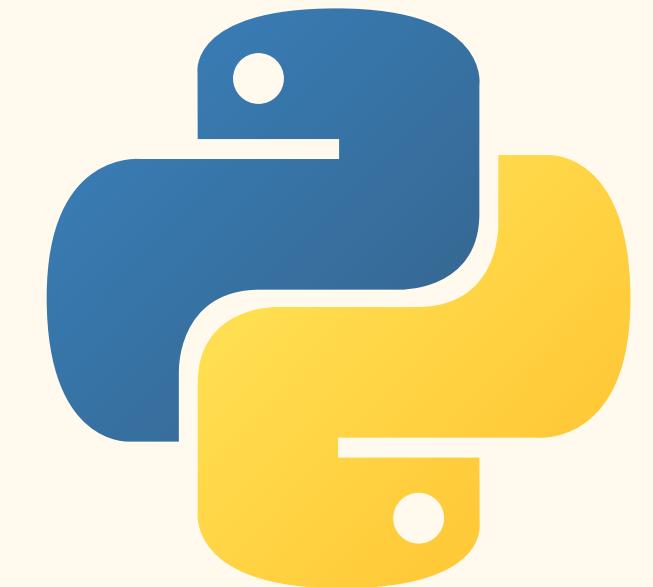
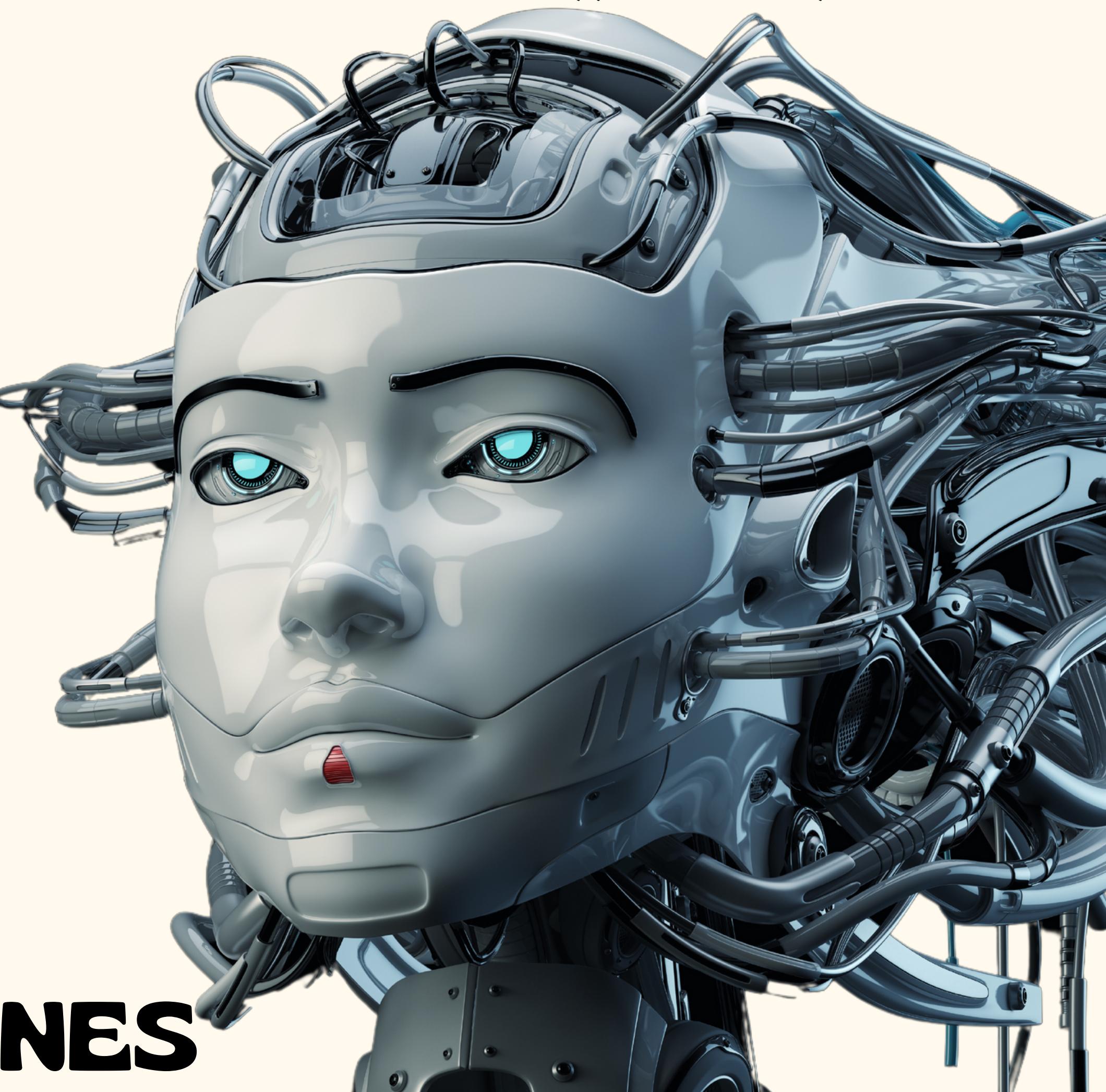


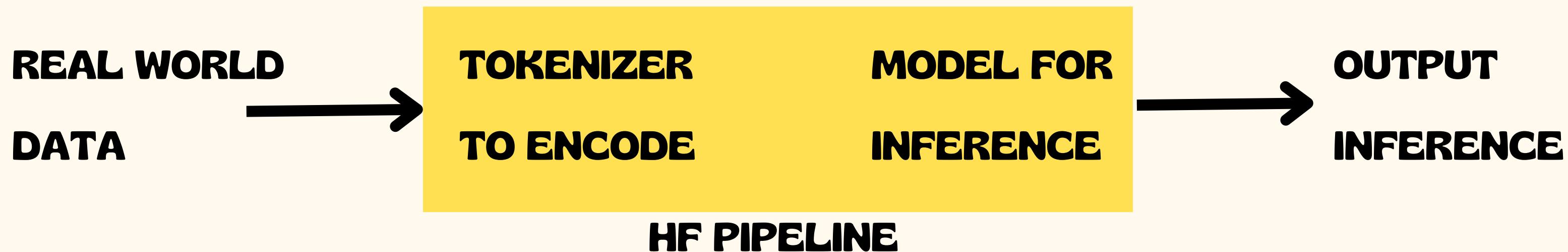
MASTERING HUGGING FACE TRANSFORMERS



A STEP-BY-STEP
GUIDE TO MODEL
FINETUNING &
INFERENCE PIPELINES



TRAINING A NLP MODEL : 5 STEPS



STEP 1: TRAINING DATA TO BE PREPROCESSED IN THE FORMAT MODEL EXPECTS
DEPENDING ON MEDIA, A TOKENIZER, FEATURE EXTRACTOR OR IMAGE PROCESSOR WILL BE USED

STEP 2: LOAD A PRE-TRAINED MODEL CHECKPOINT SPECIFIC TO YOUR TASK FOR FINE-TUNING
PROCESS

STEP 3: INSTANTIATE THE TRAINING ARGUMENTS WHICH CONTAINS THE HYPERPARAMETERS

STEP 4: INSTANTIATE EVALUATE INSTANCE THAT WILL HELP TRAINER TO CALCULATE METRICS

STEP 5: CREATE THE TRAINER INSTANCE THAT CONTAINS

- MODEL
- TRAINING ARGS + EVALUATION METRIC
- TRAIN + EVAL DATASETS

CHALLENGE SOLVED : TRANSFORMERS

WITH TRANSFORMERS, THERE IS NO NEED TO CREATE THE NEURAL NETWORK MODEL FROM SCRATCH USING THE PYTORCH/ TF/ JAX PRIMITIVES.

1. CREATE INFERENCE ENDPOINT WITH SINGLE LINE

OF CODE USING PIPELINE CLASS

2. LOAD TOKENIZER AND MODEL WITH SINGLE LINE

METHOD CALLED FROM_PRETRAINED

3. MODEL CAN BE USED AS THE STARTING POINT FOR

FINETUNING INSIDE TRAINER CLASS

4. CREATE NEW MODEL FROM SCRATCH USING THE

AUTOCONFIG CLASS.

5. SAVING, LOADING AND SHARING IS SIMPLE JSON

[HTTPS://PYTORCH.ORG/TUTORIALS/BEGINNER/BLITZ/CIFAR10_TUTORIAL.HTML](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)

[HTTPS://WWW.KAGGLE.COM/CODE/BASU369VICTOR/PYTORCH-TUTORIAL-THE-CLASSIFICATION/NOTEBOOK](https://www.kaggle.com/code/basu369victor/pytorch-tutorial-the-classification-notebook)

```
import torch.nn as nn
import torch.nn.functional as F

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = torch.flatten(x, 1) # flatten all dimensions except batch
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x

net = Net()
```

[HTTPS://GITHUB.COM/INSIGHTBUILDER](https://github.com/insightbuilder)

METHODS TO PLAY WITH DATASETS

1. AUTOTOKENIZER LOADS TOKENIZER

2. AUTOMODEL LOADS MODEL

A. MODEL CAN BE LOADED WITH W/B

**B. PLAIN ARCHITECTURE IF CONFIG IS
GIVEN**

3. AUTOCONFIG EXTRACTS THE MODEL

ARCHITECTURE

**4. AUTOMODELFOR CLASS LOADS THE
MODEL FOR A PARTICULAR TASK**

**5. TRAININGARGUMENTS WILL LOAD THE
TRAINING PARAMETERS**

**insightbuilder/
python_de_learners_data**

Repo contains the code, data and supporting documents including presentations, playbooks and additional documents to support learning

2
Contributors

0
Issues

48
Stars

20
Forks



**python_de_learners_data/huggingface_transformers_trainers.ipynb at main ·
insightbuilder/python_de_learners_data**

Repo contains the code, data and supporting documents including presentations, playbooks and additional documents to support learning - python_de_learners_data/huggingface_transformers_trainers.ip...
[GitHub](#)

THANKS FOR WATCHING

 **LIKE**

 **SHARE**

 **SUBSCRIBE**