

CS 696 Applied Large Language Models  
Spring Semester, 2025  
Doc 4 Running Models  
Jan 23, 2025


Copyright ©, All rights reserved. 2025 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/openpub/>) license defines the copyright on this document.

# Accessing via Jupyter

<https://jupyterlab.readthedocs.io/en/stable/index.html>

```
conda install conda-forge::transformers
```

# Jupyter Lab



The screenshot shows a Jupyter Lab interface with three open notebooks: 'HandsOn.ipynb', 'mlx.ipynb', and 'Class Demo.ipynb'. The 'mlx.ipynb' notebook is active and displays the following code in a cell:

```
[4]: from transformers import AutoModelForCausalLM, AutoTokenizer

# Load model and tokenizer
model = AutoModelForCausalLM.from_pretrained(
    "microsoft/Phi-3-mini-4k-instruct",
    #device_map="auto",
    attn_implementation='eager',
    torch_dtype="auto",
    trust_remote_code=True,
)
tokenizer = AutoTokenizer.from_pretrained("microsoft/Phi-3-min.
```

Below the code cell, a progress bar indicates the loading status: "Loading checkpoint shards: 100% 2/2 [00:00<00:00, 4.22it/s]".

The next cell contains the following code:

```
[2]: from transformers import pipeline

# Create a pipeline
generator = pipeline(
    "text-generation",
    model=model,
    tokenizer=tokenizer,
    return_full_text=True,
    max_new_tokens=500,
    do_sample=False
)
```

A pink notification bar below the code cell states: "Device set to use mps:0".

The third cell contains the following code:

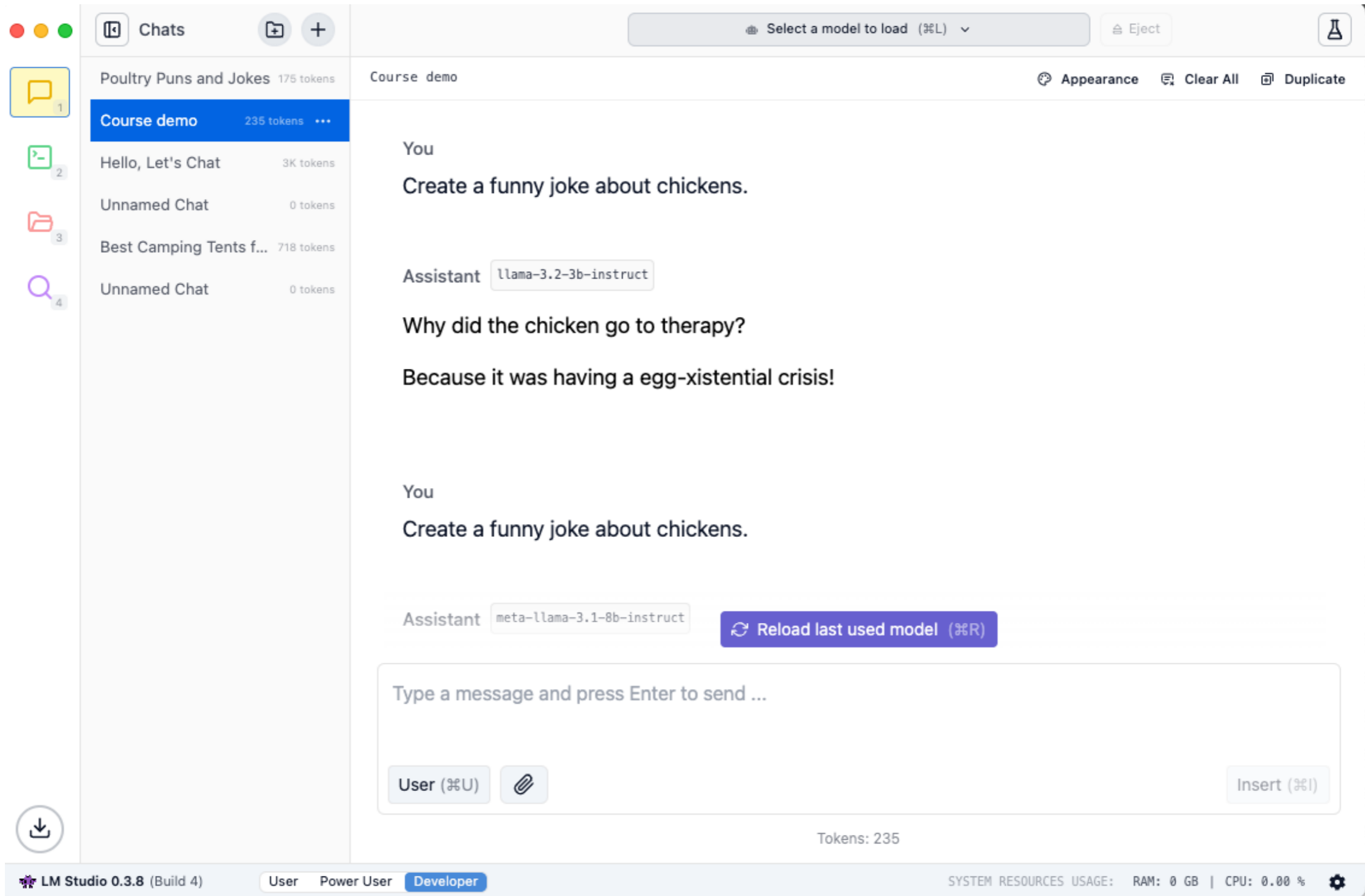
```
[3]: # The prompt (user input / query)
messages = [
```

# Jupyter Lab

The screenshot displays the Jupyter Lab interface with three open notebooks: `HandsOn.ipynb`, `mlx.ipynb`, and `Class Demo.ipynb`. The `Class Demo.ipynb` notebook is the active one. It contains the following code cells:

- Cell [ ]: (Empty code cell)
- Cell [2]: `x + y` (Code cell) with output `5` (Text output)
- Cell [1]: `x = 3`  
`y = 2` (Code cell)

The interface includes a toolbar with icons for saving, adding, deleting, and running code, and a dropdown menu for the kernel, which is currently set to `Python 3 (ipykernel)`.



# mlx\_lm

<https://pypi.org/project/mlx-lm/>

For macOS

Python

```
from mlx_lm import load, generate

model, tokenizer = load("mlx-community/Mistral-7B-Instruct-v0.3-4bit")

prompt = "Write a story about Einstein"

messages = [{"role": "user", "content": prompt}]
prompt = tokenizer.apply_chat_template(
    messages, add_generation_prompt=True
)

text = generate(model, tokenizer, prompt=prompt, verbose=True)
```

# mlx\_lm

<https://pypi.org/project/mlx-lm/>

For macOS

Python - Streaming

```
from mlx_lm import load, generate

model, tokenizer = load("mlx-community/Mistral-7B-Instruct-v0.3-4bit")

prompt = "Write a story about Einstein"

messages = [{"role": "user", "content": prompt}]
prompt = tokenizer.apply_chat_template(
    messages, add_generation_prompt=True
)

text = generate(model, tokenizer, prompt=prompt, verbose=True)
```

# mlx\_lm

<https://pypi.org/project/mlx-lm/>

Command line

mlx\_lm.generate

```
--model /Users/rwhitney/.cache/lm-studio/models/mlx-communityLlama-3.2-3B-Instruct-4bit  
--prompt "hello"
```

Prompt: <|begin\_of\_text|><|start\_header\_id|>system<|end\_header\_id|>

Cutting Knowledge Date: December 2023

Today Date: 22 Jan 2025

<|eot\_id|><|start\_header\_id|>user<|end\_header\_id|>

hello<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

Hello! How can I assist you today?

=====

Prompt: 37 tokens, 41.083 tokens-per-sec

Generation: 10 tokens, 44.464 tokens-per-sec

Peak memory: 1.856 GB



# mlx\_lm

<https://pypi.org/project/mlx-lm/>

Command line

```
mlx_lm.chat
```

```
--model mlx-communityLlama-3.2-3B-Instruct-4bit
```

```
Fetching 6 files: 100%| | 6/6 [00:00<00:00, 6267.95it/s]
```

```
[INFO] Starting chat session with mlx-community/Llama-3.2-3B-Instruct-4bit. To exit, enter 'q'.
```

```
>> Write a story about Einstein
```

```
It was a crisp winter morning in Princeton, New Jersey, and Albert Einstein was sitting in his favorite armchair, sipping a cup of coffee and staring out the window at the snow-covered trees. He had spent the previous night lost in thought, pondering the mysteries of the universe.
```

# mlx\_lm

<https://pypi.org/project/mlx-lm/>

```
def loss_fn(w, x, y):  
    return mx.mean(mx.square(w * x - y))
```

```
w = mx.array(1.0)
```

```
x = mx.array([0.5, -0.5])
```

```
y = mx.array([1.5, -1.5])
```

```
# Computes the gradient of loss_fn with respect to w:
```

```
grad_fn = mx.grad(loss_fn)
```

```
dloss_dw = grad_fn(w, x, y)
```

```
# Prints array(-1, dtype=float32)
```

```
print(dloss_dw)
```

```
# To get the gradient with respect to x we can do:
```

```
grad_fn = mx.grad(loss_fn, argnums=1)
```

```
dloss_dx = grad_fn(w, x, y)
```

```
# Prints array([-1, 1], dtype=float32)
```

```
print(dloss_dx)
```