**HUGGING FACE**

[Hugging Face](https://huggingface.co/)  is a company that maintains a huge [open-source](https://www.ibm.com/think/topics/open-source) community of the same name that builds tools, machine learning models and platforms for working with [artificial intelligence](https://www.ibm.com/think/topics/artificial-intelligence), with a focus on [data science](https://www.ibm.com/think/topics/data-science), [machine learning](https://www.ibm.com/think/topics/machine-learning) and [natural language processing](https://www.ibm.com/think/topics/natural-language-processing) (NLP). Hugging Face is notable for its NLP transformers library and a platform that allows users to share models and datasets.

Hugging Face serves multiple purposes in the AI ecosystem. Users can:

* Discover and share cutting-edge AI models
* Collaborate with AI enthusiasts globally
* Access powerful tools for AI experimentation
* Share, use, and fine-tune ML models
* Host demos and evaluate ML models

**How Does Hugging Face Work?**

The platform democratizes AI access by providing pre-trained models that anyone can use. This is particularly significant because training state-of-the-art NLP transformer models typically requires substantial financial investment, making it impractical for startups and small businesses.

By offering freely available pre-trained models, organizations can develop proprietary AI solutions without incurring the massive expenses typically associated with training large language models from scratch.

For a deeper understanding of how Hugging Face works, take a look at our [Intro to LLMs course](https://365datascience.com/courses/intro-to-llms/?utm_medium=social&utm_source=web&utm_campaign=blog&utm_content=what-is-hugging-face&utm_term=article) taught by NLP expert Lauren Newbould, a data scientist at the BBC. The course covers Hugging Face and its cutting-edge AI Engineering tools.

**What Are Hugging Face Transformers?**

One of Hugging Face's most valuable offerings is its Transformers Python library. This powerful tool provides AI developers with easy access to pre-trained models through an API. The library also facilitates efficient creation of machine learning pipelines.

A person pointing at a screen

AI-generated content may be incorrect.

**Key Features of Hugging Face Transformers**

The Transformers library includes several important features:

* State-of-the-art Natural Language Processing models
* Support for PyTorch and TensorFlow frameworks
* Easy model fine-tuning capabilities
* Comprehensive documentation and examples
* Regular updates and community support

**Is Hugging Face Open Source?**

Hugging Face takes a unique approach to open-source development. While the platform strongly champions open-source AI and encourages collaboration among developers, it maintains a hybrid model.

The core infrastructure and some enterprise features remain proprietary, allowing Hugging Face to sustain its business operations. But the platform's most valuable assets—the vast collection of use-contributed models and datasets—are entirely open-source. This means anyone can freely access, use, and modify these resources for their own projects.

This balance between proprietary infrastructure and open-source content has helped Hugging Face create a sustainable ecosystem that benefits both the company and the broader AI community.

A black background with white text

AI-generated content may be incorrect.

**Hugging Face services**

Here are Hugging Face’s primary services:

Hugging Face Hub

The Hugging Face Hub is a central web-based platform where users can share, discover and collaborate on models, datasets and applications. It acts like a "GitHub for AI," hosting thousands of publicly available resources. Model and dataset pages include documentation, examples, version tracking and live demos in many cases. The Hub also supports private repositories for teams and enterprises for secure collaboration.

Transformers Library

The Transformers library is one of the most widely used tools for NLP, [computer vision](https://www.ibm.com/think/topics/computer-vision) and [deep learning](https://www.ibm.com/think/topics/deep-learning) models. It’s a Python library that users install on their computers or servers that provides code that lets them use the models they find on the Hub. It includes model architectures, preprocessing tools, training utilities and more. Built on top of popular frameworks like [PyTorch](https://www.ibm.com/think/topics/pytorch) and TensorFlow, the Transformers library allows users to load powerful ML models like BERT, [GPT](https://www.ibm.com/think/topics/gpt) and others with just a few lines of code. It also offers extensive tools for [fine-tuning](https://www.ibm.com/think/topics/fine-tuning) open source models on custom datasets, making it more useful for research and production.

Other libraries

In addition to Transformers and the Hub, the Hugging Face ecosystem contains libraries for other tasks, such as dataset processing ("Datasets"), model evaluation ("Evaluate"), and machine learning demos ("Gradio").

**HuggingFace Spaces**

*HuggingFace Spaces* is a comprehensive ecosystem designed to facilitate creating, sharing, and deploying machine learning models. This platform is tailored to accommodate novice and experienced AI practitioners, providing tools and resources that streamline the development process.

**Building and Deploying Models**

Building and deploying machine learning models on HuggingFace Spaces is designed to be straightforward and accessible. The platform supports a variety of frameworks, including TensorFlow, PyTorch, and JAX, allowing users to work with their preferred tools.

**Building Models**

Users can start by exploring pre-trained models in the Model Hub. These models can be fine-tuned or adapted to specific tasks using transfer learning techniques. HuggingFace provides extensive documentation and tutorials to guide users through the process of building and customizing models.



**Steps to build a model:**

1. **Select a Pre-trained Model:** Browse the Model Hub and select a pre-trained model that aligns with your task.
2. **Fine-Tune the Model:** Use your own dataset to fine-tune the model. This involves adjusting the model’s parameters and training it on task-specific data.
3. **Evaluate Performance:**Assess the model’s performance using relevant metrics and validation datasets.

**Deploying Models**

Once a model is built and fine-tuned, deploying it on HuggingFace Spaces is a seamless process. The platform handles the infrastructure, ensuring that models are scalable and accessible.

**Steps to deploy a model:**

1. **Upload the Model:**Upload your model to HuggingFace Spaces by creating a new repository or updating an existing one.
2. **Configure Deployment Settings:**Specify deployment settings such as the model’s endpoint, hardware requirements, and scaling options.
3. **Launch the Model:**Deploy the model with a single click. The platform provides a URL for accessing the model’s API, allowing you to integrate it into applications.

**Loading and Preprocessing Data**

The HuggingFace Datasets library simplifies the process of loading and preprocessing data. This library is designed to handle large datasets efficiently, providing tools for data manipulation and transformation.

**Steps to load and preprocess data:**

**1. Install the Datasets Library:**Use pip to install the HuggingFace Datasets library.

pip install datasets

**2. Load a Dataset:** Load a dataset from the repository using the load\_dataset function.

from datasets import load\_dataset

dataset = load\_dataset('dataset\_name')

**3. Preprocess the Data:**Use built-in functions to preprocess the data, such as tokenization, normalization, and augmentation.

**Creating Interactive Demos**

Interactive demos are an excellent way to showcase the capabilities of your models. HuggingFace Spaces provides tools and frameworks to create engaging and interactive applications.

**Using Streamlit**

Streamlit is a popular framework for creating interactive web applications. It allows users to build custom interfaces with minimal code, making it ideal for creating demos.

**Steps to create a demo with Streamlit:**

**1. Install Streamlit:**Use pip to install the Streamlit library.

pip install streamlit

**2. Create a Streamlit App:** Write a Python script to define your Streamlit app.

import streamlit as st

​

def main():

st.title('Streamlit Example')

user\_input = st.number\_input("Enter a number", value=1)

result = user\_input \* 10

st.write(f"10 times your input is {result}")

​

if \_\_name\_\_ == '\_\_main\_\_':

main()

**3. Deploy the App:**Deploy the app on HuggingFace Spaces by uploading the script to a new repository.

**Using Gradio**

Gradio is another framework for creating interactive demos. It provides a simple API for building web-based interfaces to interact with machine learning models.

**Steps to create a demo with Gradio:**

**1. Install Gradio:**Use pip to install the Gradio library.

pip install gradio

**2. Create a Gradio Interface:**Write a Python script to define your Gradio interface.

import gradio as gr

​

def predict(input\_text):

return "Prediction: " + input\_text

​

iface = gr.Interface(fn=predict, inputs="text", outputs="text")

iface.launch()

**3. Deploy the Interface:**Deploy the interface on HuggingFace Spaces by uploading the script to a new repository.

**Community Engagement**

The HuggingFace community is vibrant and active. Users can follow other members, share their projects, and participate in discussions. The platform includes forums and chat features to facilitate communication and knowledge sharing.

**Key community features:**

* **User Profiles:**Create a profile to showcase your projects, models, and contributions.
* **Discussions:** Participate in forum discussions to seek help, share insights, and collaborate on projects.
* **Social Media Integration:**Connect with the HuggingFace community on social media platforms like Twitter, LinkedIn, and GitHub.

**Conclusion**

HuggingFace Spaces is a powerful platform that simplifies the process of building, sharing, and deploying machine learning models. Its user-friendly interface, extensive features, and strong community support make it an excellent choice for both beginners and experienced practitioners in the field of AI and machine learning.

**HuggingFace Datasets**

Datasets is a library for easily accessing and sharing AI datasets for Audio, Computer Vision, and Natural Language Processing (NLP) tasks.

Load a dataset in a single line of code, and use our powerful data processing and streaming methods to quickly get your dataset ready for training in a deep learning model. Backed by the Apache Arrow format, process large datasets with zero-copy reads without any memory constraints for optimal speed and efficiency. We also feature a deep integration with the [Hugging Face Hub](https://huggingface.co/datasets), allowing you to easily load and share a dataset with the wider machine learning community.

Hugging Face Dataset Hub is a platform that hosts an extensive collection of datasets for natural language processing (NLP) tasks and other machine learning domains like computer vision and speech recognition. It serves as a centralized repository where we can discover, download and use datasets for various ML applications.

The Hugging Face Dataset Hub offers several features that make it a go-to platform for ML practitioners:

1. **Diverse Datasets**: The Hub includes datasets for a wide range of tasks such as text classification, question answering, image captioning and much more.
2. **Easy Access:**The datasets are easily accessible via the datasets library which we can install and use in just a few lines of code.
3. **Community Contributions:** The platform encourages collaboration, allowing anyone to share datasets and improvements, promoting a rich ecosystem of publicly available resources.
4. **Integration with Models:** Datasets on the Hub are often paired with pre-trained models allowing us to fine-tune models with minimal setup.

**Accessing and Using Datasets**

We will access a dataset from the hugging face dataset hub by installing the necessary libraries.

*pip install datasets*

**1. Loading a Dataset**

Once the library is installed, we can load any available dataset with a simple line of code. For example, we will load the IMDB dataset which is frequently used for sentiment analysis.

* **load\_dataset("imdb"):** Loads the "imdb" dataset from the Hugging Face Dataset Hub.
* **dataset["train"][0]:** Accesses the first example from the training split of the dataset.

**2. Exploring the Dataset**

The Hugging Face datasets library provides useful methods to explore the loaded datasets. We can check the dataset structure, see the number of entries and access specific splits such as train, test and validation.

* **print(dataset):** Displays the structure of the entire dataset, showing its available splits (e.g., train, test, validation).
* **print(dataset["test"][0:5]):** Displays the first 5 examples from the "test" split of the dataset.

**Popular Datasets on Hugging Face Dataset Hub**

The Hugging Face Dataset Hub is home to a variety of datasets across different domains. Some of the most popular datasets include:

1. **IMDB**: A dataset commonly used for sentiment analysis.
2. **SQuAD (Stanford Question Answering Dataset):** A dataset for machine reading comprehension tasks.
3. **COCO (Common Objects in Context):** A dataset used for image captioning and object detection.
4. **LibriSpeech:** A speech dataset for automatic speech recognition (ASR) tasks.

These datasets are preprocessed and ready to be used for model training and fine-tuning.

**Advanced Features of Dataset Hub**

The Hugging Face Dataset Hub provides advanced features that further enhance the usability and accessibility of datasets:

**1. Dataset Versioning**

Each dataset in the Hub is versioned which means we can track changes made over time. This feature ensures reproducibility and allows us to use specific versions of a dataset for model training.

**2. Dataset Streaming**

Hugging Face supports dataset streaming for large datasets that may be too large to fit in memory. This feature allows us to stream data from the Hub without needing to download the entire dataset upfront. We will be loading squad dataset which is a very large dataset.

* **load\_dataset("squad", streaming=True):** Loads the "squad" dataset in streaming mode
* **for example in dataset["train"]:** The loop iterates through the "train" split of the dataset.
* **break:** Stops the loop after printing the first example

**3. Dataset Splitting**

The datasets library also supports splitting of datasets into training, validation and test sets. This is particularly useful for preparing datasets for model training.

* **dataset.keys()**: Lists the available splits (e.g., 'train', 'validation') in the dataset.
* **dataset["train"]**: Accesses the training split of the dataset.
* **dataset["validation"]:** Accesses the validation split of the dataset.
* **take(n):** Retrieves the first n examples from the dataset (in this case, 1 example).