```
# Étape 1 : Installer la bibliothèque transformers
!pip install transformers --quiet

# Étape 2 : Importer les modules nécessaires
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline

# Étape 3 : Choisir et charger un LLM depuis Hugging Face
model_name = "google/flan-t5-base"

# Charger le tokenizer et le modèle
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

# Créer un pipeline pour générer du texte
llm = pipeline("text2text-generation", model=model, tokenizer=tokenizer)

# Étape 4 : Utiliser le LLM sur une tâche simple
prompt = "Explain in simple words what machine learning is."

# Générer une réponse à partir du prompt
response = llm(prompt, max_new_tokens=100)

# Afficher le texte généré
print("Prompt :", prompt)
print("Response :", response[0]['generated_text'])
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json:  2.54k/?  [00:00<00:00,  110kB/s]
spiece.model:  100%  792k/792k  [00:00<00:00,  1.35MB/s]
tokenizer.json:  2.42M/?  [00:00<00:00,  17.0MB/s]
special_tokens_map.json:  2.20k/?  [00:00<00:00,  128kB/s]
config.json:  1.40k/?  [00:00<00:00,  82.1kB/s]
model.safetensors:  100%  990M/990M  [00:20<00:00,  45.2MB/s]
generation_config.json:  100%  147/147  [00:00<00:00,  9.95kB/s]
Device set to use cpu
Prompt : Explain in simple words what machine learning is.
```

```
from google.colab import files
import pickle

uploaded = files.upload()
```

```
Sélect. fichiers spam.csv
• spam.csv(text/csv) - 479260 bytes, last modified: 02/07/2025 - 100% done
Saving spam.csv to spam.csv
```

```
import pandas as pd

# Charger le fichier correctement
df = pd.read_csv("spam.csv", encoding='latin-1')
df = df[['target', 'text']]  # On garde les colonnes utiles
df.head()
```

	target	text
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Étapes suivantes : Générer du cod

Que puis-je vous aider à créer ?

```
from transformers import pipeline, AutoTokenizer, AutoModelForSeq2SeqLM

model_name = "google/flan-t5-base"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
llm = pipeline("text2text-generation", model=model, tokenizer=tokenizer)
```

```
Device set to use cpu
```

```
def classify_email(text):
    text = str(text)
    prompt = f"Classify this message as spam or ham. Answer only with 'spam' or 'ham'. Message: {text}"

    # Tronquer si trop long
    if len(prompt) > 1000:
        prompt = prompt[:1000]

    response = llm(prompt, max_new_tokens=10)[0]['generated_text'].lower().strip()

    # Nettoyer la sortie pour ne garder que 'spam' ou 'ham'
    if 'spam' in response:
        return 'spam'
    elif 'ham' in response:
        return 'ham'
    else:
        return 'unknown'



df_sample = df.sample(5, random_state=42)  # 5 e-mails aléatoires
df_sample['llm_class'] = df_sample['text'].apply(classify_email)
df_sample[['text', 'llm_class']]
```

	text	llm_class
3245	Funny fact Nobody teaches volcanoes 2 erupt, t...	ham
944	I sent my scores to sophas and i had to do sec...	spam
1044	We know someone who you know that fancies you....	spam
2484	Only if you promise your getting out as SOON a...	spam
812	Congratulations ur awarded either Ã¥Â£500 of C...	spam