Agentic AI: Automating Analysis with CrewAI

Transforming Data into Actionable Insights

Ronak Shah

What is Agentic AI

Autonomous AI entities that can perceive their environment, make decisions, and take actions to achieve specific goals.

Characteristics:

Autonomy: Operates independently without constant human intervention.	**Goal-Oriented:** Designed to achieve specific objectives.	**Adaptive:** Can learn and adjust behavior based on experience and new information.	**Interactive:** Can communicate and collaborate with other agents and humans.

Why Agentic AI for Complex Tasks?

Handles Complexity: Breaks down large, intricate problems into smaller, manageable tasks.

Improves Efficiency: Automates repetitive and time-consuming processes.

Enhances Accuracy: Reduces human error and bias.

Accelerates Discovery: Speeds up the process of finding insights and solutions.

Introducing CrewAI: A Framework for Agentic Systems

An open-source Python framework for building agentic AI systems.

Features:

- **Agent Management:** Easily create and manage multiple agents with different roles and skills.
- **Task Orchestration:** Define tasks and assign them to agents.
- **Collaboration:** Enable agents to communicate and work together.
- **Tool Integration:** Integrate external tools and APIs to extend agent capabilities.
- **Process Definition:** Define the workflow and interactions between agents.

Pilot Problem

Challenge: Analyzing Mutation Annotation Format (MAF) files to identify potential therapeutic targets for cancer treatment.

Complexity: MAF files contain vast amounts of genomic data, requiring expertise in bioinformatics, cancer biology, and drug-gene interactions.

Agentic AI Solution: Automate the analysis process with a CrewAI-powered system.

CrewAI Solution: A Team of Specialized Agents

Chief Cancer Genomics Analyst: The lead agent responsible for orchestrating the analysis.

Role: Analyzes MAF data based on natural language instructions and identifies potential therapeutic targets.

Tools: Natural Language Parser, Task Delegator, MAF Summarizer, Somatic Interactions Tool, Drug-Gene Interaction Tool.

Report Agent: Compiles the analysis results into a clear and concise report.

Role: Creates a professional Markdown report summarizing the key findings and potential therapeutic targets.

Tools: Access to the outputs of the other agents.

Task Breakdown: From Instruction to Insights

Input:	Natural language instruction and MAF file path.
Parsing Task:	The Chief Analyst uses the Natural Language Parser to create a plan of action.
Summarization Task:	The Chief Analyst uses the MAF Summarizer to generate a summary of the MAF file.
Somatic Interactions Task:	The Chief Analyst uses the Somatic Interactions Tool to perform somatic interaction analysis.
Drug-Gene Interaction Task:	The Chief Analyst uses the Drug-Gene Interaction Tool to identify potential therapeutic targets.
Gather Results Task:	The Chief Analyst gathers the results from all the analysis tools.
Report Generation Task:	The Report Agent compiles the results into a Markdown report.
Output:	Markdown report with key findings and potential therapeutic targets.

Tool Integration: Extending Agent Capabilities

MAF Summarizer: Provides a concise overview of the MAF file.

Somatic Interactions Tool: Identifies significant somatic interactions between genes.

Drug-Gene Interaction Tool: Identifies potential therapeutic targets based on drug-gene interactions.

Natural Language Parser: Parses natural language instructions to create a plan of action.

Task Delegator: Delegates tasks to appropriate tools based on the plan.

Summary of traditional vs this approach

Feature	Traditional Script	CrewAI/Langchain
Flexibility	Low	High
Reasoning	None	Yes
Collaboration	Difficult	Easier
Natural Language	No	Yes
Explainability	Low	High
Learning	No	Yes
Development Effort	High (coding)	High (design)

When to use a Script vs. CrewAI/Langchain

- **Script:** Use a script when you have a well-defined analysis task that is unlikely to change, and you need maximum performance and control.
- **CrewAI/Langchain:** Use CrewAI/Langchain when you need a more flexible, adaptable, and collaborative analysis workflow, and you're willing to trade some performance for those benefits.

Complexity: For very simple analyses, a script is almost always the better choice. CrewAI/Langchain adds overhead.
Reliability: LLMs can hallucinate or make incorrect inferences. Thorough testing and validation are crucial.
Cost: LLM API calls can be expensive, especially for complex workflows.

Resources

- https://www.crewai.com/
- https://python.langchain.com/
- https://platform.openai.com/docs/models