# Multi-Agent Customer Support Automation with CrewAI

Empowering AI Agents to Deliver Exceptional Support

Speaker: Mohammad Arshad

Date: December 27, 2024

# Introduction to CrewAI

What is CrewAI?

- A framework for orchestrating AI agents to tackle complex tasks.
- Enables collaboration between specialized AI agents.

Key Features:

- Role-based agents.
- Intelligent workflows.
- Memory-enabled operations.

Objective of this workshop:

- Understand and build an automated customer support system.

# Workshop Goals

Learn the six key elements of high-performing AI agents:

- Role Playing
- Focus
- Tools
- Cooperation
- Guardrails
- Memory

Build and deploy a multi-agent system for customer support.

Gain hands-on experience with CrewAI and Gradio.

# Step 1 - Setting Up

Install required libraries:

pip install crewai crewai_tools gradio

Initialize the environment:

- Import libraries.

- Configure API keys and models.

# Step 2 - Defining Agents

Support Agent:

- Role: Senior Support Representative.
- Goal: Provide the most friendly and complete support.
- Backstory: Works at CrewAI and supports important customers.

Quality Assurance Agent:

- Role: Support Quality Assurance Specialist.
- Goal: Ensure responses meet high-quality standards.
- Backstory: Collaborates to review and improve inquiries.

# Step 3 - Defining Tasks

Inquiry Resolution Task:

- Description: Respond to customer inquiries thoroughly.
- Tools: Documentation scraper for relevant data.
- Assigned Agent: Support Agent.

Quality Assurance Task:

- Description: Review the response for accuracy and completeness.
- Tools: None (relies on expertise).
- Assigned Agent: Quality Assurance Agent.

# Step 4 - Configuring Tools

Use prebuilt CrewAI tools:

- Example: ScrapeWebsiteTool for documentation lookup.

Assign tools at:

- Agent Level: Usable across all tasks.
- Task Level: Specific to a task.

Tool Example:

```
docs_scrape_tool = ScrapeWebsiteTool(

website_url="https://docs.crewai.com/how-to/Creating-a-Crew-and-kick-it-off/"
)
```

# Step 5 - Assembling the Crew

Combine agents and tasks into a Crew:

```
crew = Crew(
    agents=[support_agent, support_quality_assurance_agent],
    tasks=[inquiry_resolution, quality_assurance_review],
    memory=True,
    verbose=2,
)
```

Enable memory for context preservation.

Use guardrails for output validation.

# Step 6 - Building an Interface

Use Gradio for user interaction:

- Input: Customer details and inquiry.

- Output: AI-generated response.

*Gradio Interface Example:*

```
interface = gr.Interface(
    fn=handle_inquiry,
    inputs=[...],
    outputs=gr.Markdown(),
)
```

# Step 7 - Running the Workflow

Execute the workflow with example inputs:

```
inputs = {
    "customer": "DeepLearningAI",
    "person": "Andrew Ng",
    "inquiry": "How to add memory to my Crew?"
}
result = crew.kickoff(inputs=inputs)
```

Review outputs from:

- Support Agent.
- Quality Assurance Agent.

# Step 8 - Deployment and Testing

- Test the interface with different inputs.
- Deploy the Gradio interface locally or on a cloud platform.
- Ensure usability and reliability.

# Key Takeaways

- Agents with defined roles improve focus and output quality.
- Tools and guardrails ensure accurate and reliable responses.
- Memory enables context-aware operations.
- Collaboration between agents enhances task handling.

# Q&A

- Open the floor for questions and discussions.
- Encourage participants to share their experiences and insights.

# Thank You

- Thank participants for their engagement.
- Share additional resources:
- - CrewAI Documentation: [Link]
- - GitHub Repository: [Link]
- - Contact Information: [Your Email/Website]
- Encourage feedback and follow-up queries.