An Introduction to Hugging Face 

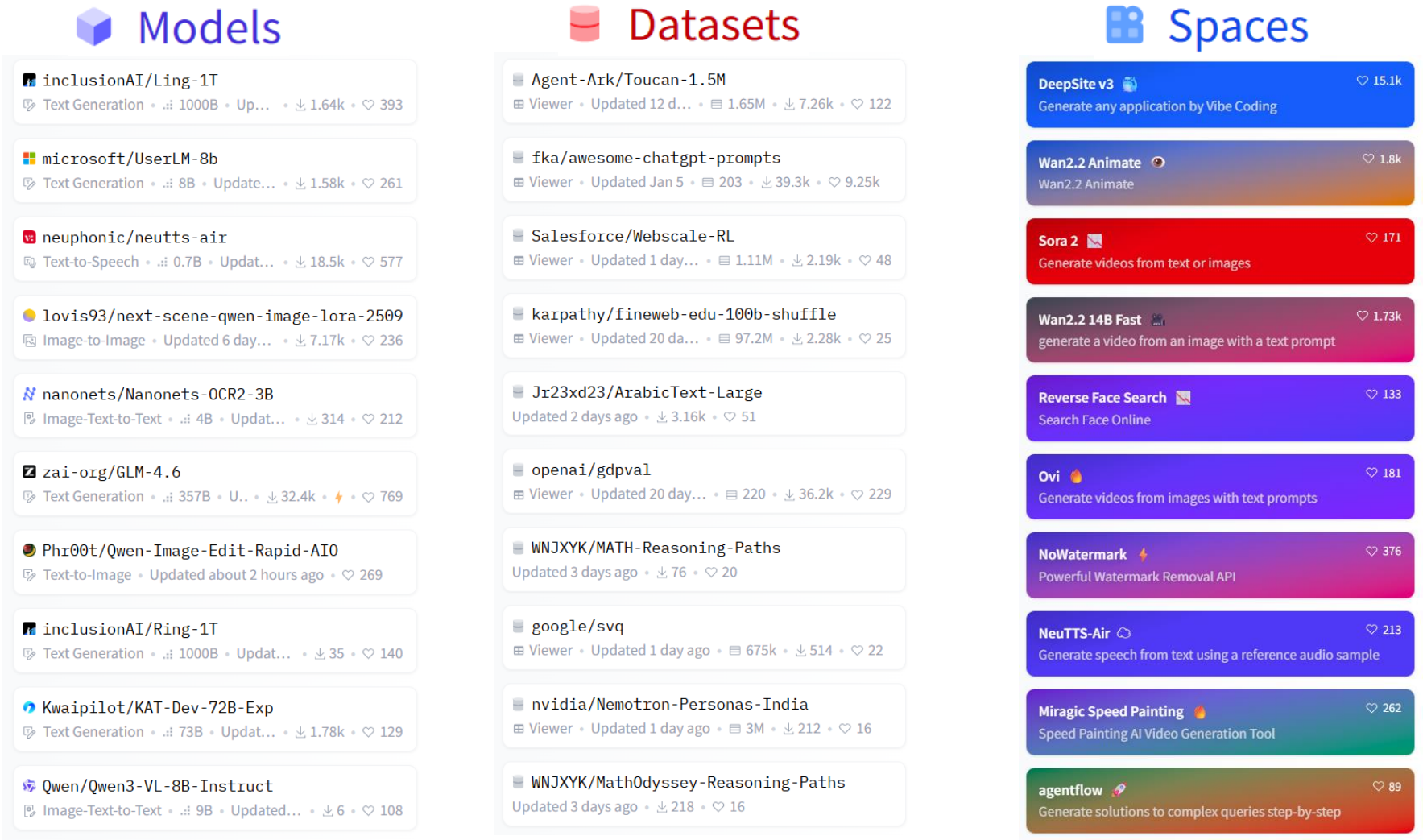
Guangzhi Xiong
Oct 16, 2025

The Hugging Face Hub: The GitHub of Machine Learning

- What is Hugging Face Hub?
  - The Hugging Face Hub is a platform with over 2M **models**, 500k **datasets**, and 1M demo apps (**Spaces**), all *open source and publicly available*, in an online platform where people can easily collaborate and build ML together.
- How to access it?
  - Link: https://huggingface.co

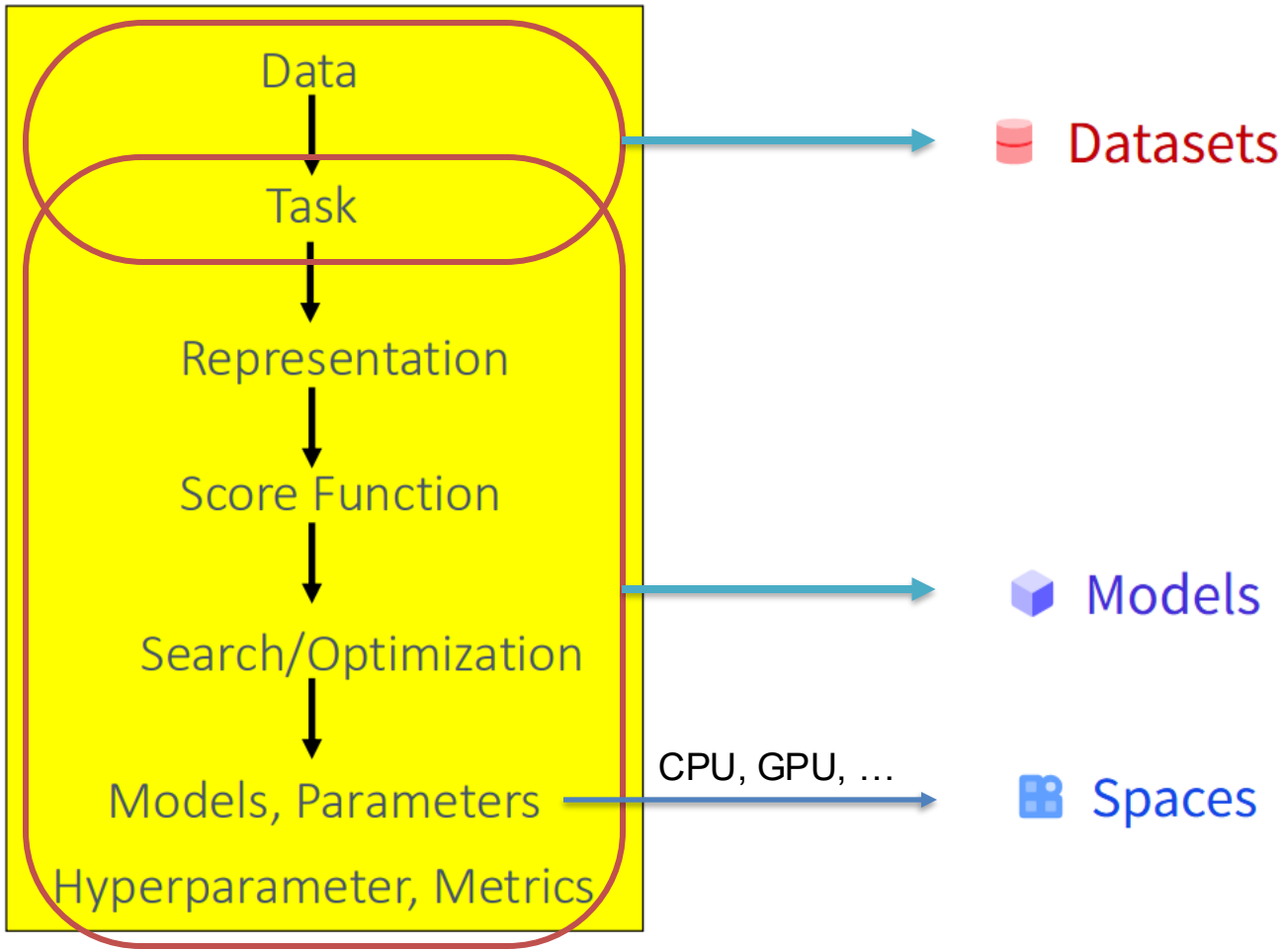
The Hugging Face Hub: The GitHub of Machine Learning

- What can we find on the Hub?



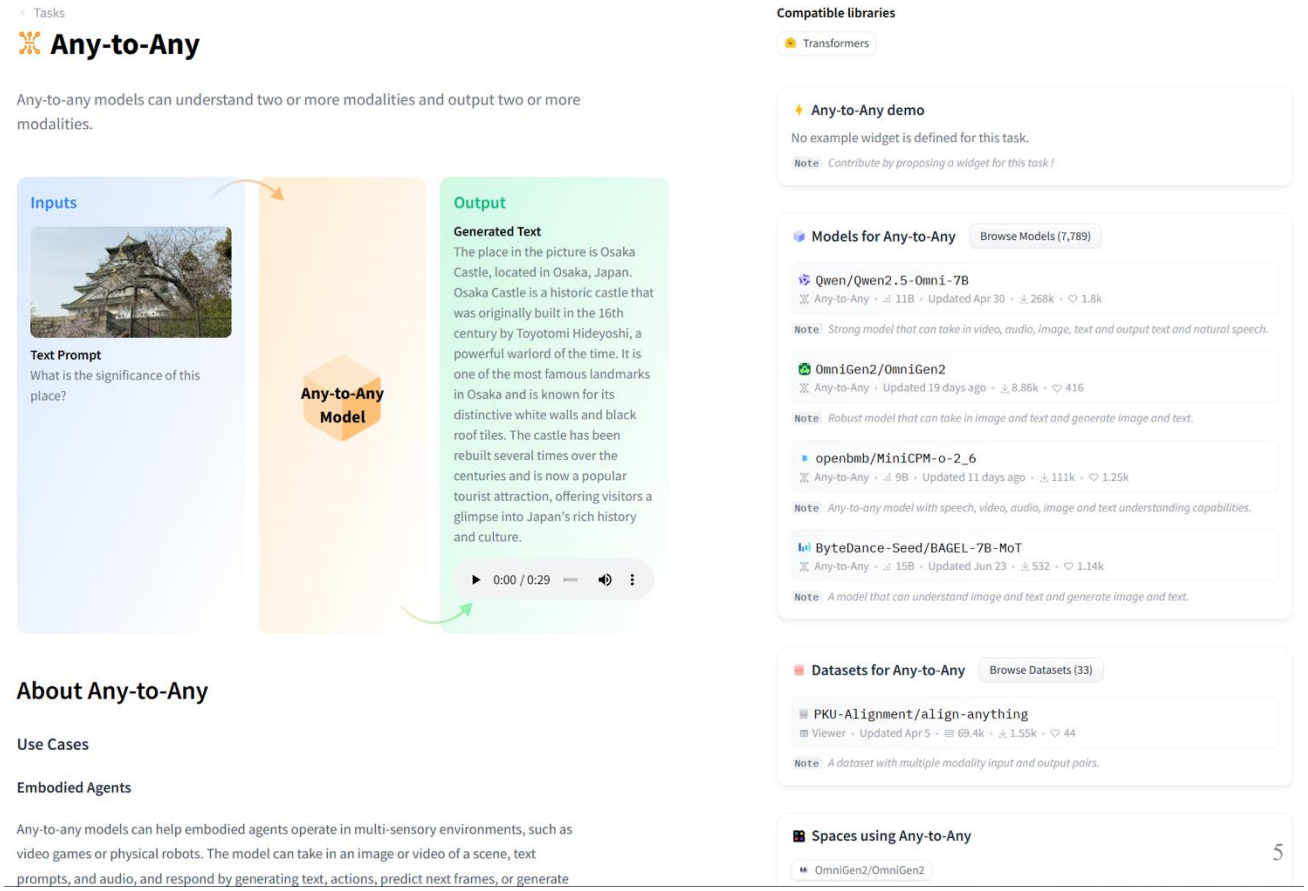
The Hugging Face Hub: The GitHub of Machine Learning

- Machine Learning in a Nutshell



Tasks in Hugging Face

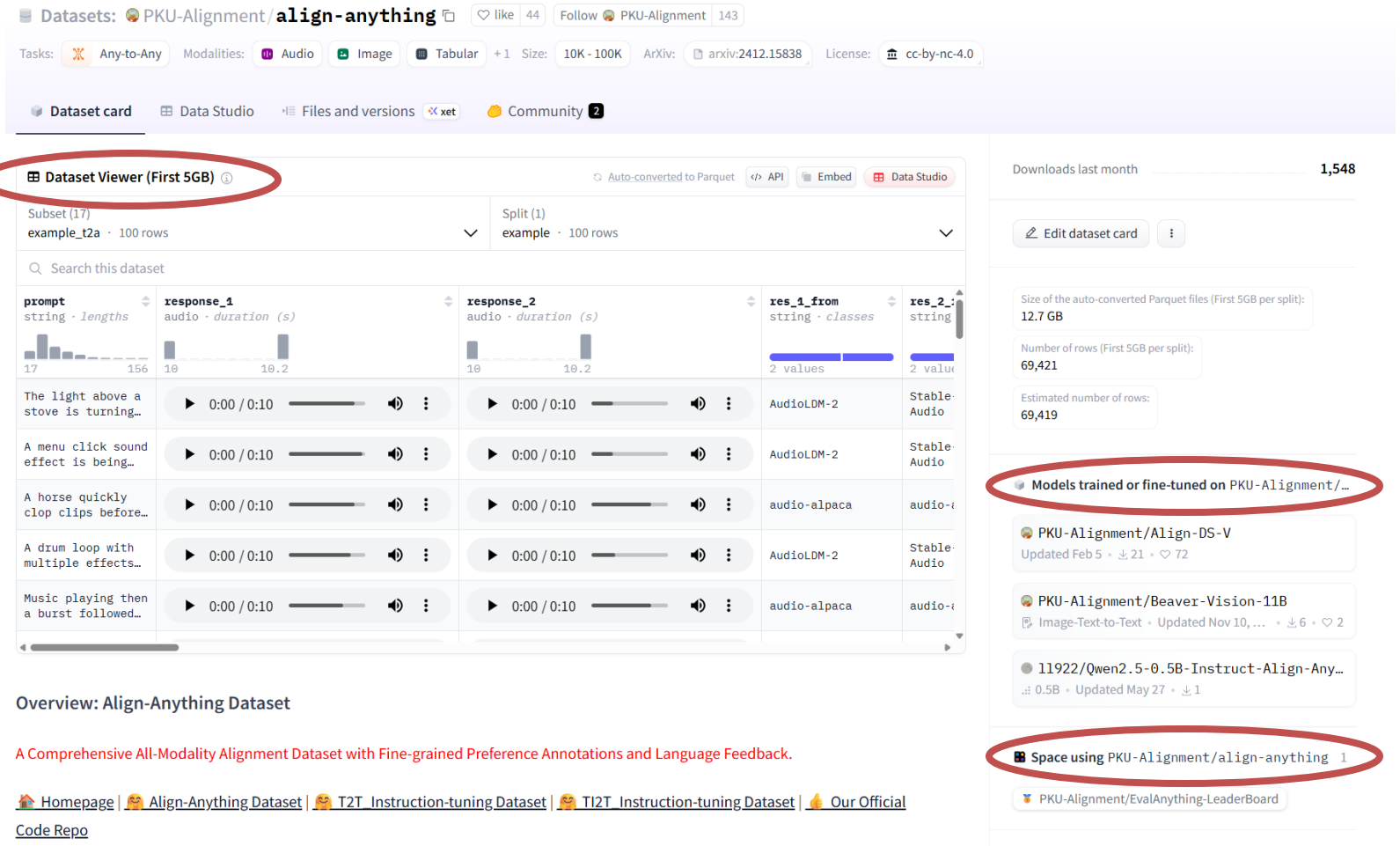
- Hugging Face is the home for all Machine Learning tasks.
  - Natural Language Processing: *Text Classification, Text Generation, …*
  - Computer Vision: *Image Classification, Text-to-Image, Object Detection, …*
  - …



https://huggingface.co/tasks

Hugging Face: Datasets

- Let's use ***Align-Anything Dataset*** as an example



Hugging Face: Datasets

- How to load datasets from Hugging Face?
  - Start by installing 🤗 Datasets: `pip install datasets`
  - Load the dataset by providing the load_dataset() function with the dataset *name*, dataset *configuration*, and dataset *split*:

```python
from datasets import load_dataset

# text-to-text
train_dataset = load_dataset('PKU-Alignment/align-anything',name='text-to-text')['train']
val_dataset = load_dataset('PKU-Alignment/align-anything',name='text-to-text')['val']

# text-image-to-text
train_dataset = load_dataset('PKU-Alignment/align-anything',name='text-image-to-text')['train']
val_dataset = load_dataset('PKU-Alignment/align-anything', name='text-image-to-text')['val']
```

Hugging Face: Datasets

- How to load datasets from Hugging Face?
  - Another example with no dataset configuration

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes")
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 8530
    })
    validation: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
})
```

Hugging Face: Datasets

- How to read instances from the loaded dataset?
  - Indexing by row

```
# Get the first row in the dataset
>>> dataset[0]
{'label': 1,
 'text': 'the rock is destined to be the 21st century\'s new " conan " and that he\'s going to make a spl
```

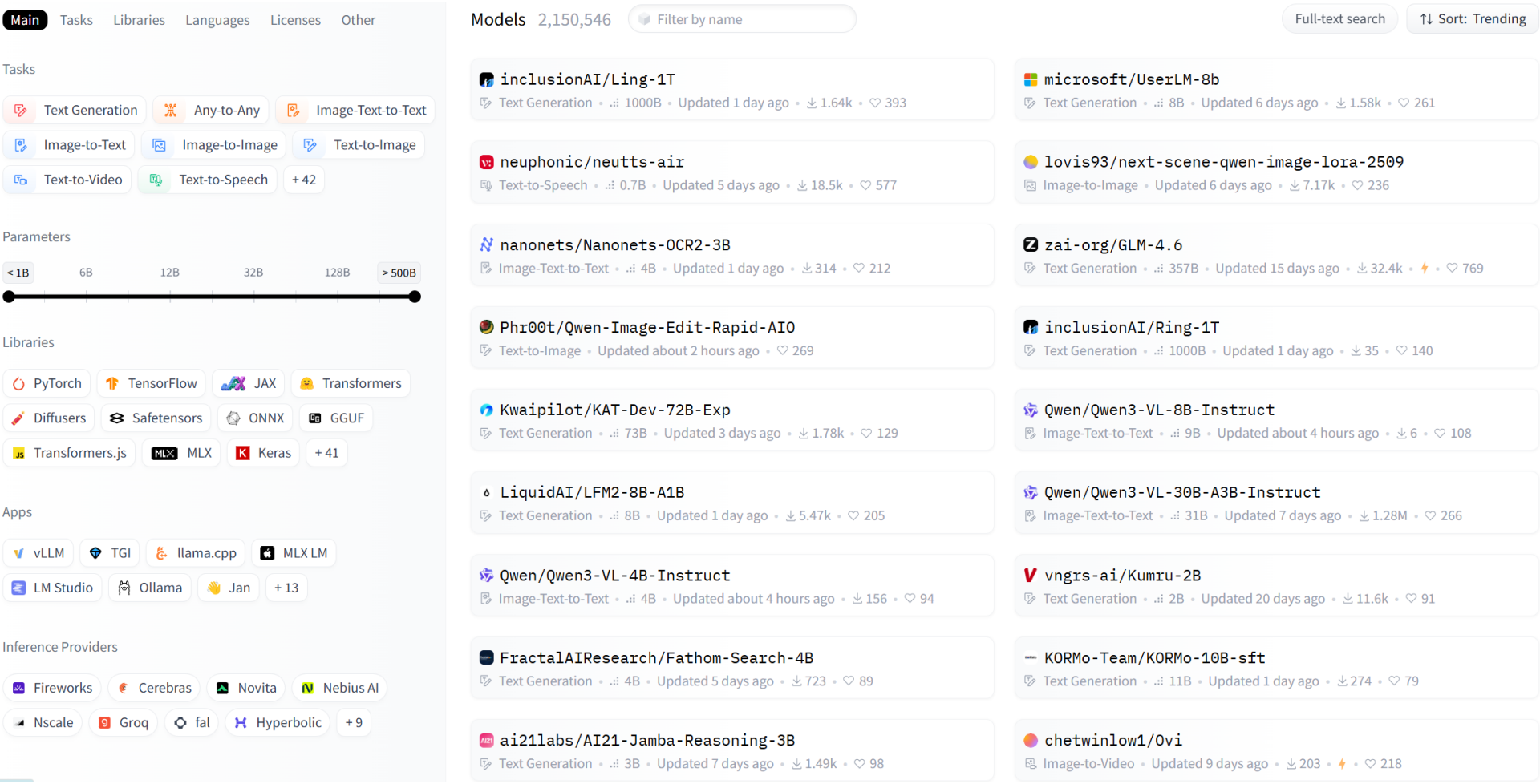
  - Indexing by column

```
>>> dataset["text"]
['the rock is destined to be the 21st century\'s new " conan " and that he\'s going to make a splash ever
 'the gorgeously elaborate continuation of " the lord of the rings " trilogy is so huge that a column of
 'effective but too-tepid biopic',
 ...,
 'things really get weird , though not particularly scary : the movie is all portent and no content .']
```

  - Slicing

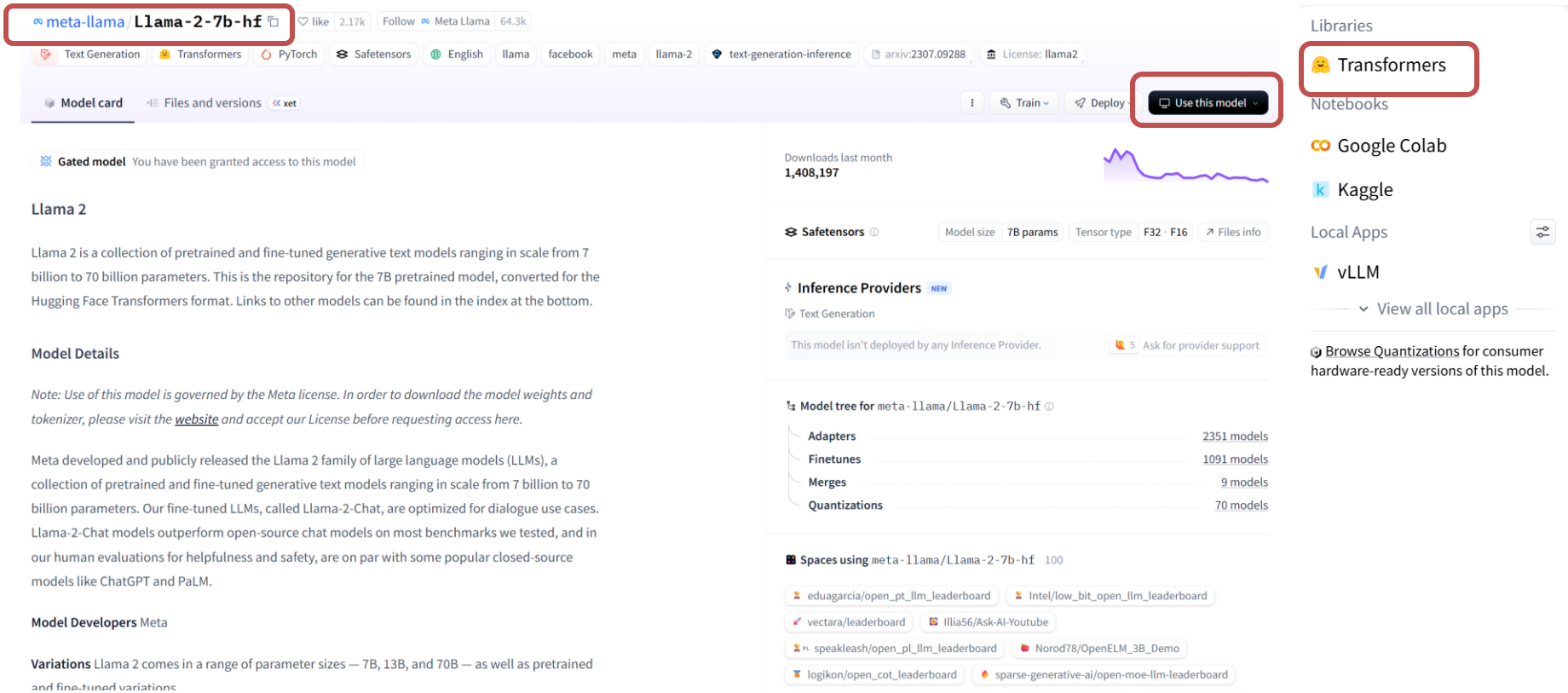
```
>>> dataset[:3]
{'label': [1, 1, 1],
 'text': ['the rock is destined to be the 21st century\'s new " conan " and that he\'s going to make a sp
  'the gorgeously elaborate continuation of " the lord of the rings " trilogy is so huge that a column of
  'effective but too-tepid biopic']}
```

Hugging Face: Models



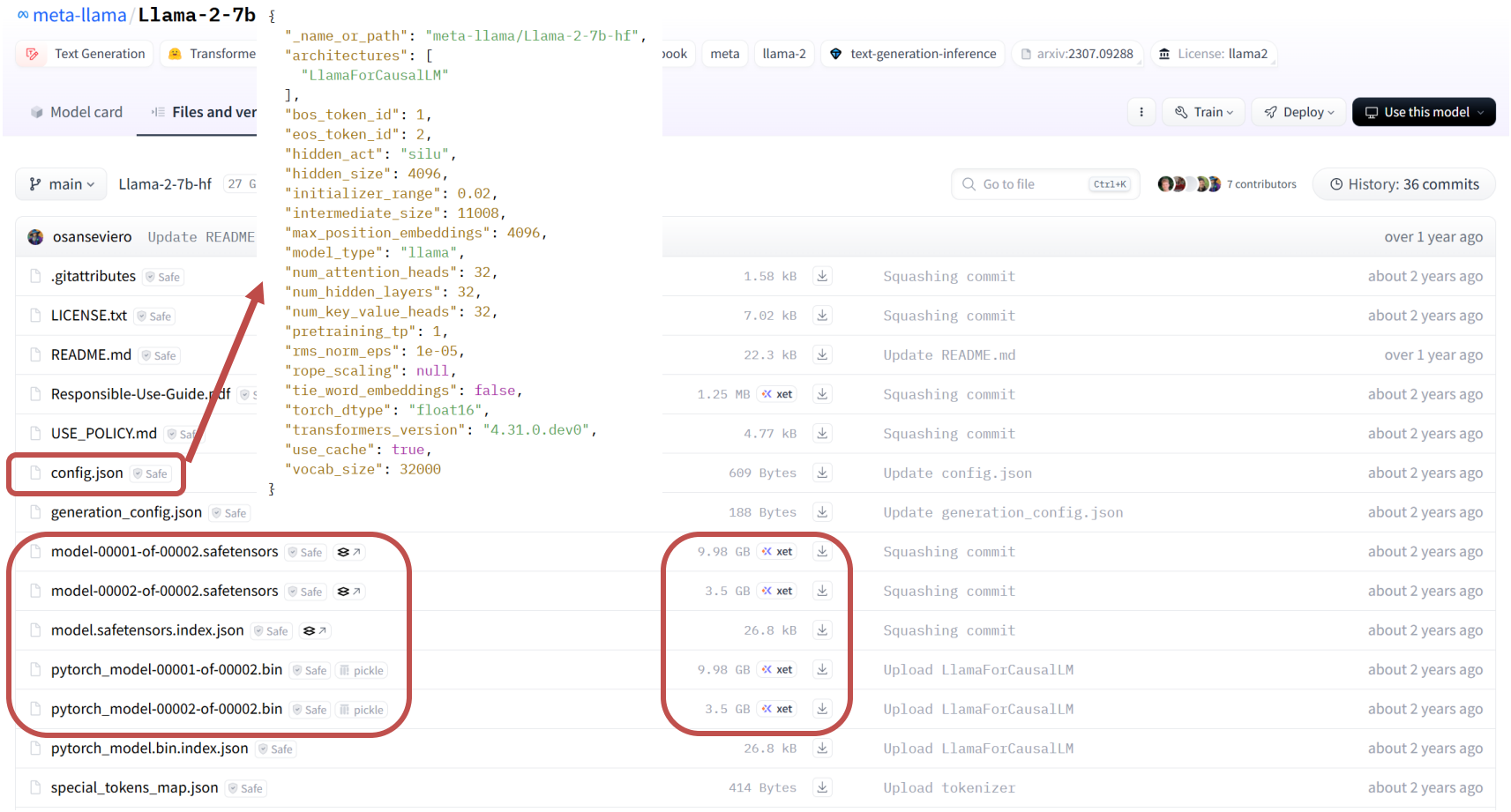
Hugging Face: Models

- Let’s check what a Hugging Face model has



Hugging Face: Models

- Let’s check what a Hugging Face model has



Hugging Face: Models

- How to use Hugging Face models?
  - Install necessary packages such as torch and transformers
  - Use from_pretrained() to load the weights and configuration file from the Hub into the model and preprocessor class.

```
from transformers import AutoModelForCausalLM, AutoTokenizer

model = AutoModelForCausalLM.from_pretrained("meta-llama/Llama-2-7b-hf", dtype="auto", device_map="auto")
tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-2-7b-hf")
```

  - For text inputs, we can tokenize the text and return PyTorch tensors with the tokenizer.

```
model_inputs = tokenizer(["The secret to baking a good cake is "], return_tensors="pt").to(model.device)
```

Hugging Face: Models

- How to use Hugging Face models?
  - For inference, pass the tokenized inputs to generate() to generate text. Decode the token ids back into text with batch_decode().

```
generated_ids = model.generate(**model_inputs, max_length=30)
tokenizer.batch_decode(generated_ids)[0]
'<s> The secret to baking a good cake is 100% in the preparation. There are so many recipes out there,'
```

  - We may also use the Pipeline as a high-level helper

```
from transformers import pipeline

pipeline = pipeline(task="text-generation", model="google/gemma-2-2b")
pipeline("the secret to baking a really good cake is ")
[{'generated_text': 'the secret to baking a really good cake is 1. the right ingredients 2. the'}]
```

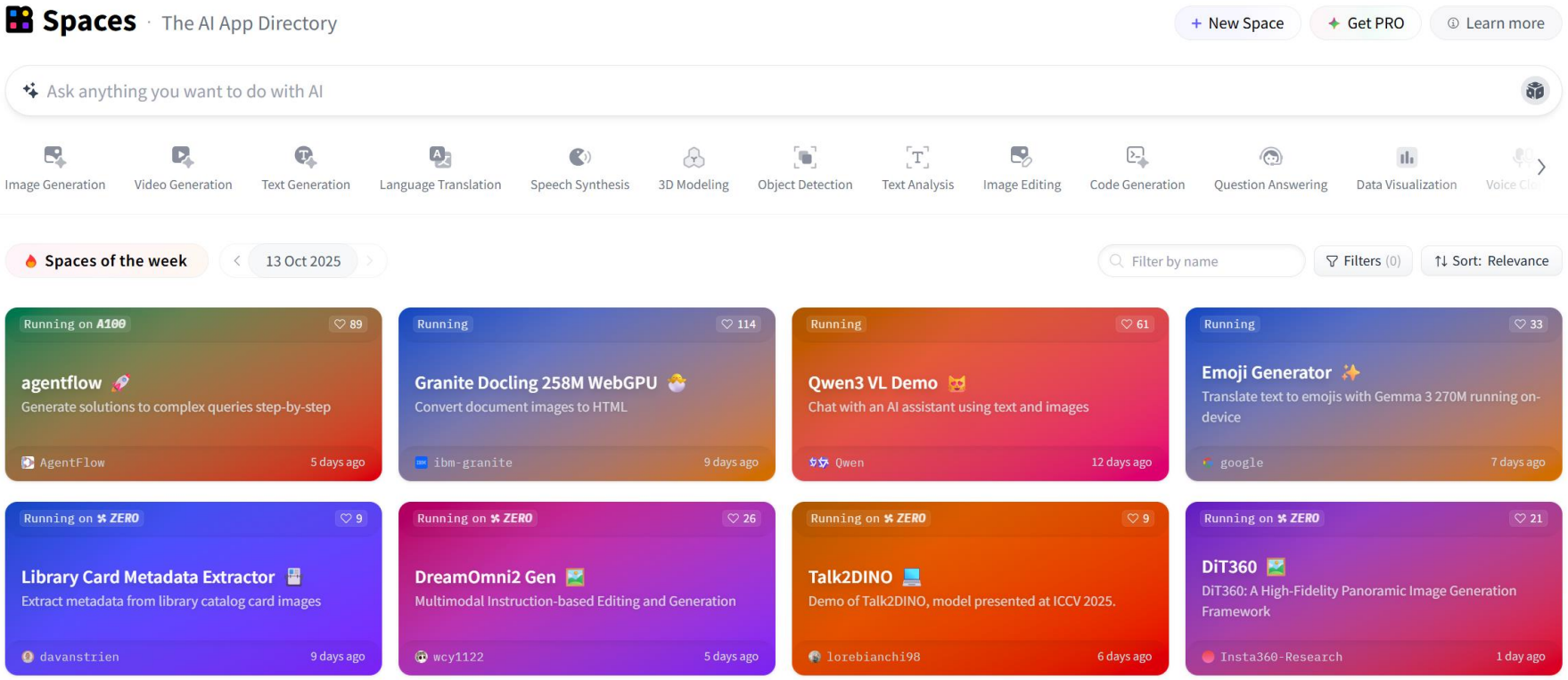
Hugging Face: Models

- Always check the model card for recommended model usage

```
model_id = "meta-llama/Meta-Llama-3-8B-Instruct"

pipeline = transformers.pipeline(
    "text-generation",
    model=model_id,
    model_kwargs={"torch_dtype": torch.bfloat16},
    device_map="auto",
)

messages = [
    {"role": "system", "content": "You are a pirate chatbot who always responds i
    {"role": "user", "content": "Who are you?"},
]

terminators = [
    pipeline.tokenizer.eos_token_id,
    pipeline.tokenizer.convert_tokens_to_ids("<|eot_id|>")
]

outputs = pipeline(
    messages,
    max_new_tokens=256,
    eos_token_id=terminators,
    do_sample=True,
    temperature=0.6,
    top_p=0.9,
)
print(outputs[0]["generated_text"][-1])
```

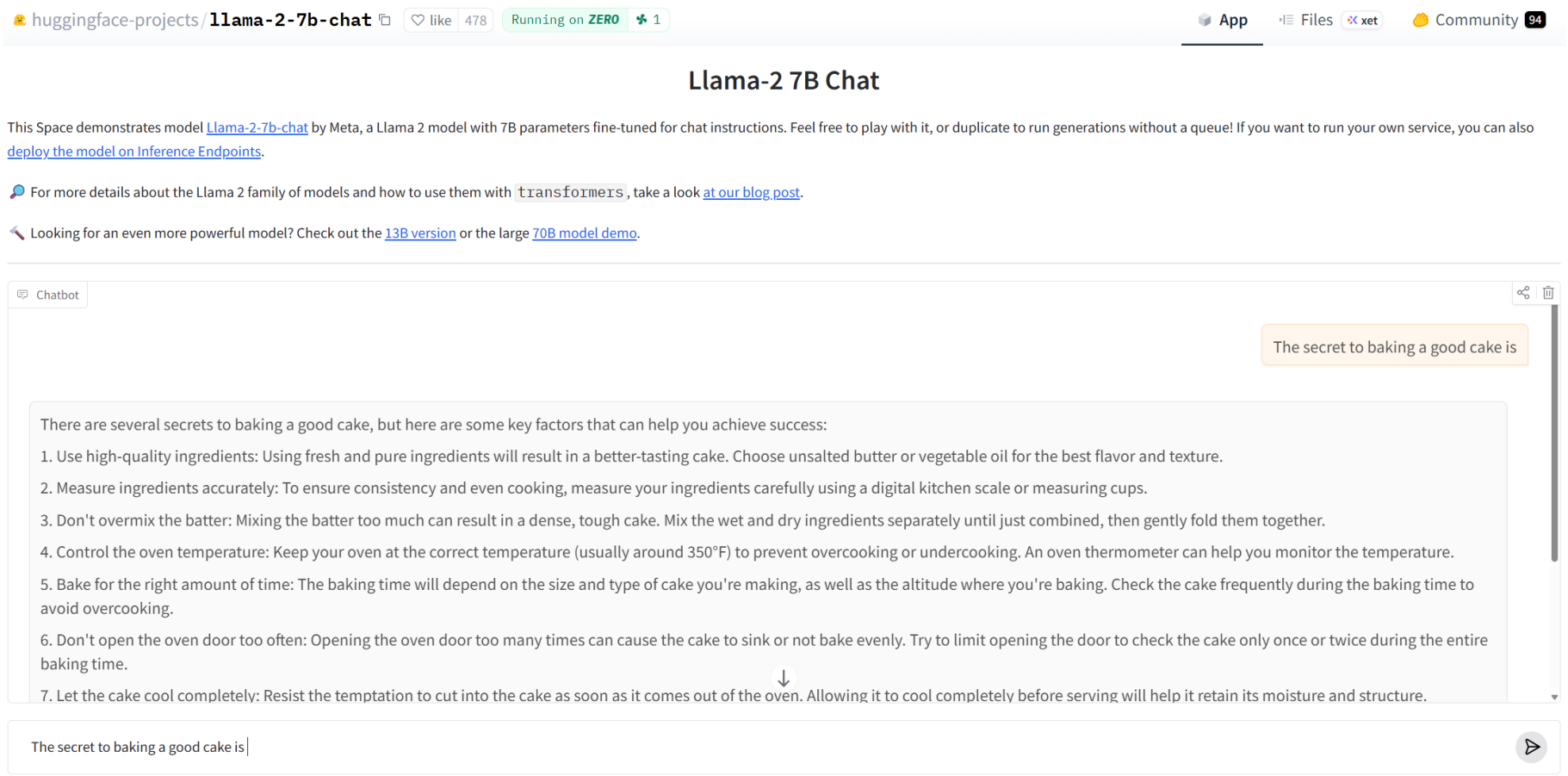
Hugging Face: Spaces

- Hugging Face Spaces offer a simple way to host ML demo apps.



Hugging Face: Spaces

- Let’s try a space for the text generation task



Hugging Face: Spaces

- We can try more spaces with different models for various tasks!