PharmaSUG 2025
Paper HT-190

AI Performing Statistical Analysis: A Major Breakthrough in Clinical Trial Data Analysis

Toshio Kimura, Arcsine Analytics
Siqi Wang, Arcsine Analytics
Songgu Xie, Regeneron Pharmaceuticals
Weiming Du, Alnylam Pharmaceuticals

ABSTRACT

Statisticians and statistical programmers in the pharmaceutical industry are dreaming about AI performing analysis, but this has not yet been demonstrated. Companies are only experimenting with AI generated code. AI is not integrated into system workflows or connected to analysis infrastructure … until now.

This paper will demonstrate a proof of concept (PoC) showing AI performing statistical analysis. The chat interface will be used to initiate, parameterize, and execute statistical analysis. To achieve this, we mapped out a detailed workflow of the current process including the dialog between the statistician and statistical programmer. After which, we then developed an AI solution to operationalize this workflow.

The solution uses Python as the backbone to seamlessly integrate ChatGPT and runs SAS as the analysis engine. AI generated responses are used in downstream processing, and with the AI system connected to SAS, it will directly execute the analysis and deliver results back to the user.

This accomplishment represents a major breakthrough and a significant milestone in the use of Generative AI for clinical trial data analysis in the pharmaceutical industry.

INTRODUCTION

We are still in the early days of AI. While statisticians and statistical programmers in the pharmaceutical industry talk and dream about AI performing statistical analysis. The statistics and statistical programming departments in the pharmaceutical industry are still mostly in the evaluation phase or at best copying and pasting AI generated code with heavy human editing. AI is not yet integrated into system workflows, nor has it been hooked up to the underlying analysis infrastructure … until now.

This paper will demonstrate a proof of concept (PoC) showing AI performing statistical analysis. The chat interface will be used to initiate, parameterize, and execute statistical analysis. AI will

evaluate the chat inputs (prompts), and the AI evaluation (response) will be used directly for downstream processing. The AI system is connected to a database and an analysis engine (SAS), and therefore can directly execute the analysis through the chat interface and present the results back to the user.

This accomplishment represents a major breakthrough and a significant milestone in the use of Generative AI for analyzing clinical trial data in the pharmaceutical industry.

SCOPE

For this PoC, the scope was primarily focused on parameterization and execution. Specification and parameterization are the key first steps in any automation process. This step is also where back and forth conversation (or a chat) between the statistician and statistical programmer takes place, so this would be the logical starting point for an AI chatbot. The collected parameters will then be executed by the analysis engine such as SAS.

THE CURRENT HUMAN-BASED WORKFLOW

How do we make the leap from AI hype to reality? The answer is that it was not a leap. It was a step-by-step process grounded in deep domain experience and expertise of the clinical trial analysis process. The solution begins with a clear articulation of the current human-based workflow. We took a deep dive into the minds of statistical programmers in the pharmaceutical industry in order to develop a systematic workflow that reflects their internal thought process. Only after establishing this systematic workflow, can we then think about integrating AI into the solution.

We broke down the human-based structured workflow into the following 5 steps:

Step 1: Identify the statistical analysis method

The very first step is to identify the statistical analysis method that is being requested. The human statistical programmer will essentially compare the analysis request from the statistician to their internal knowledge of different statistical analysis methods.

Step 2: Ask about analysis specific parameters

After identifying the statistical analysis, the statistical programmer then thinks about what information is necessary to perform that specific statistical analysis method. The statistical programmer will think about all the parameters that are needed to run the analysis and identify if the initial request from the statistician already contained that information. For each parameter, the statistical programmer will also check to make sure that the provided values are valid.

Step 3: Ask for additional information

If the initial request did not provide a complete specification for all of the parameters, the statistical programmer will ask the statistician for the missing information. When the missing information is provided, the statistical programmer will check to make sure that the provided values are valid.

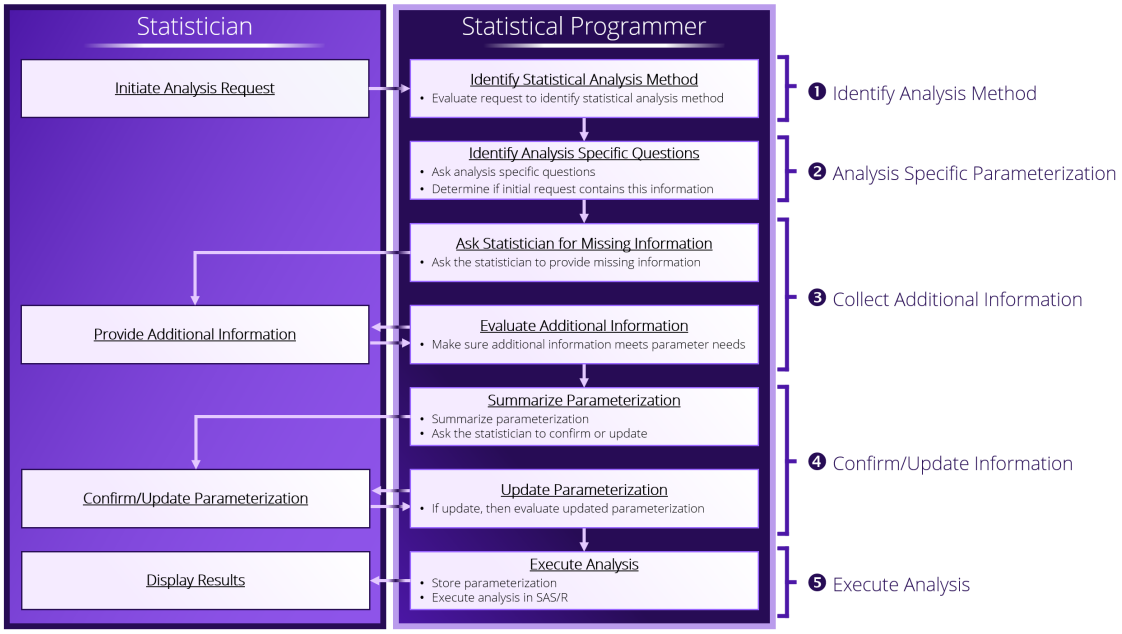
Step 4: Confirm/update information

After all of the information has been gathered, the statistical programmer will ask the statistician to either confirm or update the parameterization. If the statistician confirms, the statistical programmer will execute the analysis.

Step 5: Execute analysis

The statistical programmer will use the gathered information and execute the statistical analysis. The statistical programmer will then show the analysis results to the statistician.

The final workflow is as follows:



THE TECHNOLOGY STACK

Before delving into the AI workflow, we will first consider the technologies that we will be used in the AI solution. There is a wide constellation of technologies that can be used to implement AI solutions. There are various options for each part of the solution. While we list the primary technology that we used, our AI solution is technology agnostic. For example, we can easily select a different LLM model, a different database or a different analysis engine.

For the PoC, we used Python as the backbone for our solution (as stated earlier other languages could be used as well). The components of the technologies are as follows:

1) Database
    - Technology: PostgreSQL
    - Usage: We use the database to store information used throughout the application. Examples include the following: statistical analysis methods, statistical analysis method specific parameters, etc.

2) LLM
    a. Technology: OpenAI ChatGPT
    b. Usage: We use the ChatGPT API to send the prompt and receive the response. While we landed on ChatGPT, we also used Google Gemini, Mistral AI, Llama (through Groq) and others as part of our evaluation process.

3) Analysis Engine
    a. Technology: SAS (integrated through SASPy)
    b. Usage: We use the analysis engine to get dataset metadata and for analysis execution. Since the solution is analysis engine agnostic, R or other statistical programming languages can be used as the analysis engine as well.

In each of the steps below, we will explain the role of the various technologies used and how we leveraged them.

THE AI WORKFLOW

Having explicitly articulated the human-based workflow into concrete structured steps, we then use these steps as the roadmap for the AI solution. The system will execute the same steps as the human statistical programmer; however, each step will be operationalized by the AI system.

Step 1: Identify the statistical analysis method

To identify the statistical analysis method, the human statistical programmer evaluates the analysis request and matches that with the statistical analysis methods that the statistical programmer knows.

For the AI system to execute the same step, the following steps will be performed:

1) Intercept the prompt and supplement it with additional data (in this case, a list of known statistical analysis methods) and specific instructions, then submit the updated prompt to the LLM.

2) Retrieval-augmented generation (RAG): We used a manual RAG type process where we supplemented the initial prompt with both additional data and instructions.

    a. Data: We pull the data from the database with regards to the statistical analysis methods.
    b. Instructions: We add instructions as to what values to return, so that the LLM response can be used in downstream processing.

Figure 1 below shows the flow diagram of Step 1 to identify the statistical analysis method. In the example below, the LLM is evaluating what statistical analysis method is being requested. The user prompt (“I would like to run a MMRM analysis.”) is being compared to a list of available statistical analysis methods (Descriptive statistics, ANCOVA, MMRM, MI, etc.). The LLM is instructed to only return the name of the statistical analysis method (“MMRM”). We do not want the LLM to return value such as “The user is requesting to perform a MMRM analysis”, since we would then have to further parse the returned string.

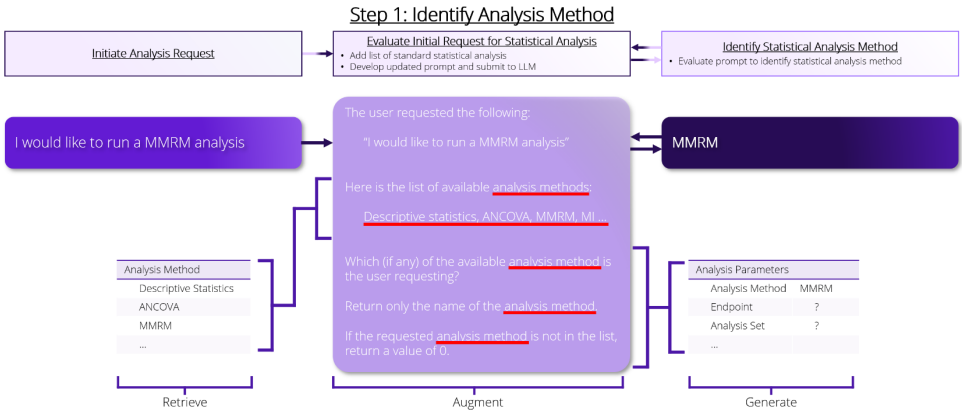


Figure 1: Flow Diagram of Step 1 to Identify the Statistical Analysis Method

Step 2: Ask about analysis specific parameters

The statistical programmer then evaluates the initial request with regards to the parameters that are needed to run the analysis. For each parameter, the statistical programmer will also check to make sure that the provided value is valid.

For the AI system to execute the same step, the following steps will be performed:

1) Analysis specific parameters: We will use the statistical method that the LLM identified and pull the analysis specific parameters from the database. We will then loop through each parameter to evaluate if the initial request contains information with regards to that parameter. Some parameter values will be defaulted (if the request does not specifically mention this parameter, then a default value will be used).

2) For each parameter, ask parameter specific questions: Intercept the prompt and supplement it with additional data (in this case, a list of values relevant to that specific parameter) and specific instructions, then submit the updated prompt to the LLM.

3) Retrieval-augmented generation (RAG): We used a manual RAG type process where we supplemented the initial prompt with both additional data and instructions.

    a. Data: The data source will depend on the parameter. Using endpoints as an example, the data will come from the ADAM datasets. Using SASPy, we will execute a SAS macro that will create a list of PARAM/PARAMCD from all of the datasets in a given folder location. For other parameters, it might come from the database.
    b. Instructions: We add instructions as to what values to return, so that the LLM response can be used in downstream processing.

Figure 2 below shows the flow diagram of Step 2 to ask analysis specific questions. The example below shows the part of the process where the system is seeking information regarding endpoints (a parameter that is required to run MMRM analysis). The endpoint data will come from the PARAM/PARAMCD values of ADAM datasets. We will loop through all of the

analysis specific parameters in the same manner. For other parameters, the data may come from the database.

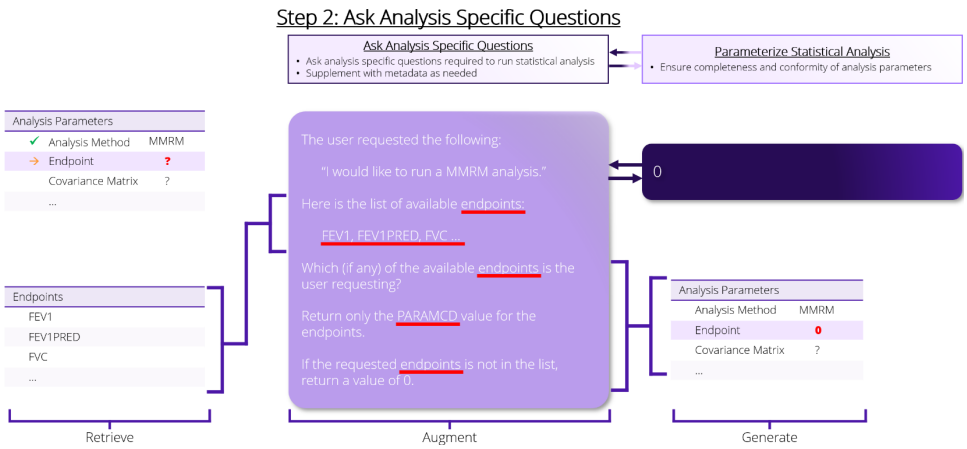


Figure 2: Flow Diagram of Step 2 to Ask Analysis Specific Questions

Step 3: Ask for additional information

If the initial request did not provide a complete specification for all parameters, the statistical programmer will ask the statistician for the missing information.

For the AI system to execute the same step, the following steps will be performed:

1) After evaluating the initial request, the system will now have a list of missing parameters. For each missing parameter, the system will ask the user for the missing information.

2) The AI system will systematically ask the user for the missing information. The same retrieval-augmented generation (RAG) process described in Step 2 will be used in evaluating the additional information provided by the user for each parameter.

Figure 3 below shows the flow diagram of Step 3 to ask for additional information. In the example below, we ask the user to provide information about the covariance matrix which was not specified in the initial request. Once the user provides this information, the same process as described in Step 2 will be used to evaluate the new information.

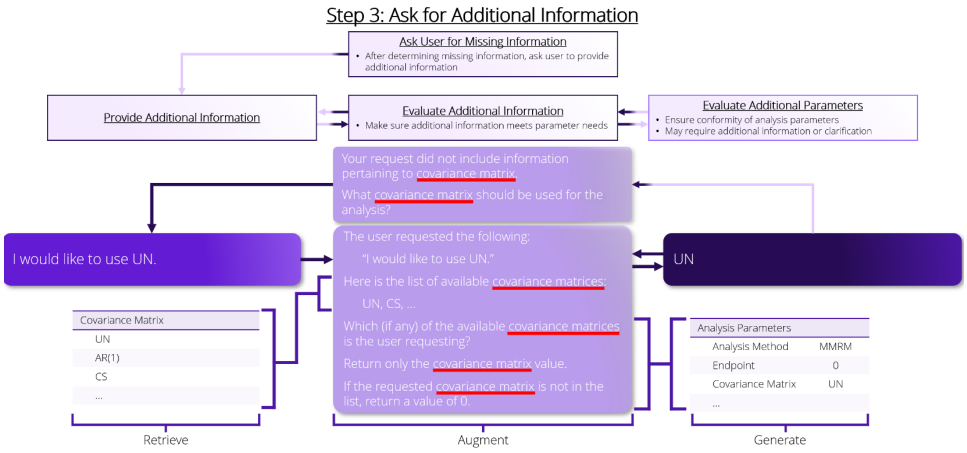


Figure 3: Flow Diagram of Step 3 to Ask for Additional Information

Step 4: Confirm/update information

After the required information are gathered, the statistical programmer will ask the statistician to either confirm or update the parameterization.

For the AI system to execute the same step, the following steps will be performed:

1) After gathering values for the required parameters, the system will ask the user to either confirm and run the analysis or to update any of the parameters.

2) If an update is required, the updated parameter will go through the evaluation process similar to Steps 3.

3) If the user confirms, then the system will proceed to the execution step.

Figure 4 below shows the flow diagram of Step 4 to ask the user for confirmation or update of the collected information. The system will first evaluate if the user is confirming and asking the system to execute the analysis or if the user would like to update the information. In the example below, the user would like to update the covariance matrix being used. After the update is made, the user is presented with the updated parameterization and is again asked if the user would like to proceed or update.

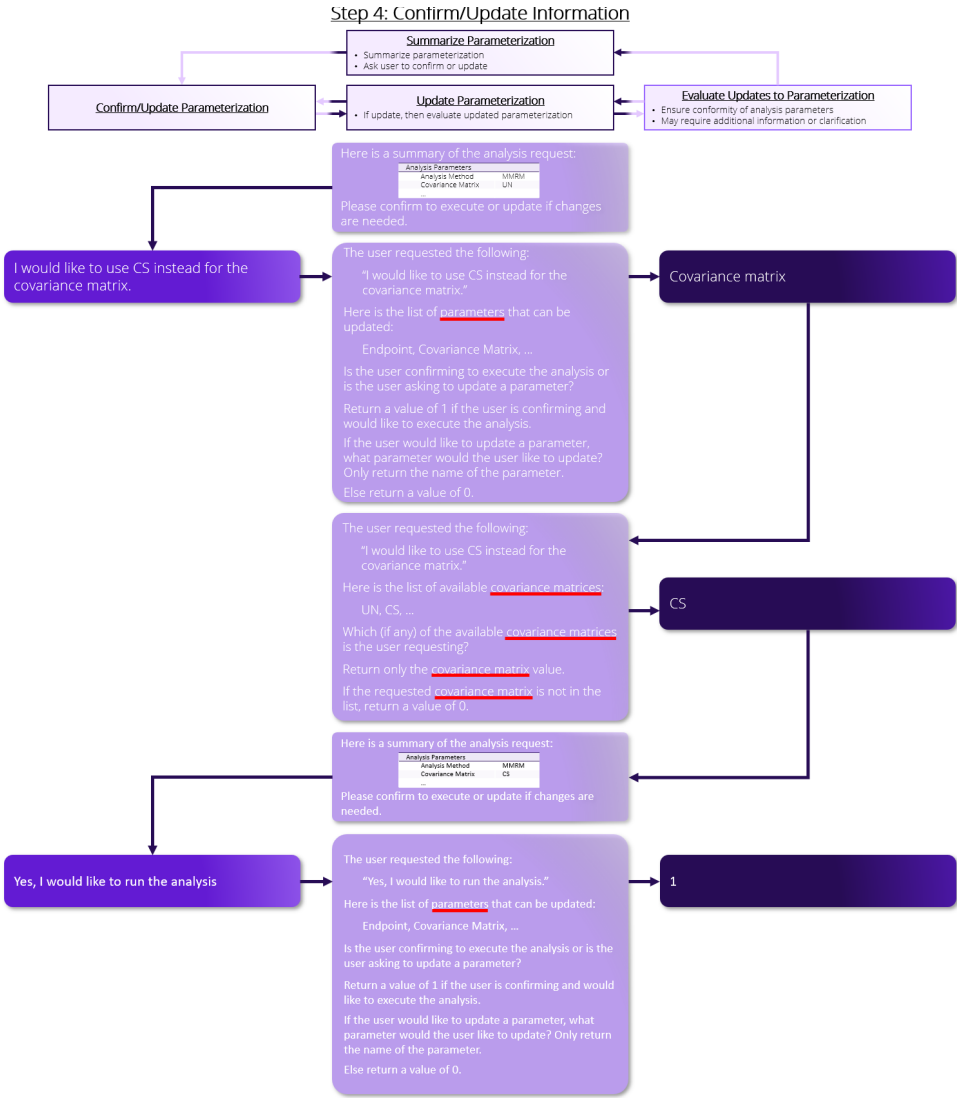


Figure 4: Flow Diagram of Step 4 to Confirm/Update Information

Step 5: Execute analysis

The statistical programmer will use the gathered information and execute the statistical analysis then provide the analysis results to the statistician.

For the AI system to execute the same step, the following steps will be performed:

1) Execute the analysis: The collected parameter values will serve as the SAS macro parameter values. The parameterized SAS macro will be run using SAS through SASPy.

2) The output will be displayed to the user

Figure 5 below shows the flow diagram of Step 5 to execute the statistical analysis and present the results to the user. The parameter values are stored as JSON which is used to make the SAS macro call. The connection to SAS is established through SASPy.

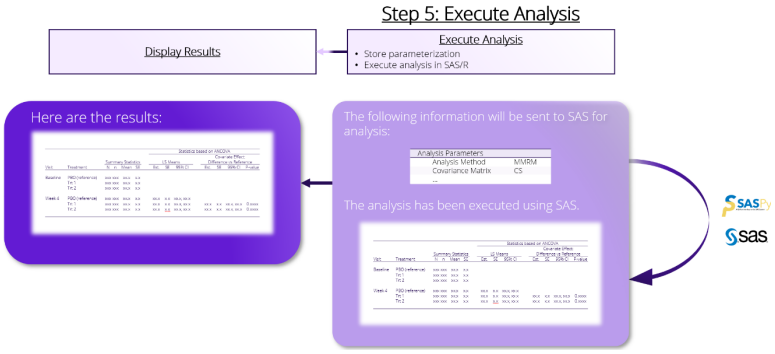
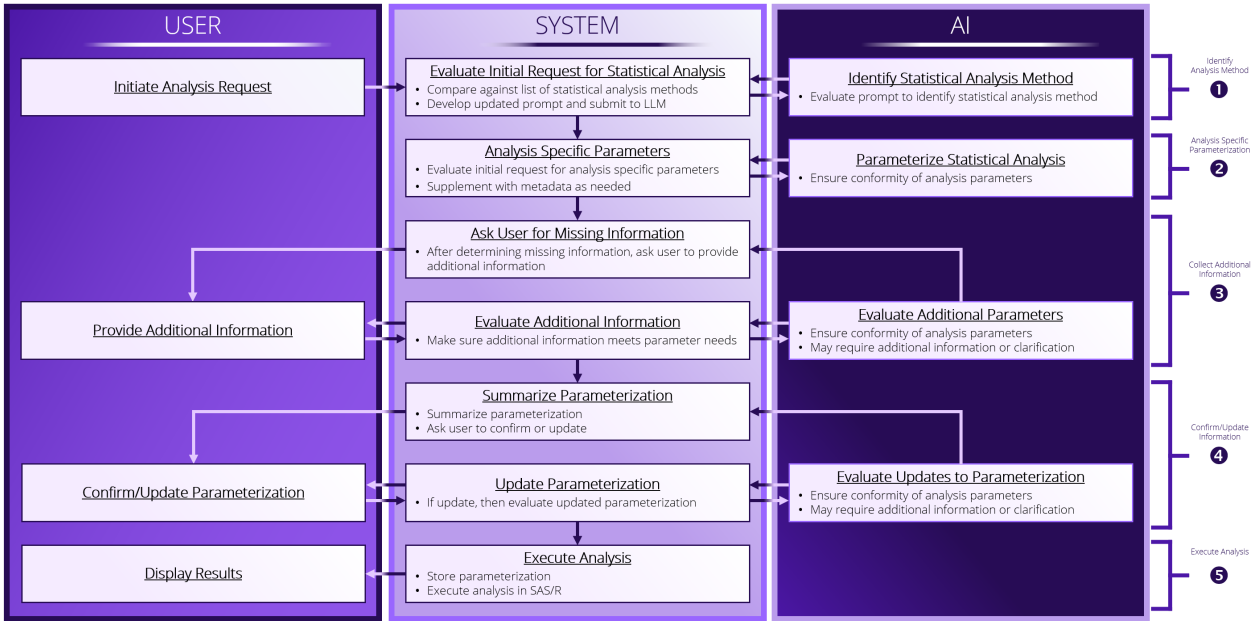


Figure 5: Flow Diagram of Step 5 to Execute Analysis

The final workflow is as follows:



AI x AI: ACUTAL INTELLIGENCE AND ARTIFICIAL INTELLGIENCE JOINING FORCES

The genesis of the AI workflow was based on human experience. The solution centers around deep domain expertise based on actual human intelligence. Only after this knowledge has been distilled into a structured workflow was AI added to supplement and develop the AI workflow. Even with AI layered into the solution, the human user is still very much in control and calls the

shots. The human user still has to answer questions to parameterize the analysis. And ultimately, the human user has to give the go ahead to run the analysis.

This solution demonstrates the power of actual intelligence and artificial intelligence joining forces to perform statistical analysis. The design of this solution goes beyond human-in-the-loop which as the name suggests keeps humans simply in the loop. Rather, this solution uses human-centric design.

CONCLUSION

Key technical insights led to the success of this PoC:

1) Understanding AI Limitations: By understanding both the power as well as the limitations of the current AI/LLMs, we are able to better leverage AI. From our previous work evaluating LLMs, we know the current limitations of AI. Based on our learnings, we purposely kept the requests made to the LLMs very simple. We are supplementing the prompts with additional information to make the task even easier.

2) Actionable Response Values from AI: Downstream processing requires actionable values. Therefore, we carefully controlled both the structure and values that the AI can output and applied checks on the response values to ensure adherence to our requirements.

Other take aways include the following:

1) Human-centric design: Domain expertise is key to leveraging AI. Before pursuing an AI solution, we must first understand the current human-based workflow. After which, we then considered integrating AI into the process. As AI gets integrated, we kept the human at the heart of the process.

2) AI x AI: Actual intelligence combined with artificial intelligence will triumph over either alone. AI is currently unable to perform the level of statistical analysis required to support the pharmaceutical industry. Therefore, actual intelligence and artificial intelligence joining forces will be the key to success.

Our successful PoC demonstrates that AI is capable of performing statistical analysis to analyze clinical trial data in the pharmaceutical industry. Continued work to expand the scope of analysis that are covered by the system and new modalities of interacting with the AI system will be added.

As we stated earlier, this accomplishment represents a major breakthrough and a significant milestone in the use of Generative AI for analyzing clinical trial data in the pharmaceutical industry.

CONTACT INFORMATION

Contact the author at:

Author Name: Toshio Kimura
Company: Arcsine Analytics
Email: toshio.kimura@arcsineanalytics.com