Introduction to Hugging Face

WORKING WITH HUGGING FACE

Jacob H. Marquez
Lead Data Engineer

datacamp

What is Hugging Face?

- Collaboration platform
- Open-source machine learning
- Text, vision, and audio tasks
- Models, datasets, frameworks
- Reduce barriers to entry

[1] https://huggingface.co/

In this course

- Navigate and use the Hugging Face Hub
- Explore models and datasets
- Build pipelines for text, image, and audio data
- Fine-tuning, generation, embeddings, and semantic search

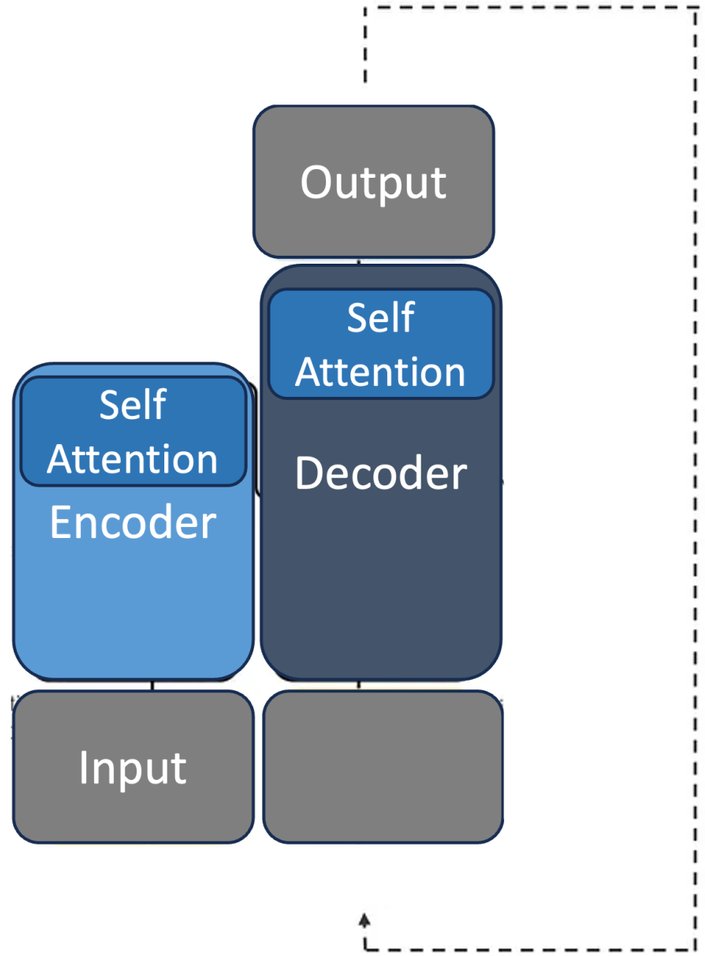
Large Language Models

- LLMs
- Understand and generate human-like text
- Massive amounts of data
- Learn patterns in sequences

[1] https://en.wikipedia.org/wiki/Large_language_model

Large Language Models

- LLMs
- Understand and generate human-like text
- Massive amounts of data
- Learn patterns in sequences
- Transformer architecture



[1] https://towardsdatascience.com/transformers-89034557de14

Large Language Models

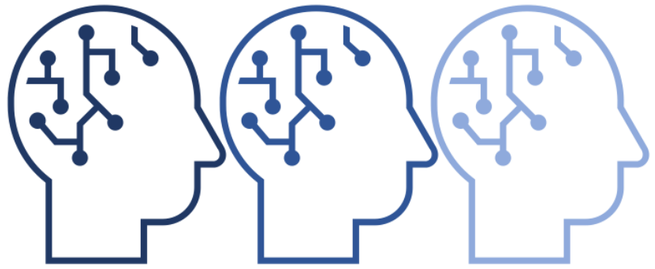
- LLMs
- Understand and generate human-like text
- Massive amounts of data
- Learn patterns in sequences
- Transformer architecture
- Popular options are GPT and Llama

ChatGPT

LLaMA

by Meta

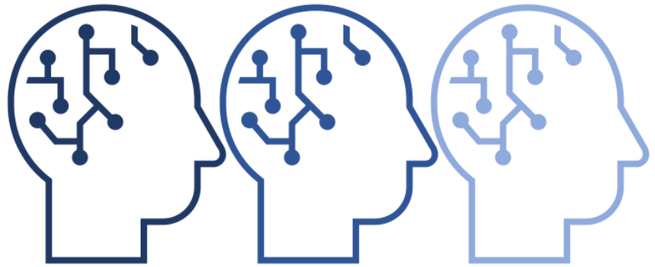
Benefits of Hugging Face



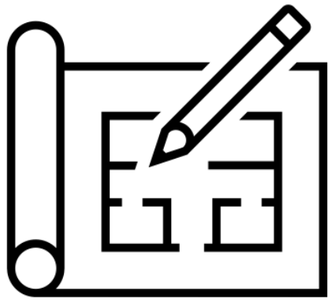
Access to Models

- Faster experimentation

Benefits of Hugging Face



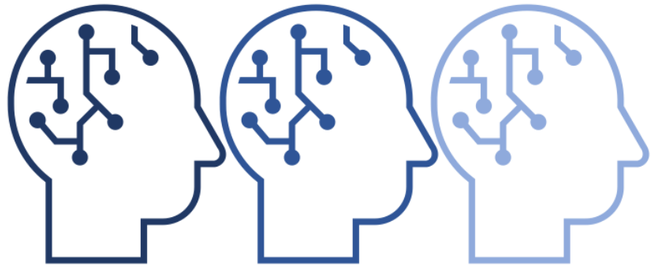
Access to Models



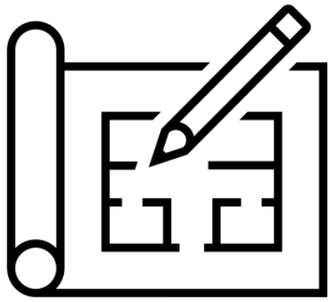
Frameworks

- Faster experimentation
- Supports every step of the process

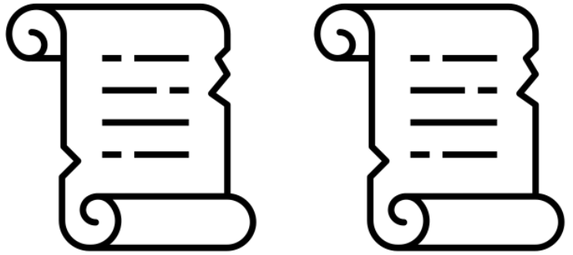
Benefits of Hugging Face



Access to Models



Frameworks



Documentation

- Faster experimentation
- Supports every step of the process
- Smoother adoption

Deciding when to use

Use Hugging Face

- Quick way to use ML tasks
- Don't have deep ML expertise
- Testing several models
- Dataset needed

Use another solution

- Slow computer
- Highly customized architectures
- Domain specific needs not yet met
- Not leveraging advanced ML techniques

Installing Hugging Face

Hugging Face

```
pip install transformers datasets
```

ML Framework

```
pip install torch torchvision torchaudio
```

[1] https://pytorch.org/

Let's practice!

WORKING WITH HUGGING FACE

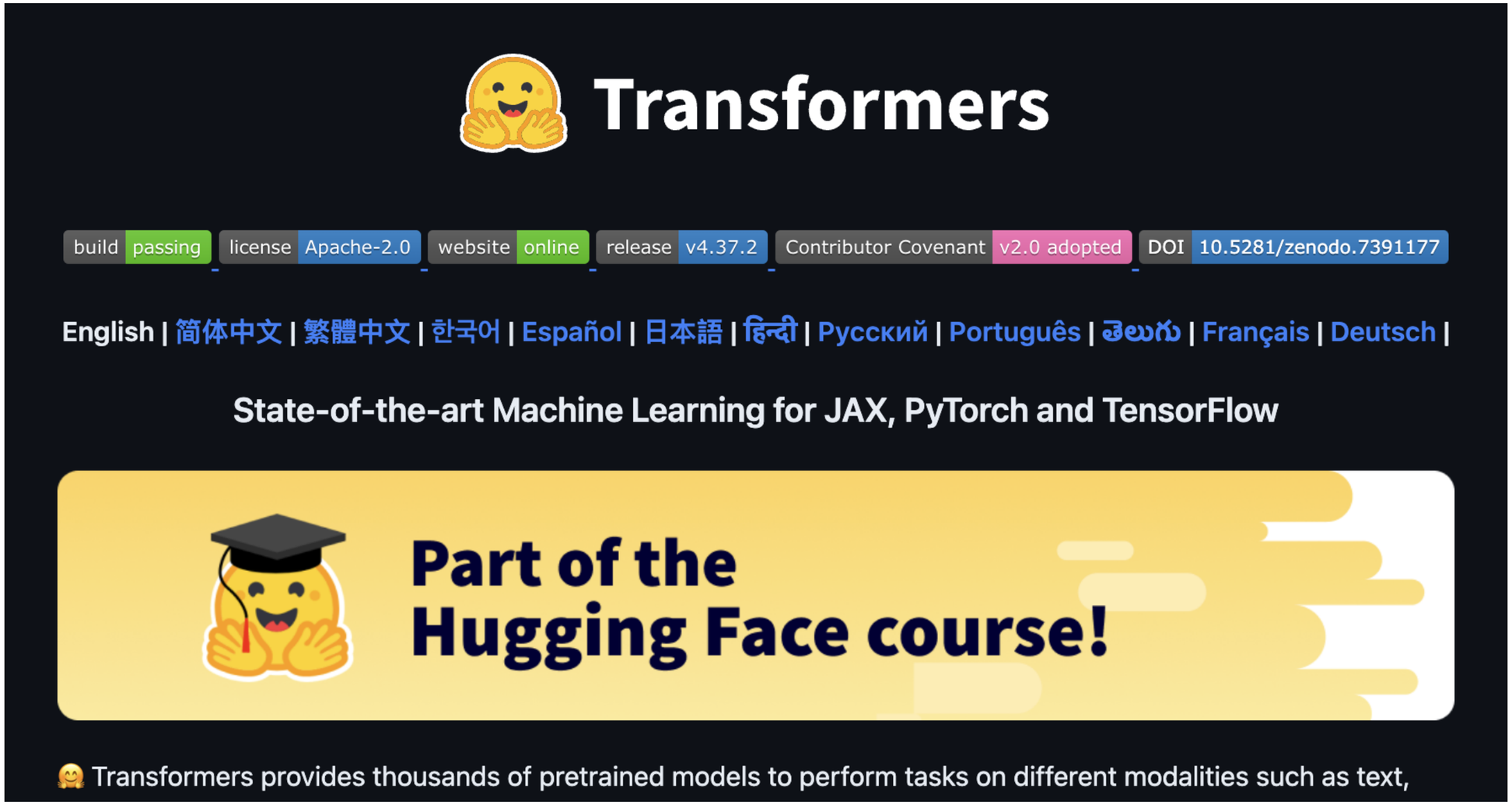
Transformers and the Hub

WORKING WITH HUGGING FACE

Jacob H. Marquez
Lead Data Engineer

datacamp

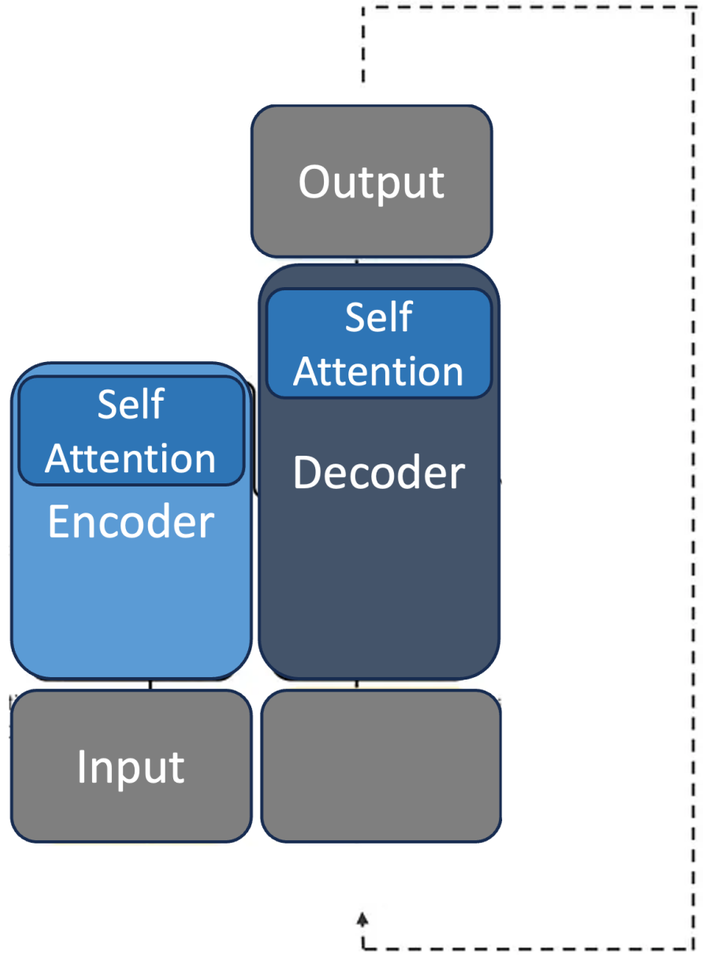
Transformers - the Hugging Face package



[1] https://github.com/huggingface/transformers

Transformers - the model architecture

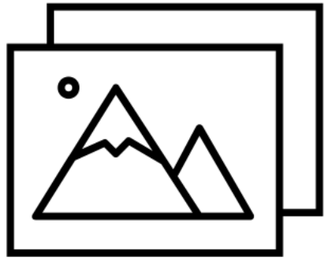
- Neural network models
- Learn context and understanding
- Core components:
  - Encoder
  - Decoder
  - Self-attention mechanism
- Transform input to numerical representations
- Helps model understand context of the input



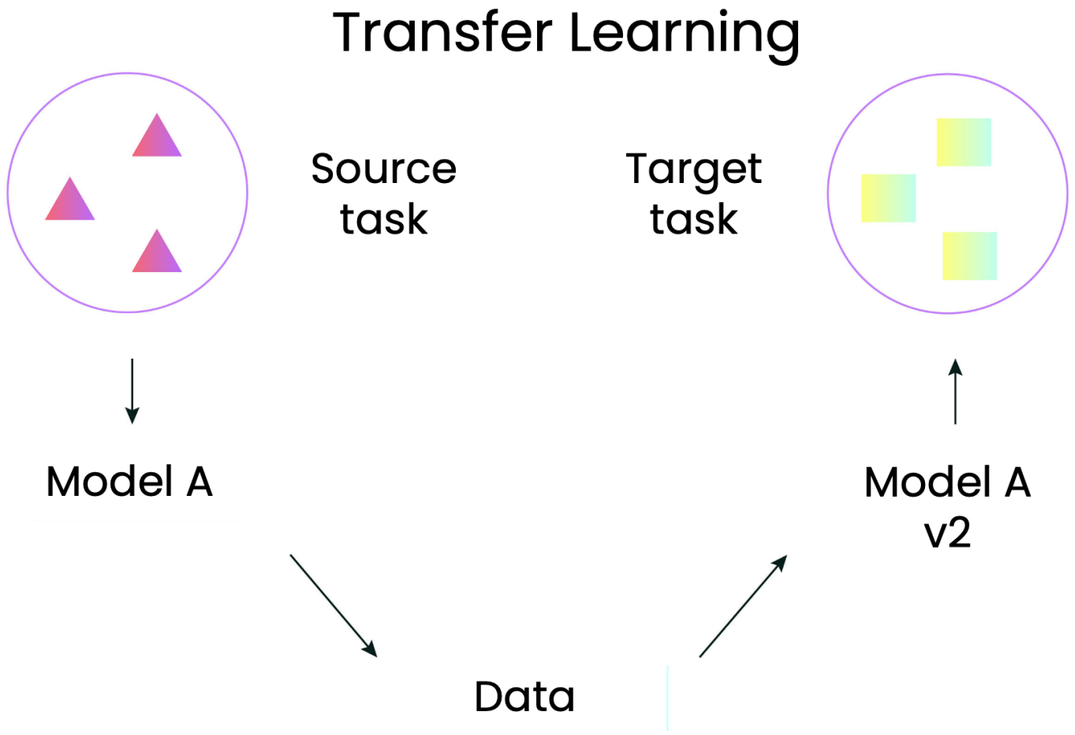
[1] https://www.turing.com/kb/brief-introduction-to-transformers-and-their-power

Uses cases of transformers

- Use cases for text, image, and vision
- Classification for all three
- Automatic speech recognition
- Text summarization
- Object detection for autonomous driving



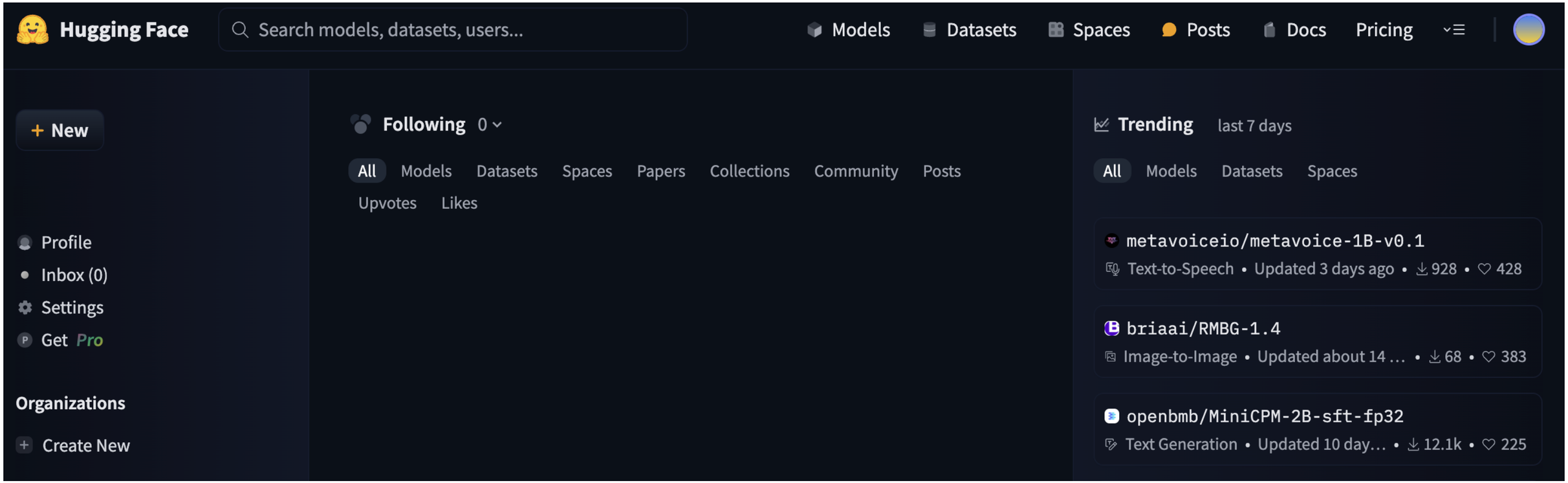
A key benefit of transformers



- Enables Hugging Face models to perform well on new tasks with little data

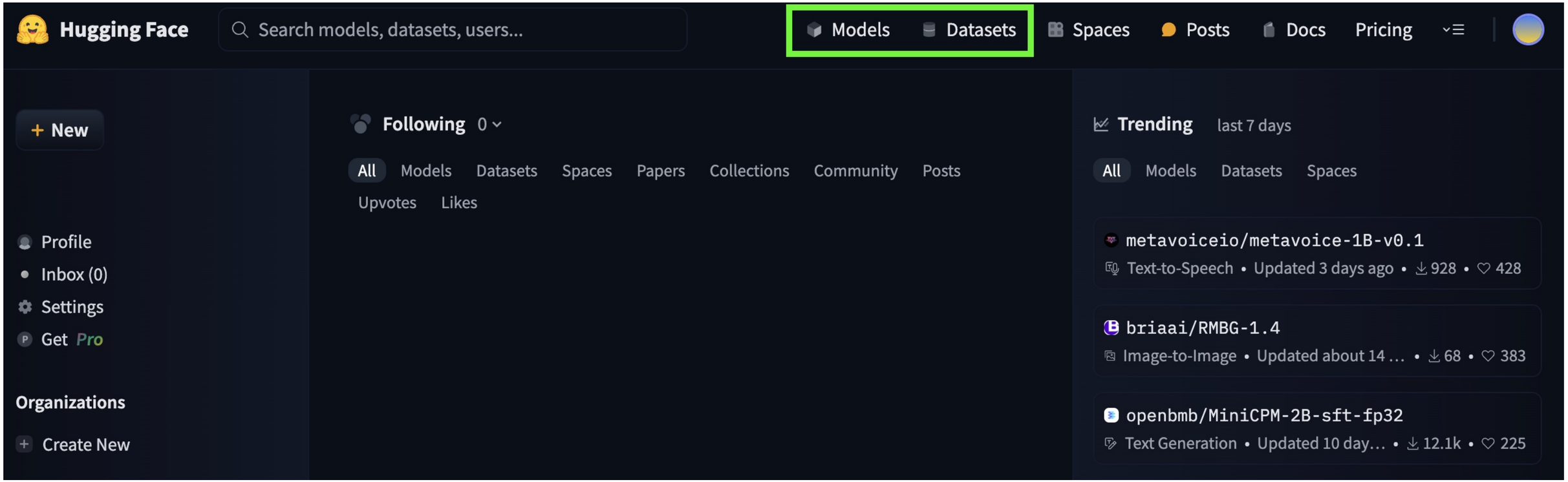
[1] https://www.topbots.com/transfer-learning-in-nlp/#transfer-learning

The Hub



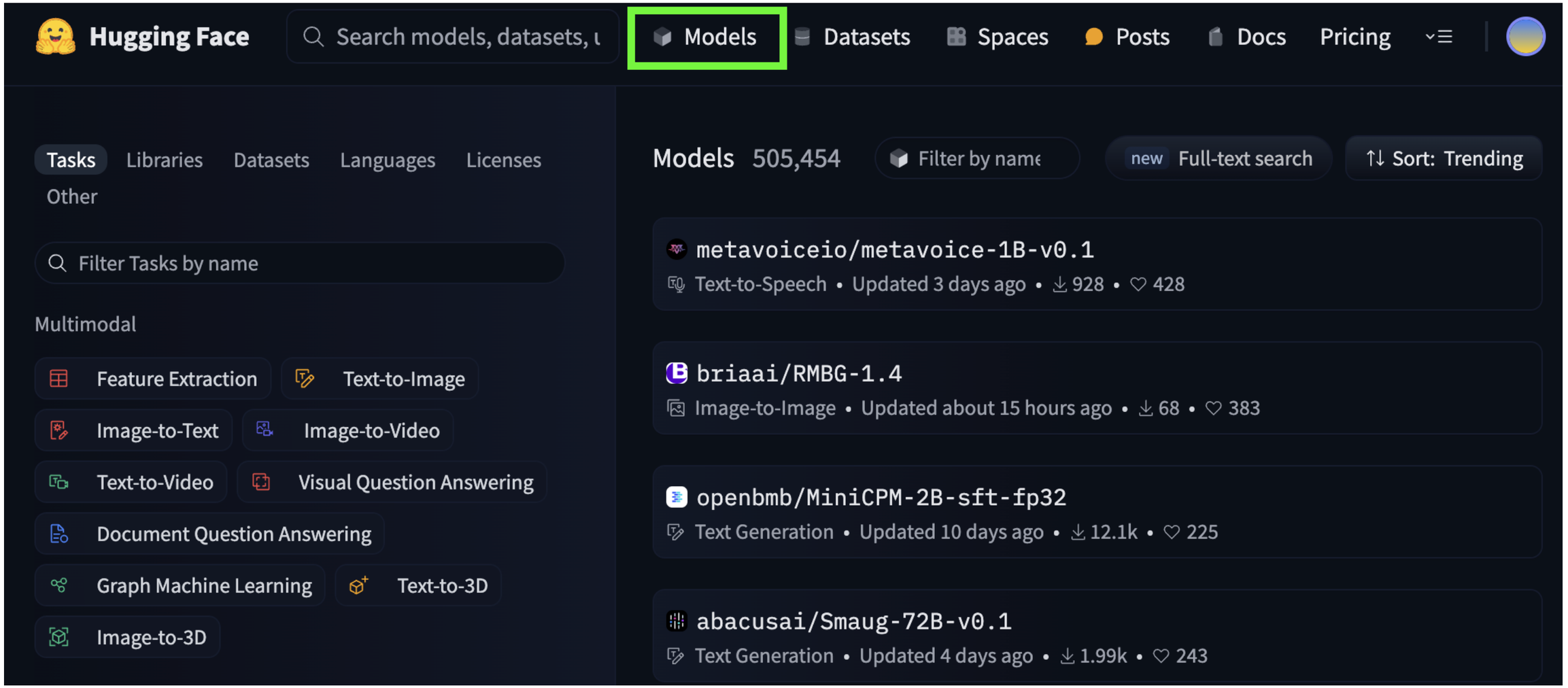
[1] https://huggingface.co/

Navigating the Hub



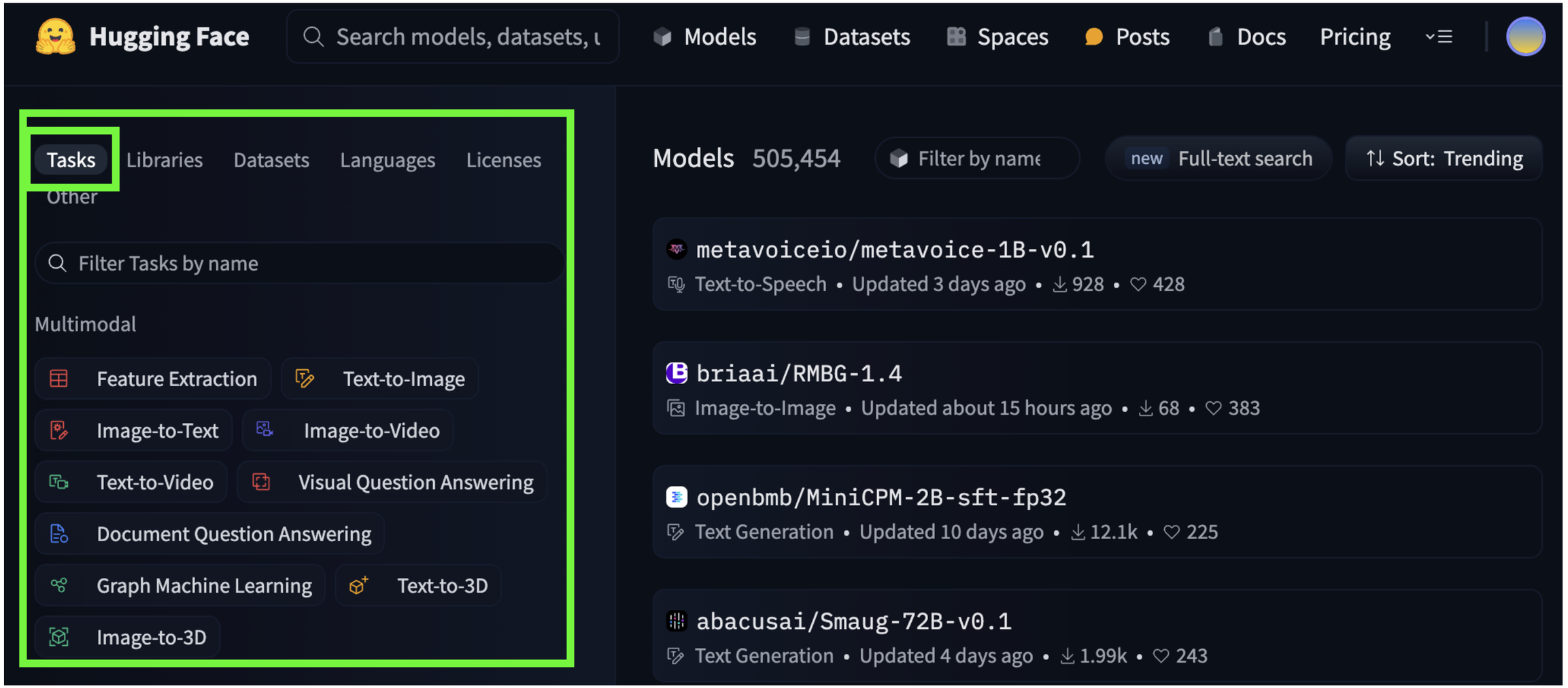
[1] https://huggingface.co/

Searching for models



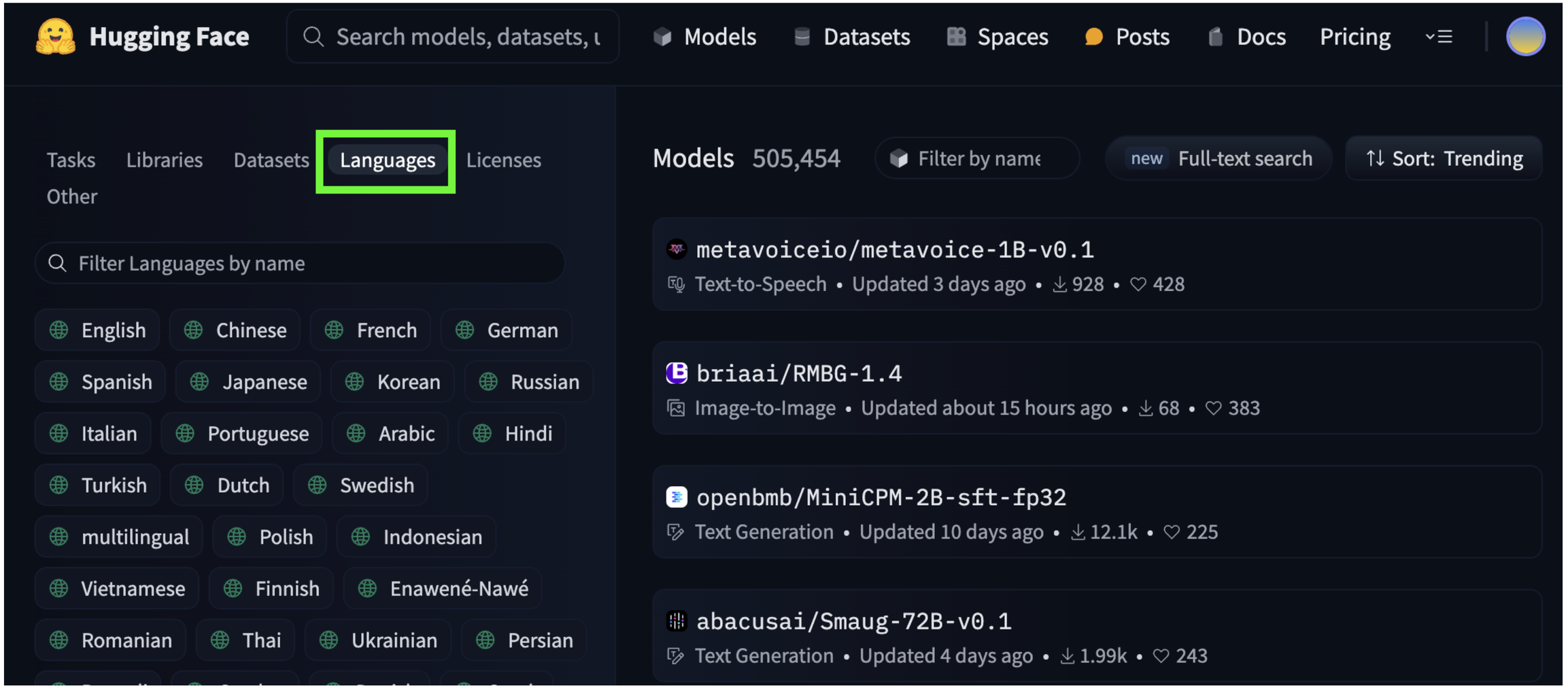
[1] https://huggingface.co/models

Searching for models



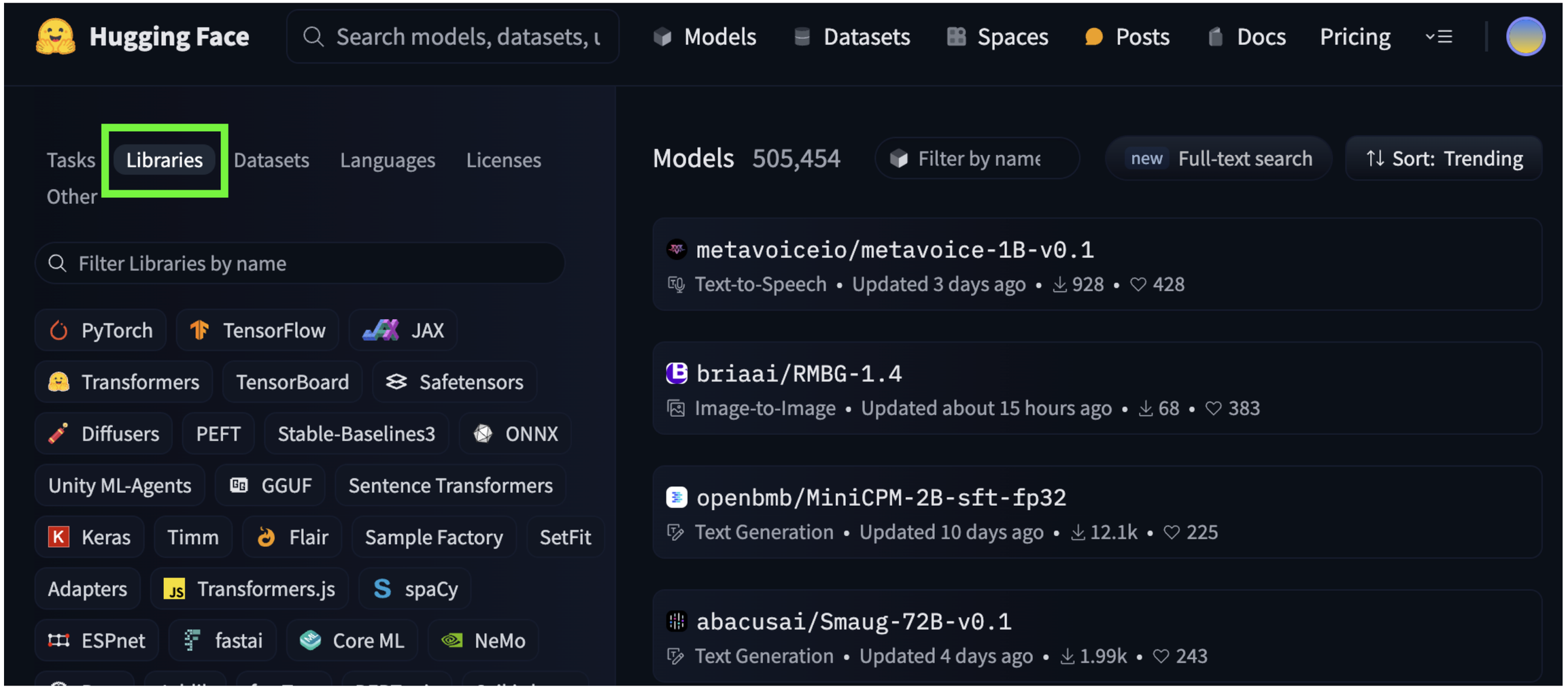
[1] https://huggingface.co/models

Searching for models



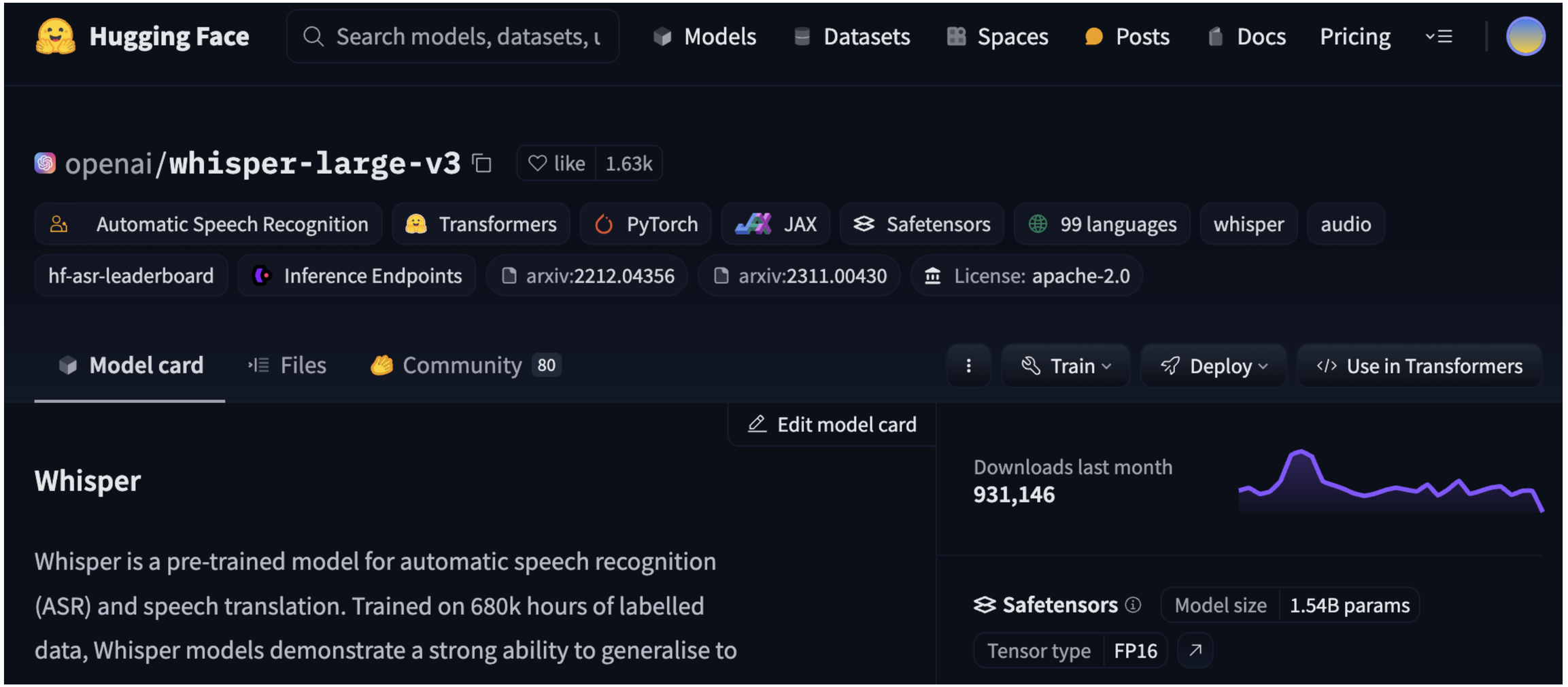
[1] https://huggingface.co/models

Searching for models



[1] https://huggingface.co/models

Model cards



[1] https://huggingface.co/openai/whisper-large-v3

Using huggingface_hub

```
pip install huggingface_hub
```

```
from huggingface_hub import HfApi
api = HfApi()
list(api.list_models())
```

```
[ModelInfo: {
  {'_id': '622fea36174feb5439c2e4be',
   'author': 'cardiffnlp',
   ...}]
```

[1] https://github.com/huggingface/huggingface_hub

Using huggingface_hub

```
models = api.list_models(
    filter=ModelFilter(
        task="text-classification"),
        sort="downloads",
        direction=-1,
        limit=5
    )
)

modelList = list(models)

print(modelList[0])
```

```
Model Name: albert/albert-base-v1, Tags: [...]
```

- `task` searches for specified task
- `sort` will order the list
- `direction` provides the direction of the sorted order
  - -1 for descending
  - all other numbers for ascending
- `limit` will limit the number of models returned

[1] https://github.com/huggingface/huggingface_hub

Saving a model locally

```
# Import AutoModel
from transformers import AutoModel

modelId = "distilbert-base-uncased-finetuned-sst-2-english"

# Download model using the modelId
model = AutoModel.from_pretrained(modelId)

# Save the model to a local directory
model.save_pretrained(save_directory=f"models/{modelId}")
```

- Be mindful of storage!

[1] https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModel

Let's practice!

WORKING WITH HUGGING FACE

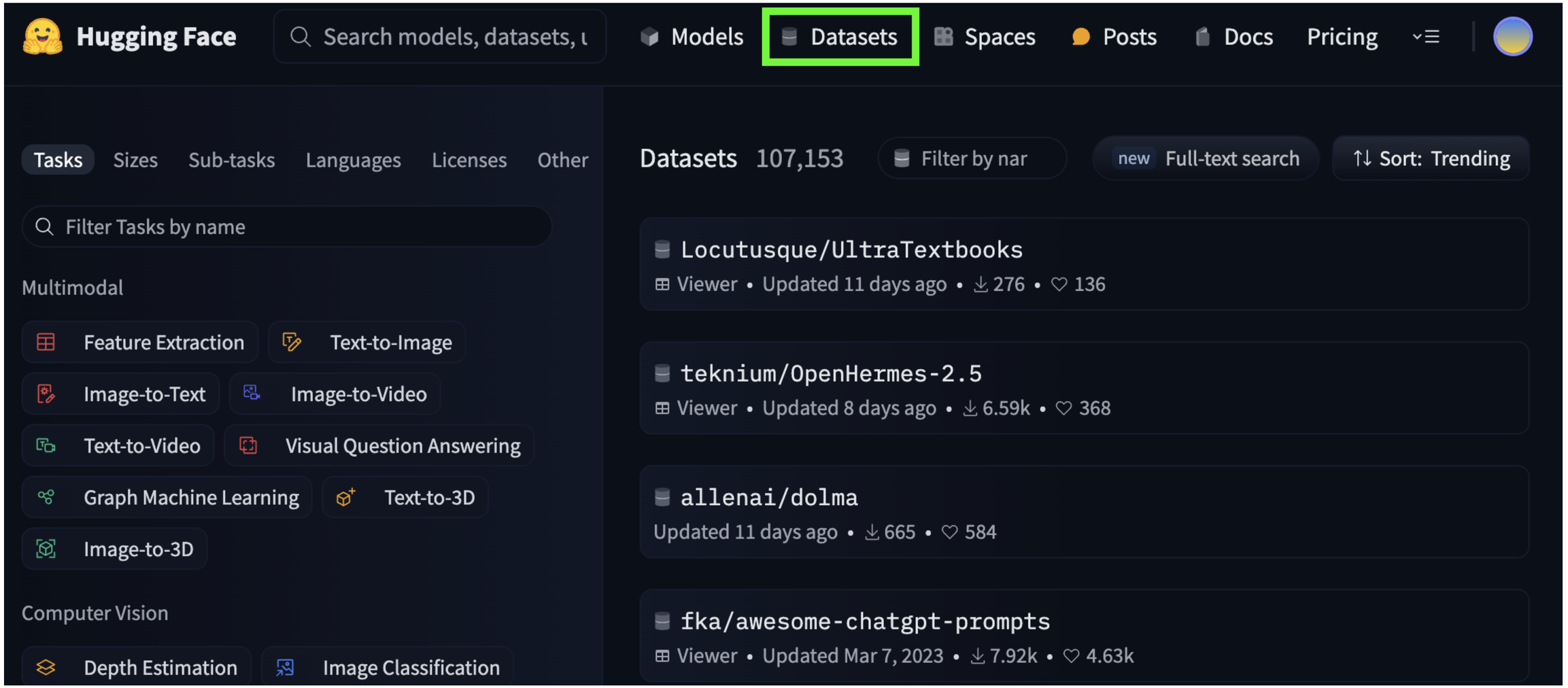
Working with datasets

WORKING WITH HUGGING FACE

Jacob H. Marquez
Lead Data Engineer

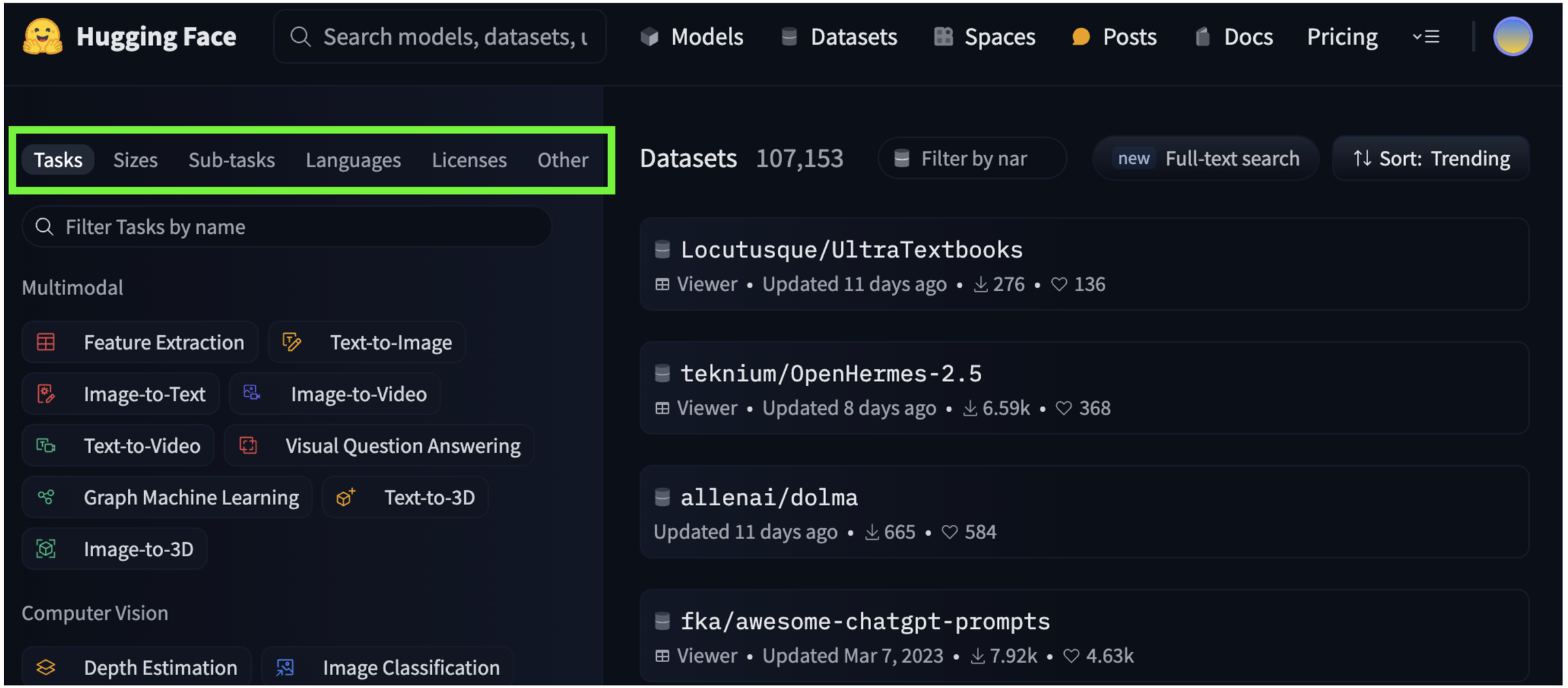
datacamp

Datasets in Hugging Face



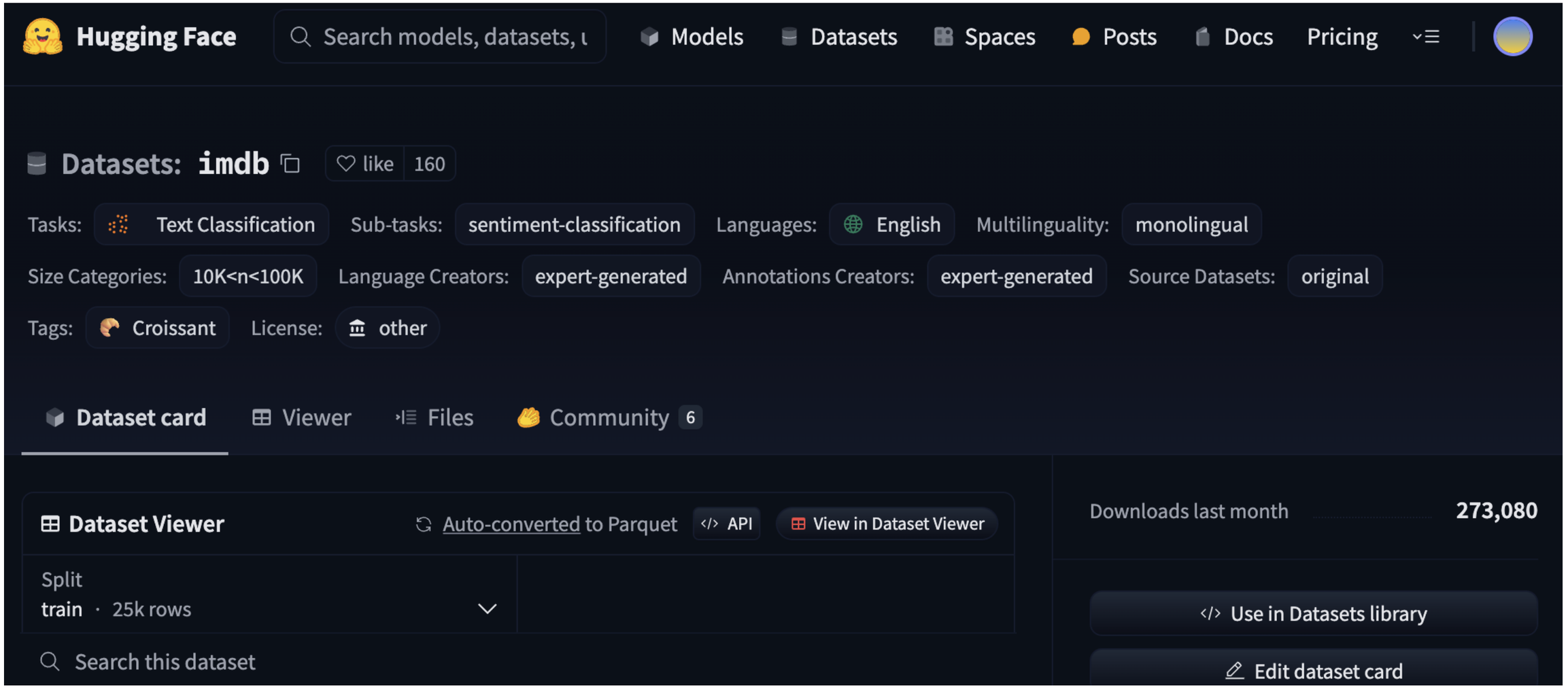
[1] https://huggingface.co/datasets

Searching for datasets



[1] https://huggingface.co/datasets

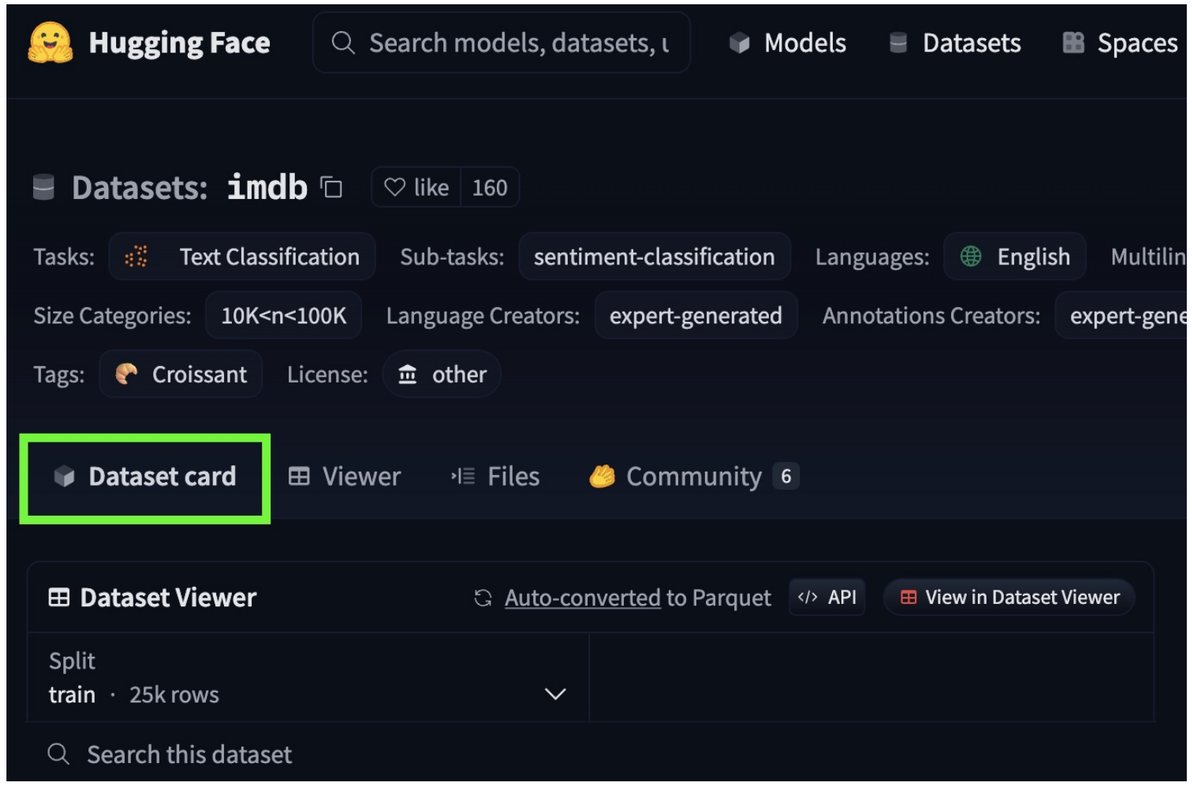
Dataset cards



[1] https://huggingface.co/datasets/imdb

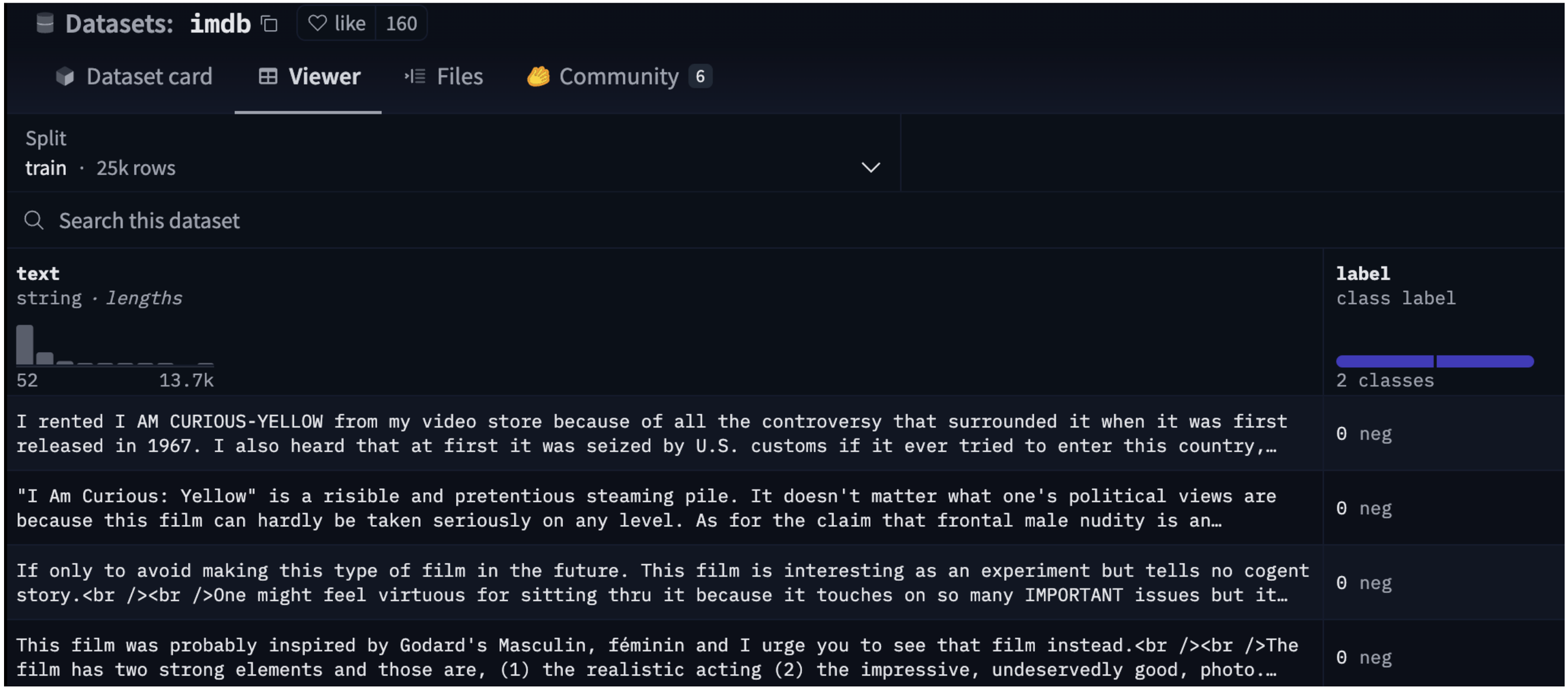
Dataset cards

- Description
- Dataset structure
- An example
- Field metadata
- Training and testing splits



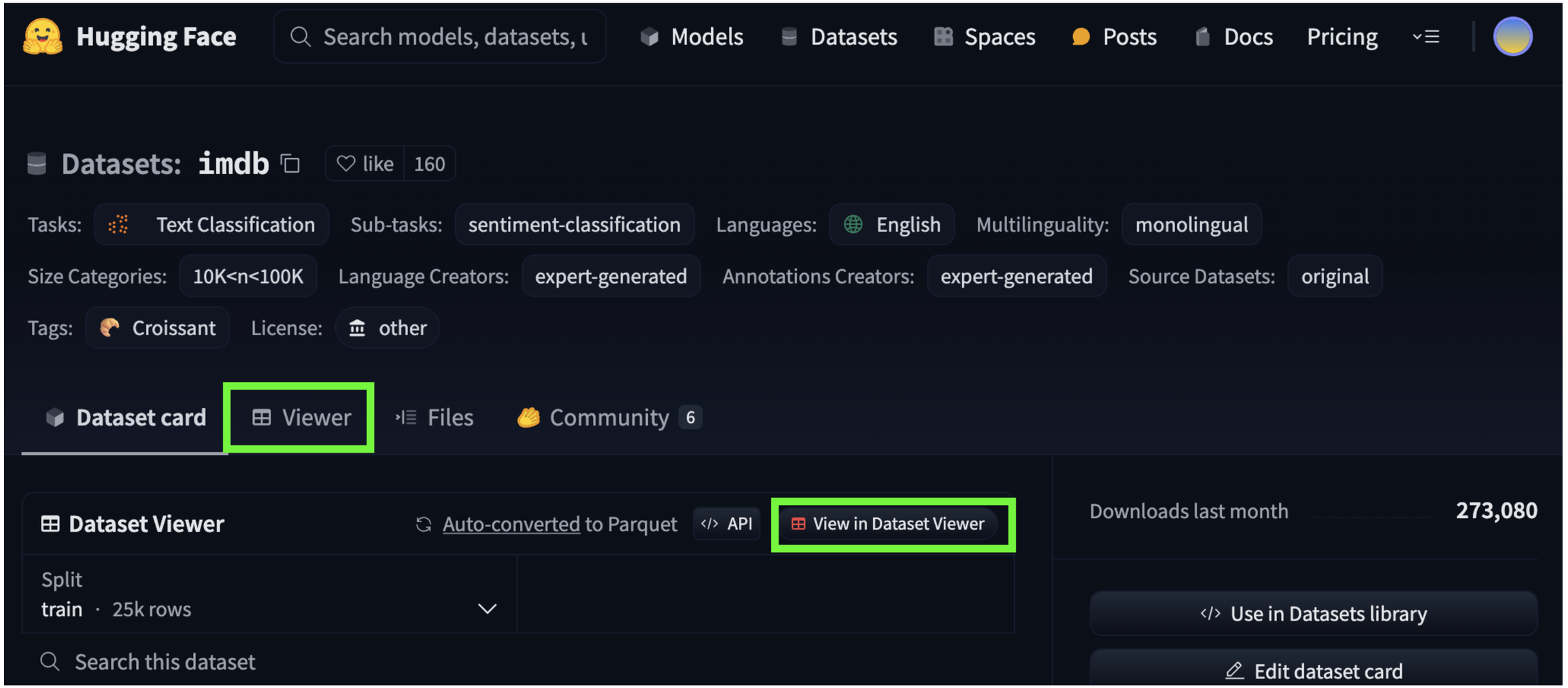
[1] https://huggingface.co/datasets/imdb

Dataset cards



[1] https://huggingface.co/datasets/imdb

Dataset cards



[1] https://huggingface.co/datasets/imdb

datasets package

```
pip install datasets
```

- Access
- Download
- Mutate
- Use
- Share

[1] https://huggingface.co/docs/datasets/index

Inspecting a dataset

```
from datasets import load_dataset_builder

data_builder = load_dataset_builder("imdb")
```

```
print(data_builder.info.description)
```

```
Large Movie Review Dataset. This is a dataset for sentiment classification...
```

```
print(data_builder.info.features)
```

```
{'text': Value(dtype='string', id=None), 'label': Value(dtype='string', id=None)}
```

[1] https://huggingface.co/docs/datasets/load_hub

Downloading a dataset

```
from datasets import load_dataset


data = load_dataset("imdb")
```

Split parameter

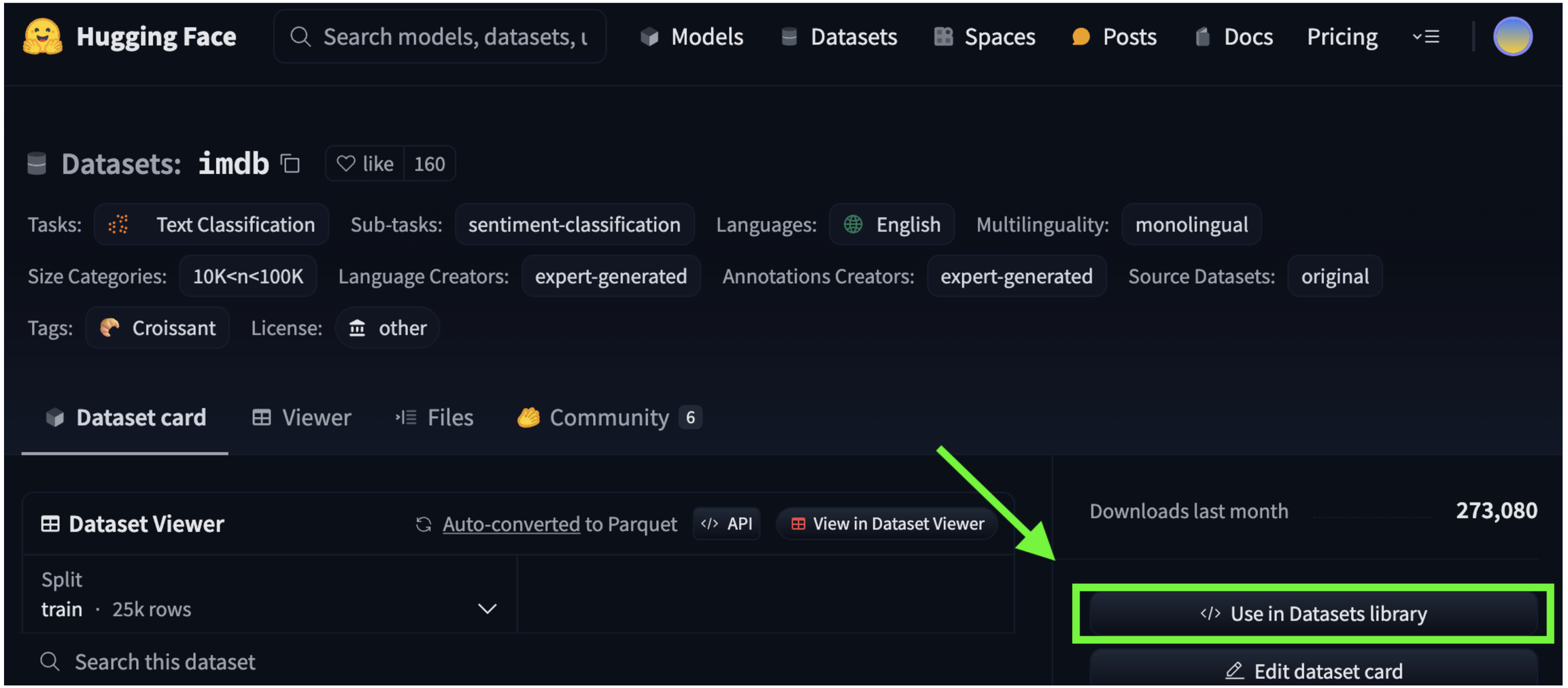
```
data = load_dataset("imdb", split="train")
```

Configuration parameter

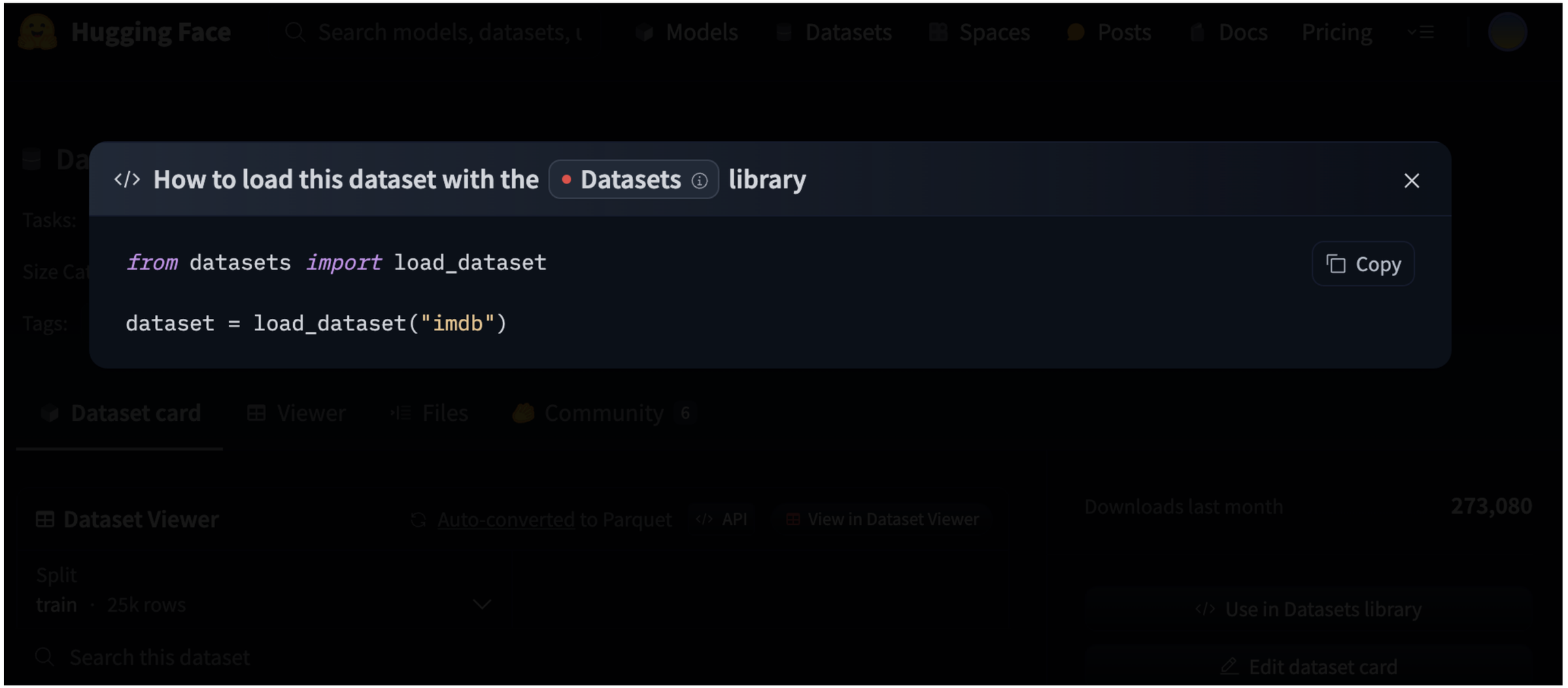
```
data = load_dataset("wikipedia", "20231101.en")
```

[1] https://huggingface.co/docs/datasets/v2.15.0/loading

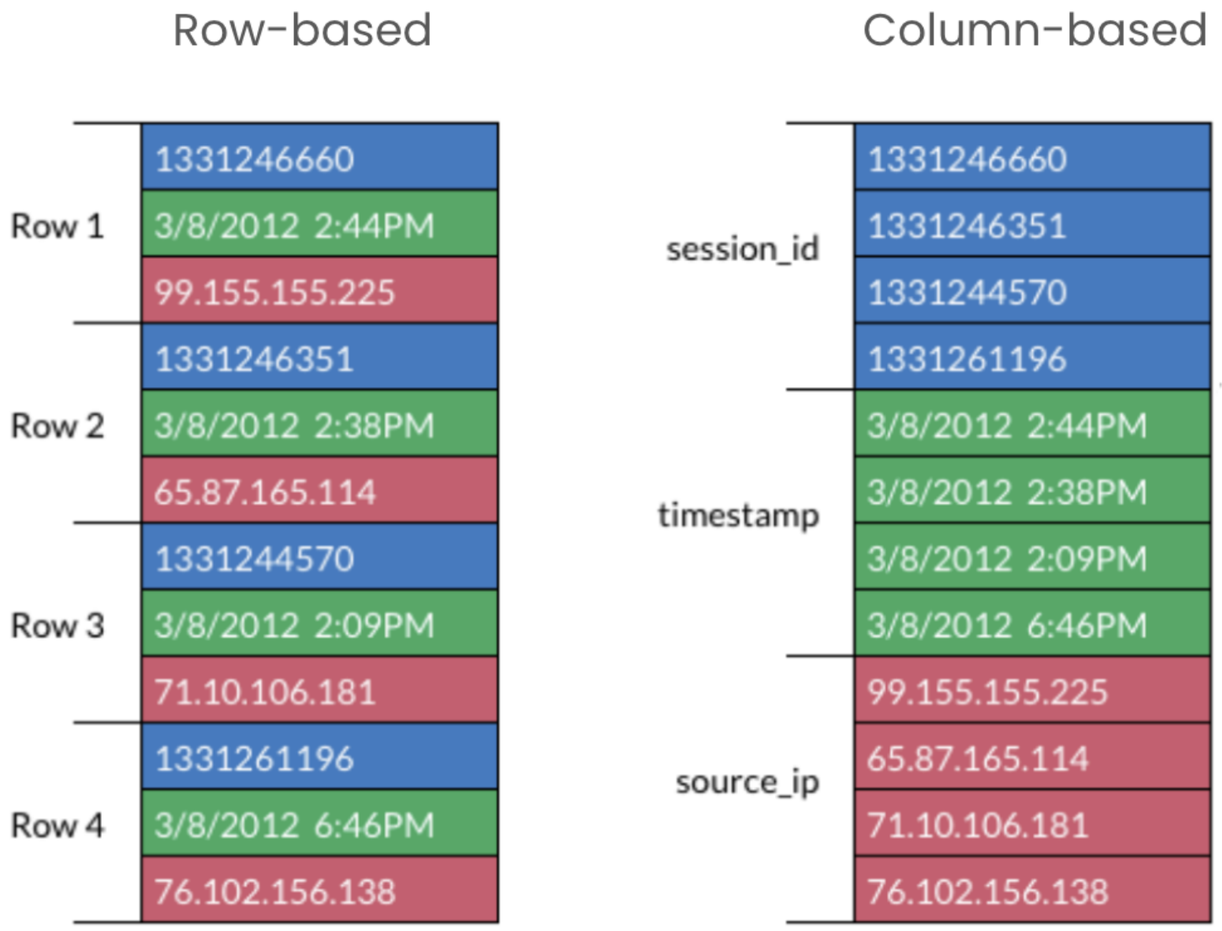
Use in datasets



Use in datasets



Apache Arrow dataset formats



[1] https://arrow.apache.org/overview/

Mutating a dataset

```
imdb = load_dataset("imdb", split="train")


# Filter imdb
filtered = imdb.filter(lambda row: row['label']==0)
```

```
{'text': 'I rented I AM CURIOUS-YELLOW...''}
```

[1] https://huggingface.co/docs/datasets/process#select-and-filter

Mutating a dataset

```
# Slicing
sliced = filtered.select(range(2))
```

```
print(sliced)
```

```
Dataset({features: ['id', 'url', 'title', 'text'], num_rows: 2})
```

```
print(sliced[0]['text'])
```

[1] https://huggingface.co/docs/datasets/process#select-and-filter

Benefits of datasets

- Accessible and shareable
- Relevant to common ML tasks
- Efficient processing on large data
- Faster querying
- Convenient complimentary `datasets` package

Let's practice!

WORKING WITH HUGGING FACE