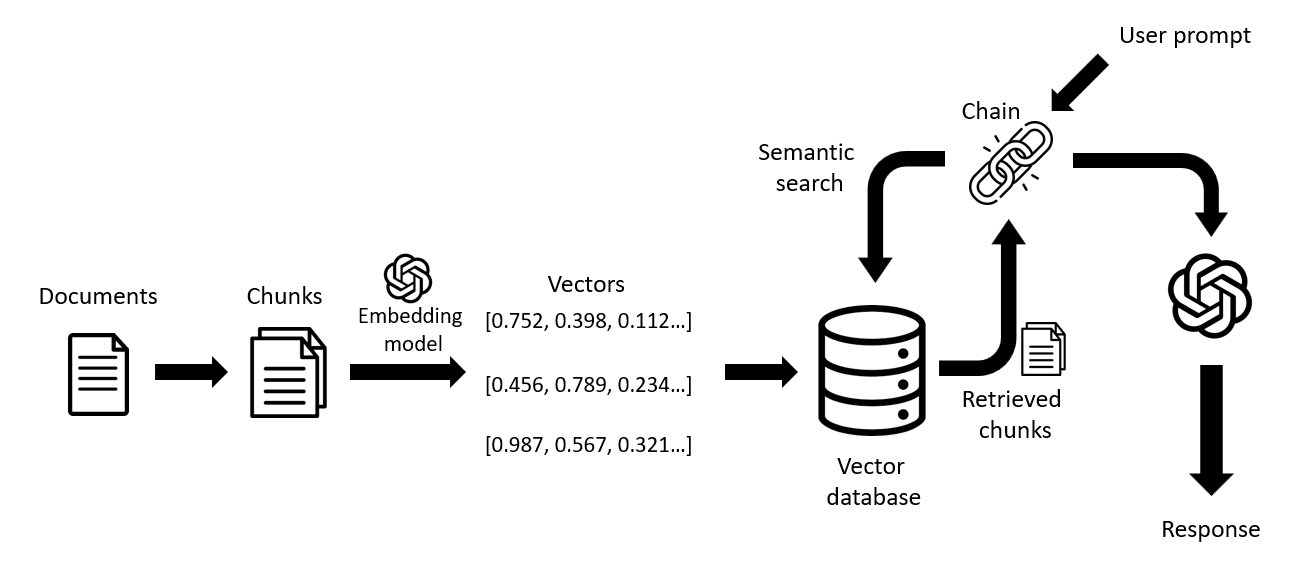
Movie Recommendation System Using Embeddings and Language Models



#### **1. Introduction**

Recommender systems are essential tools for enhancing user experiences across industries like entertainment, e-commerce, and education. This project builds a personalized movie recommendation system by combining text embeddings, vector search, and large language models (LLMs). The system retrieves relevant movie details based on user queries and generates human-like responses tailored to their preferences.

#### **2. Objectives**

The primary objectives of this project are:

* To preprocess and embed movie metadata into high-dimensional vectors for efficient retrieval.
* To implement a vector-based search using Pinecone for similarity matching.
* To leverage Groq’s Llama language model for natural language-based recommendations.
* To design a robust query-answering pipeline (QA Chain) that combines embedding search and LLM capabilities.

#### **3. Dataset Description**

* **Dataset Source**: IMDB dataset provided as a .csv file.
* **Columns Used**:
  + movie\_name: The title of the movie.
  + description: A short synopsis of the movie.
  + genre: The genre(s) associated with the movie.
  + certificate: Age certification for the movie.
* **Processed Column**: combined\_info — a concatenated representation of the relevant fields.
* **Preprocessing**:
  + Tokenized text data using tiktoken.
  + Filtered out entries exceeding the model's token limit (8000 tokens).

**4. Methodology**

##### **4.1 Preprocessing and Embedding**

1. **Text Tokenization**:
   * Tokenized the combined\_info field using the tiktoken library to ensure compatibility with the LLM.
   * Removed entries with token counts exceeding the defined maximum limit.
2. **Vector Embeddings**:
   * Converted preprocessed text into embeddings for numerical representation.
   * Stored these embeddings in a vector database (Pinecone) for fast similarity-based searches.

##### **4.2 Vector Search**

* Utilized Pinecone's vector store for efficient retrieval of similar movie embeddings based on user queries.
* Embedded user queries into the same vector space as the movie metadata for compatibility.

##### **4.3 Language Model Integration**

* Set up the **Llama 3-8b-8192** model via the Groq API.

Used a structured prompt template to contextualize the query and retrieved data:  
You are a movie recommender system that helps users find movies that match their preferences.

Use the following pieces of context to answer the question at the end.

Context:

{context}

Question:

{query}

Answer:

##### **4.4 Question-Answering Pipeline (QA Chain)**

* Combined vector search and LLM into a single pipeline:
  + **Retriever**: Extracted the most relevant movie metadata from Pinecone.
  + **LLM**: Used the retrieved context and user query to generate recommendations.

#### 

#### **5. Implementation**

The project was implemented in Python using the following tools:

* **Data Handling**: pandas for data preprocessing.
* **Tokenization**: tiktoken for efficient text tokenization.
* **Embedding Storage**: Pinecone as the vector database.
* **LLM Integration**: langchain and ChatGroq for interfacing with the Groq-based Llama model.

The notebook workflow included:

1. **Dataset loading**: IMDB dataset read into a DataFrame and preprocessed.
2. **Embedding generation**: Text converted to embeddings and stored.
3. **Query handling**: User queries processed to retrieve relevant movies and generate recommendations.

#### **6. Results**

The system successfully generates movie recommendations by combining:

* **Vector Search**: Retrieves contextually relevant movie information.
* **Language Model Responses**: Generates coherent and personalized recommendations in natural language.

**Example**:  
**Query**: Recommend a comedy movie with family-friendly themes.  
**Response**:

🎥 Movie Recommendations:

Based on your request for comedy movies with family-friendly themes, I'd like to recommend the following three movies:

Housefull 2

This Bollywood comedy is a hilarious tale of four men trying to outsmart their prospective father-in-laws. With a star-studded cast, including Akshay Kumar and John Abraham, this movie is full of action, comedy, and drama. The plot is light-hearted and family-friendly, making it a great choice for a fun night in with the family.

My Big Fat Greek Wedding

This romantic comedy tells the story of a Greek woman who falls in love with a non-Greek man and the cultural clashes that ensue. The movie is a heartwarming and hilarious exploration of family values and traditions. With a relatable plot and lovable characters, this movie is a great choice for families to enjoy together.

The Princess Diaries

In this beloved comedy, a high school student discovers she's the princess of a small European country and must learn to navigate royal protocol. With a talented young cast, including Anne Hathaway and Julie Andrews, this movie is a fun and lighthearted take on royal family life. The movie's themes of self-discovery and family are perfect for a family movie night.

I hope you enjoy these recommendations!

#### **7. Challenges and Solutions**

1. **Token Limit Exceedance**:
   * Issue: Some movie descriptions exceeded the maximum token limit.
   * Solution: Filtered such entries during preprocessing.
2. **Embedding Quality**:
   * Ensured consistency between query embeddings and movie metadata embeddings by using the same embedding model.
3. **Response Quality**:
   * Improved natural language responses by refining the prompt template.

#### **8. Future Work**

* **Enhancements**:
  + Integrate user feedback to improve recommendation accuracy.
  + Add sentiment analysis to consider user mood in recommendations.
* **Scalability**:
  + Expand the system to handle multilingual queries and datasets.
  + Support real-time updates in the vector database for new movies.
* **Explainability**:
  + Include justifications for recommendations to improve transparency.

#### **9. Conclusion**

The movie recommendation system combines state-of-the-art embedding techniques and LLMs to deliver accurate, personalized movie recommendations. The integration of Pinecone and Groq’s Llama model ensures both speed and coherence. This approach demonstrates the potential of combining vector search with advanced NLP capabilities to create intelligent, scalable systems.