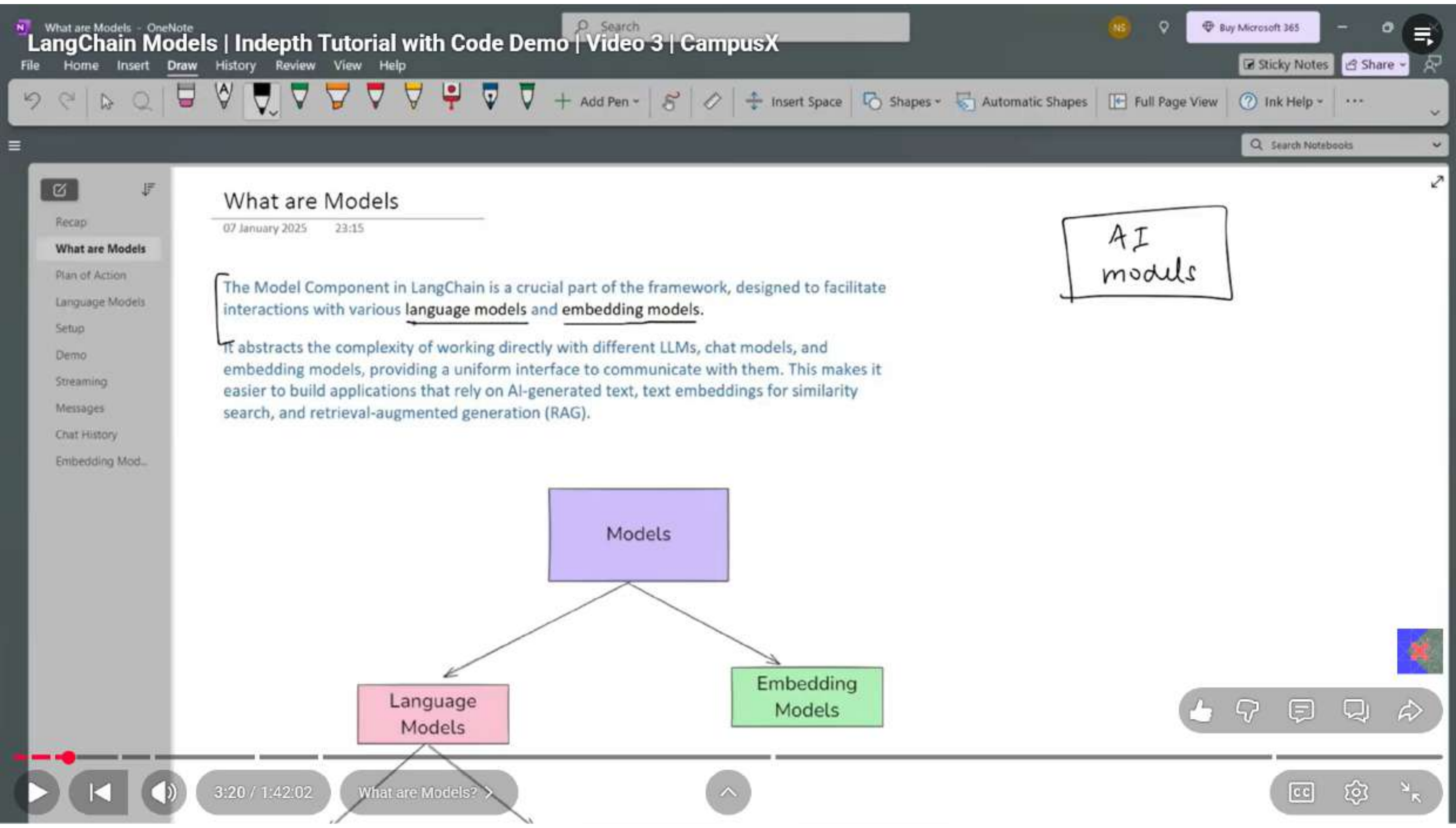


LangChain Models | Indepth Tutorial with Code Demo | Video 3 | CampusX

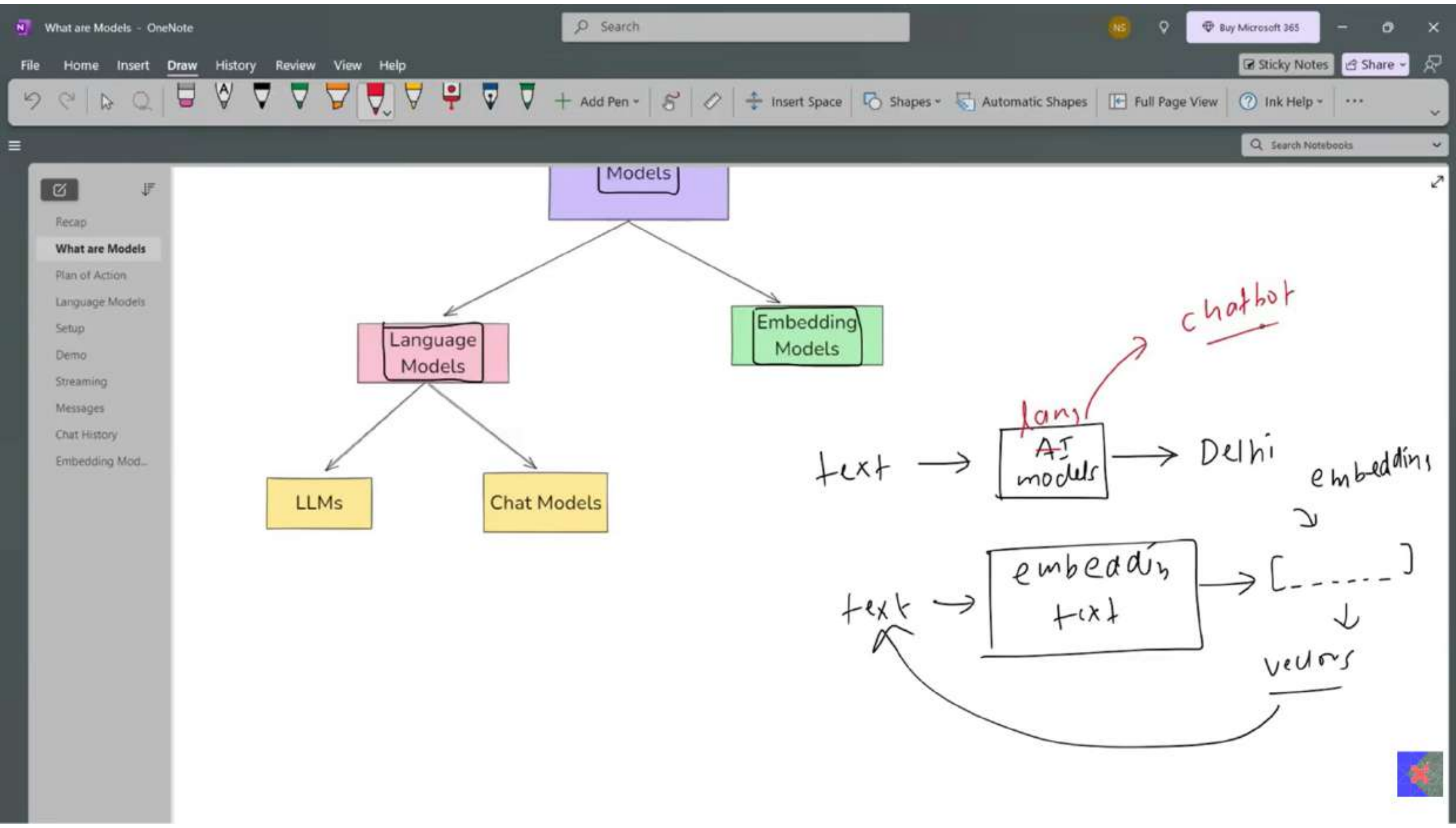
What are Models

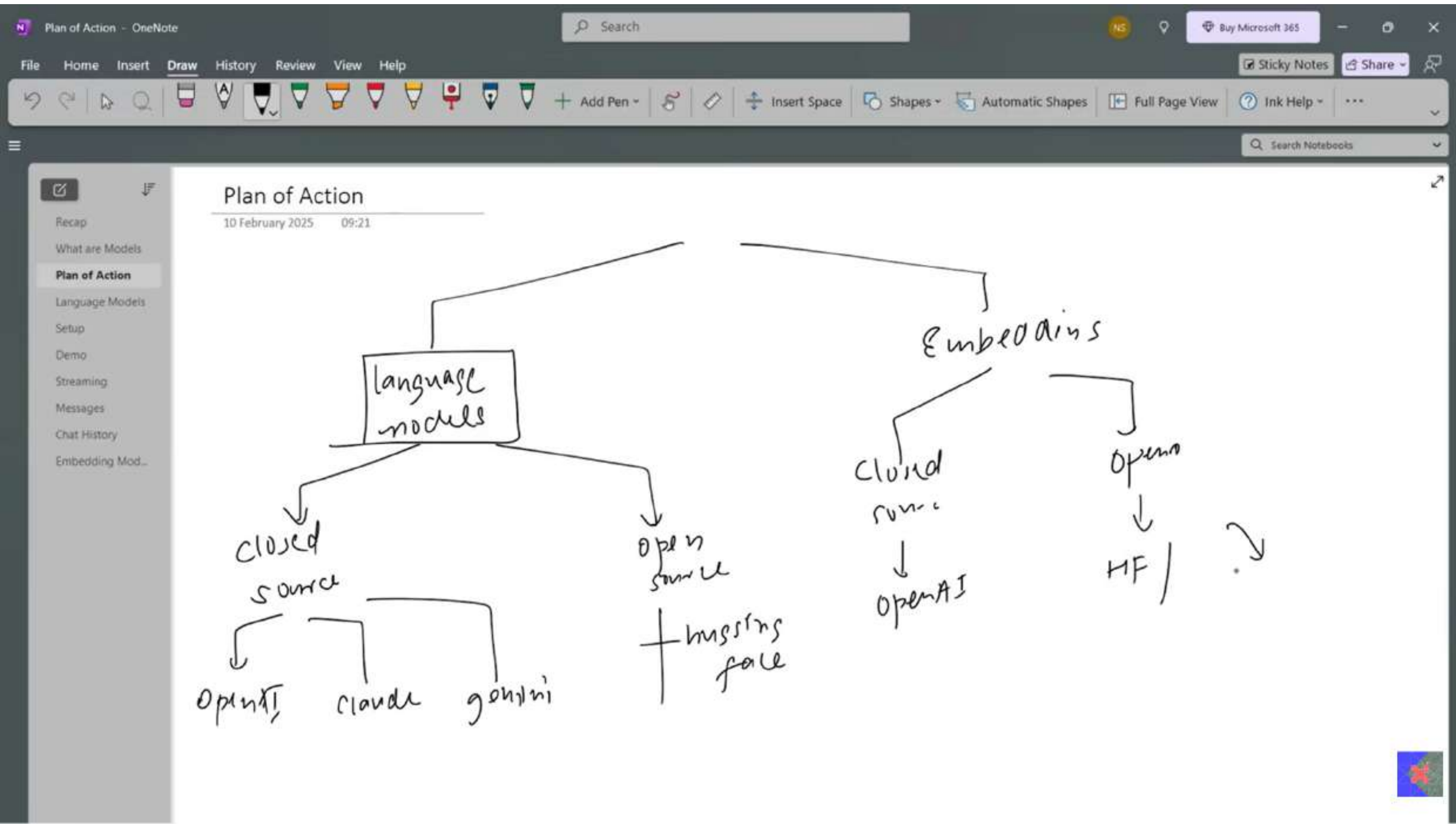
07 January 2025 23:15

The Model Component in LangChain is a crucial part of the framework, designed to facilitate interactions with various language models and embedding models.

It abstracts the complexity of working directly with different LLMs, chat models, and embedding models, providing a uniform interface to communicate with them. This makes it easier to build applications that rely on AI-generated text, text embeddings for similarity search, and retrieval-augmented generation (RAG).

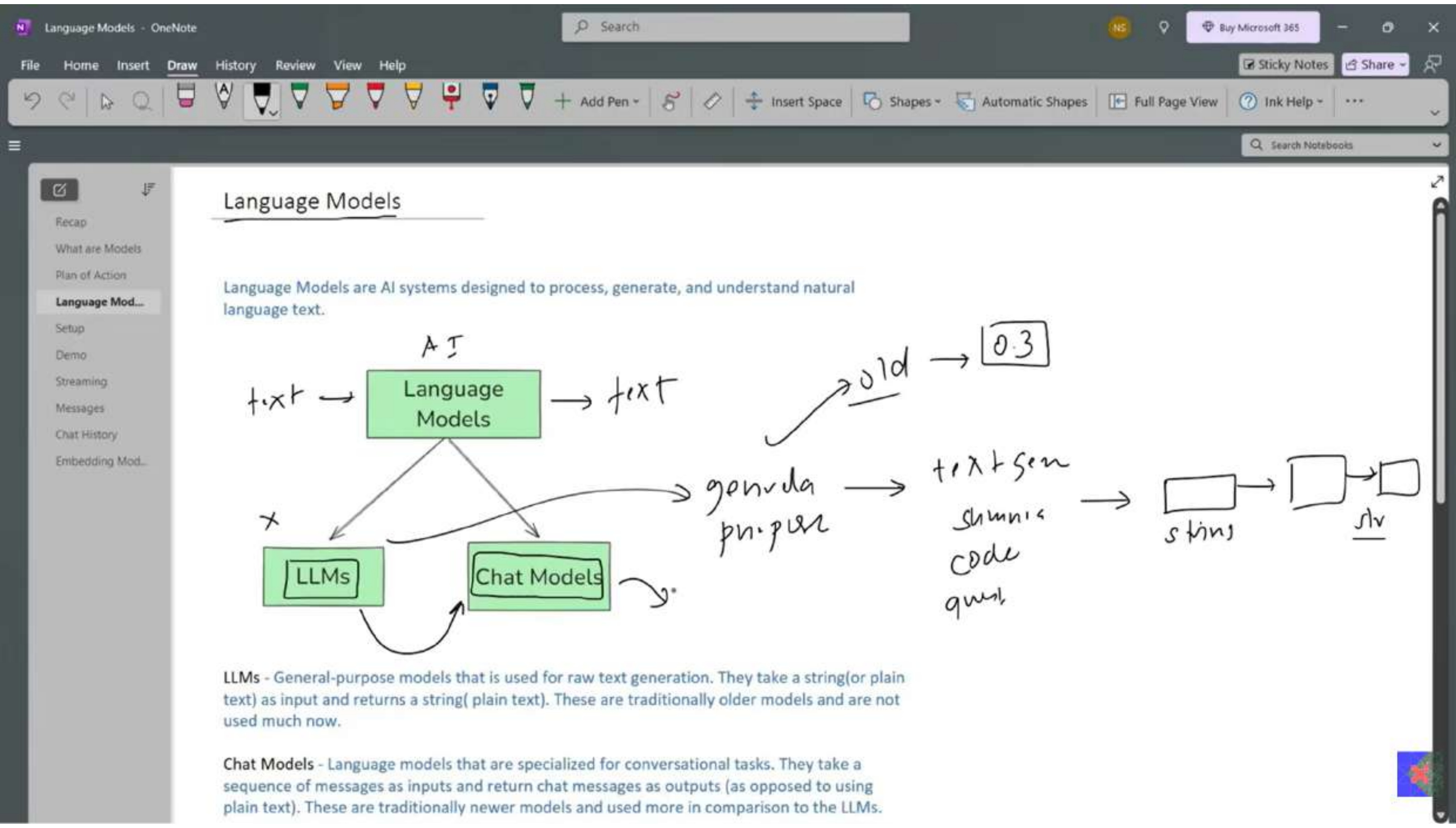
AI models


What are Models - OneNote
Search
Buy Microsoft 365
File
Home
Insert
Draw
History
Review
View
Help
Sticky Notes
Share
Add Pen
Insert Space
Shapes
Automatic Shapes
Full Page View
Ink Help
Search Notebooks
Recap
What are Models
Plan of Action
Language Models
Setup
Demo
Streaming
Messages
Chat History
Embedding Mod...
Models
Language Models
Embedding Models
LLMs
Chat Models
chatbot
langs
~~AI~~ models
text → → Delhi
embeddings
text → embeddin text → [------]
vectors



Plan of Action

10 February 2025 09:21

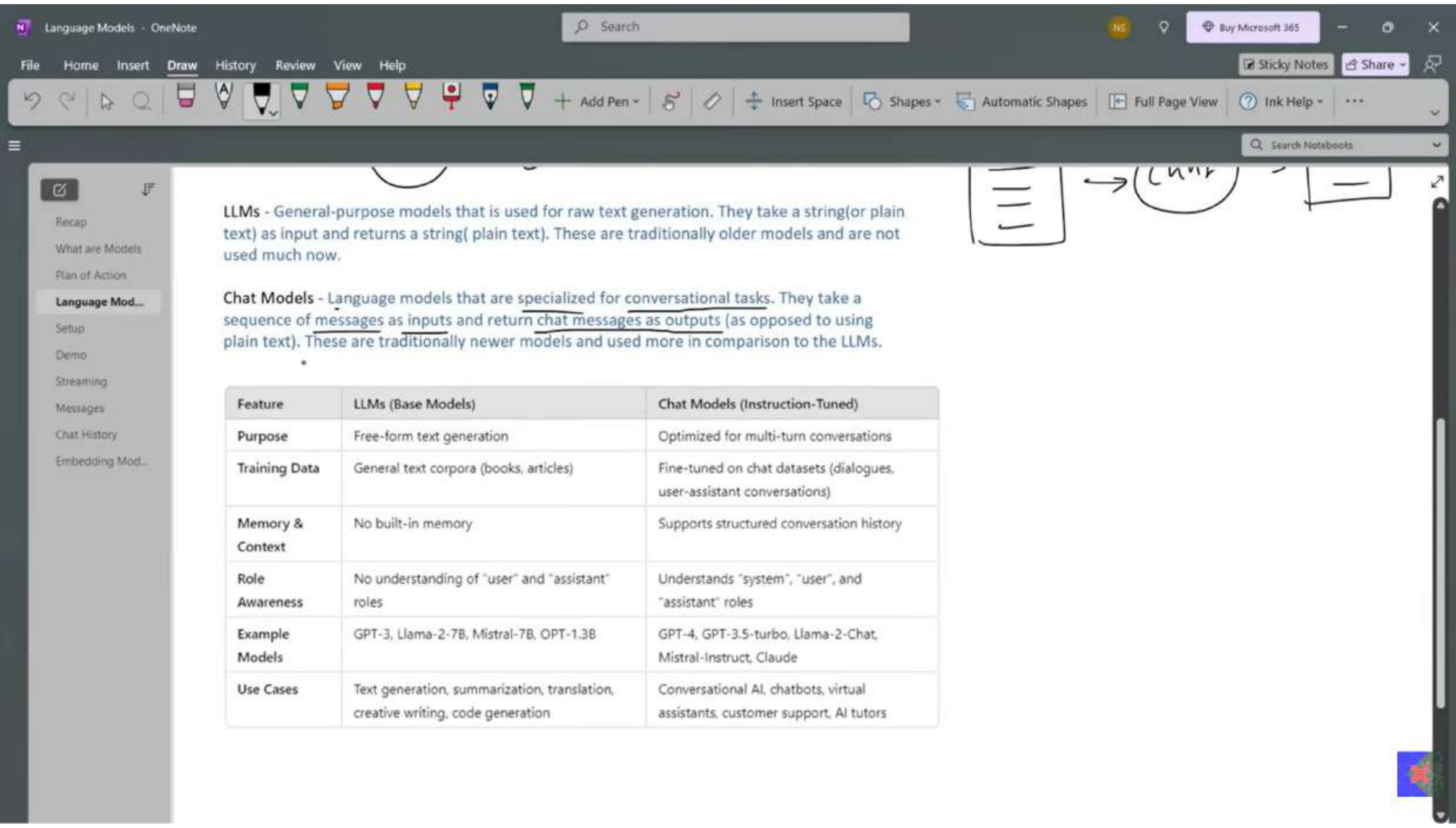


Language Models

Language Models are AI systems designed to process, generate, and understand natural language text.

LLMs - General-purpose models that is used for raw text generation. They take a string(or plain text) as input and returns a string(plain text). These are traditionally older models and are not used much now.

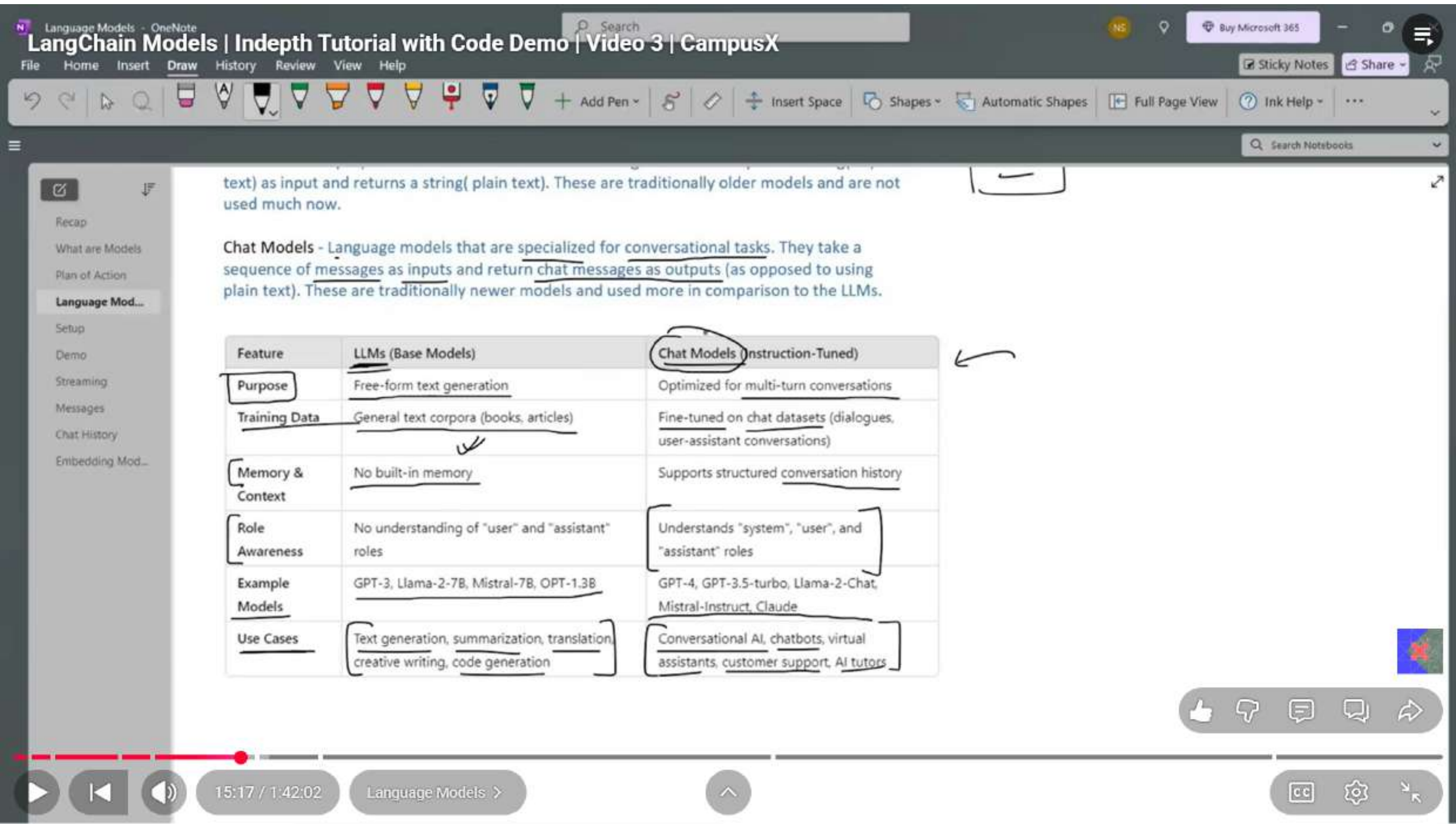
Chat Models - Language models that are specialized for conversational tasks. They take a sequence of messages as inputs and return chat messages as outputs (as opposed to using plain text). These are traditionally newer models and used more in comparison to the LLMs.



LLMs - General-purpose models that is used for raw text generation. They take a string(or plain text) as input and returns a string(plain text). These are traditionally older models and are not used much now.

Chat Models - Language models that are specialized for conversational tasks. They take a sequence of messages as inputs and return chat messages as outputs (as opposed to using plain text). These are traditionally newer models and used more in comparison to the LLMs.

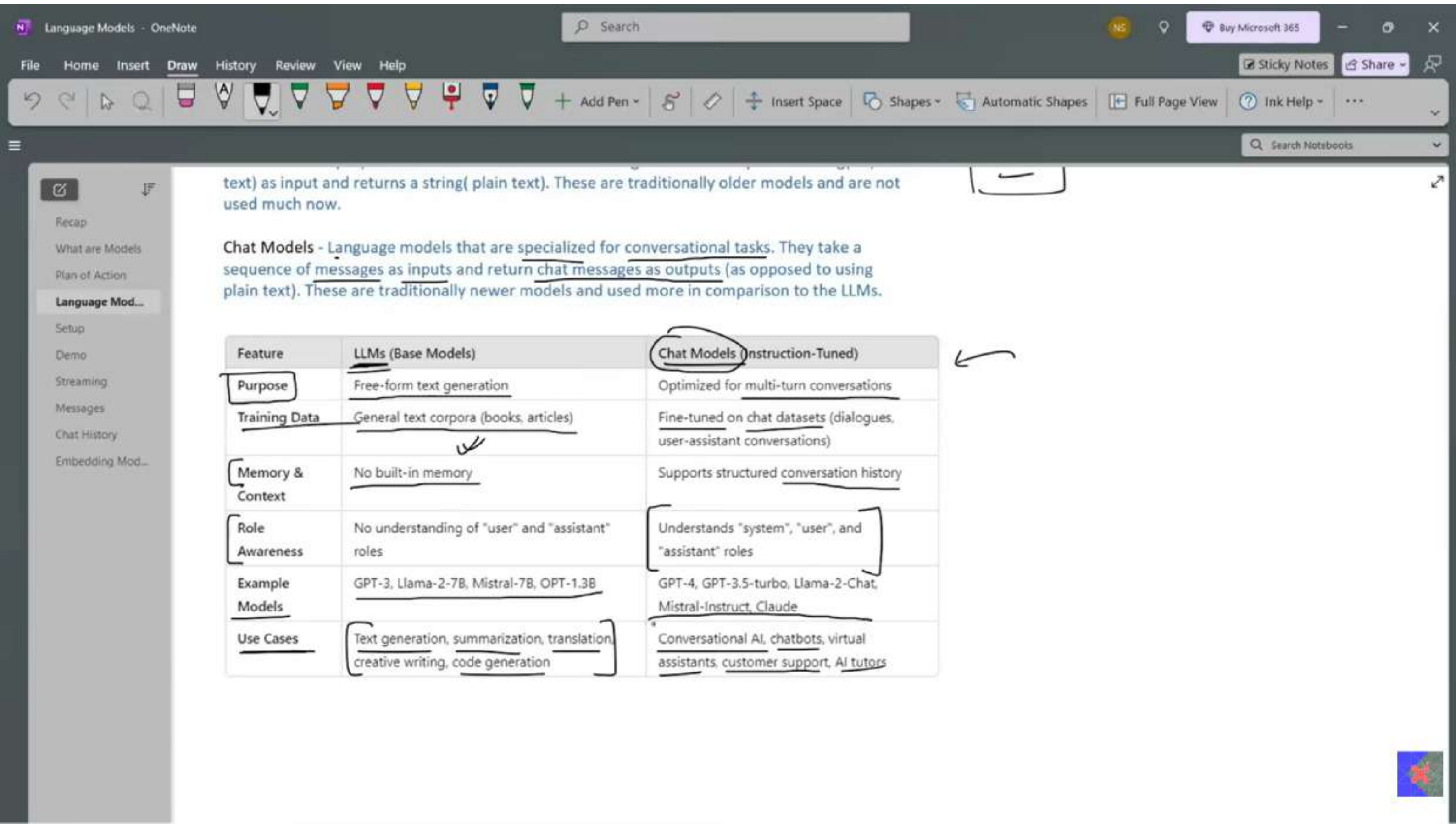
Feature	LLMs (Base Models)	Chat Models (Instruction-Tuned)
Purpose	Free-form text generation	Optimized for multi-turn conversations
Training Data	General text corpora (books, articles)	Fine-tuned on chat datasets (dialogues, user-assistant conversations)
Memory & Context	No built-in memory	Supports structured conversation history
Role Awareness	No understanding of "user" and "assistant" roles	Understands "system", "user", and "assistant" roles
Example Models	GPT-3, Llama-2-7B, Mistral-7B, OPT-1.3B	GPT-4, GPT-3.5-turbo, Llama-2-Chat, Mistral-Instruct, Claude
Use Cases	Text generation, summarization, translation, creative writing, code generation	Conversational AI, chatbots, virtual assistants, customer support, AI tutors



text) as input and returns a string(plain text). These are traditionally older models and are not used much now.

Chat Models - Language models that are specialized for conversational tasks. They take a sequence of messages as inputs and return chat messages as outputs (as opposed to using plain text). These are traditionally newer models and used more in comparison to the LLMs.

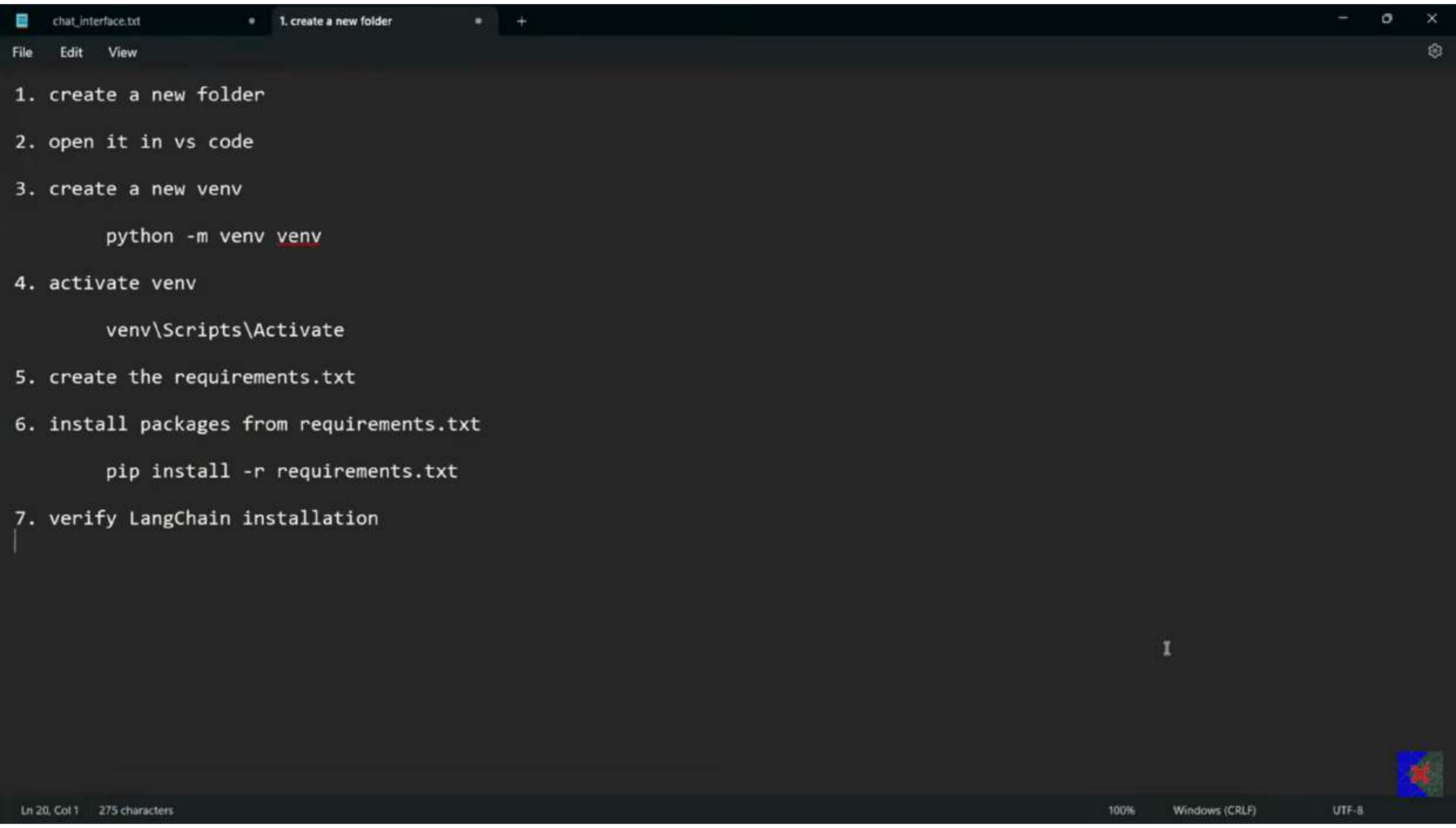
Feature	LLMs (Base Models)	Chat Models (Instruction-Tuned)
Purpose	Free-form text generation	Optimized for multi-turn conversations
Training Data	General text corpora (books, articles)	Fine-tuned on chat datasets (dialogues, user-assistant conversations)
Memory & Context	No built-in memory	Supports structured conversation history
Role Awareness	No understanding of "user" and "assistant" roles	Understands "system", "user", and "assistant" roles
Example Models	GPT-3, Llama-2-7B, Mistral-7B, OPT-1.3B	GPT-4, GPT-3.5-turbo, Llama-2-Chat, Mistral-Instruct, Claude
Use Cases	Text generation, summarization, translation, creative writing, code generation	Conversational AI, chatbots, virtual assistants, customer support, AI tutors



text) as input and returns a string(plain text). These are traditionally older models and are not used much now.

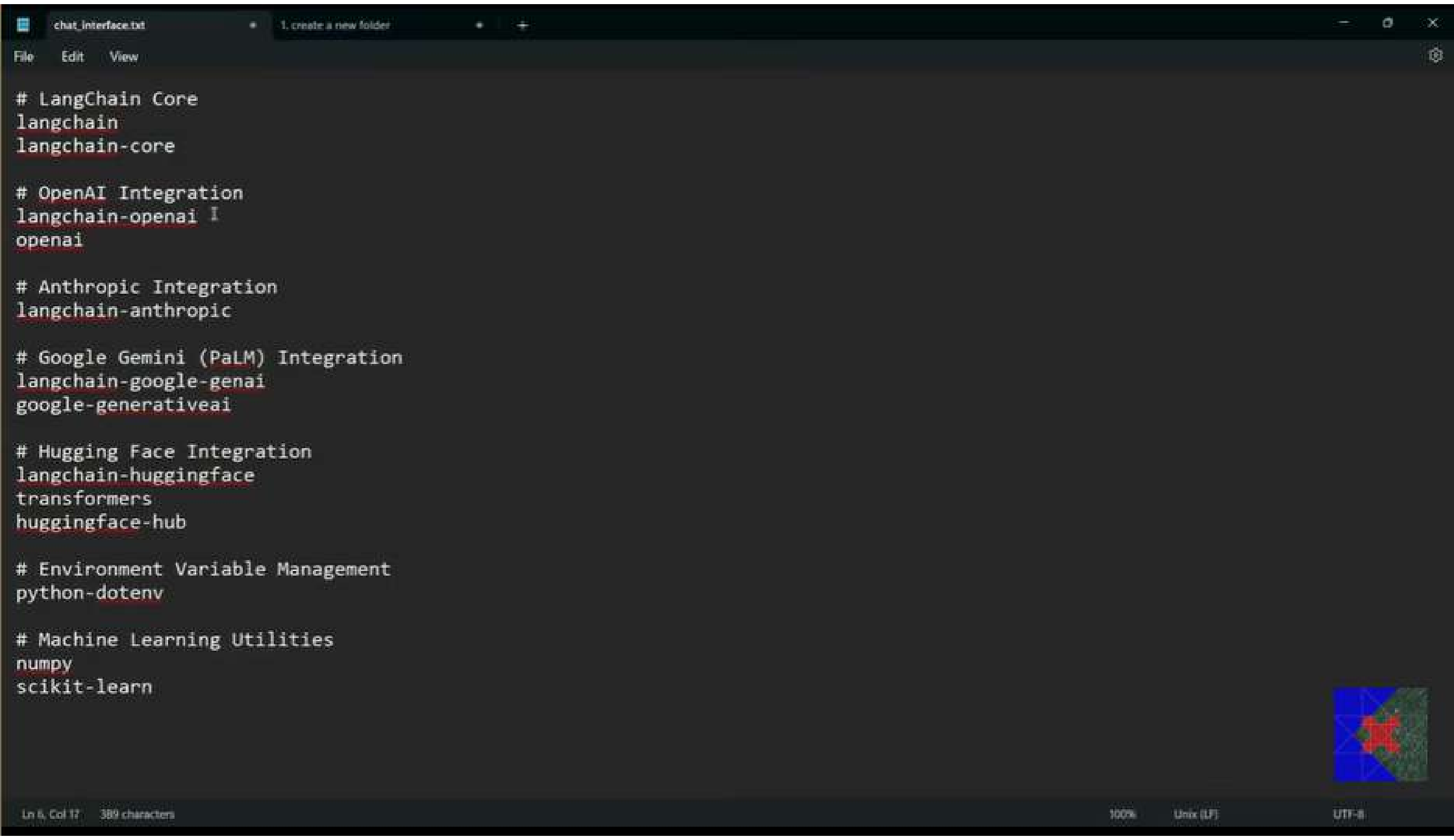
Chat Models - Language models that are specialized for conversational tasks. They take a sequence of messages as inputs and return chat messages as outputs (as opposed to using plain text). These are traditionally newer models and used more in comparison to the LLMs.

Feature	LLMs (Base Models)	Chat Models (Instruction-Tuned)
Purpose	Free-form text generation	Optimized for multi-turn conversations
Training Data	General text corpora (books, articles)	Fine-tuned on chat datasets (dialogues, user-assistant conversations)
Memory & Context	No built-in memory	Supports structured conversation history
Role Awareness	No understanding of "user" and "assistant" roles	Understands "system", "user", and "assistant" roles
Example Models	GPT-3, Llama-2-7B, Mistral-7B, OPT-1.3B	GPT-4, GPT-3.5-turbo, Llama-2-Chat, Mistral-Instruct, Claude
Use Cases	Text generation, summarization, translation, creative writing, code generation	Conversational AI, chatbots, virtual assistants, customer support, AI tutors

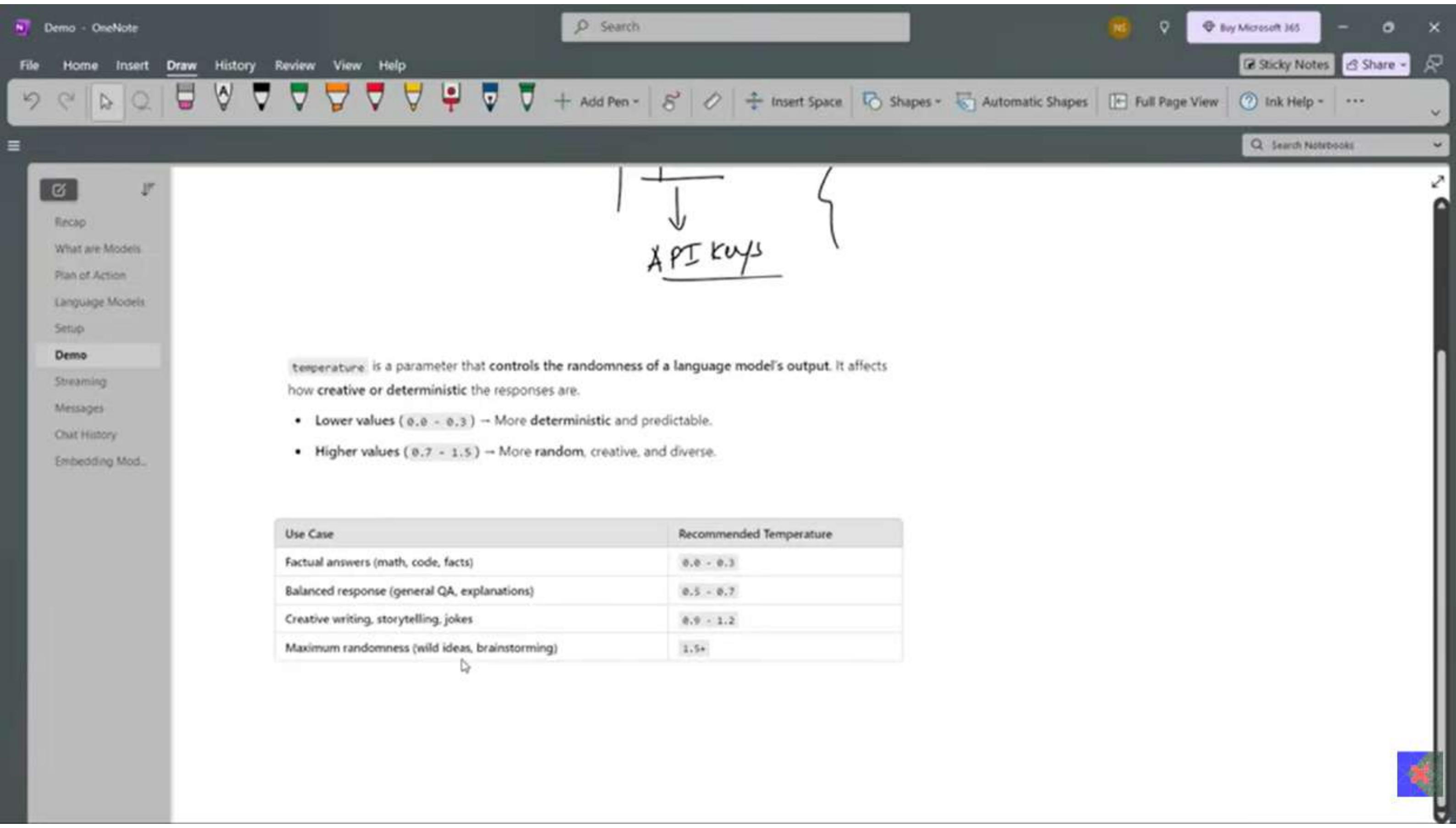


```
1. create a new folder

2. open it in vs code

3. create a new venv

        python -m venv venv

4. activate venv

        venv\Scripts\Activate

5. create the requirements.txt

6. install packages from requirements.txt

        pip install -r requirements.txt

7. verify LangChain installation
```

chat_interface.txt
1. create a new folder
File
Edit
View



```
# LangChain Core
langchain
langchain-core

# OpenAI Integration
langchain-openai
openai

# Anthropic Integration
langchain-anthropic

# Google Gemini (PaLM) Integration
langchain-google-genai
google-generativeai

# Hugging Face Integration
langchain-huggingface
transformers
huggingface-hub

# Environment Variable Management
python-dotenv

# Machine Learning Utilities
numpy
scikit-learn
```

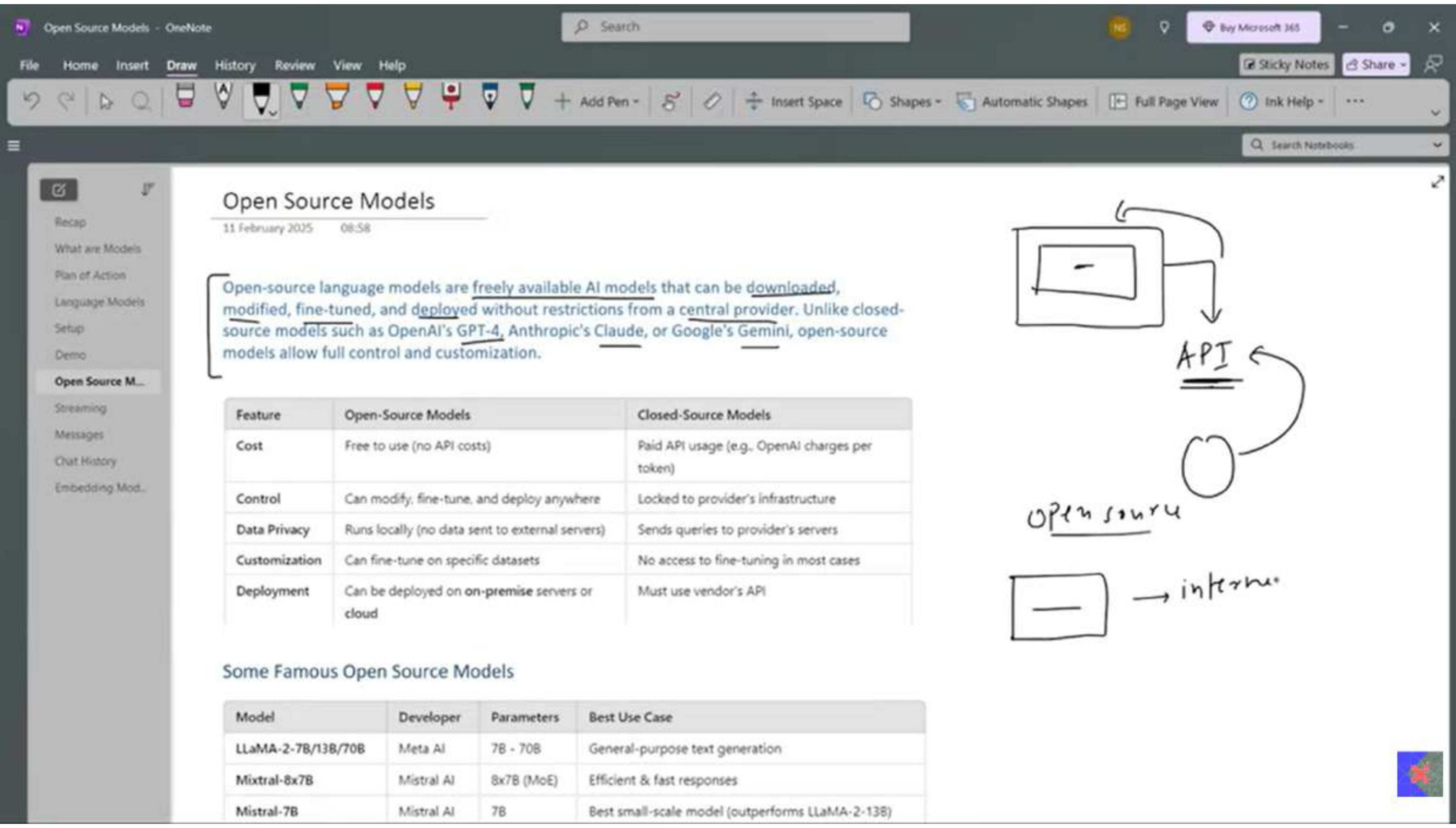


API keys

`temperature` is a parameter that **controls the randomness of a language model's output**. It affects how **creative or deterministic** the responses are.

- **Lower values** (`0.0 - 0.3`) → More **deterministic** and predictable.
- **Higher values** (`0.7 - 1.5`) → More **random**, creative, and diverse.

Use Case	Recommended Temperature
Factual answers (math, code, facts)	`0.0 - 0.3`
Balanced response (general QA, explanations)	`0.5 - 0.7`
Creative writing, storytelling, jokes	`0.9 - 1.2`
Maximum randomness (wild ideas, brainstorming)	`1.5+`



Open Source Models

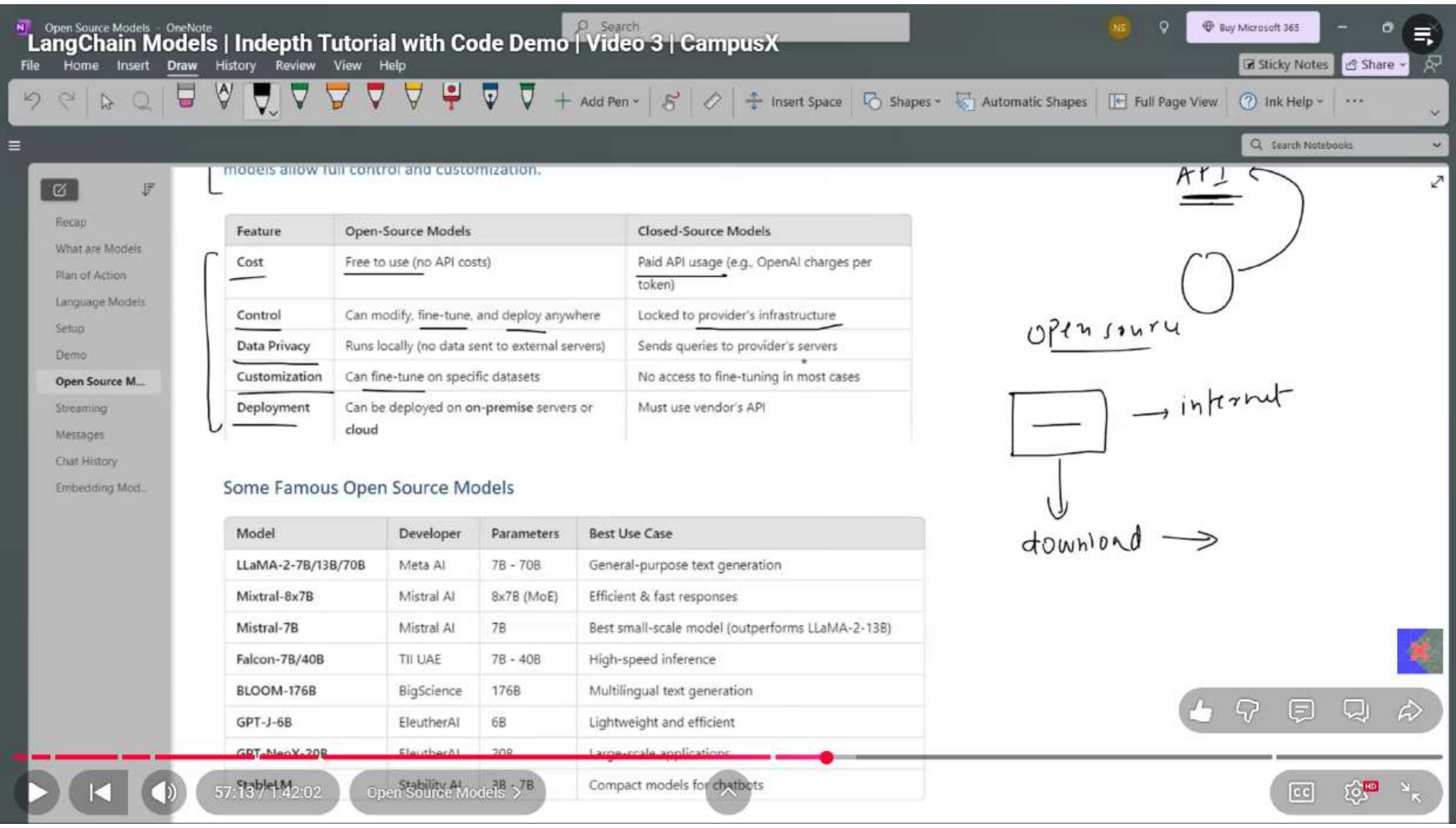
11 February 2025 08:58

Open-source language models are freely available AI models that can be downloaded, modified, fine-tuned, and deployed without restrictions from a central provider. Unlike closed-source models such as OpenAI's GPT-4, Anthropic's Claude, or Google's Gemini, open-source models allow full control and customization.

Feature	Open-Source Models	Closed-Source Models
Cost	Free to use (no API costs)	Paid API usage (e.g., OpenAI charges per token)
Control	Can modify, fine-tune, and deploy anywhere	Locked to provider's infrastructure
Data Privacy	Runs locally (no data sent to external servers)	Sends queries to provider's servers
Customization	Can fine-tune on specific datasets	No access to fine-tuning in most cases
Deployment	Can be deployed on **on-premise** servers or **cloud**	Must use vendor's API

Some Famous Open Source Models

Model	Developer	Parameters	Best Use Case
LLaMA-2-7B/13B/70B	Meta AI	7B - 70B	General-purpose text generation
Mixtral-8x7B	Mistral AI	8x7B (MoE)	Efficient & fast responses
Mistral-7B	Mistral AI	7B	Best small-scale model (outperforms LLaMA-2-13B)



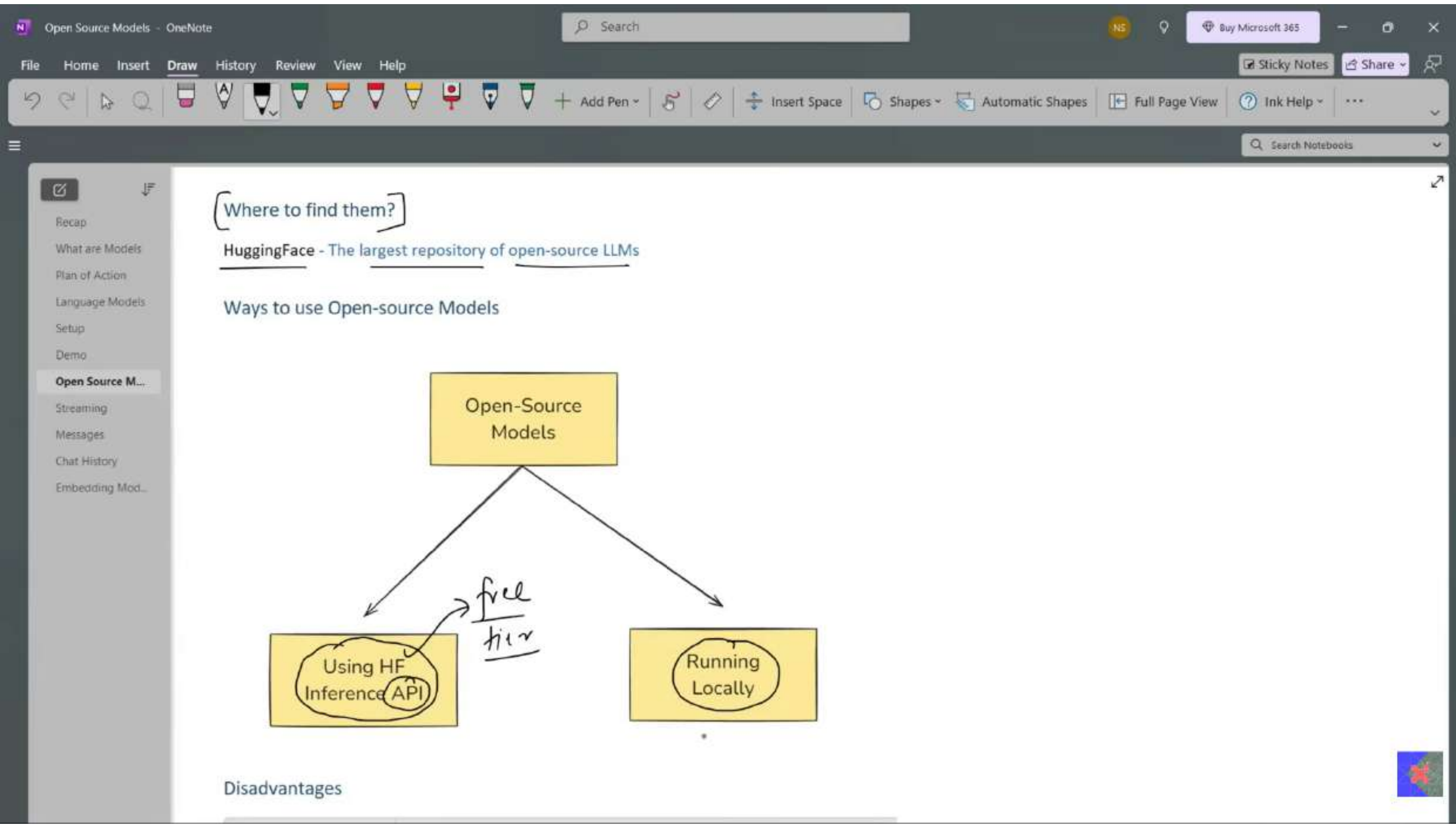
LangChain Models | Indepth Tutorial with Code Demo | Video 3 | CampusX

models allow full control and customization.

Feature	Open-Source Models	Closed-Source Models
Cost	Free to use (no API costs)	Paid API usage (e.g., OpenAI charges per token)
Control	Can modify, fine-tune, and deploy anywhere	Locked to provider's infrastructure
Data Privacy	Runs locally (no data sent to external servers)	Sends queries to provider's servers
Customization	Can fine-tune on specific datasets	No access to fine-tuning in most cases
Deployment	Can be deployed on **on-premise** servers or **cloud**	Must use vendor's API

Some Famous Open Source Models

Model	Developer	Parameters	Best Use Case
LLaMA-2-7B/13B/70B	Meta AI	7B - 70B	General-purpose text generation
Mixtral-8x7B	Mistral AI	8x7B (MoE)	Efficient & fast responses
Mistral-7B	Mistral AI	7B	Best small-scale model (outperforms LLaMA-2-13B)
Falcon-7B/40B	TII UAE	7B - 40B	High-speed inference
BLOOM-176B	BigScience	176B	Multilingual text generation
GPT-J-6B	EleutherAI	6B	Lightweight and efficient
GPT-NeoX-20B	EleutherAI	20B	Large-scale applications
StableLM	Stability AI	3B - 7B	Compact models for chatbots



Where to find them?

HuggingFace - The largest repository of open-source LLMs

Ways to use Open-source Models

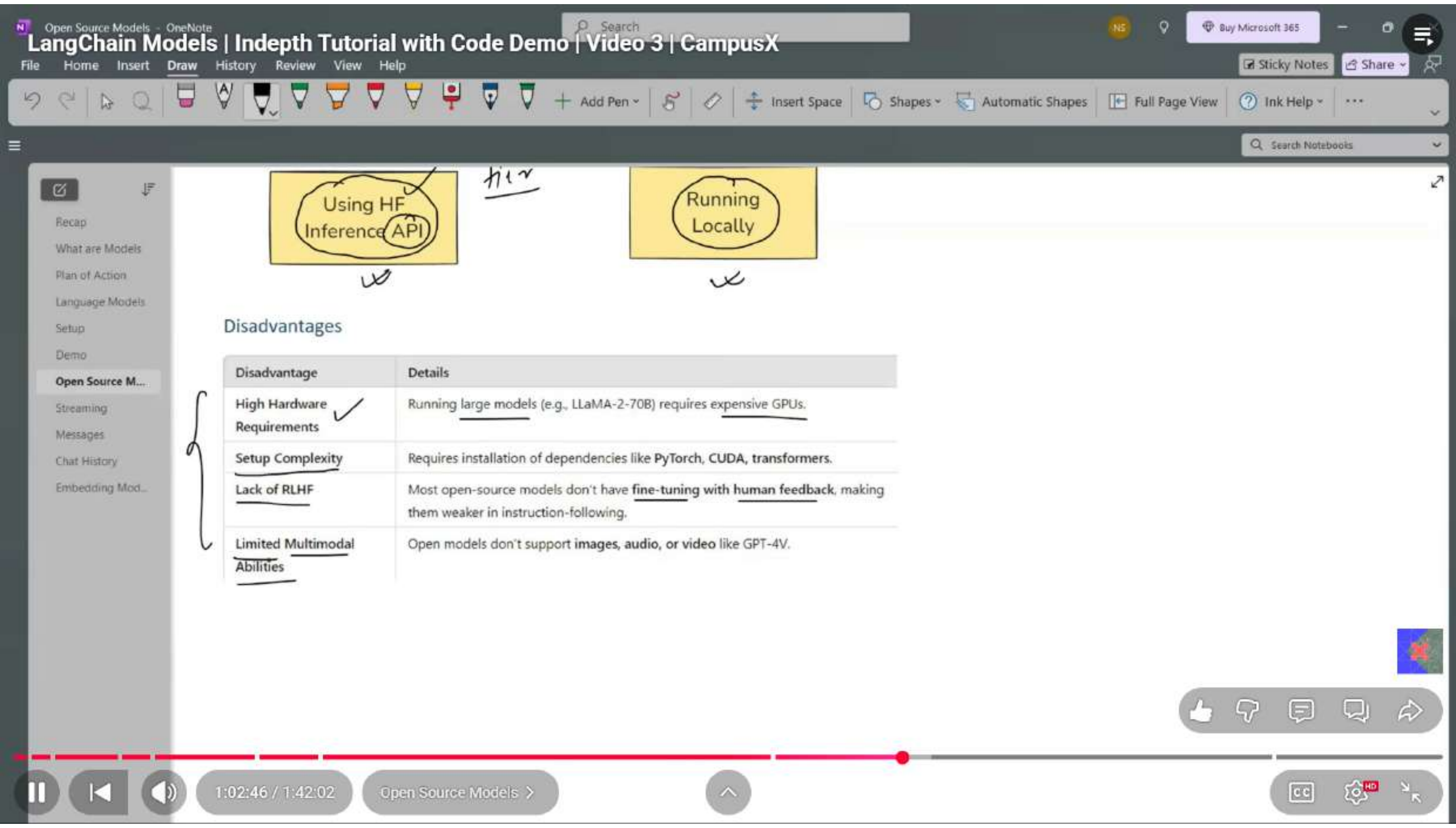
Open-Source Models

Using HF Inference API

free tier

Running Locally

Disadvantages



Disadvantages

Disadvantage	Details
High Hardware Requirements ✓	Running large models (e.g., LLaMA-2-70B) requires expensive GPUs.
Setup Complexity	Requires installation of dependencies like **PyTorch, CUDA, transformers**.
Lack of RLHF	Most open-source models don't have **fine-tuning with human feedback**, making them weaker in instruction-following.
Limited Multimodal Abilities	Open models don't support **images, audio, or video** like GPT-4V.