RAG using the Monogo DB and Pinecone

1 Download the required Libarires

```
[ ]: !pip install -q datasets pandas sentence_transformers pymongo pinecone-client
     langchain-google-genai
```

2 preprocessing of data

```
[ ]: from datasets import load_dataset
```

```
[ ]: dataset=load_dataset("MongoDB/embedded_movies")
```

```
Downloading readme: 0%|          | 0.00/6.18k [00:00<?, ?B/s]
Downloading data:   0%|          | 0.00/42.3M [00:00<?, ?B/s]
Generating train split:    0%|          | 0/1500 [00:00<?, ? examples/s]
```

```
[ ]: dataset
```

```
[ ]: DatasetDict({
         train: Dataset({
             features: ['directors', 'title', 'imdb', 'plot_embedding', 'awards',
     'rated', 'countries', 'type', 'writers', 'metacritic', 'languages', 'fullplot',
     'genres', 'poster', 'cast', 'runtime', 'plot', 'num_mflix_comments'],
             num_rows: 1500
         })
     })
```

```
[ ]: import pandas as pd
     data=pd.DataFrame(dataset["train"])
```

```
[ ]: data.shape
```

```
[ ]: (1500, 18)
```

```
[ ]: data.head()
```

```
[ ]:                              directors     … num_mflix_comments
     0  [Louis J. Gasnier, Donald MacKenzie]   …                   0
     1       [Alfred J. Goulding, Hal Roach]   …                   0
     2                     [Herbert Brenon]    …                   0
     3                      [Albert Parker]    …                   1
     4                         [Sam Taylor]    …                   0

     [5 rows x 18 columns]
```

```
[ ]: data=data.sample(80)
```

```
[ ]: data.shape
```

```
[ ]: (80, 18)
```

```
[ ]: data.columns
```

```
[ ]: Index(['directors', 'title', 'imdb', 'plot_embedding', 'awards', 'rated',
            'countries', 'type', 'writers', 'metacritic', 'languages', 'fullplot',
            'genres', 'poster', 'cast', 'runtime', 'plot', 'num_mflix_comments'],
           dtype='object')
```

```
[ ]: data.isnull().sum()
```

```
[ ]: directors            0
     title                0
     imdb                 0
     plot_embedding       1
     awards               0
     rated               22
     countries            0
     type                 0
     writers              0
     metacritic          51
     languages            0
     fullplot             4
     genres               0
     poster               3
     cast                 0
     runtime              1
     plot                 1
     num_mflix_comments   0
     dtype: int64
```

```
[ ]: data=data.dropna(subset=["fullplot"])
```

```
[ ]: data=data.drop(columns=["plot_embedding"])
```

```
[ ]: data.isnull().sum()
```

```
[ ]: directors           0
     title               0
     imdb                0
     awards              0
     rated              21
     countries           0
     type                0
     writers             0
     metacritic         48
     languages           0
     fullplot            0
     genres              0
     poster              3
     cast                0
     runtime             1
     plot                0
     num_mflix_comments  0
     dtype: int64
```

3   Embedding of the model

```
[ ]: from sentence_transformers import SentenceTransformer
     embedding_model = SentenceTransformer("thenlper/gte-large")
```

modules.json: 0%| | 0.00/385 [00:00<?, ?B/s]

README.md: 0%| | 0.00/67.9k [00:00<?, ?B/s]

sentence_bert_config.json: 0%| | 0.00/57.0 [00:00<?, ?B/s]

/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132:
FutureWarning: `resume_download` is deprecated and will be removed in version
1.0.0. Downloads always resume when possible. If you want to force a new
download, use `force_download=True`.
  warnings.warn(

config.json: 0%| | 0.00/619 [00:00<?, ?B/s]

model.safetensors: 0%| | 0.00/670M [00:00<?, ?B/s]

tokenizer_config.json: 0%| | 0.00/342 [00:00<?, ?B/s]

vocab.txt: 0%| | 0.00/232k [00:00<?, ?B/s]

tokenizer.json: 0%| | 0.00/712k [00:00<?, ?B/s]

special_tokens_map.json: 0%| | 0.00/125 [00:00<?, ?B/s]

1_Pooling/config.json: 0%| | 0.00/191 [00:00<?, ?B/s]

4 Store the data in monogo DB

```python
from pymongo.mongo_client import MongoClient

uri = "mongodb+srv://snshrivas:Snshrivas@cluster0.u141hkk.mongodb.net/?
  retryWrites=true&w=majority&appName=Cluster0"

# Create a new client and connect to the server
client = MongoClient(uri)

# Send a ping to confirm a successful connection
try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

Pinged your deployment. You successfully connected to MongoDB!

```python
db=client["moviemydb"]
```

```python
collection=db["moviemycollection"]
```

```python
document=data.to_dict("records")
```

```python
collection.insert_many(document)
```

```
InsertManyResult([ObjectId('663ce86a57b4141aa769a43f'),
ObjectId('663ce86a57b4141aa769a440'), ObjectId('663ce86a57b4141aa769a441'),
ObjectId('663ce86a57b4141aa769a442'), ObjectId('663ce86a57b4141aa769a443'),
ObjectId('663ce86a57b4141aa769a444'), ObjectId('663ce86a57b4141aa769a445'),
ObjectId('663ce86a57b4141aa769a446'), ObjectId('663ce86a57b4141aa769a447'),
ObjectId('663ce86a57b4141aa769a448'), ObjectId('663ce86a57b4141aa769a449'),
ObjectId('663ce86a57b4141aa769a44a'), ObjectId('663ce86a57b4141aa769a44b'),
ObjectId('663ce86a57b4141aa769a44c'), ObjectId('663ce86a57b4141aa769a44d'),
ObjectId('663ce86a57b4141aa769a44e'), ObjectId('663ce86a57b4141aa769a44f'),
ObjectId('663ce86a57b4141aa769a450'), ObjectId('663ce86a57b4141aa769a451'),
ObjectId('663ce86a57b4141aa769a452'), ObjectId('663ce86a57b4141aa769a453'),
ObjectId('663ce86a57b4141aa769a454'), ObjectId('663ce86a57b4141aa769a455'),
ObjectId('663ce86a57b4141aa769a456'), ObjectId('663ce86a57b4141aa769a457'),
ObjectId('663ce86a57b4141aa769a458'), ObjectId('663ce86a57b4141aa769a459'),
ObjectId('663ce86a57b4141aa769a45a'), ObjectId('663ce86a57b4141aa769a45b'),
ObjectId('663ce86a57b4141aa769a45c'), ObjectId('663ce86a57b4141aa769a45d'),
ObjectId('663ce86a57b4141aa769a45e'), ObjectId('663ce86a57b4141aa769a45f'),
ObjectId('663ce86a57b4141aa769a460'), ObjectId('663ce86a57b4141aa769a461'),
ObjectId('663ce86a57b4141aa769a462'), ObjectId('663ce86a57b4141aa769a463'),
ObjectId('663ce86a57b4141aa769a464'), ObjectId('663ce86a57b4141aa769a465'),
```

```
ObjectId('663ce86a57b4141aa769a466'),  ObjectId('663ce86a57b4141aa769a467'),
ObjectId('663ce86a57b4141aa769a468'),  ObjectId('663ce86a57b4141aa769a469'),
ObjectId('663ce86a57b4141aa769a46a'),  ObjectId('663ce86a57b4141aa769a46b'),
ObjectId('663ce86a57b4141aa769a46c'),  ObjectId('663ce86a57b4141aa769a46d'),
ObjectId('663ce86a57b4141aa769a46e'),  ObjectId('663ce86a57b4141aa769a46f'),
ObjectId('663ce86a57b4141aa769a470'),  ObjectId('663ce86a57b4141aa769a471'),
ObjectId('663ce86a57b4141aa769a472'),  ObjectId('663ce86a57b4141aa769a473'),
ObjectId('663ce86a57b4141aa769a474'),  ObjectId('663ce86a57b4141aa769a475'),
ObjectId('663ce86a57b4141aa769a476'),  ObjectId('663ce86a57b4141aa769a477'),
ObjectId('663ce86a57b4141aa769a478'),  ObjectId('663ce86a57b4141aa769a479'),
ObjectId('663ce86a57b4141aa769a47a'),  ObjectId('663ce86a57b4141aa769a47b'),
ObjectId('663ce86a57b4141aa769a47c'),  ObjectId('663ce86a57b4141aa769a47d'),
ObjectId('663ce86a57b4141aa769a47e'),  ObjectId('663ce86a57b4141aa769a47f'),
ObjectId('663ce86a57b4141aa769a480'),  ObjectId('663ce86a57b4141aa769a481'),
ObjectId('663ce86a57b4141aa769a482'),  ObjectId('663ce86a57b4141aa769a483'),
ObjectId('663ce86a57b4141aa769a484'),  ObjectId('663ce86a57b4141aa769a485'),
ObjectId('663ce86a57b4141aa769a486'),  ObjectId('663ce86a57b4141aa769a487'),
ObjectId('663ce86a57b4141aa769a488'),  ObjectId('663ce86a57b4141aa769a489'),
ObjectId('663ce86a57b4141aa769a48a')], acknowledged=True)
```

5 Store in the data in Pinecone

```python
from pinecone import Pinecone
PINECONE_API_KEY="cef794a5-a590-4234-99ff-e14b50257f93"
pc = Pinecone(api_key=PINECONE_API_KEY)
index = pc.Index("mongomovie")
```

```python
def get_result(query,similar_result=3):
    embedding=embedding_model.encode(query)
    embedding=embedding.tolist()

    result=index.query(
      vector=embedding,
      top_k=similar_result,
    )
    return result
```

```python
query="what is the best horror movie to watch and why?"
```

```python
result=get_result(query)
```

```python
result
```

```python
{'matches': [{'id': '663ce86a57b4141aa769a475',
              'score': 0.792062521,
              'values': []},
```

```
              {'id': '663ce86a57b4141aa769a44b',
               'score': 0.77806145,
               'values': []},
              {'id': '663ce86a57b4141aa769a482',
               'score': 0.767377675,
               'values': []}],
    'namespace': '',
    'usage': {'read_units': 5}}
```

```python
from bson.objectid import ObjectId
```

```python
mylist=[]
for i in   range(len(result["matches"])):
  value=result["matches"][i]['id']
  mylist.append(collection.find_one({"_id": ObjectId(value)}))
```

```python
mylist
```

```
[{'_id': ObjectId('663ce86a57b4141aa769a475'),
     'directors': ['Stephen Sommers'],
     'title': 'Van Helsing',
     'imdb': {'id': 338526, 'rating': 6.0, 'votes': 177269},
     'awards': {'nominations': 13, 'text': '3 wins & 13 nominations.', 'wins': 3},
     'rated': 'PG-13',
     'countries': ['USA', 'Czech Republic'],
     'type': 'movie',
     'writers': ['Stephen Sommers'],
     'metacritic': 35.0,
     'languages': ['English', 'Latin'],
     'fullplot': "Van Helsing is in the world to rid all evil, even if not everyone
agrees with him. The Vatican sends the monster hunter and his ally, Carl, to
Transylvania. They have been sent to this land to stop the powerful Count
Dracula. Whilst there they join forces with a Gypsy Princess called Anna
Valerious, who is determined to end an ancient curse on her family by destroying
the vampire. They just don't know how!",
     'genres': ['Action', 'Adventure', 'Fantasy'],
     'poster': 'https://m.media-amazon.com/images/M/MV5BODRmY2NhNDItOWViNi00OTIyLTk
3YjYtYzY0YTFlMDg1YzQ0L2ltYWdlL2ltYWdlXkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SY1000_SX6
77_AL_.jpg',
     'cast': ['Hugh Jackman',
     'Kate Beckinsale',
     'Richard Roxburgh',
     'David Wenham'],
     'runtime': 131.0,
     'plot': "The notorious monster hunter is sent to Transylvania to stop Count
Dracula who is using Dr. Frankenstein's research and a werewolf for some
sinister purpose.",
```

'num_mflix_comments': 0},
{'_id': ObjectId('663ce86a57b4141aa769a44b'),
  'directors': ['Avi Nesher'],
  'title': 'Timebomb',
  'imdb': {'id': 105597, 'rating': 5.5, 'votes': 1040},
  'awards': {'nominations': 0, 'text': '1 win.', 'wins': 1},
  'rated': 'R',
  'countries': ['USA'],
  'type': 'movie',
  'writers': ['Avi Nesher'],
  'metacritic': nan,
  'languages': ['English', 'Hungarian'],
  'fullplot': "When someone tries to murder watchmaker Eddy Kay, the incident triggers a barrage of nightmares and flashbacks into a past that isn't his own. Fearing for his sanity, Eddy contacts psychiatrist Dr. Anna Nolmar for help. Anna thinks he's hallucinating until another attack proves the dangers are all too real. The two of them go on the run, trying to discover the truth about Eddie's past and true identity before it kills them.",
  'genres': ['Action', 'Drama', 'Sci-Fi'],
  'poster': 'https://m.media-amazon.com/images/M/MV5BODIzMDQzMjQ3NF5BMl5BanBnXkFtZTcwMzExODM5NA@@._V1_SY1000_SX677_AL_.jpg',
  'cast': ['Michael Biehn', 'Patsy Kensit', 'Tracy Scoggins', 'Robert Culp'],
  'runtime': 96.0,
  'plot': "When someone tries to murder watchmaker Eddy Kay, the incident triggers a barrage of nightmares and flashbacks into a past that isn't his own. Fearing for his sanity, Eddy contacts ...",
  'num_mflix_comments': 1},
{'_id': ObjectId('663ce86a57b4141aa769a482'),
  'directors': ['Paul W.S. Anderson'],
  'title': 'Resident Evil',
  'imdb': {'id': 120804, 'rating': 6.7, 'votes': 185410},
  'awards': {'nominations': 4, 'text': '4 nominations.', 'wins': 0},
  'rated': 'R',
  'countries': ['UK', 'Germany', 'France', 'USA'],
  'type': 'movie',
  'writers': ['Paul W.S. Anderson'],
  'metacritic': 33.0,
  'languages': ['English'],
  'fullplot': 'A virus has escaped in a secret facility called "The Hive," turning the staff into hungry zombies and releasing the mutated Lab "Animals" that they were studying. The complex computer shuts down the base to prevent infection. The parent corporation sends in an elite military unit, where they meet Alice, who is suffering from amnesia due to exposure to nerve gas. The military team must shut down the computer and get out, fighting their way past zombies, mutants, and the computer itself, before the virus escapes and infects the rest of the world. Alice must also come to terms with her slowly-returning memories.',

    'genres': ['Action', 'Fantasy', 'Horror'],
    'poster': 'https://m.media-amazon.com/images/M/MV5BN2Y2MTljNjMtMDRlNi00ZWNhLTh
mMWltYTlmZjYyZDk4NzYxXkEyXkFqcGdeQXVyNjQ2MjQ5NzM@._V1_SY1000_SX677_AL_.jpg',
    'cast': ['Ryan McCluskey', 'Oscar Pearce', 'Indra Ovè', 'Anna Bolt'],
    'runtime': 100.0,
    'plot': 'A special military unit fights a powerful, out-of-control
supercomputer and hundreds of scientists who have mutated into flesh-eating
creatures after a laboratory accident.',
    'num_mflix_comments': 1}]

```python
combined_information = ""
for i in range(len(mylist)):
    fullplot=mylist[i]["fullplot"]
    title=mylist[i]["title"]
    combined_information += f"Title:{title}, fullplot: {fullplot}\n"
```

```python
print(combined_information)
```

Title:Van Helsing, fullplot: Van Helsing is in the world to rid all evil, even if not everyone agrees with him. The Vatican sends the monster hunter and his ally, Carl, to Transylvania. They have been sent to this land to stop the powerful Count Dracula. Whilst there they join forces with a Gypsy Princess called Anna Valerious, who is determined to end an ancient curse on her family by destroying the vampire. They just don't know how!
Title:Timebomb, fullplot: When someone tries to murder watchmaker Eddy Kay, the incident triggers a barrage of nightmares and flashbacks into a past that isn't his own. Fearing for his sanity, Eddy contacts psychiatrist Dr. Anna Nolmar for help. Anna thinks he's hallucinating until another attack proves the dangers are all too real. The two of them go on the run, trying to discover the truth about Eddie's past and true identity before it kills them.
Title:Resident Evil, fullplot: A virus has escaped in a secret facility called "The Hive," turning the staff into hungry zombies and releasing the mutated Lab "Animals" that they were studying. The complex computer shuts down the base to prevent infection. The parent corporation sends in an elite military unit, where they meet Alice, who is suffering from amnesia due to exposure to nerve gas. The military team must shut down the computer and get out, fighting their way past zombies, mutants, and the computer itself, before the virus escapes and infects the rest of the world. Alice must also come to terms with her slowly-returning memories.

```python
query
```

```python
'what is the best horror movie to watch and why?'
```

```python
prompt = f"Query: {query}\nContinue to answer the query by using the fullplot
ₛonly:\n{combined_information}."
```

```
print(prompt)
```

Query: what is the best horror movie to watch and why?
Continue to answer the query by using the fullplot only:
Title:Van Helsing, **fullplot**: Van Helsing **is** in the world to rid **all evil,** even
if not everyone agrees with him. The Vatican sends the monster hunter and his
ally, Carl, to Transylvania. They have been sent to this land to stop the
powerful Count Dracula. Whilst there they join forces with a Gypsy Princess
called Anna Valerious, who **is** determined to end an ancient curse on her family
by destroying the vampire. They just don't know how!
Title:Timebomb, **fullplot**: When someone **tries** to murder watchmaker Eddy Kay, the
incident triggers a barrage of nightmares and flashbacks into a past that **isn't**
his own. Fearing for his sanity, Eddy contacts psychiatrist Dr. Anna Nolmar for
help. Anna thinks he's hallucinating **until** another attack proves the dangers are
all too **real.** The two of them go on the run, trying to discover the truth about
Eddie's past and true identity before **it kills** them.
Title:Resident Evil, fullplot: A virus has escaped in a secret facility called
"The Hive," turning the staff into hungry zombies and releasing the mutated Lab
"Animals" that they were studying. The complex computer shuts down the base to
prevent infection. The parent corporation sends in an elite military unit, where
they meet Alice, who **is** suffering from amnesia due to exposure to nerve gas. The
military team must shut down the computer and get out, fighting their way past
zombies, mutants, and the computer itself, before the virus escapes and infects
the rest of the world. Alice must also come to terms with her slowly-returning
memories.

```
import os
from google.colab import userdata
GOOGLE_API_KEY=userdata.get('GOOGLE_API_KEY')
os.environ["GOOGLE_API_KEY"] = GOOGLE_API_KEY
```

```python
from langchain_google_genai import ChatGoogleGenerativeAI
def load_model(model_name):
    if model_name=="gemini-pro":
        llm = ChatGoogleGenerativeAI(model="gemini-pro")
    else:
        llm=ChatGoogleGenerativeAI(model="gemini-pro-vision")

    return llm
```

```python
model_text=load_model("gemini-pro")
```

```python
model_text.invoke(prompt).content
```

'The best horror movie to watch depends on individual preferences. "Van Helsing"
is a classic monster movie with a thrilling plot and visually stunning effects.
```

"Timebomb" offers a unique and suspenseful experience with its twisty plot and psychological elements. "Resident Evil" is a popular horror franchise that combines action, horror, and science fiction elements. Ultimately, the best horror movie to watch is the one that resonates most with your personal interests and scares.'

[ ]: