Project Report: IMDB Semantic Movie Recommender

Author: Vishal Mohan Puri　　Date: 12 April 2025

Technology Stack: Streamlit, Pinecone, Hugging Face Transformers, Python

Dataset: IMDB Top 1000 Movies (Kaggle)

🎯 Objective

The goal of this project was to build an intelligent movie recommendation system using semantic search. Instead of relying on traditional keyword-based or genre-based search, this system allows users to describe what kind of movie they want to watch in natural language and returns the most semantically relevant matches from a dataset of the top 1000 IMDB movies.

? Components

1. Dataset

- Source: Kaggle - IMDB Top 1000 Movies
- Fields Used: Series_Title, Genre, Overview
- Preprocessing: Combined title and overview, filled missing values

2. Text Embeddings

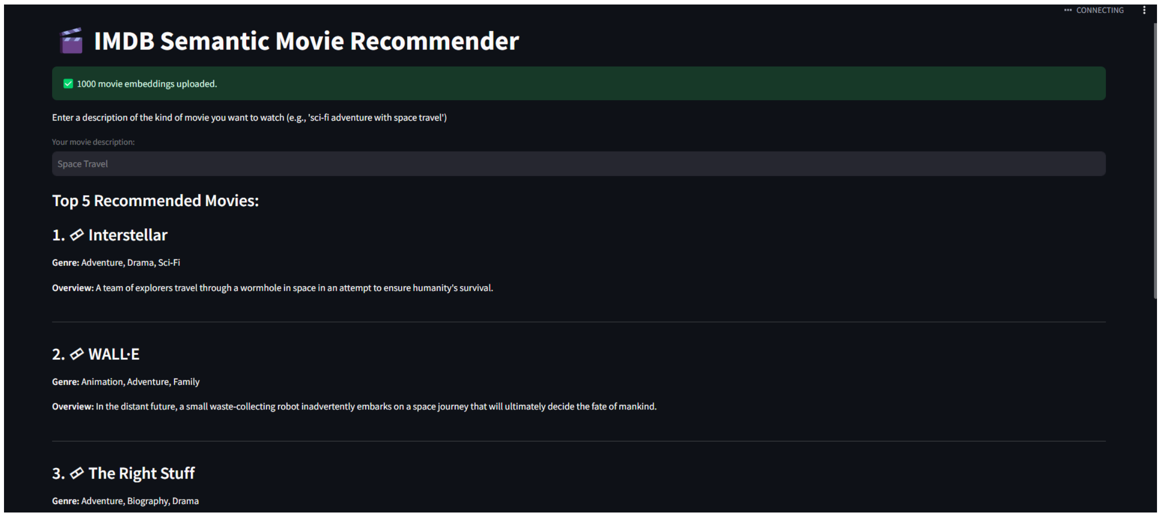
- Model: all-MiniLM-L6-v2 from Hugging Face
- Vector Dimension: 384
- Purpose: Converts descriptions into semantic vectors

3. Vector Database

- Tool: Pinecone
- Index Name: imdb
- Configuration: 384 dimensions, cosine similarity, serverless
- Metadata includes: title, genre, overview

4. Streamlit UI

- Input: Natural language text box
- Output: Top 5 similar movies with title, genre, and overview



🎬 IMDB Semantic Movie Recommender

✅ 1000 movie embeddings uploaded.

Enter a description of the kind of movie you want to watch (e.g., 'sci-fi adventure with space travel')

Your movie description:

Space Travel

Top 5 Recommended Movies:

1. 🎟 Interstellar

Genre: Adventure, Drama, Sci-Fi

Overview: A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.

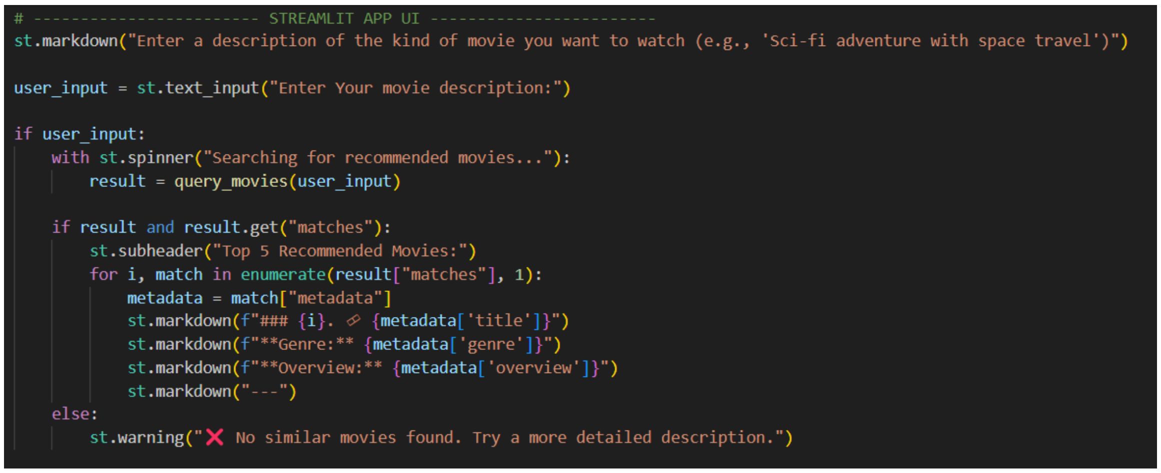
2. 🎟 WALL·E

Genre: Animation, Adventure, Family

Overview: In the distant future, a small waste-collecting robot inadvertently embarks on a space journey that will ultimately decide the fate of mankind.

3. 🎟 The Right Stuff

Genre: Adventure, Biography, Drama



```python
# ------------------------ STREAMLIT APP UI ------------------------
st.markdown("Enter a description of the kind of movie you want to watch (e.g., 'Sci-fi adventure with space travel')")

user_input = st.text_input("Enter Your movie description:")

if user_input:
    with st.spinner("Searching for recommended movies..."):
        result = query_movies(user_input)

    if result and result.get("matches"):
        st.subheader("Top 5 Recommended Movies:")
        for i, match in enumerate(result["matches"], 1):
            metadata = match["metadata"]
            st.markdown(f"### {i}. 🎬 {metadata['title']}")
            st.markdown(f"**Genre:** {metadata['genre']}")
            st.markdown(f"**Overview:** {metadata['overview']}")
            st.markdown("---")
    else:
        st.warning("❌ No similar movies found. Try a more detailed description.")
```

🔁 Workflow

1. Load and preprocess the dataset
2. Generate sentence embeddings
3. Initialize Pinecone index
4. Upsert vectors into Pinecone
5. Query using cosine similarity
6. Display results in Streamlit

✅ Features Implemented

- Semantic search using embeddings
- Vector search with Pinecone
- Streamlit UI
- Upsert safeguard
- Performance caching

📌 Summary

This project demonstrates the power of combining NLP with vector search to build intelligent recommender systems. Users can now describe what they want to watch, and the app semantically understands their intent to recommend relevant movies.