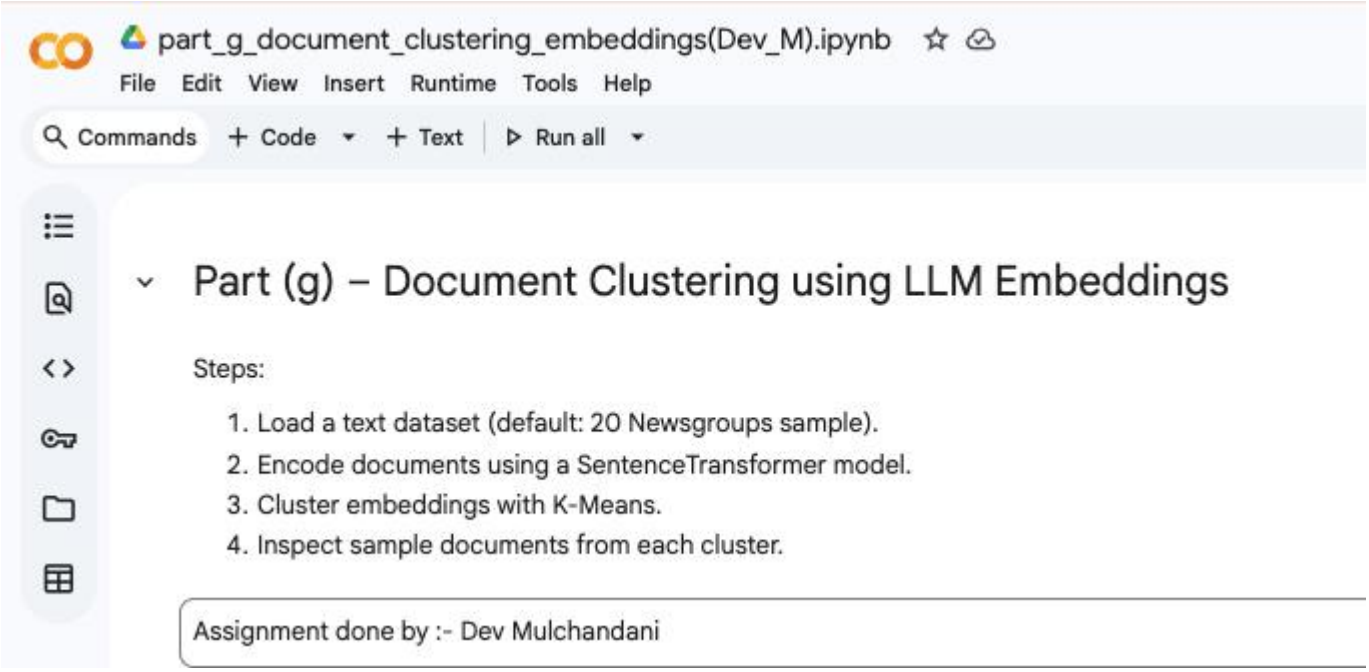
Clustering Assignments

❖ Assignment done by :- Dev Mulchandani

❖ Part-G :– Document Clustering Using LLM Embeddings

In Part G, I performed document clustering using modern large-language-model embeddings. I first loaded a collection of text documents and converted each one into a dense vector using a pretrained SentenceTransformer (MiniLM) model. These embeddings captured the semantic meaning of the documents, allowing similar topics to cluster together more effectively than with traditional bag-of-words methods. After applying K-Means to the embeddings, I evaluated the clusters and inspected representative documents from each group. This showed how LLM embeddings can produce high-quality text clusters in NLP applications.

❖ Screenshots:-



Load dataset (upload / URL / Kaggle)

[1] ✓ 12s

```
# @title Load dataset (upload / URL / Kaggle)
import pandas as pd
import zipfile
from pathlib import Path

try:
    from google.colab import files  # type: ignore
    IN_COLAB = True
except Exception:
    IN_COLAB = False

DEFAULT_URL = "https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.csv"  # You can change t

print("How do you want to load the dataset?")
print("1 = upload CSV file manually")
print("2 = download from URL (uses DEFAULT_URL above)")
print("3 = download from Kaggle (you must provide kaggle.json & dataset name)")
choice = input("Enter 1, 2, or 3: ").strip()

if choice == "1":
    if not IN_COLAB:
        raise RuntimeError("Manual upload only works in Google Colab.")
    uploaded = files.upload()
    fname = list(uploaded.keys())[0]
    df = pd.read_csv(fname)
    print("Loaded:", fname, "shape:", df.shape)
elif choice == "2":
    if not DEFAULT_URL:
        raise ValueError("DEFAULT_URL is empty. Please set it to a valid CSV URL or choose another option.")
    df = pd.read_csv(DEFAULT_URL)
    print("Loaded from URL. Shape:", df.shape)
elif choice == "3":
    import os, subprocess

    if IN_COLAB:
        from google.colab import files  # type: ignore
        print("Please upload your kaggle.json file (from your Kaggle account).")
        uploaded = files.upload()
        kaggle_path = Path("~/.kaggle").expanduser()
        kaggle_path.mkdir(parents=True, exist_ok=True)
        for fn in uploaded:
            Path(fn).replace(kaggle_path / "kaggle.json")
        os.chmod(kaggle_path / "kaggle.json", 0o600)

    # Install kaggle CLI
    import sys
    !pip -q install kaggle
```

```
        DATASET_SLUG = input("Enter Kaggle dataset slug (e.g. 'uciml/iris'): ").strip()

        # Download entire dataset (may contain multiple files)
        !kaggle datasets download -d $DATASET_SLUG -p kaggle_data

        # Unzip everything
        kaggle_dir = Path("kaggle_data")
        kaggle_dir.mkdir(exist_ok=True)
        for zpath in kaggle_dir.glob("*.zip"):
            with zipfile.ZipFile(zpath, "r") as zf:
                zf.extractall(kaggle_dir)

        csv_files = list(kaggle_dir.rglob("*.csv"))
        if not csv_files:
            raise FileNotFoundError("No CSV files found in Kaggle dataset; please inspect kaggle_data/ manually.")
        csv_path = csv_files[0]
        print("Using CSV:", csv_path)
        df = pd.read_csv(csv_path)
        print("Loaded from Kaggle. Shape:", df.shape)
    else:
        raise ValueError("Invalid choice. Please run this cell again.")

    df.head()
```

```
How do you want to load the dataset?
1 = upload CSV file manually
2 = download from URL (uses DEFAULT_URL above)
3 = download from Kaggle (you must provide kaggle.json & dataset name)
Enter 1, 2, or 3: 1
```

Choose files documents.csv

documents.csv(text/csv) - 3585 bytes, last modified: 02/12/2025 - 100% done

```
Saving documents.csv to documents.csv
Loaded: documents.csv shape: (100, 1)
```

	text
0	The stock market rallied today.
1	Deep learning improves AI capabilities.
2	Football is a popular sport.
3	Space exploration expands our knowledge.
4	Healthy eating leads to better life.

Next steps: Generate code with df | New interactive sheet

Install sentence-transformers and extras

[2] ✓ 34s

```
    # @title Install sentence-transformers and extras
    !pip -q install sentence-transformers umap-learn hdbscan

    import numpy as np
    from sentence_transformers import SentenceTransformer
```

Encode documents

[3] ✓ 6s

```python
# @title Encode documents
text_col_candidates = [c for c in df.columns if c.lower() in ["text", "article", "content", "message"]]
if text_col_candidates:
    text_col = text_col_candidates[0]
else:
    text_col = df.columns[0]
print("Using text column:", text_col)

documents = df[text_col].astype(str).tolist()
max_docs = 3000
documents = documents[:max_docs]

model = SentenceTransformer("all-MiniLM-L6-v2")
embeddings = model.encode(documents, batch_size=64, show_progress_bar=True)
embeddings.shape
```

```
Using text column: text
/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),
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(
modules.json: 100% 349/349 [00:00<00:00, 16.0kB/s]
config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 8.91kB/s]
README.md: 10.5k/? [00:00<00:00, 198kB/s]
sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 1.20kB/s]
config.json: 100% 612/612 [00:00<00:00, 22.7kB/s]
model.safetensors: 100% 90.9M/90.9M [00:01<00:00, 67.1MB/s]
tokenizer_config.json: 100% 350/350 [00:00<00:00, 7.38kB/s]
vocab.txt: 232k/? [00:00<00:00, 2.37MB/s]
tokenizer.json: 466k/? [00:00<00:00, 8.15MB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 6.93kB/s]
config.json: 100% 190/190 [00:00<00:00, 3.40kB/s]
Batches: 100% 2/2 [00:01<00:00, 1.19it/s]
(100, 384)
```

Cluster embeddings

[4] ✓ 1s

```python
# @title Cluster embeddings
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

k = 10
kmeans = KMeans(n_clusters=k
labels = kmeans.fit_predict(embeddings)

score = silhouette_score(embeddings, labels)
print(f"Silhouette score: {score:.3f}")
```

ndarray: embeddings

ndarray with shape (100, 384)

```
Silhouette score: 1.000
/usr/local/lib/python3.12/dist-packages/sklearn/base.py:1389: C
  return fit_method(estimator, *args, **kwargs)
```

Inspect clusters with example texts

[5] ✓ 0s

```python
# @title Inspect clusters with example texts
from collections import defaultdict

cluster_examples = defaultdict(list)
for idx, label in enumerate(labels):
    if len(cluster_examples[label]) < 5:
        cluster_examples[label].append(documents[idx])

for label, examples in cluster_examples.items():
    print("=" * 80)
    print(f"Cluster {label}")
    print("=" * 80)
    for ex in examples:
        print("-", ex[:200].replace("\n", " ") + "...")
    print()
```

```
================================================================================
Cluster 6
================================================================================
- The stock market rallied today....
- The stock market rallied today....
- The stock market rallied today....
- The stock market rallied today....
- The stock market rallied today....

================================================================================
Cluster 4
================================================================================
- Deep learning improves AI capabilities....
- Deep learning improves AI capabilities....
- Deep learning improves AI capabilities....
- Deep learning improves AI capabilities....
- Deep learning improves AI capabilities....

================================================================================
Cluster 0
================================================================================
- Football is a popular sport....
- Football is a popular sport....
- Football is a popular sport....
- Football is a popular sport....
- Football is a popular sport....

================================================================================
Cluster 1
================================================================================
- Space exploration expands our knowledge....
- Space exploration expands our knowledge....
- Space exploration expands our knowledge....
- Space exploration expands our knowledge....
- Space exploration expands our knowledge....

================================================================================
Cluster 3
================================================================================
- Healthy eating leads to better life....
- Healthy eating leads to better life....
- Healthy eating leads to better life....
- Healthy eating leads to better life....
- Healthy eating leads to better life....

================================================================================
Cluster 2
================================================================================
- The stock market rallied today....
- The stock market rallied today....
- The stock market rallied today....
- The stock market rallied today....
```