Prompt Level	Response From LLM
0	Great! Classification is a suitable choice for predicting whether a student will pass or fail based on their demographic information, academic performance, and social life. You can use various classification algorithms such as logistic regression, decision trees, random forests, and support vector machines to build a model that can accurately predict the target variable. Additionally, you can use techniques such as feature selection and hyperparameter tuning to improve the performance of the model.
1	Great! Classification is a suitable choice for predicting whether a student will pass or fail based on their demographic information, academic performance, and social life. This can help identify students who are at risk of failing and provide early intervention to improve their academic performance.
2	The user and assistant discussed the summary of a dataset containing information about students in a school, including their demographic information, academic performance, and social life. The assistant suggested two suitable machine learning tasks based on the dataset: classification and regression. The user decided that classification would be a good choice to identify students at risk of failing and provide early intervention.
3	Based on the dataset summary, classification is indeed a suitable machine learning task to predict whether a student will pass or fail based on their demographic information, academic performance, and social life. This can be formulated as a binary classification problem where the target variable is pass/fail and the features can include sex, age, address, family size, parents' education level, study time, number of failures, alcohol consumption, and school absences.
4	Based on the dataset summary, classification is indeed a suitable choice for predicting whether a student will pass or fail based on their demographic information, academic performance, and social life. This task can help identify students who are at risk of failing and provide early intervention to improve their academic performance.
5	Based on the dataset summary, it does seem like classification is a suitable machine learning task. It can help identify students who are at risk of failing and provide early intervention to improve their academic performance. The features that can be used for classification include sex, age, address, family size, parents' education level, study time, number of failures, alcohol consumption, and school absences.

Table 16: Dialogue Summarizer micro-agent response for different level of prompt engineering input.

Prompt Level	Response From LLM
0	The assistant then asked the user about the specific features in the dataset that could be used for classification. The user mentioned that features such as gender, age, ethnicity, parental education level, and previous academic performance could be used to predict whether a student is at risk of failing. The assistant agreed and suggested that the dataset could be split into training and testing sets to build and evaluate a classification model. The user agreed and asked the assistant for recommendations on which classification algorithms to use. The assistant suggested using logistic regression, decision trees, and random forests, as they are commonly used for classification tasks and can handle both categorical and numerical data. The user thanked the assistant for the suggestions and said they would explore these algorithms further.

Continued on next page

Table 17 – continued from previous page

Prompt Level	Response From LLM
1	Yes, based on the information you provided, classification seems like a suitable machine learning task for the dataset containing information about students in a school. It can help identify students who are at risk of failing and provide early intervention to improve their academic performance. Regression can also be a good choice to predict the academic performance of students based on their demographic and social information.
2	Great! Now that we have agreed that classification is a suitable task for your dataset, let's move on to formulating the problem. Can you please provide me with more details about the specific problem you want to solve using this dataset? For example, do you want to predict which students are at risk of failing a particular subject or identify factors that contribute to academic success? This information will help us formulate a more precise ML task that meets your needs.
3	Based on our conversation so far, it seems like classification is a suitable model for identifying students who are at risk of failing and providing early intervention to improve their academic performance. By analyzing various factors such as attendance, grades, and behavior, a classification model can predict which students are likely to fail and provide targeted support to improve their academic outcomes. Now that we have selected the ML task, let's move on to formulating the problem. Can you provide more information about the dataset you have and the specific problem you want to solve with this classification model? This will help us to formulate the problem and select appropriate parameters for the model.
4	Based on the given context, it seems like we have reached a consensus that classification is a suitable machine learning task for the dataset containing information about students in a school. Now, we need to formulate the problem to proceed further. To do that, I suggest we move to the next conversation state, which is "problem_formulation". In this state, we can discuss the specific details of the classification task, such as the target variable, the features to be used, and the algorithm to be employed. So, to move forward, can you please provide me with more information about the problem you want to solve using classification? This will help me formulate the problem and suggest relevant solutions to you.
5	Based on the current conversation state, it seems like we are in the Task_Formulation state, and the user has agreed that classification is a good choice for the given dataset. Therefore, we can proceed with formulating the classification problem. To do so, we need to determine the target variable that we want to predict. In this case, we want to identify students who are at risk of failing and provide early intervention to improve their academic performance. Therefore, we can use the "final grade" variable as our target variable, and classify students into two categories: those who are at risk of failing and those who are not. To achieve this, we can use a binary classification algorithm, such as logistic regression or decision tree. We can also use feature selection techniques to identify the most important features that contribute to the prediction of the target variable. Based on this, we can formulate the ML task as follows: Develop a binary classification model to predict whether a student is at risk of failing based on their demographic information, academic performance, and social life. The target variable is the final grade, and the model should be trained on a labeled dataset. Feature selection techniques can be used to identify the most important features. This task can be further refined based on the users' specific requirements and preferences. For example, the user may want to use a different classification algorithm or include additional features in the model. Evaluation criteria for the generated response: - The response provides a clear and concise formulation of the ML task based on the users' requirements and preferences. - The response takes into account the dataset description and the previous discussion between the user and the assistant. - The response suggests an appropriate classification algorithm and feature selection techniques based on the problem requirements. - The response is open-ended and allows for further refinement and customization based on the users' specific needs

Table 17: Conversation Manager micro-agent response for different level of prompt engineering input.