Clone and Perform Inference with Hugging Face Text-to-Speech (TTS) Model

1. Objective

- Learn how to clone a pre-trained Text-to-Speech (TTS) model from Hugging Face.
- Perform inference to convert input text into spoken audio using the cloned model.
- (Optional) Understand the basics of hosting TTS models with Hugging Face Spaces.

2. Problem Statement

- Text-to-Speech (TTS) technology converts written text into natural-sounding speech, enabling applications such as voice assistants, accessibility tools, and multimedia content creation.
- In this exercise, you will clone a pre-trained TTS model from Hugging Face and perform inference to generate audio from text.

3. Inputs / Shared Artifacts

- No starter code or data provided.
- You will clone a publicly available TTS model from Hugging Face (any suitable model, not restricted to fastspeech2-en-ljspeech).
- Use Python and required libraries (transformers, torch, etc).
- Input text for TTS can be any sentence you choose.

4. Expected Outcome

- Successfully clone a pre-trained TTS model from Hugging Face.
- Perform inference to synthesize audio from input text.
- Play or save the generated audio waveform.

5. Concepts Covered

- Model cloning and loading from Hugging Face Hub.
- Using the Transformers TTS interface for audio synthesis.
- Basic audio processing and playback in Python.
- (Optional) Model deployment and hosting on Hugging Face Spaces.

6. Example: Step-by-Step Instructions with Code

```
# Step 1: Install required packages (run in terminal or notebook)

# pip install transformers torch IPython

# Step 2: Import required libraries

from transformers import VitsModel, AutoTokenizer
import torch
from IPython.display import Audio

# Step 3: Clone and load the pre-trained TTS model from Hugging Face
model = VitsModel.from_pretrained("facebook/mms-tts-vie")  # You may replace
this with any compatible TTS model
tokenizer = AutoTokenizer.from_pretrained("facebook/mms-tts-vie")



# Step 4: Prepare input text

text = "Xin chào anh em đến với bài tập của khoá AI Application Engineer"  #
Example text in Vietnamese

# Step 5: Tokenize the input text

inputs = tokenizer(text, return_tensors="pt")

# Step 6: Perform inference to generate the waveform
```

```
with torch.no_grad():
    output = model(**inputs).waveform

# Step 7: Play the generated audio in Jupyter Notebook

Audio(output.numpy(), rate=model.config.sampling_rate)

# Optional: Save audio to file (requires soundfile)

# import soundfile as sf

# sf.write('output.wav', output.numpy(), model.config.sampling_rate)
```

7. Final Submission Checklist

- Submit your notebook(.ipynb) or .py file containing the full code for cloning and running inference with the TTS model.
- Include preview generated audio in Notebook or audio file.
- Add comments explaining each step of the code and any challenges faced.