# Introduction to AI Agents and Large Language Models

## For Non-Experts

**Daniel Morgan**

**Friday 13 Boo!**

# What Are Large Language Models (LLMs)?

- Very large AI trained on almost all text on the internet
- Core skill: predict the next word (extremely well)
- Result: can write emails, explain concepts, translate, write code, answer questions

**Examples everyone knows:**

- ChatGPT, Claude, Grok, Gemini

Like a super-smart autocomplete that "understands" a huge amount of knowledge.

# From Chatbot to Agent: The Big Jump

**Normal LLM (Chat):**

- You ask a question
- It gives one answer
- Conversation ends (or continues linearly)

**AI Agent:**

- Has a goal
- Thinks step-by-step
- Uses tools (search, code, files, calculators...)
- Remembers previous steps
- Loops until goal achieved

$$\text{Agent} = \text{LLM} + \text{Tools} + \text{Memory} + \text{Reasoning Loop}$$

# How an AI Agent Actually Works

## Simple Loop Diagram

1. Get a clear goal (e.g., "Clean this messy CSV and summarize insights")
2. LLM thinks: "What do I know? What do I need to do first?"
3. Choose action/tool (e.g., load file, check missing values)
4. Execute $\rightarrow$ see result
5. Think again: "Did that work? Next step?"
6. Repeat until done $\rightarrow$ final output (report, cleaned data, etc.)

This is often called ReAct (Reason + Act) pattern.

# Real-World Agent Examples (Simple  Useful)

- Research agent: searches web, reads pages, writes summary report
- Data agent: loads CSV $\rightarrow$ finds problems $\rightarrow$ cleans $\rightarrow$ makes charts $\rightarrow$ explains insights
- Coding agent: writes, tests, fixes Python code for you
- Travel agent: checks flights, hotels, builds itinerary

**Today's demo:** We build a data-cleaning & EDA agent that works mostly by itself.

# Why Should Data Scientists Care?

## Agents = Your New Super Teammate

- Automate boring / repetitive work (data wrangling, basic EDA)
- Handle multi-step problems without constant supervision
- Scale analysis: one person $\rightarrow$ many parallel agents
- Evolve fast: better LLMs $\rightarrow$ instantly smarter agents

Limits exist (hallucinations, unsafe code, cost), but huge potential.

Let's build one right now!

# What is MCP? (Model Context Protocol)

- Lightweight protocol connecting LLMs to external tools/servers
- Enables reliable use of:
  - code interpreters
  - file access
  - web search
  - databases
- Keeps model context clean — heavy computation lives outside
- Widely used in Claude Code / Claude Cowork era

**Core benefit for agents**

- Turns stateless LLM into stateful, tool-equipped agent

**Simple analogy**

LLM = brain  MCP = reliable remote hands that never forget

Current status (2026):
Native tool-calling + very long context in newer models
→ custom MCP now less essential for many use cases