## **📘 Teaching Notes: Hugging Face from Scratch**

### **🔀 Prerequisites**

Before beginning with Hugging Face, ensure learners are familiar with:

1. Python Programming (functions, classes, loops)
2. Basics of NLP (tokenization, embeddings, sequence models)
3. PyTorch or TensorFlow basics
4. Transformers (encoder-decoder concept)
5. Jupyter/Colab usage
6. Git and GitHub basics

## Understanding LLMs

-----------------------

This course will teach you about large language models (LLMs) and natural language processing (NLP) using libraries from the [Hugging Face](https://huggingface.co/) ecosystem — [🤗 Transformers](https://github.com/huggingface/transformers), [🤗 Datasets](https://github.com/huggingface/datasets), [🤗 Tokenizers](https://github.com/huggingface/tokenizers), and [🤗 Accelerate](https://github.com/huggingface/accelerate) — as well as the [Hugging Face Hub](https://huggingface.co/models).

### **📚 Course Overview**

|  |  |  |
| --- | --- | --- |
| **Module** | **Topic** | **Description** |
| 1 | Introduction to Hugging Face | Ecosystem, history, libraries overview |
| 2 | Hugging Face Hub | Models, datasets, spaces |
| 3 | Transformers Library | Pipelines, tokenizers, pretrained models |
| 4 | Tokenizers | Types and usage |
| 5 | Inference with Transformers | Classification, generation, QA |
| 6 | Fine-tuning | Trainer API, metrics, custom datasets |
| 7 | Datasets Library | Preprocessing and loading |
| 8 | Accelerate & PEFT | Efficient training |
| 9 | Hugging Face Spaces | App deployment with Gradio/Streamlit |
| 10 | Real-World Projects | Practical implementations |
| 11 | Deployment & Sharing | Model cards, hub push, API usage |

### **🛋️ Detailed Modules**

#### **Module 1: Introduction to Hugging Face**

* History and mission
* Libraries: transformers, datasets, accelerate, etc.
* Use cases: NLP, CV, Audio, Multimodal

#### **Module 2: Hugging Face Hub**

* Model browsing
* Dataset browsing
* Spaces overview
* Model card metadata

#### **Module 3: Transformers Library**

* Installing library
* pipeline() usage for zero-shot tasks

from transformers import pipeline

classifier = pipeline("sentiment-analysis")

print(classifier("Hugging Face is amazing!"))

* AutoModel, AutoTokenizer

#### **Module 4: Tokenizers**

* Pretrained vs custom
* WordPiece, BPE, Unigram
* Encoding and decoding examples

#### **Module 5: Inference with Transformers**

* Sentiment analysis
* NER
* Text generation
* QA and summarization

#### **Module 6: Fine-tuning**

* Trainer and TrainingArguments
* Loading and preprocessing datasets
* Evaluation metrics
* Saving and uploading models

#### **Module 7: Datasets Library**

* load\_dataset() usage
* Exploring and filtering
* Mapping and tokenizing

#### **Module 8: Accelerate & PEFT**

* accelerate library for multi-GPU
* LoRA and quantization

#### **Module 9: Hugging Face Spaces**

* Gradio/Streamlit apps
* Hosting models
* Private/public options

#### **Module 10: Real-World Projects**

* Project 1: Sentiment App (Gradio)
* Project 2: Summarizer (T5)
* Project 3: Chatbot (DialoGPT)
* Project 4: Q&A system (BERT)

#### **Module 11: Deployment & Sharing**

* push\_to\_hub()
* Model card creation
* Using Inference API

### **🧪 Lab Exercises**

* Test sentiment model with pipeline
* Fine-tune on custom CSV dataset
* Deploy summarizer on Gradio
* Upload model and create space

### **📖 Resources**

* Hugging Face: <https://huggingface.co/learn/nlp-course>
* DeepLearning.AI: <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>
* Coursera NLP Specialization
* Fast.ai, Karpathy’s Zero to Hero
* [Hugging Face Course](https://huggingface.co/course)
* [Transformers Docs](https://huggingface.co/docs/transformers)
* [HF Notebooks GitHub](https://github.com/huggingface/notebooks)
* [YouTube Tutorials](https://www.youtube.com/c/HuggingFace)