


```
top_k=50, top_p=0.95)
```

```
print("\n--- Generated Creative Content ---\n")  
print(tokenizer.decode(output[0], skip_special_tokens=True))
```

Output :

```
/usr/local/lib/python3.12/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 secret in your Google Colab and restart the notebook.  
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: 100% ██████████ 26.0/26.0 [00:00<00:00, 2.08kB/s]  
config.json: 100% ██████████ 762/762 [00:00<00:00, 71.2kB/s]  
vocab.json: 100% ██████████ 1.04M/1.04M [00:00<00:00, 2.47MB/s]  
merges.txt: 100% ██████████ 456k/456k [00:00<00:00, 1.07MB/s]  
tokenizer.json: 100% ██████████ 1.36M/1.36M [00:00<00:00, 1.60MB/s]  
model.safetensors: 100% ██████████ 353M/353M [00:05<00:00, 88.1MB/s]  
generation_config.json: 100% ██████████ 124/124 [00:00<00:00, 10.0kB/s]  
Setting 'pad_token_id' to 'eos_token_id':50256 for open-end generation.  
--- Generated Creative Content ---  
  
Imagine a magical forest where books can talk. Write a short story: "I'm in the middle of my writing career. I've been doing my best to get my book published." You might think it's a bit silly, but that's the book I'm writing right now, and it's the best I've done. It's a really great book, and it's a really great book, and it's really funny. I'm really excited,
```

2. Implement simple probabilistic models like Gaussian Mixture Models.

Practical 2: Gaussian Mixture Models (GMM)

```
!pip install scikit-learn matplotlib --quiet import
```

```
numpy as np  
import matplotlib.pyplot as plt  
from sklearn.mixture import GaussianMixture  
from sklearn.datasets import make_blobs
```

Generate sample 2D dataset

```
X, y_true = make_blobs(n_samples=400, centers=3, cluster_std=1.0,  
random_state=42)
```

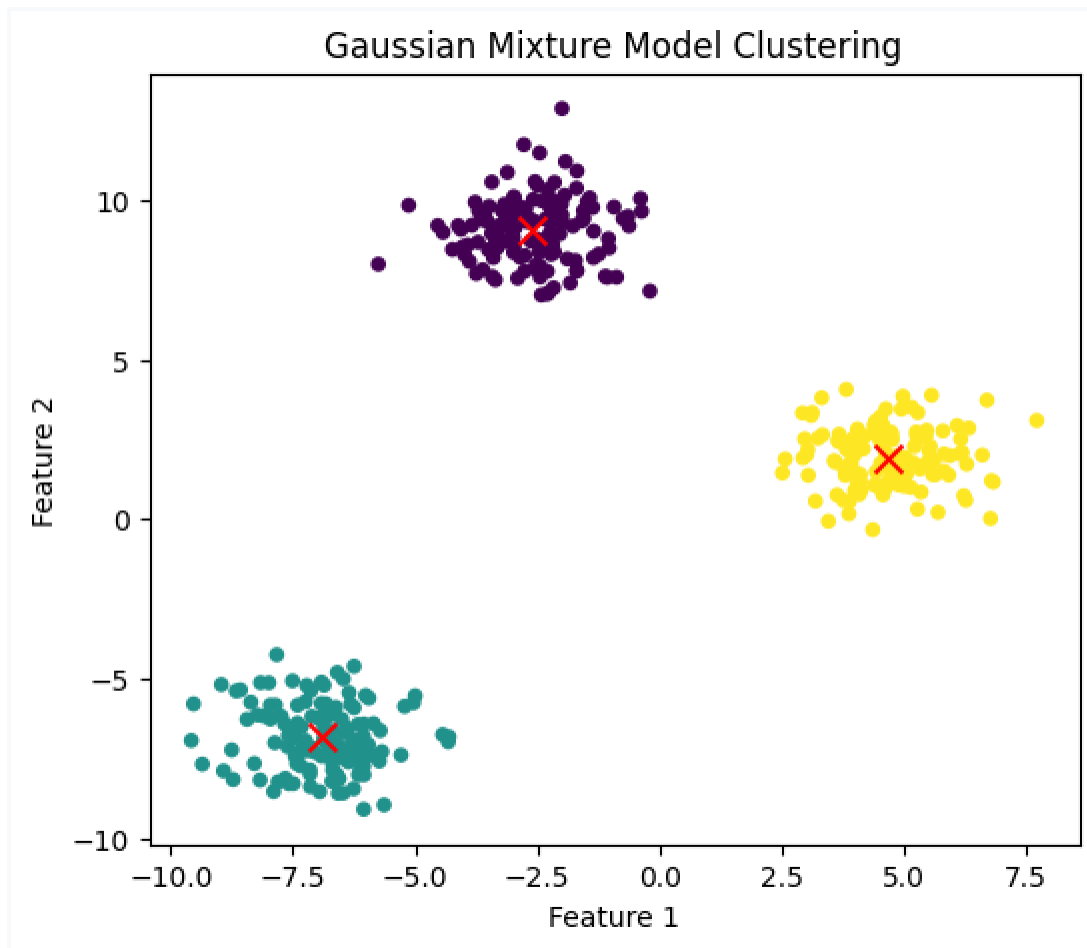
Fit GMM with 3 components

```
gmm = GaussianMixture(n_components=3, covariance_type='full',  
random_state=42)
```

```
gmm.fit(X)
labels = gmm.predict(X)
```

```
#Plot clustering results
```

```
plt.figure(figsize=(6,5))
plt.scatter(X[:,0], X[:,1], c=labels, s=20, cmap='viridis')
plt.scatter(gmm.means_[0], gmm.means_[1], c='red', marker='x', s=100)
plt.title("Gaussian Mixture Model Clustering")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()
```



Code Link :

<https://colab.research.google.com/drive/1-XRAHC-akZojnArKibVxKBklqaBZPBI?usp=sharing>