Collaborating with AI

MGMT 675: Generative AI for Finance

Kerry Back

The Mindset Shift

- AI is not a search engine—it's a **collaborator**
- Do not try to craft the "perfect prompt"
- Instead: have a **conversation**
- Think of AI as a capable colleague, not a vending machine
- The goal is *iterative refinement*, not one-shot perfection

Conversation Over Perfect Prompts

Don't Do This

- Spend 30 minutes crafting one prompt
- Try to anticipate every edge case
- Give up when the first response is wrong
- Treat AI as a one-shot tool

Do This Instead

- Start with a rough request
- Refine based on the response
- Ask follow-up questions
- Iterate until you get what you need

Ask the AI What It Can Do

- AI systems know their own capabilities
- Ask: "What information do you need from me to do this?"
- Ask: "What are the different ways you could approach this?"
- Ask: "What should I consider before we start?"
- Let the AI **guide the conversation**—it often knows what questions to ask

"I want to build a portfolio optimizer. What do you need to know to help me?"

Explain What, Not How

- Describe your **goal**, not the implementation steps
- Let the AI suggest the approach
- You provide domain expertise; AI provides technical execution
- Focus on outcomes: what does success look like?

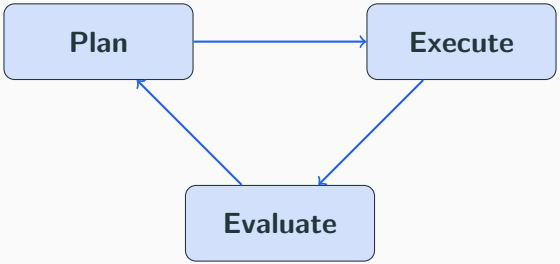
Too Prescriptive

"Write a Python function that uses scipy.optimize.minimize with SLSQP to find portfolio weights..."

Better

"I want to find the portfolio that maximizes the Sharpe ratio. Here are my expected returns and covariance matrix..."

The Plan-Execute-Evaluate Cycle



- **Plan:** Define the task, gather requirements, outline approach
- **Execute:** Let AI do the work with your guidance
- **Evaluate:** Check results, identify gaps, refine
- Repeat until satisfied

Phase 1: Planning

- Don't dive straight into execution
- First, **brainstorm with the AI**
- Ask it to help you define requirements
- Request a step-by-step plan before starting work
- Identify edge cases and potential problems upfront

"Before we start coding, can you outline a plan? What are the key steps and what could go wrong?"

Phase 2: Execution

- Work through the plan step by step
- Provide feedback after each step
- If something doesn't look right, say so immediately
- Share context: constraints, preferences, examples
- Let the AI iterate—errors are opportunities to improve

"That looks good, but we also need to handle the case where returns are negative. Can you update it?"

Phase 3: Evaluation

- Don't assume the output is correct
- **Test** the results against known cases
- Ask the AI to explain its reasoning
- Check for edge cases and assumptions
- Consider: does this actually solve my problem?

"Can you walk me through how you calculated these weights? What assumptions did you make?"

Cross-Evaluation with Other AI Conversations

- Start a **new conversation** to evaluate the output
- Even with the *same model*, a fresh context can catch errors
- Ask the new conversation to review, critique, or verify
- Different phrasing may reveal blind spots
- Consider using a *different model* for additional perspective

"I received this analysis from another session. Can you review it for errors or questionable assumptions?"

Why Cross-Evaluation Works

Same Model, Fresh Context

- No accumulated assumptions
- No confirmation bias from earlier turns
- Approaches the problem fresh
- May spot logical gaps

Different Model

- Different training emphasis
- Different reasoning patterns
- May catch model-specific blind spots
- Provides true second opinion

Provide Context

- AI works better with **context about you and your situation**
- Share your role, goals, and constraints
- Explain why you're doing this, not just what
- Context guides the AI toward relevant knowledge
- More context = more tailored responses

"I'm a finance student working on a case competition. We need to present our portfolio recommendation to a panel of judges who are industry professionals..."

Learn Through Experimentation

- There's no substitute for **hands-on practice**
- Experiment with different approaches
- Push the boundaries—see what works and what doesn't
- Each domain (coding, writing, analysis) has different patterns

The best prompters are those who have experimented the most

Resources

- Ethan Mollick: Working with AI - Two Paths to Prompting
- MIT Sloan: Effective Prompts for AI
- Five Tips for Collaborating with AI
- Collaborative Prompt Technique
- Prompt Engineering in 2025: Latest Best Practices