RESOURCES

1 Datagen

Publication: No official academic publication is linked to this repository

Github repository link: https://github.com/starpig1129/DATAGEN

DATAGEN is an advanced AI-powered data analysis and research tool that utilizes multiple specialized agents to streamline tasks such as data analysis, visualization, and report generation. It leverages cutting-edge technologies including LangChain, OpenAI's GPT models, and LangGraph to handle complex research processes, integrating diverse AI architectures for optimal performance.

In the MIRAI section, only the hypothesis generation component was extracted as-is, and it can be used directly through the interface.

Operation:

- DATAGEN first inspects the dataset schema, variable types, basic statistics, and the user-defined research objective to build a structured analytical context.

- Based on this context, the hypothesis agent uses an LLM to generate multiple domain-relevant, explicitly testable hypotheses, each defined by clear variable relationships, underlying assumptions, and suggested evaluation metrics or statistical tests.

Usage:

- It is mandatory to provide a .csv file containing the dataset (the data can be of any type).

- The input must also include some guidance, for example a prompt instructing the system to generate hypotheses based on the given dataset.

Example output:

Research Hypotheses for Analysis of Online Sales Data

Hypothesis 1: Impact of Product Category on Total Sales Amount

Statement: The product category significantly influences the total sales amount, with electronics generating the highest revenue compared to clothing, home & garden, and other categories.

Steps to Test:

1. *Perform an ANOVA analysis to compare total sales amounts across different product categories*
2. *Visualize the data using boxplots to illustrate the sales distribution within each category.*
3. *Use post-hoc tests (like Tukey's HSD) to determine which specific categories differ.*

Uniqueness and Feasibility: This hypothesis targets the potential sales strategy implications for businesses. The feasibility is high due to the structured nature of the data.

References:

- *Montgomery, D. C. (2017). Design and Analysis of Experiments. DOI: 10.1007/s00500-017-3095-9.*

2 Denario

Publication: https://arxiv.org/abs/2510.26887
Github repository link: https://github.com/AstroPilot-AI/Denario

Denario is a specialized module designed to automate the generation of novel scientific hypotheses by simulating an academic debate. It employs a competitive "Maker-Hater" multi-agent architecture to iteratively refine ideas, combined with an autonomous literature search agent that verifies novelty against real-world publications using Semantic Scholar. This ensures that generated hypotheses are unique.

Operation:

- **Contextual Ingestion:** Denario scans user-uploaded files (datasets, PDF papers, code scripts, etc.) to build a detailed "Data Description." This ensures the hypothesis is grounded in the available data and tools.
- **Maker-Hater Loop:** A "Maker" agent proposes a hypothesis based on the data context. A "Hater" agent then critiques it for feasibility, impact, and flaws. The Maker uses this feedback to refine the idea over several iterations.
- **Novelty Verification:** An autonomous "Researcher" agent takes the refined hypothesis and queries Semantic Scholar. It iteratively refines its search terms to find conflicting or overlapping literature, ultimately producing a verdict on the idea's novelty.

Usage:

- It is mandatory to upload files to serve as context (e.g., a .csv dataset, a .pdf literature review, or .py code files).
- The input must include a research goal or direction (e.g., "Generate a novel hypothesis connecting these variables").

Example output:

denario_hypothesis.md: A structured document containing the final hypothesis, including its rationale, expected variables, and experimental approach.

Title: Digital Distractions and the Myth of Multitasking: A Comprehensive Analysis

Abstract: This research seeks to investigate the multifaceted impact of digital interruptions on both perceived and actual productivity by utilizing a complete and diverse dataset encompassing various demographics and job types. We aim to explore not only the correlation between the number of notifications, daily social media usage, and productivity scores but also examine how factors such as stress levels, sleep patterns, and job type interact with digital interruptions. By employing a mixed-methods approach that includes experimental design and longitudinal analysis, we hypothesize that frequent digital interruptions create an illusion of multitasking efficiency, leading to inflated self-assessments of productivity. Furthermore, the study will delineate clear operational definitions of productivity measures and propose practical strategies for enhancing digital wellbeing and workplace efficiency in light of technological advancements.

literature_summary.md: A validation report summarizing relevant papers found during the novelty check and explaining why the generated hypothesis is considered novel (or why it is not).

3 AstroAgents

Publication:*" A Multi-Agent AI for Hypothesis Generation from Mass Spectrometry Data" by Daniel Saeedi and others (arXiv:2503.23170v1).*

Github repository link: https://github.com/amirgroup-codes/AstroAgents

AstroAgents is an AI-based tool that helps scientists generate new scientific hypotheses from complex mass spectrometry data. It works by using multiple AI agents that repeatedly analyze the data, read relevant research papers, and evaluate the generated ideas. This looped process allows the system to refine its results over time and produce more realistic and innovative hypotheses about the origins of life. AstroAgents is built on the LangChain framework and mainly uses OpenAI models, but it can also work with other models such as DeepSeek or any model compatible with the OpenAI API.

Operation:

The AstroAgents hypothesis generation tool first loads the provided literature and stores it as a markdown context file. It then creates all required agents. During each loop, the Data Analyst agent identifies important patterns and relationships in the data and passes this analysis to the Planner agent, which creates research instructions for the Researcher agents. The Researchers use the data, literature, and instructions to generate structured hypotheses. These hypotheses are then collected and refined by the Accumulator agent, which removes duplicates and merges similar ideas. The resulting hypotheses are reviewed using the Semantic Scholar API to filter out irrelevant content. Finally, the Critic agent evaluates the hypotheses using the literature and context, and this feedback is sent back to the Data Analyst for the next iteration. At the end of the process, the system produces a final hypothesis and a summary.

Usage:

· The system can run without provided data or literature, but this is not recommended.

· At a minimum, it is recommended to provide literature in PDF format to give proper context.

· Providing data is strongly recommended, either: directly in the prompt (e.g., in LaTeX format), or as a CSV file.

· Give a clear and specific instruction describing what kind of hypothesis you want to generate.

· Providing data, literature, and precise instructions helps avoid repeated or non-unique hypotheses.

· All provided PDF and CSV files are loaded, but very large files may exceed the context limit, which can cause loss of information.

Output files:

· Markdown formatted summary explaining the new ideas

· JSON formatted hypothesis

Example Output:

{
"hypothesis": [
{

"H_one": "Organic molecule preservation in carbon-rich meteorites",

"statement": "The presence of simple amino acids in carbon-rich meteorites suggests that these bodies experienced low-temperature aqueous processes that allowed organic molecules to form and remain stable over long periods.",

"key_datapoints": "Glycine, Alanine; Meteorite: Murchison (Sample ID: MUR-01)"

},

{

"H_two": "Early solar system water activity indicated by mineral signatures",

"statement": "Hydrated mineral peaks observed in mass spectrometry data indicate that liquid water was present in the early solar system, potentially supporting prebiotic chemical reactions.",

"key_datapoints": "Phyllosilicates; Meteorite: Orgueil (Sample ID: ORG-03)"

}

]

}

4 Openai Hypothesis

OpenAI Hypothesis: a simple and efficient tool for generating hypotheses with LLMs, connecting to OpenAI-compatible models via the LangChain framework. It is designed to be fast and easy to use, making it ideal for quickly generating basic hypotheses. While it works well with standard models, it can perform even better with more advanced or cutting-edge LLMs. The tool has been mainly tested with GPT-4, but it is also compatible with other models, such as DeepSeek, or any model that supports the OpenAI API. As a general-purpose hypothesis generator, it can be used for any topic and can process both context documents and data files to support its analysis.

Operation:

This tool is designed for simplicity and does not use an agent-based system. It can load PDF, markdown, and text files as context, as well as CSV files for data. The LLM is set up with a system prompt and a user prompt template. It can also perform web searches to expand its knowledge. The tool calls a web search function that returns basic information from web pages, which is then used as additional context for the LLM. Using all this information, the LLM can generate hypotheses, referencing both the provided context and web results as key data points.

Usage:

· The system can run without providing data or literature, but this is not recommended.

· Only upload compatible files: .txt, .md, .pdf, .csv.

· To improve hypothesis generation, provide all relevant data and context, while keeping in mind the model's maximum context length.

· Write a clear and specific prompt to avoid producing obvious or unhelpful hypotheses.

Output files:

· Markdown summary, explaining the research outcome

· JSON formatted hypotheses

· JSON formatted short summary and evaluation score

Example Output:

{

"hypothesis": [

{

"H_one": "Tool wear affects vibration amplitude during milling",

"statement": "Increasing tool wear leads to higher vibration amplitudes in milling operations, which can reduce surface quality and increase the risk of machine damage over time.",

"key_datapoints": "Vibration amplitude measurements; Tool: HSS End Mill (Sample ID: TM-01); Workpiece: Aluminum Alloy"

}

]

}

5 NCBI Agent

NCBI Agent is a research and hypothesis generation tool designed to explore biological and biomedical data using the NCBI Entrez API. It supports basic research tasks such as literature exploration, question answering, and hypothesis generation. The tool is built using LangChain and works with LLM models compatible with OpenAI API.

The system combines user provided documents with data retrieved from NCBI databases, stores this information in a RAG system, and uses an iterative research loop to refine results and generate hypotheses.

Operation:

When the system starts, it first reads all uploaded context files, including .pdf, .csv, .txt, and .md documents, and merges them into a single knowledge context. If a .pkl file from a previous run is provided, it is loaded as the initial RAG database, allowing the system to reuse earlier research.

All context is then split into chunks and stored in a vector-based RAG database. Based on the user's input prompt, the system generates search queries to retrieve relevant information from both the RAG database and the NCBI Entrez API. Available Entrez databases are identified, and research instructions are created by the Coordinator Agent. Any new records retrieved from NCBI are added to the RAG database, expanding the system's knowledge during runtime.

Multiple researcher processes start one by one. The Researcher Agent responds for each task based on the instructions and context returned from the RAG. The Critic Agent evaluates the collected results and can suggest additional research steps, causing the system to run multiple iterations if needed. Once the research loop finishes, the Responder Agent produces a short summary and generates one or more hypotheses based on the collected evidence.

Usage:

· The module can operate without uploaded data or documents, as it always queries NCBI.

· Providing a detailed and structured input prompt strongly improves results.

· Upload only supported file formats: .pdf, .csv, .txt, .md

· You may upload one .pkl file to reuse knowledge from previous research

· Clearly describe: the research topic, what information is important, and what type of hypothesis you want to generate.

· You may include explicit research instructions or ideas in the prompt.

· Complex research questions may take longer to complete due to multiple iterations.

Output files:

· Markdown summary (.md): A brief, easy to read summary of the research findings.

· Hypothesis file (.json): Structured hypotheses generated from NCBI data and retrieved literature.

· RAG database (.pkl): A reusable knowledge base created during the research process (recommended to save).

Example Output:

{

"hypothesis": [

{

"H_one": "BRCA1 gene mutations are associated with increased DNA repair instability",

"statement": "Mutations in the BRCA1 gene disrupt normal DNA repair mechanisms, leading to increased genomic instability and a higher risk of tumor development.",

"key_datapoints": "Gene: BRCA1; NCBI Gene ID: 672; Studies on homologous recombination deficiency"

},

{

"H_two": "TP53 gene expression levels influence cancer progression rates",

"statement": "Reduced expression or loss-of-function mutations in the TP53 gene are associated with faster cancer progression due to impaired cell cycle regulation and apoptosis.",

"key_datapoints": "Gene: TP53; NCBI Gene ID: 7157; Cancer progression studies in PubMed"

}

]

}

6 Sakana-ai

Publication: https://arxiv.org/abs/2408.06292
Github repository link: https://github.com/SakanaAI/AI-Scientist-v2

The AI Scientist v2 (from SakanaAI) is a comprehensive system designed to fully automate the scientific discovery process. It uses Large Language Models to independently generate novel research ideas, write code, execute experiments, and author full scientific manuscripts. The v2 iteration improves upon the original by removing reliance on fixed templates and introducing an agentic tree-search methodology for more open-ended exploration.

For this project, only the **hypothesis generation component** was extracted as-is, and it can be used directly through the MIRAI and chat interface to brainstorm and validate research proposals.

Operation:

- The system takes a high-level research topic from the user and "brainstorms" an initial set of research directions using an LLM (e.g., GPT-5o).
- It automatically queries the **Semantic Scholar API** to retrieve existing literature, ensuring the generated ideas are novel and not duplicative of prior work.
- The system performs multiple rounds of "self-reflection" (3 rounds by default in this integration) to refine the hypothesis, improving its clarity, feasibility, and interestingness before finalizing it into a structured plan.

Usage:

- A text prompt is required describing the research topic or area of interest (e.g., "Generate a hypothesis for a video game decompilation thesis").
- No specific file input is strictly required, and the system treats the text input as a "workshop description" to guide the ideation process.

Example output:

{

"Name": "educational_game_decompilation",

"Title": "Decompiling Games for Learning: A Novel Approach to Game Design Education",

"Short Hypothesis": "This proposal investigates whether decompiling popular video games can improve students' understanding of game design principles...",

"Related Work": "While there is significant research on video game localization... existing studies do not investigate using decompilation as a means of teaching software development...",

"Abstract": "This research proposal aims to explore the potential of video game decompilation as an innovative educational tool...",

"Experiments": [

"Conduct a workshop series where students decompile a selected video game...",

"Post-workshop, have students create their own modified versions...",

"Evaluate students' understanding through assessments..."

],

"Risk Factors and Limitations": "Potential copyright issues with decompiling proprietary games..."

}

7 Syxplain — *Hollós Roland*

Syxplain is an automated symbolic regression framework designed to bridge the gap between data-driven discovery and established scientific theory. As illustrated in the system architecture, the workflow splits raw data processing into two parallel streams: a "Computational Discovery Path" and a "Theoretical Research Path." The computational stream leverages statistical analysis (SHAP), Vision LLMs for visual reasoning, and symbolic regression algorithms (pySR) to identify the Pareto frontier of optimal mathematical formulas. Simultaneously, the theoretical stream performs deep research and literature reviews to establish theoretical priors. These pathways converge in a research synthesis phase, generating a final report that validates computational findings against existing scientific knowledge.

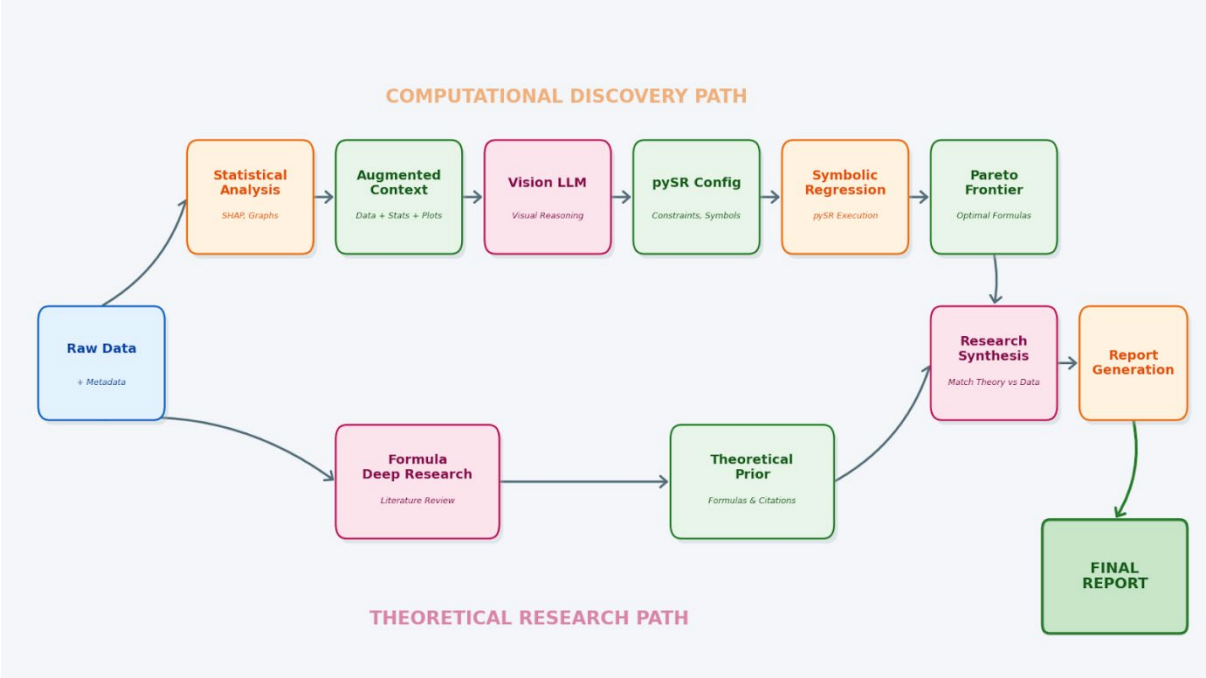

Figure 4.7.1: Syxplain System Architecture. The framework operates via two parallel streams—the Computational Discovery Path and the Theoretical Research Path—which converge in a synthesis phase to validate discovered formulas against existing scientific literature.

Syxplain also has an advanced software interface which provides a robust environment for managing this workflow. Users can initiate analyses on datasets (e.g., CSV format) with configurable depth, choosing between "Quick" and "Deep" analysis types to balance computational intensity with search exhaustiveness. The platform includes a history management system for tracking run status (e.g., "Completed") and a dedicated profile section for monitoring usage statistics. Results can be shared publicly, providing detailed views of run configurations, analysis summaries, and specific term breakdowns.

Availability

The software is free to use after registration on the following website: https://syxplain.ai1science.net/

Usage

The process involves three primary phases: Data upload, Problem description, and Reporting. Detailed information is provided below for each of these steps.

Data preparation

Following registration, users are required to upload a CSV, XLSX or XLS file. This file must contain a tabular dataset with all features represented by numerical data, and no missing values are permitted.

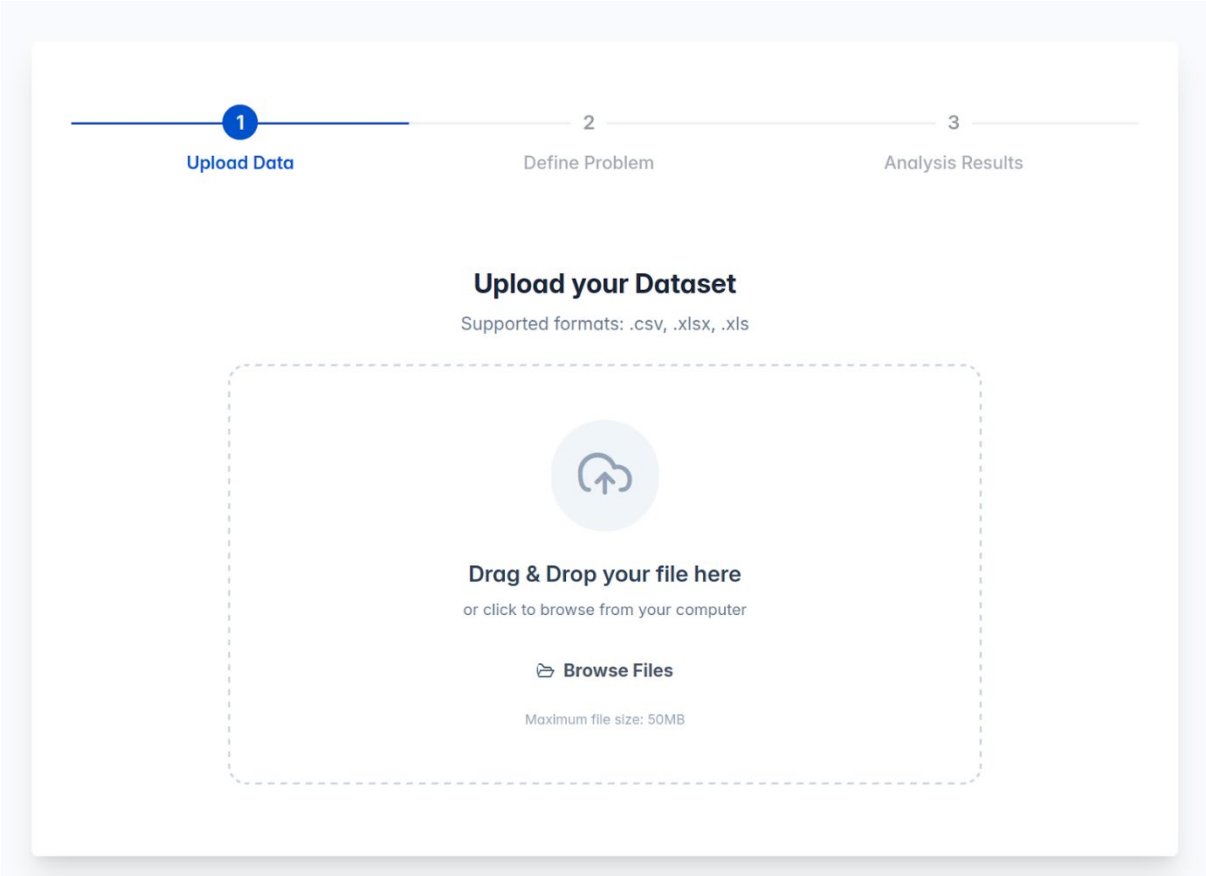

Figure 4.7.2: Syxplain Data upload

Problem Description

Providing the appropriate metadata—such as the project description, the prediction target variable, and the features in the data—is just as crucial as the data itself. Users have two methods for supplying this necessary information: either through an interactive environment (Figure 4.7.2) or via a JSON file. The JSON file is structured in the following way:

```
{
 "project_description": "TEXT",
 "target_variable": "TEXT",
 "features": {
  "feature1": "TEXT",
  "feature2": "TEXT",
```

```
  "feature3": "TEXT"
  // ... additional features
 }
}
```

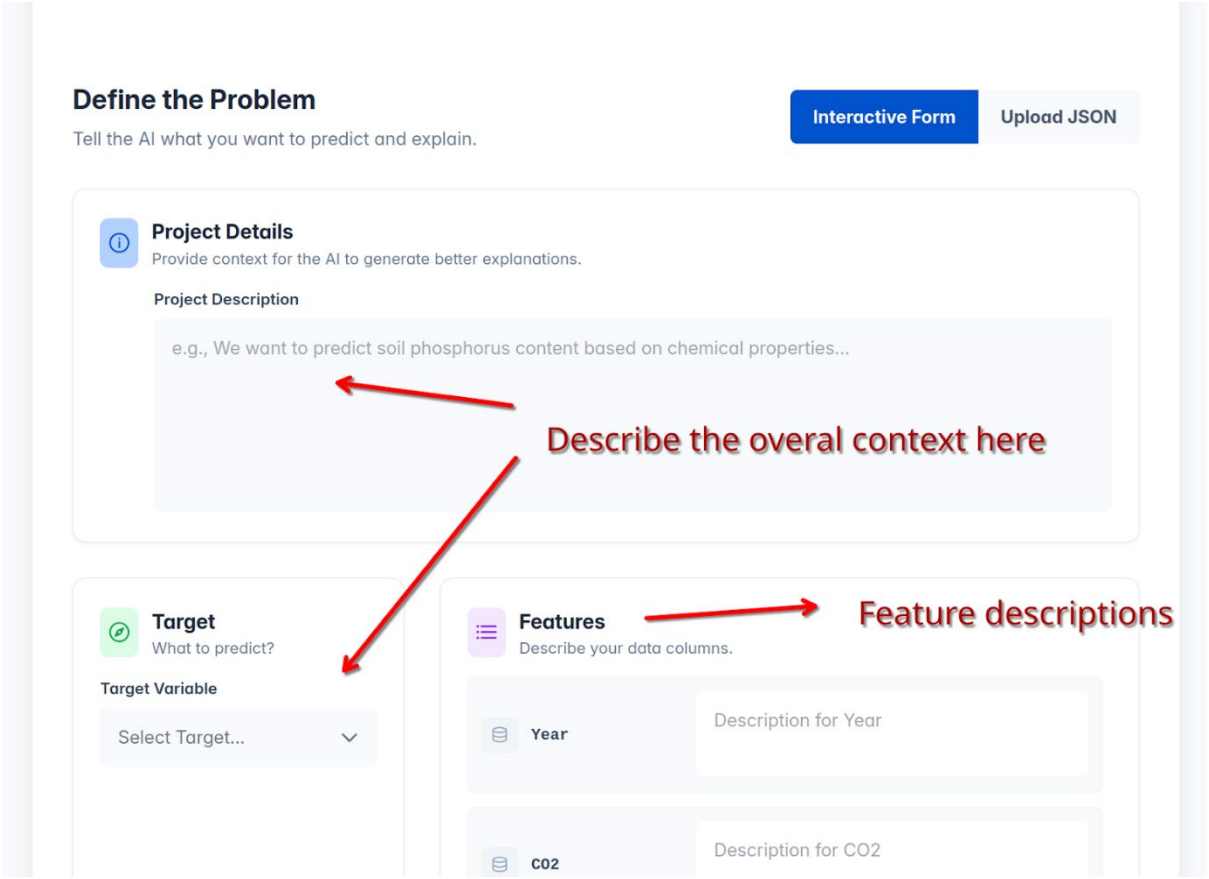

Figure 4.7.3: Syxplain Data upload

Reporting (and analyzing)

Once the problem is described, the user can begin the analysis. By default, the system offers **Quick analysis**, which prioritizes speed and cost savings by performing simple web searches instead of deep research. The user can initiate a **Deep Analysis** by clicking the chooser button (downward arrow) to change the analysis depth.

Because the analysis can be slow, it runs in the background. Users can navigate away from the screen and check the progress of the analysis in the history menu and open the report for further inspection.

From the Analysis Results, you have the option to generate a shareable link for specific findings or to export the complete report as a PDF.

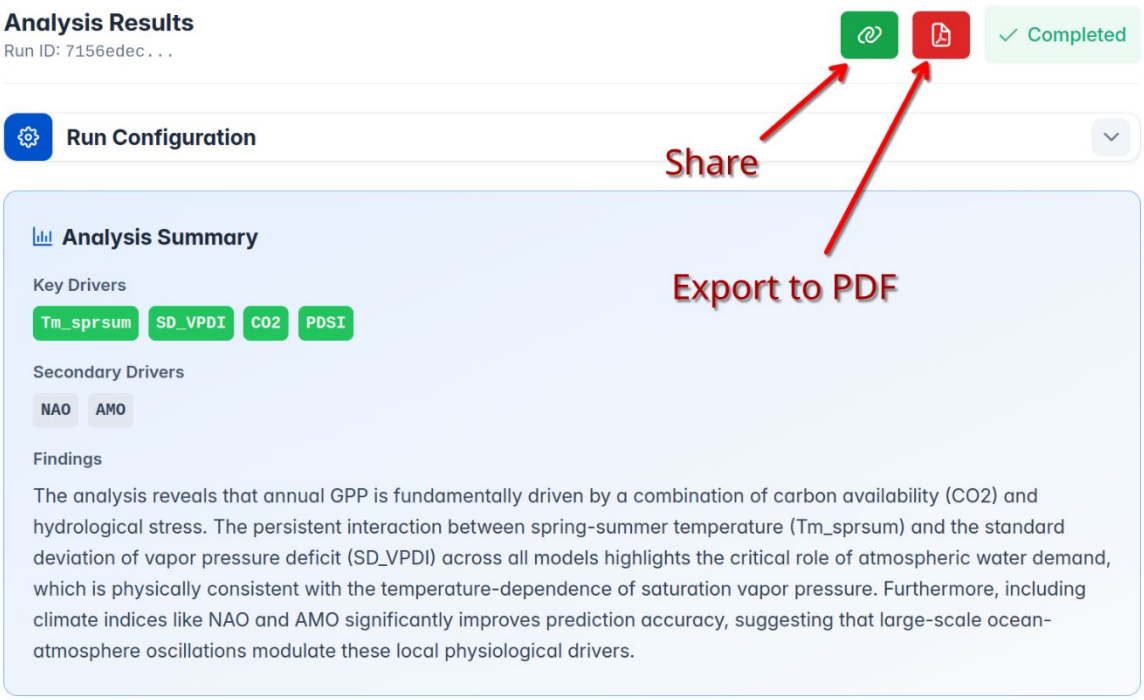

Figure 4.7.3: Syxplain Analysis result - sharing and exporting

The analysis encompasses key and secondary influencing factors, alongside the primary conclusions, complete with supporting citations. Recommended equations are also provided, arranged in ascending order of complexity.

Detailed supplementary materials are available upon request (by clicking on the show detauls button). These include a scatterplot illustrating the model's data fit, with the default loss metric (Mean Squared Error or MSE) reported—this metric will be explicitly labeled in subsequent releases. The detailed information also offers further explanations and a term-by-term breakdown. Additionally, modelled values can be exported for external analysis after accessing the 'show details' section within Syxplain.

Limitations

Syxplain's capabilities are limited because it uses pySR symbolic regression, which is designed for standard tabular data. Its primary function is to discover mathematical expressions that operate on a row-by-row basis.

Consequently, Syxplain cannot handle models that require modeling autoregressive processes, such as time series data, nor can it fit differential equations.

To overcome these limitations, users must preprocess their data. This involves adding necessary predictors like lagged features, SAVGOL derivatives, or smoothed variables.

This equally applies to all categorical or textual data, which requires necessary preprocessing and meaningful encoding.

8 Everything Else

Home (The Entry Point)

The Home page is your launchpad for quick testing and system discovery. It is designed to get you interacting with the agentic system immediately without a complex setup.

- **Agentic Chat Field:** A streamlined text area where you can send direct instructions to the AI.
- **Prompt Library:** Located directly beneath the input field, these categorized examples show you prompts for different research modules.
- **Tools Dashboard:** A visual summary of the tools currently integrated into the system, letting you see what the AI has in its "toolbox" at any given moment.

Tools (Technical Directory)

The Tools page is the reference manual for the platform's capabilities. While the Home page lists the tools, this page explains the "how" and "why."

- **In-Depth Documentation:** Each tool features a detailed breakdown of its function, the parameters it accepts, and its specific role in the scientific workflow.
- **System Transparency:** This section is vital for researchers who need to understand the underlying mechanics and data sources the agents use to generate their findings.

Chat (Advanced Research Interface)

We recommend using this page for your primary work. The Chat menu provides a robust, ChatGPT-style environment optimized for complex, multi-turn scientific discovery.

- **Multi-Module Support:** Unlike the basic input on the Home page, the Chat interface allows you to engage with the full suite of agents, not just for hypothesis generation, but also for data analysis, literature reviews, and more.
- **Contextual Depth:** The UI is designed to handle long-form dialogues, allowing you to iterate on a single project, ask follow-up questions, and refine the AI's output through ongoing conversation.

MIRAI (Parallel Discovery Engine)

MIRAI is the platform’s high-throughput research module. It is a specialized environment for users who want to compare different scientific "opinions" simultaneously.

- **One-to-Many Processing:** In the standard Chat, you work with one module at a time. In MIRAI, you submit your research description and dataset once, and the system broadcasts it to **all** available hypothesis generation agents at once.
- **Side-by-Side Comparison:** Results are returned in parallel, making it easy to identify consensus across different AI architectures or to find unique, "out-of-the-box" ideas that a single agent might have missed.

Future Roadmap (Under Development)

The following sections are currently in a **Beta/Mockup state**. They provide a glimpse into the platform's development path and will be populated with your real data in future updates:

- **My Researches:** This will become your personal database, where you can save, tag, and revisit successful hypotheses or previous chat logs.
- **About:** This section will host the project's mission statement, the methodology behind our "AI Scientist" vision, and information about the development team.

FAQ: A growing knowledge base that will address common technical questions, prompt engineering tips, and troubleshooting guides.