SDS PODCAST EPISODE 876: HUGGING FACE'S SMOLAGENTS: AGENTIC AI IN PYTHON MADE EASY

Jon Krohn:	00:02	This is episode number 876 on Hugging Face's smolagents.
	00:19	Welcome back to the SuperDataScience Podcast. I'm your host, Jon Krohn. Today we're diving into Hugging Face's smolagents, S-M-O-L agents, one word, a new development that gives AI models more autonomy.
	00:35	Hugging Face, the open source AI powerhouse behind technologies like the Transformers library has now turned its attention to AI agents, programs where AI models can plan and execute tasks on their own. And their latest library, smolagents makes building these agents simpler than ever. In this short episode, I'll break down what smolagents are for you, how they work, and why they're a big deal for developers, businesses, and researchers alike.
	01:02	All right, so let's start off by describing what exactly smolagents is. In a nutshell, it's a new Python library from Hugging Face that simplifies the creation of AI agents, making them more accessible to developers and data scientists. Hugging Face themselves describe smolagents as a library that unlocks agentic capabilities for language models. It is exactly that. This means you can take an LLM, a large language model, such as a proprietary OpenAI model, an open-weight Llama model from Meta, or a fully open source model from DeepSeek and easily give it the ability to act, to use tools, call APIs, and perform multi-step reasoning to accomplish a task.
	01:42	Under the hood, smolagents is designed with minimal complexity. The core logic fits in only a few thousand lines of code, keeping abstractions light and above raw python. The simplicity is intentional. It lets developers understand and tweak agent behaviors without wading through overly complex framework code. Smolagents is actually the successor to Hugging Face's earlier

Transformers.agents module, which it will eventually replace, reflecting lessons learned to make this library lean and user-friendly.

02:11 One of the standout features is its code agent approach. Unlike some agent frameworks that represent actions in abstract JSON or TXT, smolagents agents literally write Python code as their way of reasoning and taking actions. In other words, the AI agent decides what to do next by generating code, for example, calling a tool or performing a calculation, which is then executed. This code-centric approach is powerful because it leverages the full flexibility of Python for the agent's actions. To keep things safe, these code executions can be sandboxed, run in isolated environments so that the agent doesn't do anything harmful on your actual system. So, you get the flexibility of letting the agent write code with safeguards in place.

02:53 And if you prefer a more traditional style, smolagents still supports a standard tool calling agent that uses structured text for actions so you've got options. Smolagents also comes with first-class tool integrations. You can easily turn any Python function into a tool that the agents can use just by decorating it with @tool, so Python code decorator @tool, and adding some type hints and doc strings into your function.

03:20 These tool integrations could be anything from a web search function to a database query or a calculator. Even better, Hugging Face has integrated this with their Hugging Face Hub. This means you can share and load tools from the Hugging Face Hub, and so a community of users can publish, is publishing useful agent tools, say a weather API fetcher or a text translator, and you can plug into those agents that they've created and upload it to the hub in seconds. It's kind of like an app store for AI agent tools.

03:51 Another key feature is that smolagents is model agnostic. You're not locked into a single AI model or provider. It supports any large language model. You can use open source models from the Hugging Face Hub that I just described with the library loading them via the Hugging Face transformers library or via the Hugging Face inference API. Or you can use API based models from again, OpenAI, Anthropic or others through an integration called LiteLLM. You could even hook up local models if you have them. Essentially, whether it's a 7 billion parameter open source model running on your laptop or a cutting edge model running via a cloud API, you can plug it into smolagents.

04:32 This flexibility is great for developers who want to experiment or deploy agents in different environments. To recap the technical highlights, smolagents offers the following five features, simplicity, so a lightweight clear code-based on the order of thousands of lines that's easy to understand. Minimal abstraction means you work almost directly with Python code, not obscure configuration. Two, the agents that write and execute Python code to do things have code-based actions, enabling complex actions and multi-step reasoning in a familiar language with sandboxing or safety. This is a distinctive agents that write code design.

05:12 Three, smolagents has tool integrations and sharing. This is a convenient way to define tools, just Python functions, and an ability to share those tools or entire agents via the Hugging Face Hub. This fosters reusability and community collaboration. Four, smolagents offers any model support, so compatibility with a wide range of LLMs, from local models to the most popular APIs through a unified interface. You choose the model that fits your needs or constraints. And then fifth, finally, smolagents has out of the box usability. So, it even provides a command line interface, the smolagent CLI, to

run an agent with one command and a specialized web agent for web browsing tasks. This allows you to quickly test and deploy agents, making it easy to get started without writing a lot of boilerplate code.

06:04 All right. So, hopefully this sounds cool, but what can you actually do with smolagents? Well, quite a lot. So, effectively, limitless options, but let me get some ideas in your head. Because smolagents [inaudible 00:06:17] an LLM use tools and perform multi-step reasoning, it opens up many real world applications across industries. So, here are a few interesting examples, and actually all of these apply to, really, any kind of agentic AI framework, but yeah, specifically here we're going to be talking about smolagents.

06:33 So, as a specific example, you could do complex question answering and research. You could build an agent that given a hard question, will search the web, find information and then compose an answer. For instance, Hugging Face provided a demo that showed an agent using a search tool to figure out how many seconds would it take for a leopard at full speed to run through the Pont des Arts. That's, I think, in Paris, and Hugging Face, the founders are originally from Paris.

06:57 You could use something like OpenAI's $200 per month deep research functionality to answer that kind of question out of the box, or you could experiment with using smolagents to be a kind of research assistant, fetching facts and data from various sources to answer your queries or to write reports. As another example, because no piece of content on agents is complete without a travel planning example. Here we go. Suppose you need to plan a trip or an event with many parameters, with smolagents you could create an agent that takes a high level goal, like plan a trip to Tokyo, Kyoto, and Osaka between March 28th and April 7th, and then smolagents

will call tools to gather information, search for flights, check train schedules, look up hotel reviews and so on. In fact, you can use the smolagent command line interface to do this in a single line, where you just type smolagent, one word, and then in natural language, say, plan a trip to Tokyo, Kyoto, and Osaka between March 28th and April 7th. That's it.

08:00 The agent could break down the task, perhaps use a maps API for distances or a Web Scraper for attractions, and return an organized itinerary for you. This kind of multi-step planning agent could be useful for you, for travel agencies, event planners, or personal assistants. Yeah. Again, kind of infinite possibilities there.

08:19 In addition to calling APIs, smolagents can also drive a web browser, like a graphical web browser to perform actions. Hugging Face provides something called webagent, one word, to do that. And so it can take instructions like, "Go to an e-commerce website, navigate to the sales section. Click the first product and get its details and price."

08:39 And then kind of final real world example for you here is that because smolagents can execute Python code, you could have them perform data analysis on the fly. Imagine an agent given a prompt to analyze last month's sales figures and identify the top three products by growth. It could load a CSV using Pandas, a Python library for data frames, for tabular data, and then it could compute the results and output a summary. It essentially can combine data processing and language generation together. That's pretty damn cool.

09:12 These examples, they barely scratch the surface. The key is that smolagents allows LLMs to interact with the world through tools and code in a controlled way. This means any task that involves finding information, transforming

data, or making decisions based on intermediate results could potentially be handled by an AI agent. And since Hugging Face has made it easy to share components, the community can rapidly build out a library of tools and agents for countless niche applications.

09:39 All right, now let's zoom out. How do smolagents fit into the broader AI ecosystem, and what do they mean for businesses and data scientists? In 2024 and particularly 2025, there's been a lot of buzz about AI agents from autonomous research assistants to AI that can use software like humans do. Smolagents is part of this movement, and its introduction is likely to have a non-negligible impact on all that. For businesses, smolagents could lower the barrier to automating complex tasks with AI. Instead of needing a whole team of engineers to stitch together a custom solution, a company could use smolagents to prototype an AI agent that, say, handles customer inquiries or performs market research with just a few lines of Python code.

10:22 Because it supports local and open source models, companies worried about data privacy can keep everything in-house on their own secure servers on-prem. This flexibility means businesses can choose between using powerful proprietary models or cheaper free open models, whatever suits their needs and budget, all within the same framework. In short, smolagents can help companies inject AI-driven automation into their workflows faster and more cost-effectively.

10:48 For data scientists and software developers, smolagents is a new powerful tool in your toolbox that's both accessible and versatile. If you've ever used frameworks like LangChain or worked with OpenAI's function calling, you'll find smolagents gives you similar capabilities. But it's also lightweight and transparent. It's all open source, so you can inspect how it works, debug it, or extend it.

The minimalist design means you can probably learn it quickly as well. This means that you, as an individual, not just big teams, can experiment with building your own agents.

11:22 And finally, for researchers, smolagents provides a convenient platform to study how AI models behave when given more autonomy. Researchers interested in reasoning, tool use and multi-step decision making can use the library to test new ideas. And the fact that you can swap in any model means it's great for benchmarking. For example, how does a smaller open model versus a larger proprietary model perform in an agent role? That also sounds like a useful kind of question for benchmarking any kind of tools you're building commercially.

11:50 All right. To sum things up here, Hugging Face's smolagent is a convenient new framework that brings the concept of AI agents to the mainstream in a small, simple and accessible form. In today's smol episode, I covered how smolagents works and its key features from code-writing agents and easy tool integration to support for any model you want. We explored examples of what you can build, like travel planners, web scrapers, and research assistants, all of which could be applied across any industry.

12:18 If you've been waiting to get started with your own Agentic AI application in your particular commercial or academic niche, with smolagents now available, it could be the right time to get going. Looking to brainstorm ideas on products or features that you could build with AI agent functionality? No problem. Talk to your colleagues or friends or your favorite conversational LLM, like Claude 3.7 Sonnet, Google Gemini, or OpenAI's GPT 4.5.

12:45 And I should point out at the end of this episode, I'm just a big fan of this library. These are my own opinions. There's no Hugging Face sponsorship of this episode, or actually any episode of this show ever. So, yeah, I just think it's something great to check out. It could make getting AI agents out into the real world for you very easy.

13:09 And on the note of not being tied to Hugging Face, in particular as the agentic framework that you should be working with, there are lots of other options out there for you for developing and deploying AI agents. So, earlier in this episode, I already mentioned LangChain, which is the most popular agentic framework today. Robust and feature-rich LangChain surpasses smolagents in scalability, memory management, and built-in integrations, making it ideal for complex agentic workflows.

13:40 LlamaIndex is another solid option. It excels at managing large datasets and memory-intensive retrieval tasks, offering out-of-the-box data integration capabilities not present in comparatively lightweight smolagents. PydanticAI provides structured type-safe agent outputs through explicit schema validation, contrasting smolagents simpler, but less controlled output handling.

14:03 And then in terms of multi-agent systems, Microsoft AutoGen specializes in collaborative multi-agent systems with dynamic coordination and communication, whereas smolagents focuses primarily on single agent tasks. And finally, you can't talk about multi-agent agentic systems without the popular CrewAI framework, which was built explicitly for orchestrating multi-agent teams through clearly defined roles and tasks, making it superior to smolagents in coordinated agent collaborations.

14:34 We've got a link to all of these frameworks, of course, smolagents, LangChain, LlamaIndex, Pydantic, Microsoft AutoGen, and Crew all in the show notes. All right.

14:44 That's it for today's episode. If you enjoyed it or know someone who might consider sharing this episode with them, leave a review of the show on your favorite podcasting platform. Tag me in a LinkedIn or Twitter post with your thoughts. And if you aren't already, obviously subscribe to the show. The most important thing, though, is that I hope you'll just keep on listening. Until next time, keep on rocking it out there, and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.