Project Goal:

Create a small app that uses a basic Generative AI model to generate new motivational quotes based on a set of examples.

How it works:

1. Input: A small set of real motivational quotes (5-10 examples).
2. Model: A lightweight text generation model like GPT-2 or a simple fine-tuned model
3. Output: motivational quotes generated by AI

Why it is easy

- No need for massive datasets
- No cost
- Easy to run on free tools like Jupiter or Colab
- Introduction to AI – Generating text with small model
- “wow” project that you can talk about
- The model generates quotes automatically if using gpt-2 or hugging face model and quotes dynamically every time

Real World Best Practices

- Pre-Generated Quotes
- Manually Review
- Clean them for quality
- Store them in a database
- Display to users

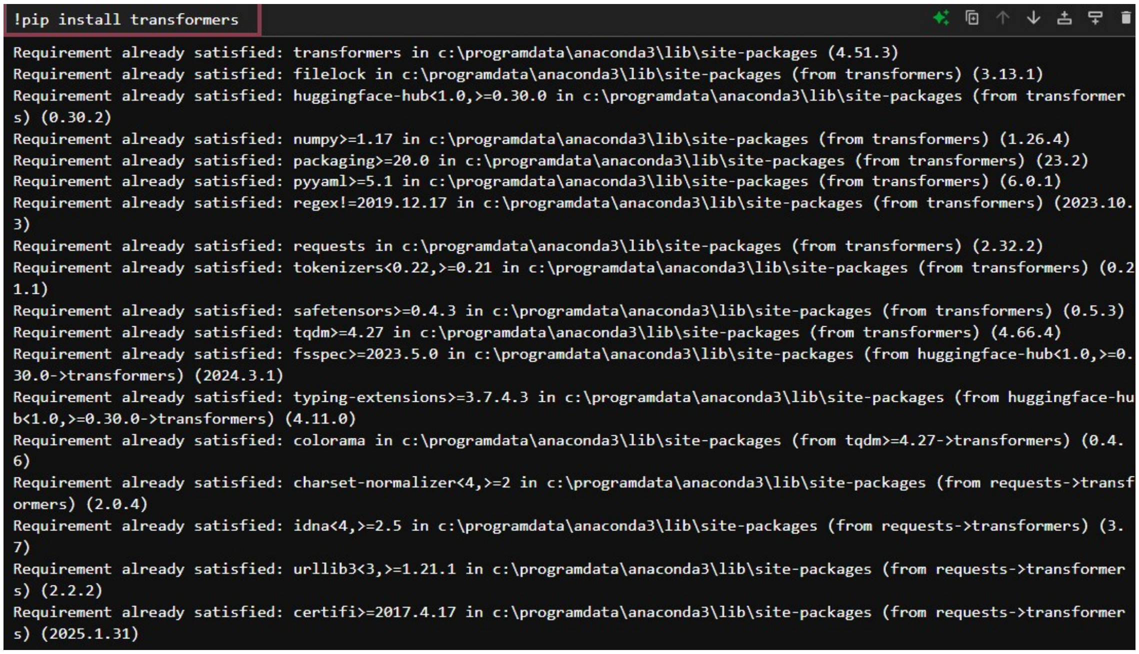
Note: AI outputs sometimes are weird

Disclaimer: The quotes displayed here are generated by an AI model and are for motivational purposes. They may not reflect the views or opinions of any individual.

Steps:

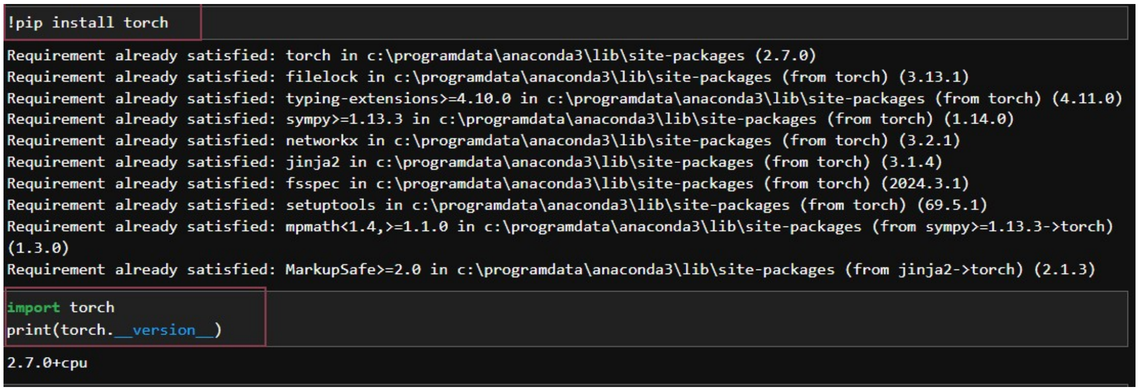
1. Open jupyter notebook
2. Select Python3 as kernel
3. Install Hugging Face Transformers (!pip install transformers) which has GPT-2 model to be used.
4. Install the Pytorch library **!pip install torch** (or) **!pip install tensor flow**. This is a small project and Pytorch is lighter for this project)
5. Verify the installation if torch is available
6. import torch
7. print(torch._version_)

Installation Steps:



```
!pip install transformers
```

```
Requirement already satisfied: transformers in c:\programdata\anaconda3\lib\site-packages (4.51.3)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in c:\programdata\anaconda3\lib\site-packages (from transformer
s) (0.30.2)
Requirement already satisfied: numpy>=1.17 in c:\programdata\anaconda3\lib\site-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site-packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in c:\programdata\anaconda3\lib\site-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in c:\programdata\anaconda3\lib\site-packages (from transformers) (2023.10.
3)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from transformers) (2.32.2)
Requirement already satisfied: tokenizers<0.22,>=0.21 in c:\programdata\anaconda3\lib\site-packages (from transformers) (0.2
1.1)
Requirement already satisfied: safetensors>=0.4.3 in c:\programdata\anaconda3\lib\site-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in c:\programdata\anaconda3\lib\site-packages (from transformers) (4.66.4)
Requirement already satisfied: fsspec>=2023.5.0 in c:\programdata\anaconda3\lib\site-packages (from huggingface-hub<1.0,>=0.
30.0->transformers) (2024.3.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\programdata\anaconda3\lib\site-packages (from huggingface-hu
b<1.0,>=0.30.0->transformers) (4.11.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from tqdm>=4.27->transformers) (0.4.
6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->transf
ormers) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->transformers) (3.
7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests->transformer
s) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->transformer
s) (2025.1.31)
```



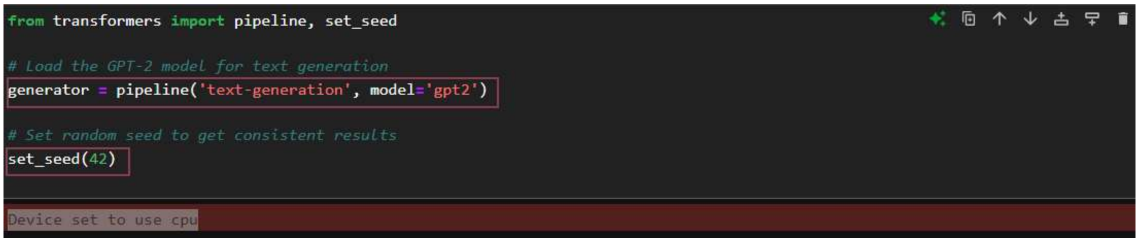
```
!pip install torch
```

```
Requirement already satisfied: torch in c:\programdata\anaconda3\lib\site-packages (2.7.0)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\site-packages (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\programdata\anaconda3\lib\site-packages (from torch) (4.11.0)
Requirement already satisfied: sympy>=1.13.3 in c:\programdata\anaconda3\lib\site-packages (from torch) (1.14.0)
Requirement already satisfied: networkx in c:\programdata\anaconda3\lib\site-packages (from torch) (3.2.1)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\site-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in c:\programdata\anaconda3\lib\site-packages (from torch) (2024.3.1)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from torch) (69.5.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\programdata\anaconda3\lib\site-packages (from sympy>=1.13.3->torch)
(1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\site-packages (from jinja2->torch) (2.1.3)
```

```
import torch
print(torch.__version__)
```

```
2.7.0+cpu
```

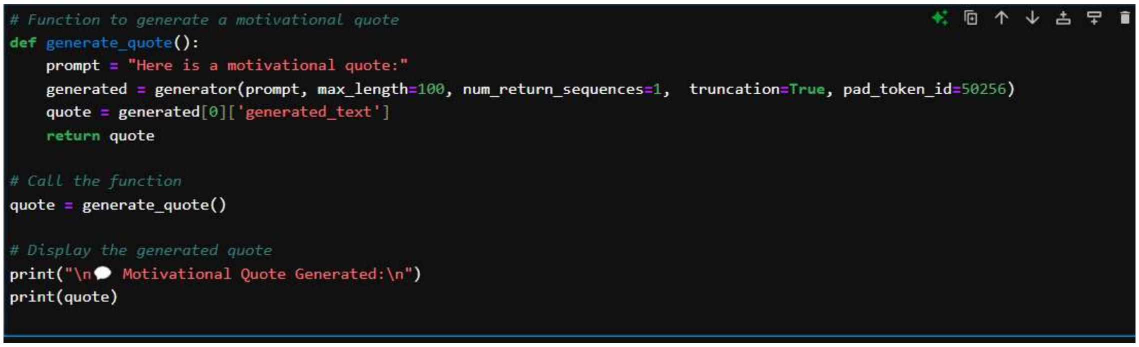
8. Load the GPT-2 model for text generation
9. The code pipeline('text-generation') load a read to use GPT-2 model
10. Set a random seed for consistent output every time



```
from transformers import pipeline, set_seed

# Load the GPT-2 model for text generation
generator = pipeline('text-generation', model='gpt2')

# Set random seed to get consistent results
set_seed(42)
```

```
Device set to use cpu
```

11. The CPU indicates that your machine does not have the GPU or not configured. For the demo project it is sufficient.
12. Create a function to generate motivational quote

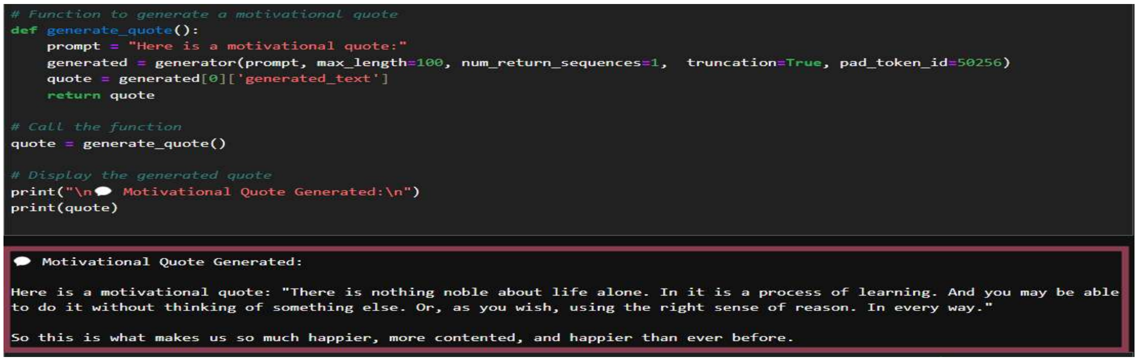


```
# Function to generate a motivational quote
def generate_quote():
    prompt = "Here is a motivational quote:"
    generated = generator(prompt, max_length=100, num_return_sequences=1,  truncation=True, pad_token_id=50256)
    quote = generated[0]['generated_text']
    return quote

# Call the function
quote = generate_quote()

# Display the generated quote
print("\n💬 Motivational Quote Generated:\n")
print(quote)
```

Code Explanation

1. generate_quote is the function name and does not take any parameters
2. The prompt is the starting text when generating the quote
3. The code calls the function generator which is a model as shown in the previous screenshot
4. The quote line extracts the generated quote from the result stored in generated
5. max_length=50: The maximum length of the generated quote is set to 50.characters. **This can be set to higher value if the quotes are not complete**

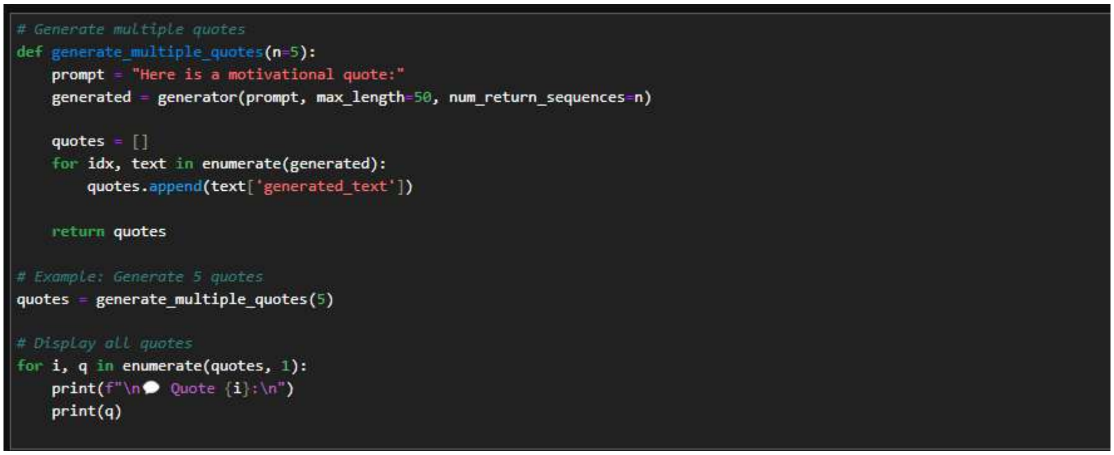
6. Multiple quotes can be generated in one go and for that modify the code
7. Adding truncation=true and pad_token=50526 – no more warnings. Pad-tokens are like sequences to ensure they are of the same length in a batch and 50526 is like a tokenizier special id used for specific models likle GPT-2



```
# Function to generate a motivational quote
def generate_quote():
    prompt = "Here is a motivational quote:"
    generated = generator(prompt, max_length=100, num_return_sequences=1,  truncation=True, pad_token_id=50256)
    quote = generated[0]['generated_text']
    return quote

# Call the function
quote = generate_quote()

# Display the generated quote
print("\n💬 Motivational Quote Generated:\n")
print(quote)
```

```
💬 Motivational Quote Generated:

Here is a motivational quote: "There is nothing noble about life alone. In it is a process of learning. And you may be able
to do it without thinking of something else. Or, as you wish, using the right sense of reason. In every way."

So this is what makes us so much happier, more contented, and happier than ever before.
```

Additional code for multiple quotes



```
# Generate multiple quotes
def generate_multiple_quotes(n=5):
    prompt = "Here is a motivational quote:"
    generated = generator(prompt, max_length=50, num_return_sequences=n)

    quotes = []
    for idx, text in enumerate(generated):
        quotes.append(text['generated_text'])

    return quotes

# Example: Generate 5 quotes
quotes = generate_multiple_quotes(5)

# Display all quotes
for i, q in enumerate(quotes, 1):
    print(f"\n💬 Quote {i}:\n")
    print(q)
```

Disclaimer: The quotes displayed here are generated by an AI model and are for motivational purposes. They may not reflect the views or opinions of any individual.