Microsoft

Operating AI Agents with Azure AI Foundry and Pydantic AI

By Charunthon Limseelo
Microsoft Learn Student Ambassador

This presentation is using for the event of
Global Azure 2025 – Thailand on May 10th, 2025 – Digital Event

1

Session Speaker

Charunthon Limseelo (Boat)

- Microsoft Learn Student Ambassador and Data Engineer Intern at Seven Peaks Software, Klong Toei, Bangkok, TH
- Speaker on Various International Stages such as PyCon TH, FOSSASIA, JavaScript Bangkok, etc.

boatchrnthn

Charunthon Limseelo

Charunthon Limseelo

2

Operating AI Agents with Azure AI Foundry and Pydantic AI

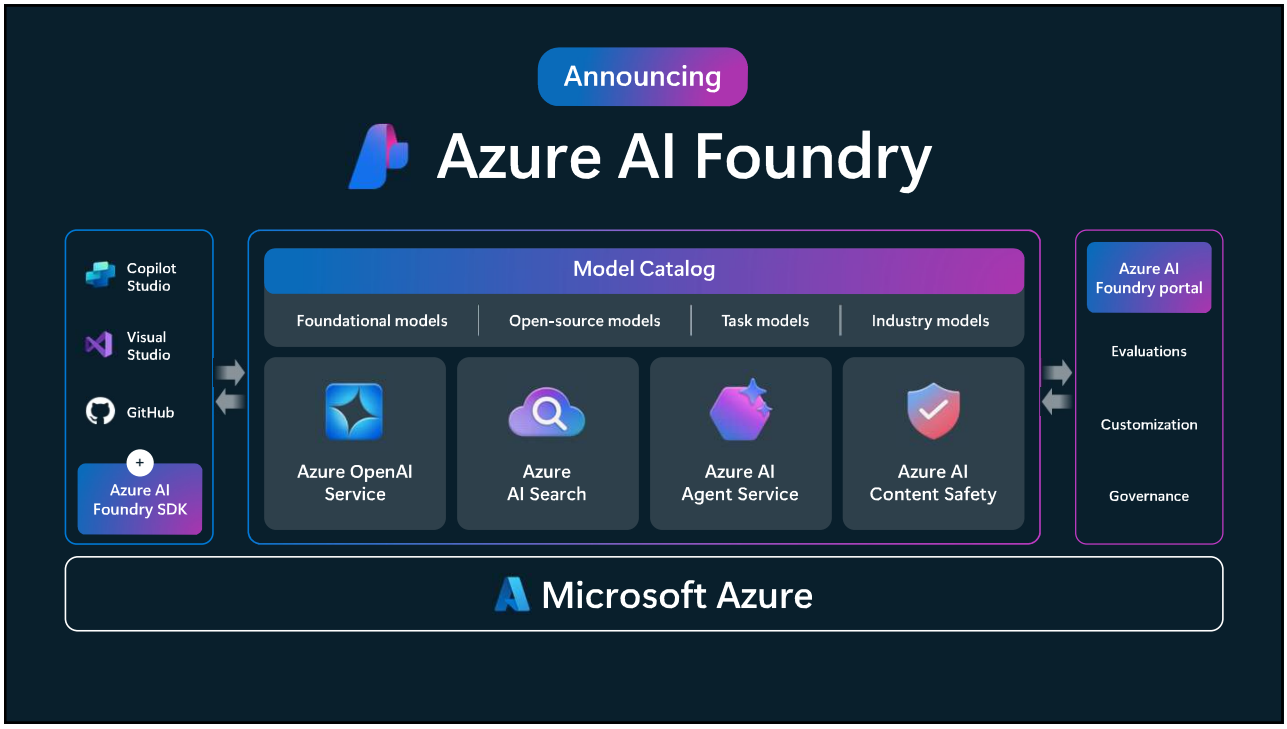
Text generation
GPT-4.1 to generate text, or other models for reasoning tasks

Multimodal to text
GPT-4.1, or other models to explore the different types of input

Pydantic Framework
Explore the capabilities of building agents to perform tasks

Explore More Multimodal Models and usage of
GitHub Models

3



4

Generally Available

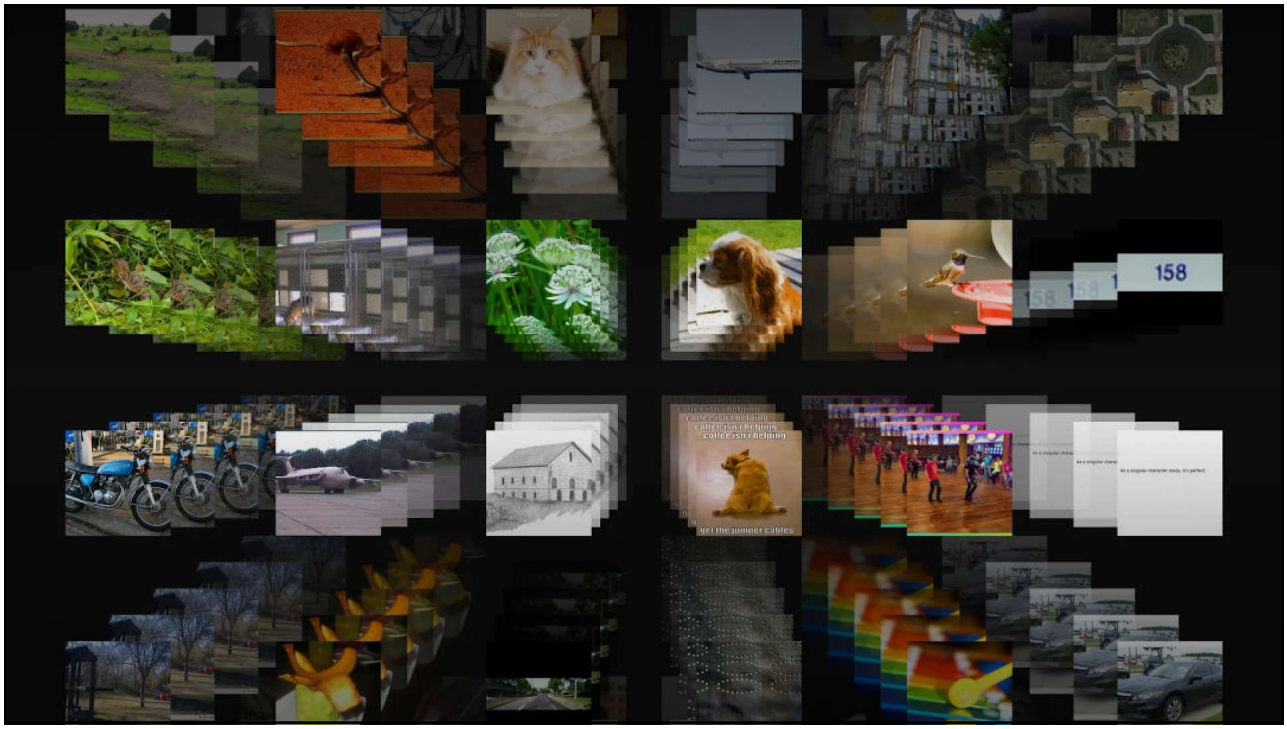
Foundation Models in

GitHub Models Marketplace

Including the usage of Azure AI SDK

aka.ms/githubmodels

5



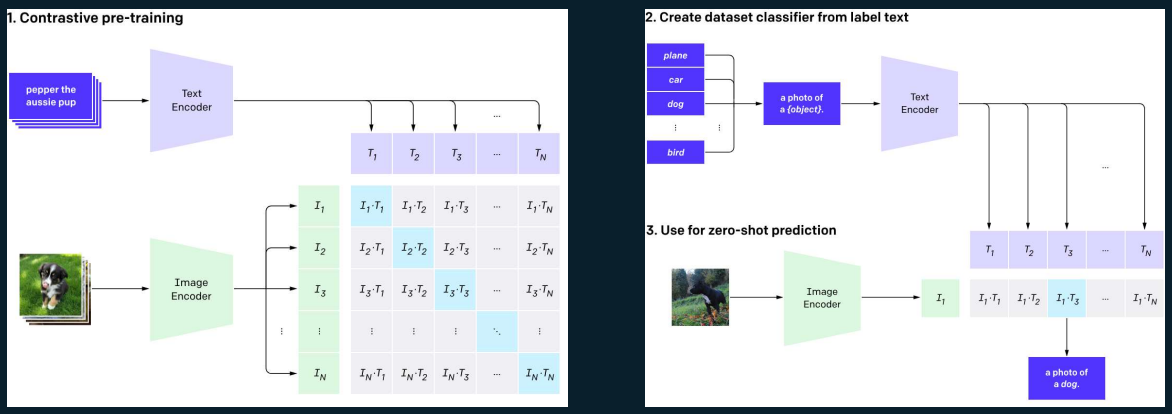
6

Section 1

Overall Picture of Developing Vision Models in Nowadays

7

CLIP: Contrastive Language-Image Pre-training

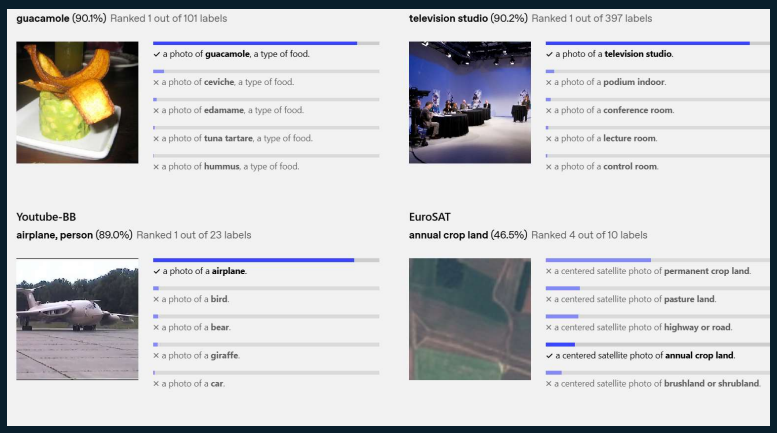


CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

https://openai.com/index/clip/

8

CLIP: Contrastive Language-Image Pre-training

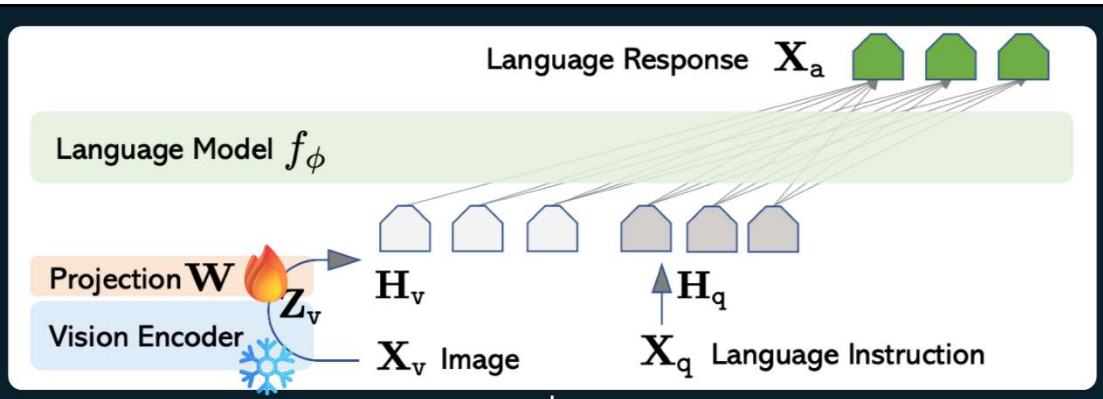


CLIP performs well in recognizing common objects but struggles with abstract and systematic tasks like counting objects or estimating distances in images. In these cases, its zero-shot performance is only slightly better than random guessing. It also underperforms in fine-grained classification, such as distinguishing between car models, aircraft variants, or flower species.

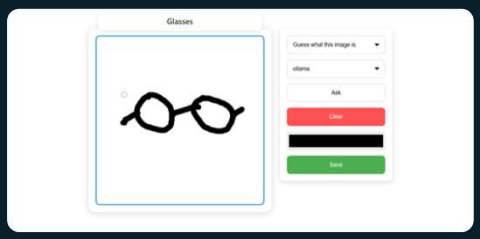
Additionally, CLIP has poor generalization to images outside its pre-training dataset. For example, while it has strong OCR capabilities, its accuracy on handwritten digits in the MNIST dataset is only 88%, significantly lower than human accuracy of 99.75%. Furthermore, CLIP's zero-shot classifiers are sensitive to wording and phrasing, sometimes requiring trial-and-error prompt engineering to achieve optimal performance

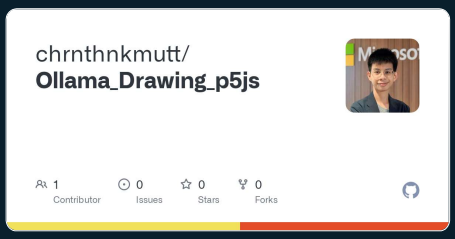
https://openai.com/index/clip/

9



https://llava-vl.github.io/





https://github.com/chrnthnkmutt/Ollama_Drawing_p5js

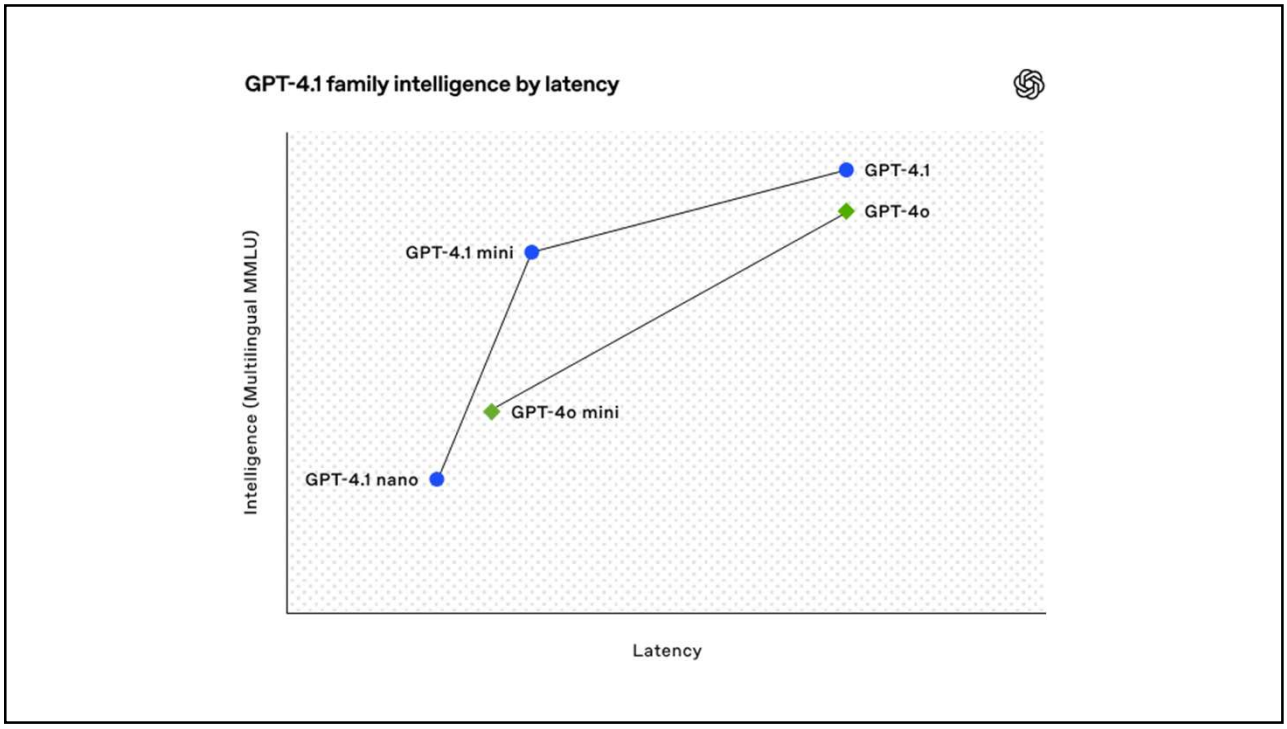
10

Section 2

Introducing OpenAI GPT-4.1

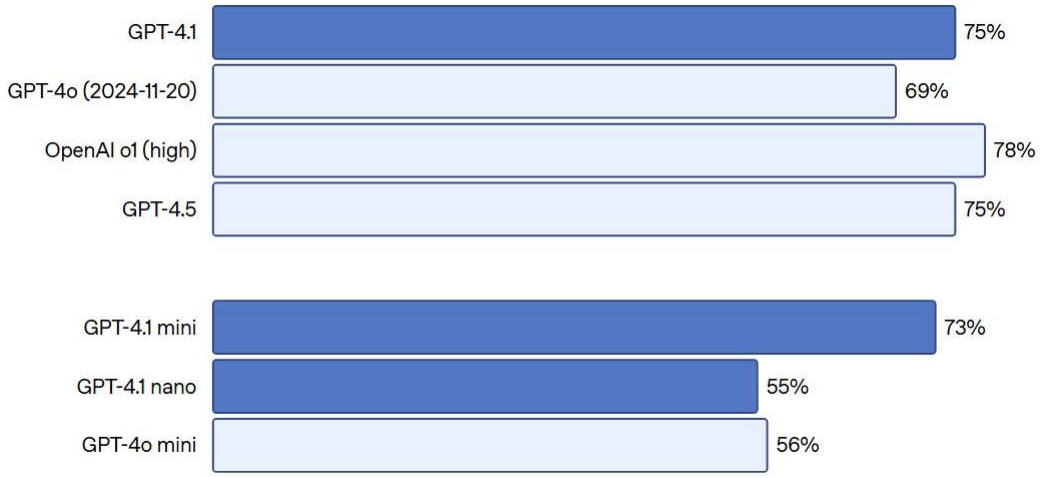
The latest iteration of the gpt-4o model family, specifically targeted for better coding and instruction following, making it better at handling complex technical and coding problems.

11



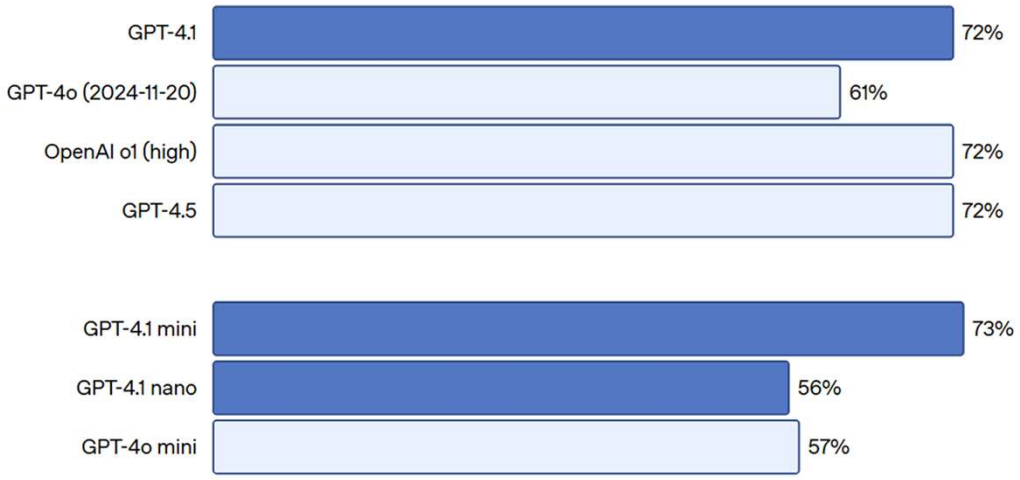
12

MMMU: Massive Multi-discipline Multimodal Understanding



13

MathVista Accuracy (Act like Photomath, but better)



14

Section 3

AI Agents Development, Pydantic: Python Data Validator, and PydanticAI: Pydantic for LLMs

15

Agent

An agent in LLM-based applications is an autonomous software entity leveraging large language models to perform specific tasks through natural language interaction.

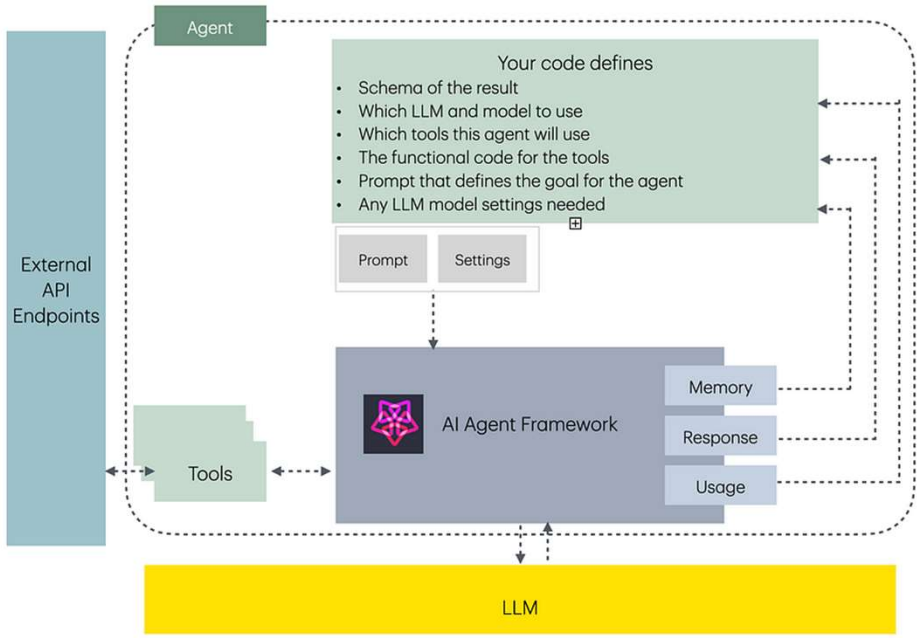
- Independent
- Goal-focused
- Interactive

16

What is Pydantic and Pydantic AI?

- **Pydantic** is the most widely used data validation library for Python.
- **PydanticAI** is a Python agent framework designed to make it less painful to build production grade applications with Generative AI.
- FastAPI revolutionized web development by offering an innovative and ergonomic design, built on the foundation of Pydantic.
- Similarly, virtually every agent framework and LLM library in Python uses Pydantic, yet when we began to use LLMs in Pydantic Logfire, we couldn't find anything that gave us the same feeling.

17



18

Example Syntax of Inferencing OpenAI Models in with Pydantic AI

```python
from pydantic import BaseModel
from pydantic_ai import Agent
from pydantic_ai.models.openai import OpenAIModel
from pydantic_ai.providers.openai import OpenAIProvider
import os # Import the os module
from dotenv import load_dotenv
from openai import OpenAI

# Load environment variables
load_dotenv()

class CityLocation(BaseModel):
    city: str
    country: str

openai_api_key = os.environ.get("OPENAI_API_KEY")

openai_provider = OpenAIProvider(
    api_key=openai_api_key
)
```

Environment Setup/Import Packages

Environment Declaration

19

Example Syntax of Inferencing OpenAI Models in with Pydantic AI

```python
openai_model = OpenAIModel(
    model_name='gpt-4o', # Changed model name for official OpenAI
    provider=openai_provider
)

agent = Agent(openai_model, output_type=CityLocation)

# Now this call will go to the official OpenAI API
result = agent.run_sync('Where were the olympics held in 2012?')

print(result.output)
# Expected output might look similar depending on the model response
# > city='London' country='United Kingdom'

print(result.usage())
# Usage details will reflect tokens used on the official OpenAI API
# """
# Usage(requests=1, request_tokens=XX, response_tokens=YY, total_tokens=ZZ, details=None)
# """
```

20

Proposed Syntax of Inferencing Azure AI Models in with Pydantic AI

```python
from pydantic_ai import Agent
from pydantic_ai.models.openai import OpenAIModel
from pydantic_ai.providers.azure import AzureProvider

model = OpenAIModel(
    'gpt-4o',
    provider=AzureProvider(
        azure_endpoint='your-azure-endpoint',
        api_version='your-api-version',
        api_key='your-api-key',
    ),
)
agent = Agent(model)
...
```

21

Demo Section

Running Multimodal Models with Normal Pydantic and Pydantic AI for Azure OpenAI SDK

22

Section 4

Key Takeaways + Study More

23

Key Takeaways

- **Multimodal Capabilities**: Azure AI Foundry enables seamless interaction with AI models that process multiple types of data (text, images, etc.), enhancing AI-powered applications.
- **GitHub Model Integration**: Leveraging pre-trained AI models from GitHub simplifies deployment and experimentation, accelerating AI development within Azure.
- **Agent-Based AI Interactions**: Intelligent agents in Azure AI Foundry provide dynamic responses and automate workflows, creating more intuitive user experiences.
- **Custom Model Adaptation**: Developers can fine-tune and customize models to align with specific business needs, ensuring optimal AI performance.
- **Efficiency & Scalability**: Azure AI Foundry offers robust infrastructure to scale AI solutions efficiently, supporting enterprise-grade implementations.

24

Resource

- Pydantic Documentation (OpenAI API on Ollama)
  - https://ollama.com/blog/structured-outputs
- Pydantic AI Documentation
  - https://ai.pydantic.dev/models/openai/#azure-ai-foundry
- GitHub Marketplace
  - github.com/marketplace/models
- GitHub Models in .NET with Semantic Kernel Blog
  - gh.io/ModelsSemanticKernel
- Build Generative AI apps with .NET and GitHub Models
  - gh.io/GitHubModelsandDotnet
- Microsoft Reactor: Prototyping AI Agents with GitHub Models
  - https://www.youtube.com/watch?v=LflBM3ntUSY

25

Thank you 🙏

boatchrnthn

Charunthon Limseelo

Charunthon Limseelo

Scan QR for resources

26