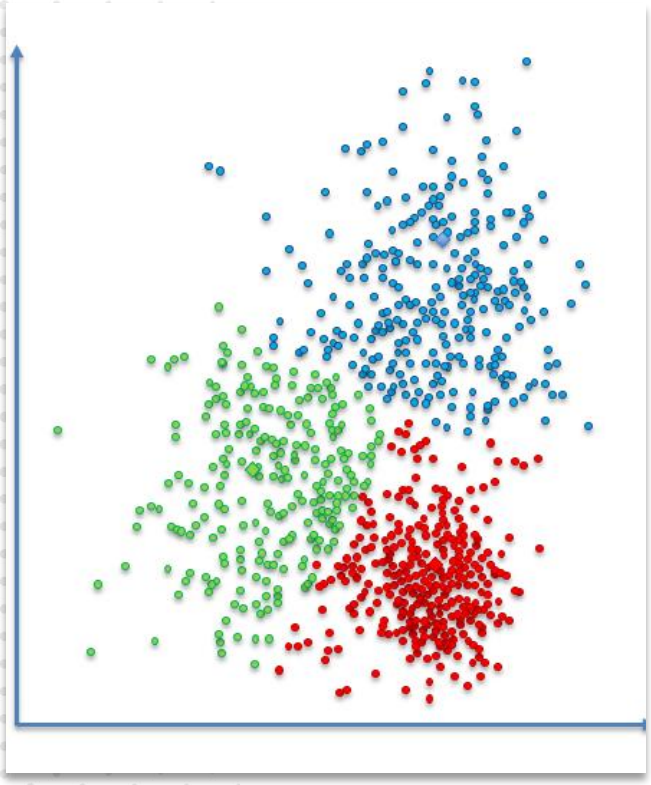
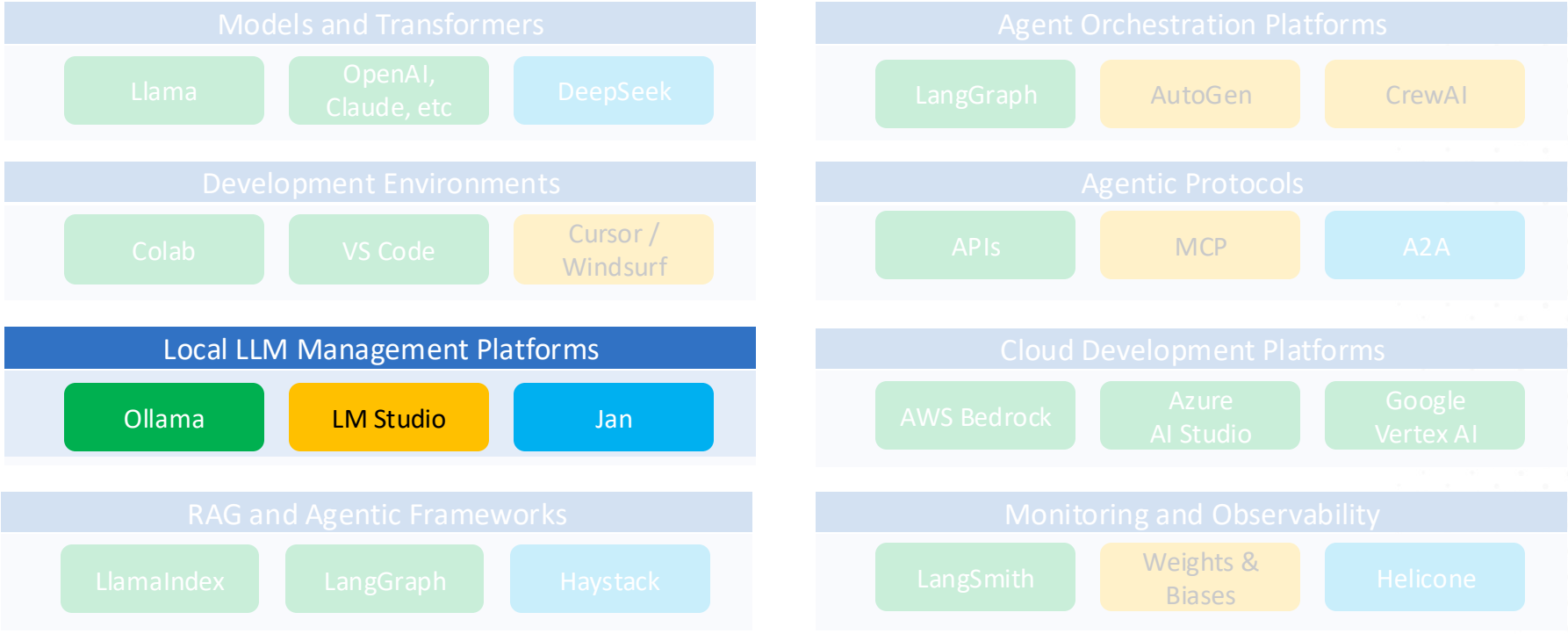


Lesson #2:
Model Management Platforms

Objectives:

- Hosting LLMs Locally
- Tools for working with LLMs: Ollama and LMStudio
- Working with Ollama
- Tools and Function Calling
- Deploying a Local Chatbot with OpenWebUI

Local LLM Management Platforms



Mature | Strong Contenders | Emerging

Pearson

Working with Open-Source Models Locally

- Local models are typically open-source (e.g. downloaded from Hugging Face)
- When downloaded, they run on a local local machine, meaning:
  - You can fine tune them
  - Build your own LLM applications
  - Add security controls to the model
  - Don’t pay a per-token cost

Tools for Working with Local Models

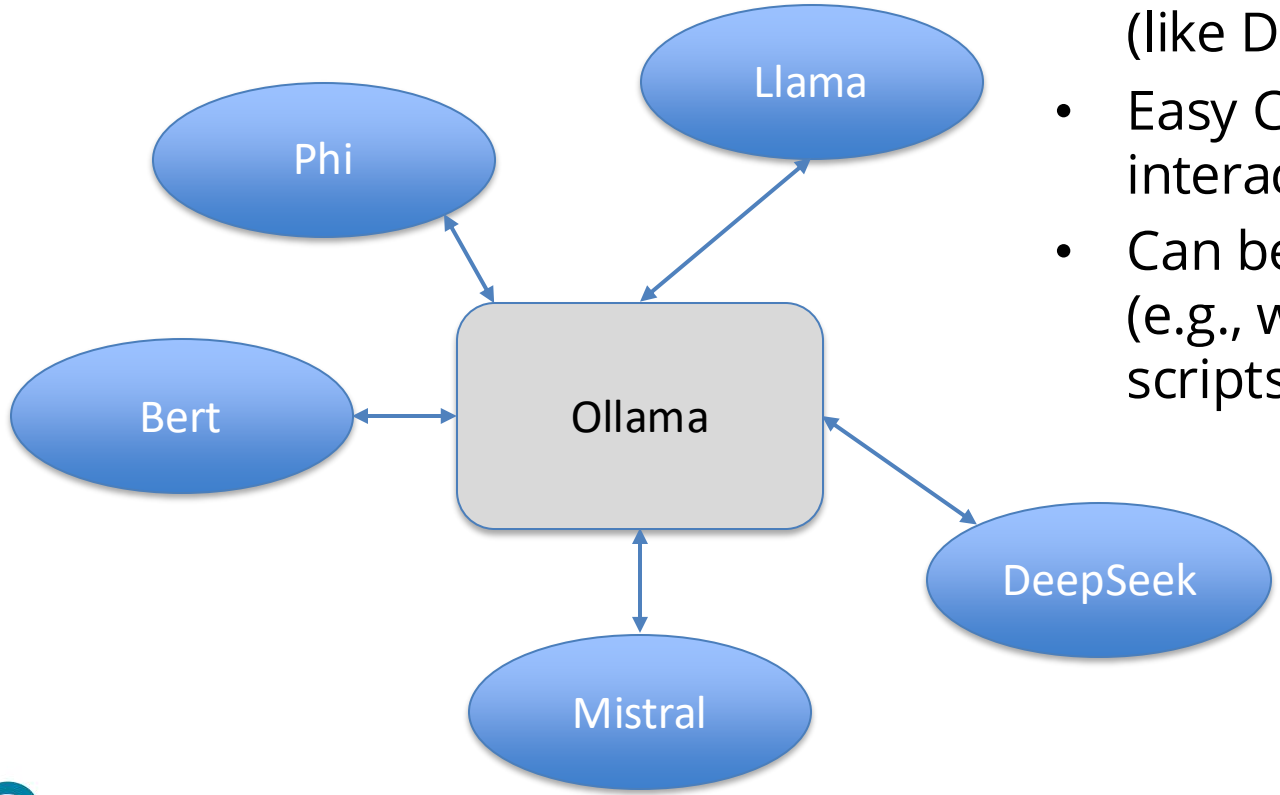
- Python and a code editor
  - VS Code, or Vibe tools like Cursor / Windsurf

- Applications built to support local models:
  - LM Studio and Ollama are good examples

Why Run Models Locally?

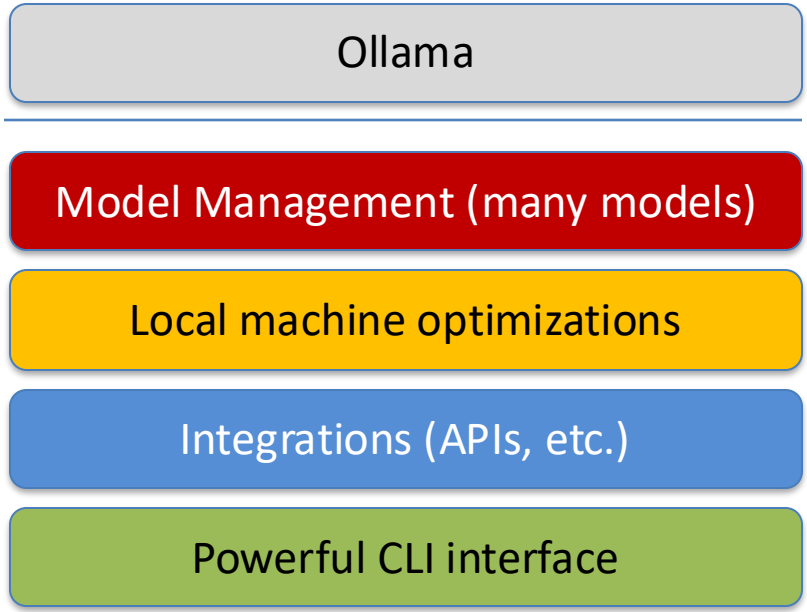
- Cost: Cloud-based GenAI solutions are becoming a lucrative source of recurring revenue for OpenAI, Google, Anthropic, etc.
- Privacy and Security: commercially available LLMs are a “black box” for users. The users have no control over how they were trained, bias, etc.
- Fine Tuning: Cloud-based LLMs are “foundation models.” Local models can be customized.
- Easy to use – with the right tools (Ollama and LM Studio)
- Flexibility of agentic integrations and RAG systems

Ollama Overview



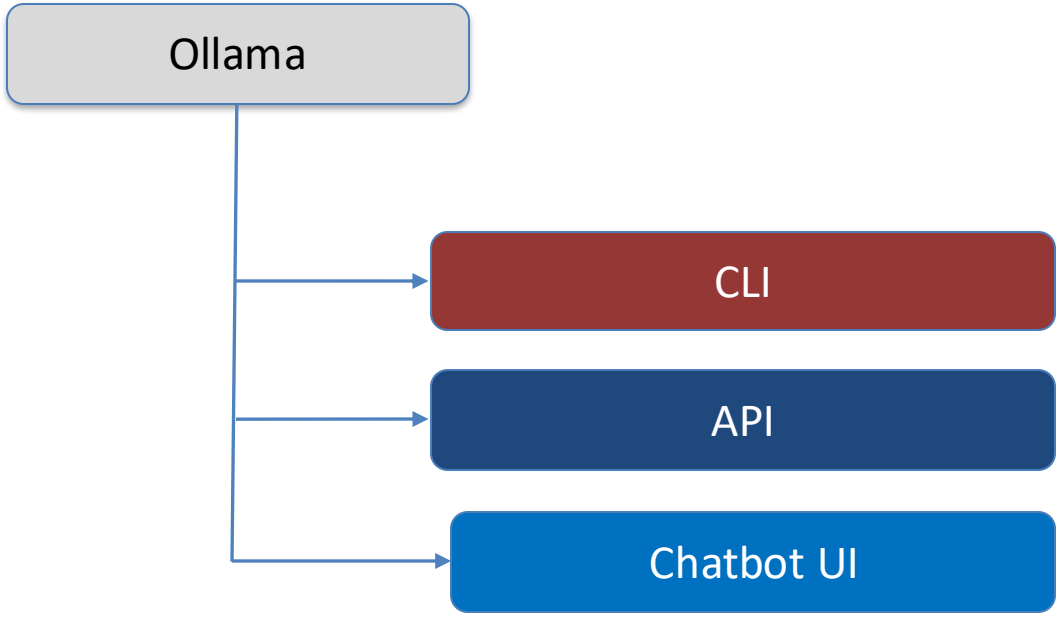
- Pull and run models locally with a single command (like Docker for LLMs)
- Easy CLI and API for interacting with models.
- Can be used in workflows (e.g., with LangChain or scripts).

Ollama Capabilities



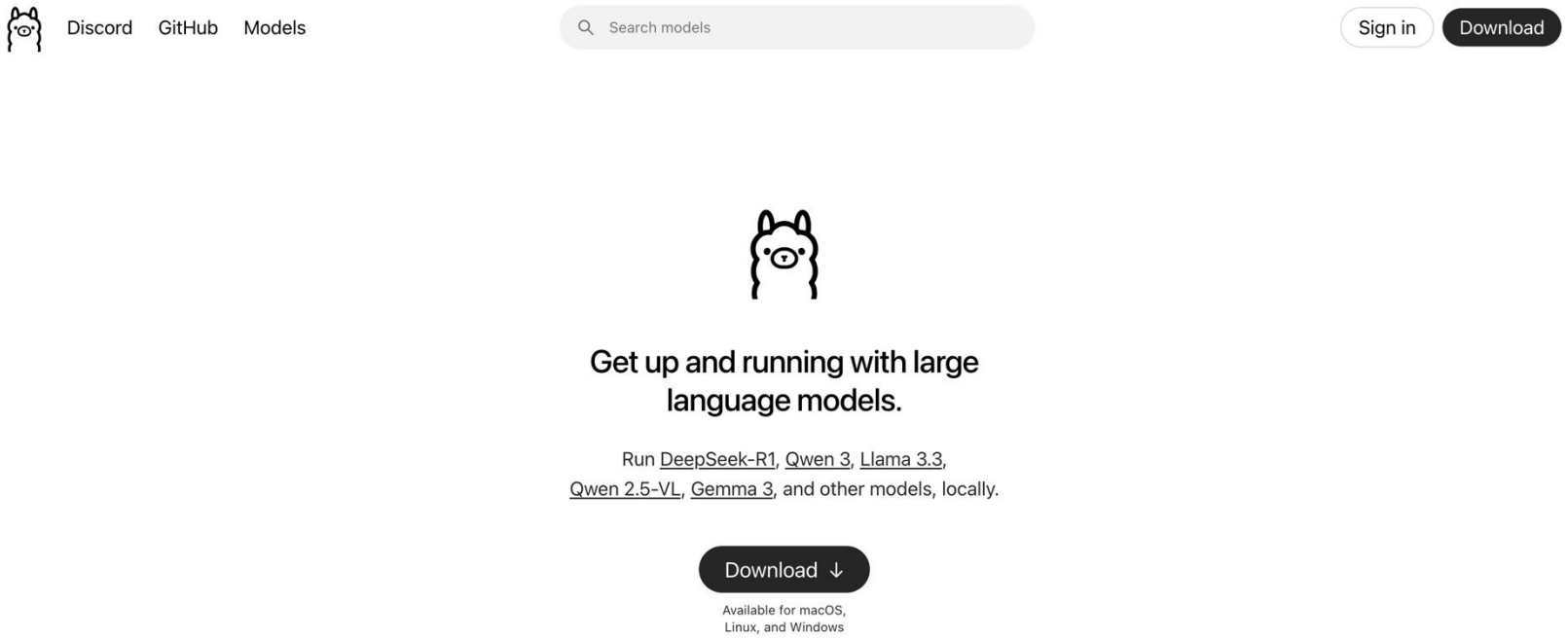
- Excellent tool for model development testing
- Security guardrail testing / red teaming
- Research and experimentation

Ways to use Ollama

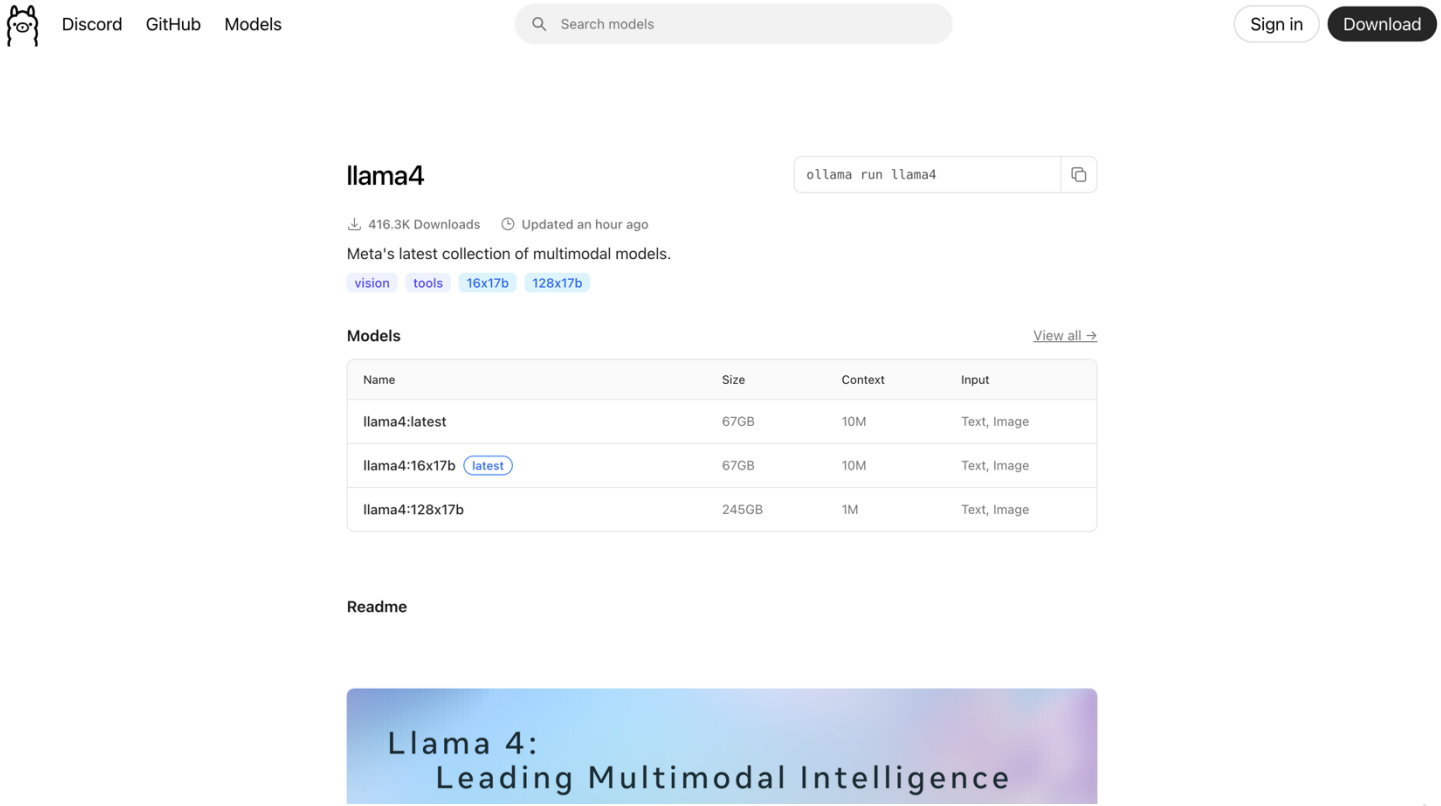


Getting Started With Ollama

DEMO



Show the types of models



Pearson

Ollama Startup

DEMO

```
ollama run tinyllama

pulling manifest
pulling 2af3b81862c6: 100%                                   637 MB
pulling af0ddbdaaa26: 100%                                    70 B
pulling c8472cd9daed: 100%                                    31 B
pulling fa956ab37b8c: 100%                                    98 B
pulling 6331358be52a: 100%                                   483 B
verifying sha256 digest
writing manifest
success
```

```
ollama list

NAME                 ID              SIZE      MODIFIED
tinyllama:latest     2644915ede35    637 MB    About a minute ago
mistral:latest       f974a74358d6    4.1 GB    2 months ago
llava:7b             8dd30f6b0cb1    4.7 GB    3 months ago
deepseek-r1:1.5b     a42b25d8c10a    1.1 GB    3 months ago
```

Ollama Terminal Commands

DEMO

```
ollama

Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  stop        Stop a running model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command
```

Ollama Show Commands

DEMO

```
ollama run tinyllama:latest

>>> /show
Available Commands:
  /show info          Show details for this model
  /show license       Show model license
  /show modelfile     Show Modelfile for this model
  /show parameters    Show parameters for this model
  /show system        Show system message
  /show template      Show prompt template

>>> Send a message (/? for help)
```

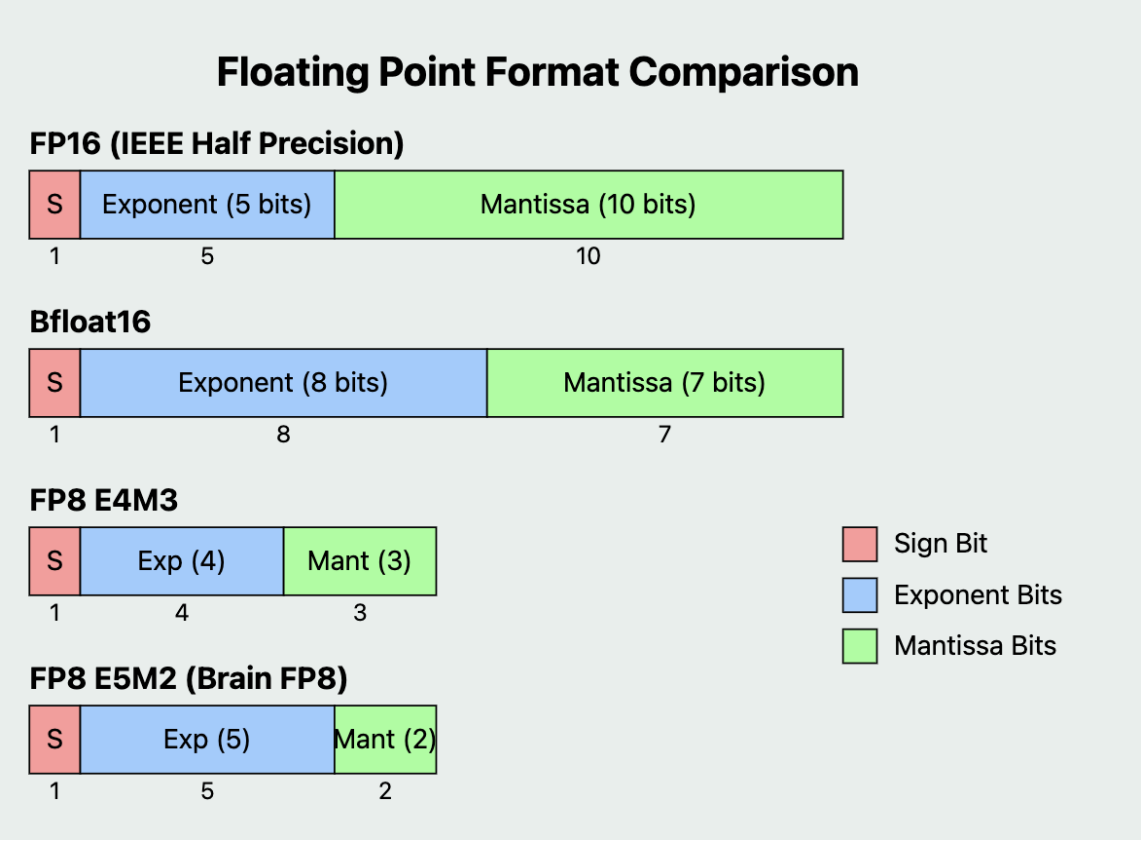
Show details of the model

DEMO

```
>>> /show info
  Model
    architecture        llama
    parameters          1.1B
    context length      2048
    embedding length    2048
    quantization        Q4_0
```

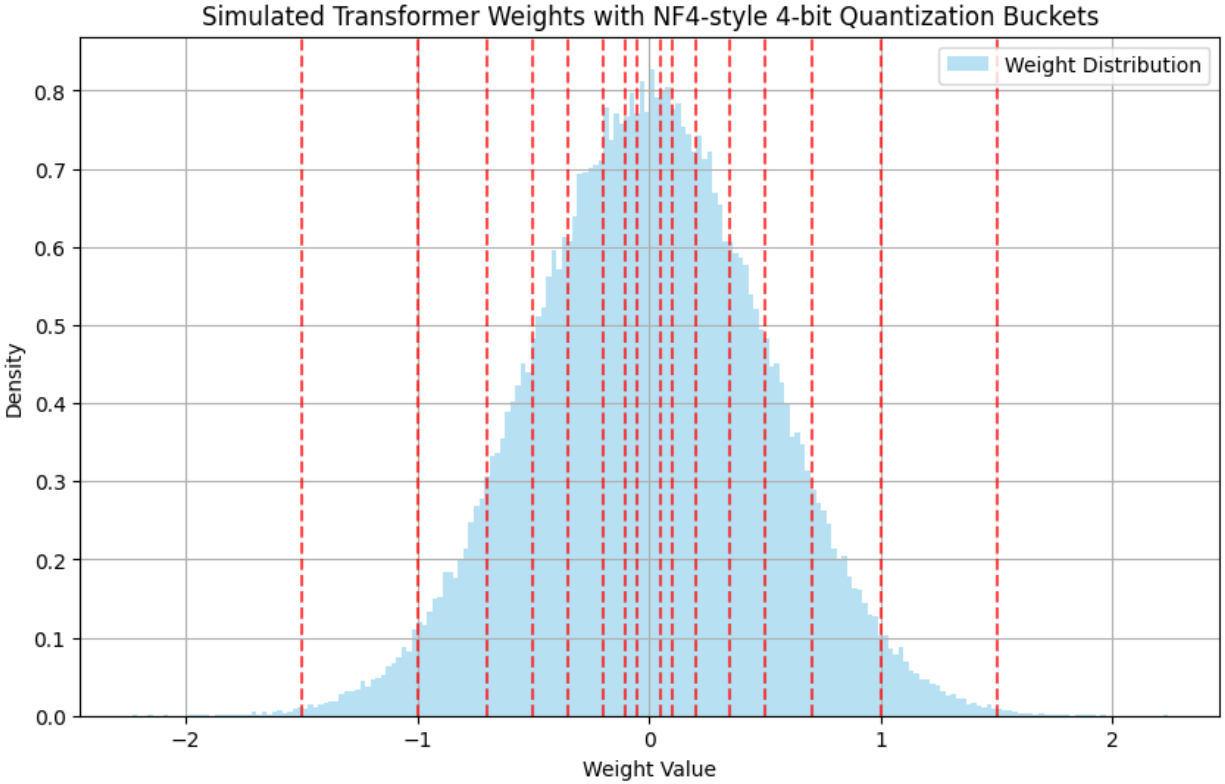
- Model Name
- Number of parameters (weights)
- How many tokens you can input into the model (more is better)
- Number of dimensions in the embedding vectors (e.g. ChatGPT uses 12,288)
- Number of bits used to represent the weights

A Few Words About Floating Point Numbers



Pearson

4-bit Quantization (QLoRA)



Pearson

Multi-Modal Models

gemma3

```
ollama run gemma3
```

6.2M Downloads Updated 2 months ago

The current, most capable model that runs on a single GPU.

vision 1b 4b 12b 27b

Models

View all →

Name	Size	Context	Input
gemma3:latest	3.3GB	128K	Text, Image
gemma3:1b	815MB	32K	Text
gemma3:4b latest	3.3GB	128K	Text, Image
gemma3:12b	8.1GB	128K	Text, Image
gemma3:27b	17GB	128K	Text, Image

Model Alignment: System Prompts vs. User Prompts

System Prompt	User Prompt
Hidden from users	Lower priority than the system prompt
Sets the LLM's personality and style	This is what you, or the API tells the model to do
Defines how the LLM responds, including rules and guardrails	Gives instructions and context
Remains constant during conversations	Drives the conversation, but changes with each new prompt
Controls format and overall style	

Examples:

SYSTEM PROMPT

```
You are a helpful coding assistant. Always provide clear,
well-commented code examples. Focus on best practices and
explain your reasoning. Never provide malicious code.
```

USER PROMPT

```
Can you show me how to create a simple to-do list in
Python using a class?
```

Ollama Modelfiles

Defines how a custom LLM is built, configured, and run within Ollama:

- **Model Selection**: Specifies the base model to use (e.g., gemma3:4b, mistral).
- **Hyperparameter Customization**: Allows you to set parameters such as temperature, top-p, or other inference settings to control the model's behavior.
- **System Prompt**: Lets you define a persistent system prompt/persona, so the model always responds with a particular style or context (e.g., as a helpful assistant, or with a specific tone).
- **Reusability**: Makes it easy to share, version, and manage different model configurations for various use cases or projects.

Interacting with the Ollama REST API

DEMO

- Runs on *localhost:11434*

- For example:

```
curl http://localhost:11434/api/generate -d '{"model": "c-3po",
"prompt": "how is R2D2 doing?", "stream": false}'
```

- Check out:
  `https://github.com/ollama/ollama/blob/main/docs/api.md`

Using Ollama in Python Code (running agents)

```
import requests
inpurt json

url = "http://localhost:11434/api/generate"

data = {
"model": "gemma3:4b",
"prompt": "tell me a short story about Ireland and make it funny"
}
```

Open WebUI

- An open-source, self-hosted web interface for running LLMs locally.
- Provides a ChatGPT-like interface that works with various local LLM backends – designed to work with Ollama

- Can install via:
  - pip
  - Docker image

- Will run on http://localhost:8080 (internal, something else for Docker)
- Download here . . https://github.com/open-webui/open-webui

Open WebUI Demo

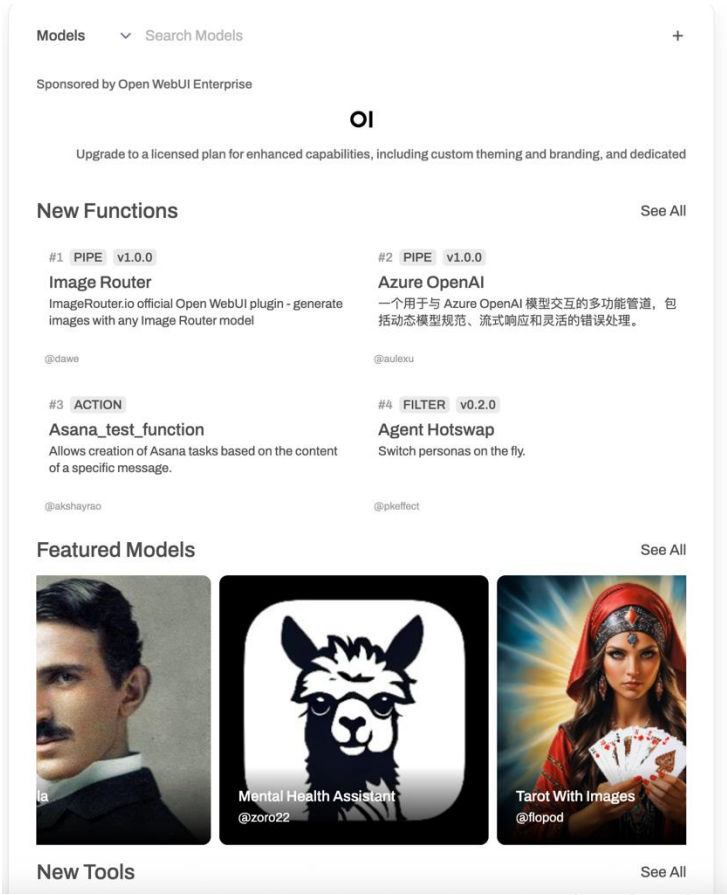
DEMO

1. Launching a model
2. Comparing two models
3. Adjusting model parameters
4. Launching a Workspace
    1. Prompts (e.g. use “/” to access your prompt template)
    2. Knowledge bases (use the “#” to load your kb)

Open WebUI Community

http://openwebui.com

Offers community supported functions, tools, models, and much more



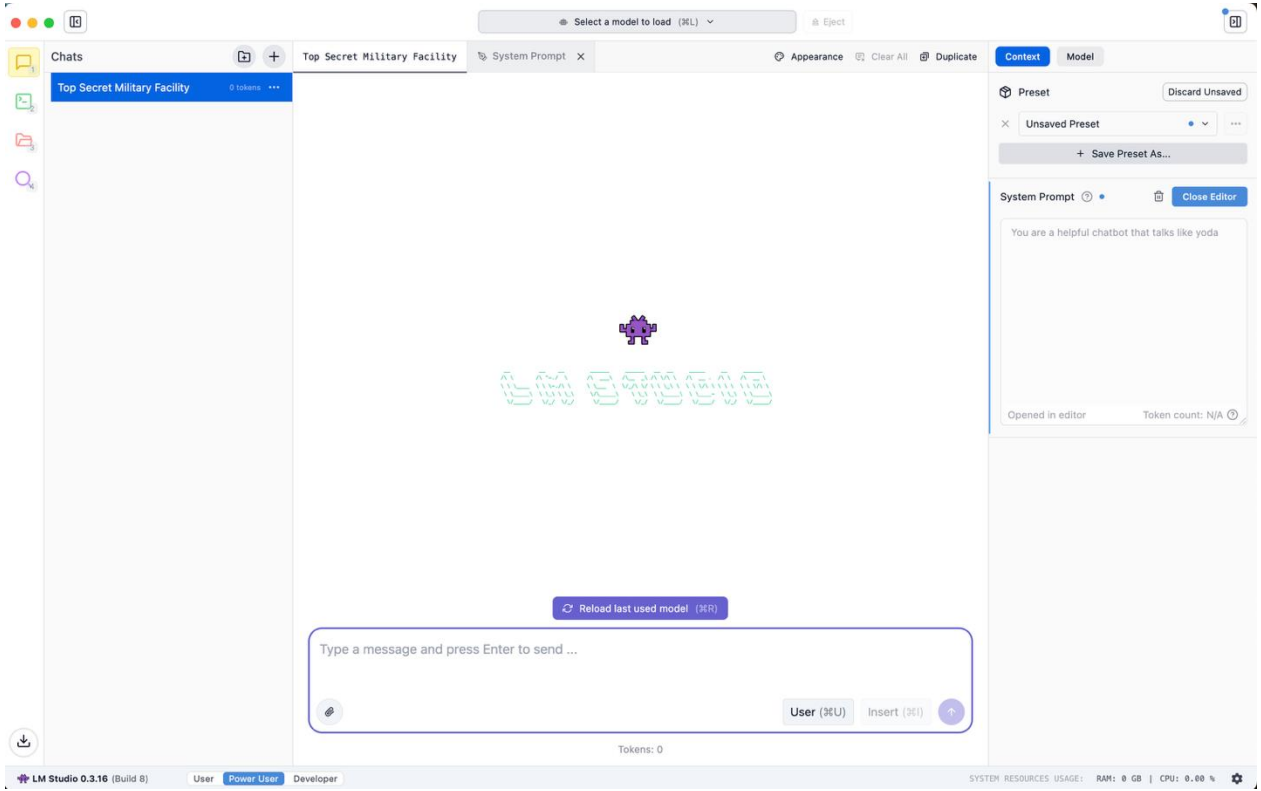
Pearson

Comparing LM Studio vs. Ollama

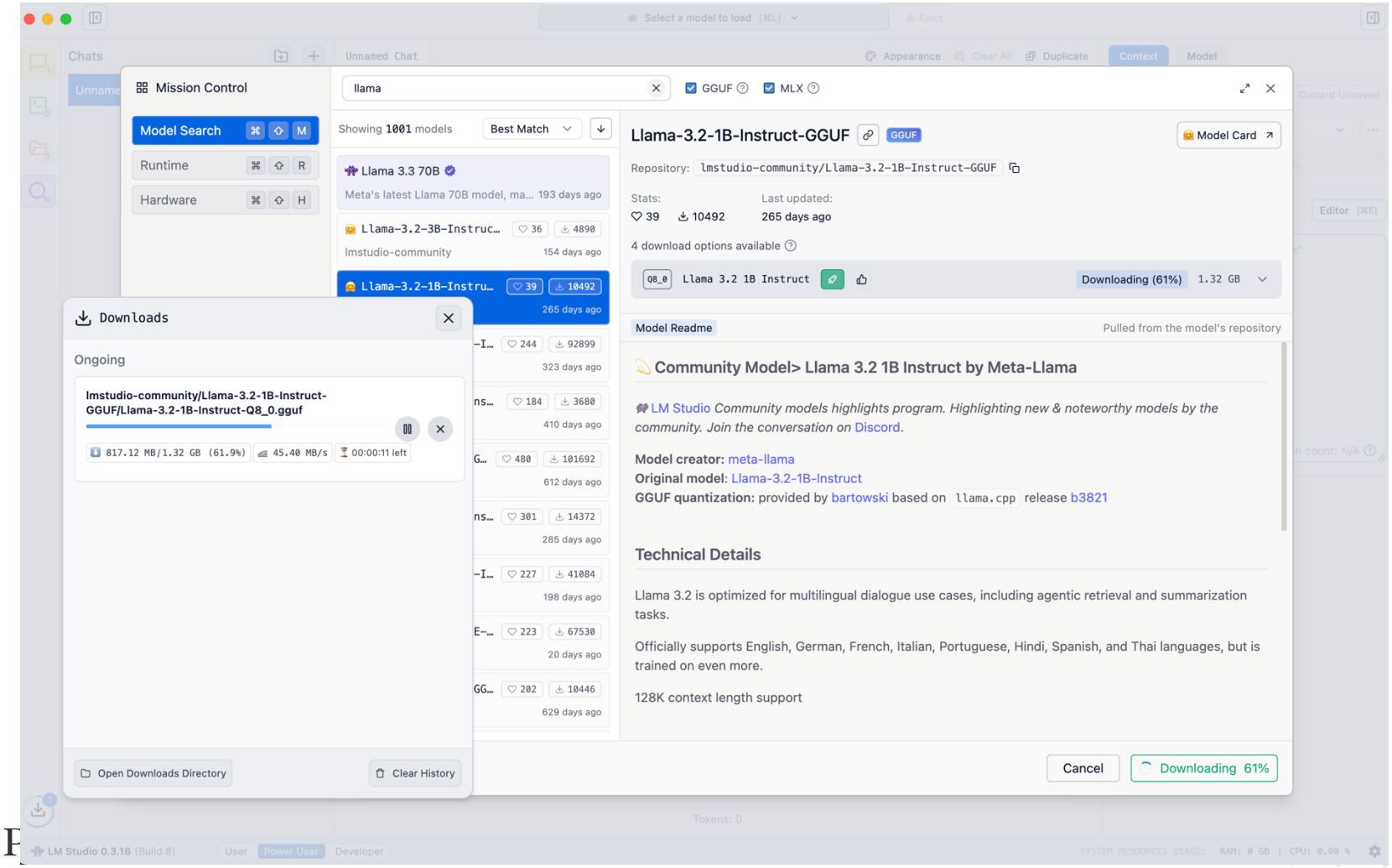
Feature	LM Studio	Ollama
Interface	Graphical user interface (GUI)	Command-line interface (CLI) + REST API
Ease of Use	Very user-friendly, ideal for non-coders	Requires basic CLI knowledge
Model Management	Browse/download models via GUI (Hugging Face)	Pull models via terminal using ollama run
Model Sources	Hugging Face	Ollama library + custom Modelfile builds
Offline Usage	Yes	Yes
Customization	Limited (basic system prompts only)	High (custom Modelfile, parameters, settings)
Developer Features	Focused on chat only	Built for dev workflows, supports API integration
Integrations	Limited API integrations	Easily used with LangChain, custom apps, scripts
Ideal For	Casual users, educators, researchers	Developers, tinkerers, power users

LM Studio Quick Overiew

DEMO



Pearson


Select a model to load (⌘L)
Eject
Chats
Unnamed Chat
Appearance
Clear All
Duplicate
Context
Model
Mission Control
Model Search ⌘ M
Runtime ⌘ R
Hardware ⌘ H
llama
GGUF
MLX
Showing 1001 models
Best Match
Llama 3.3 70B
Meta's latest Llama 70B model, ma... 193 days ago
Llama-3.2-3B-Instruc... 36 4890
lmstudio-community
154 days ago
Llama-3.2-1B-Instru... 39 10492
265 days ago
Llama-3.2-1B-Instruct-GGUF
GGUF
Model Card
Repository: lmstudio-community/Llama-3.2-1B-Instruct-GGUF
Stats:
39 10492
Last updated:
265 days ago
4 download options available
Q8_0 Llama 3.2 1B Instruct
Downloading (61%) 1.32 GB
Model Readme
Pulled from the model's repository
Community Model> Llama 3.2 1B Instruct by Meta-Llama
LM Studio Community models highlights program. Highlighting new & noteworthy models by the community. Join the conversation on Discord.
Model creator: meta-llama
Original model: Llama-3.2-1B-Instruct
GGUF quantization: provided by bartowski based on llama.cpp release b3821
Technical Details
Llama 3.2 is optimized for multilingual dialogue use cases, including agentic retrieval and summarization tasks.
Officially supports English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai languages, but is trained on even more.
128K context length support
Downloads
Ongoing
lmstudio-community/Llama-3.2-1B-Instruct-GGUF/Llama-3.2-1B-Instruct-Q8_0.gguf
817.12 MB/1.32 GB (61.9%)
45.40 MB/s
00:00:11 left
Open Downloads Directory
Clear History
Cancel
Downloading 61%
Discard Unsaved
Editor (⌘E)
Tokens: 0
LM Studio 0.3.16 (Build 8)
User
Power User
Developer
SYSTEM RESOURCES USAGE:
RAM: 0 GB | CPU: 0.00 %