AI-powered Recommendation System using OpenAI and Vector Stores

By:

AANCHAL CHOUDHARY ANSHIKA SINHA

Department of Information Technology Vidyalankar Institute of Technology University of Mumbai

(October 2025)

Table of Contents

1. Introduction

1.1 Background

In the current digital landscape, Over-The-Top (OTT) and movie platforms are a primary source of entertainment. The effectiveness of these platforms heavily relies on their ability to suggest relevant content to keep users engaged. Traditionally, many recommendation systems rely on collaborative filtering or content-based filtering, which often depend on large volumes of user data (like ratings) or simple metadata matching (like genres).

This project proposes a more modern approach, leveraging the power of large language models (LLMs) and vector databases to understand the *semantic* meaning of content and user preferences, leading to more intelligent and personalized recommendations.

1.2 Problem Statement

The project aims to address several key limitations of existing recommendation systems:

- **Limited Data Dependency:** Platforms often recommend content based on limited or superficial user data.
- **Generic Suggestions:** Users frequently receive generic or popular-based suggestions rather than truly personalized recommendations.
- **Lack of Semantic Understanding:** Existing systems often lack a deep semantic understanding of user preferences and content nuances.
- **Need for Smarter Pipelines:** There is a clear need for a smarter, AI-driven recommendation pipeline to enhance user experience.

1.3 Objectives

The primary objectives of this project are:

- To build an intelligent and automated movie recommendation workflow.
- To utilize OpenAI embeddings for deep semantic similarity matching between content and user queries.
- To integrate the recommendation system within a real-time AI chat interface for interactive suggestions.
- To automate the entire process using n8n (a workflow automation tool) and vector databases (like Qdrant).

1.4 Scope

The scope of this project is to develop a complete, end-to-end movie recommendation workflow. While the current implementation, "Fright Finder," focuses on horror movies, the underlying architecture is designed to be scalable and adaptable. The same principles and pipeline can be extended to other domains, such as music, e-commerce products, or articles.

2. Proposed Methodology

2.1 System Overview

The proposed solution is an automated pipeline that combines the capabilities of OpenAI for intelligence, Qdrant as a vector store for efficient search, and n8n for workflow automation.

The core idea is to process movie data (like descriptions, plots, and genres) into high-dimensional vector embeddings using OpenAI. These embeddings capture the semantic essence of the movies. These vectors are then stored in a Qdrant vector database, which is optimized for high-speed similarity search.

When a user requests a recommendation (e.g., "find a movie about a haunted house"), their query is also converted into an embedding. The system then searches the Qdrant database for movie vectors that are "closest" in the vector space to the user's query vector, resulting in highly relevant, semantic-based recommendations.

2.2 Core Technologies

- **OpenAI:** Used to generate vector embeddings for movie data and user queries. This allows the system to understand language nuances and context.
- **Qdrant Vector Store:** A high-performance vector database used to store the movie embeddings and perform efficient similarity searches.
- **n8n:** A workflow automation tool used to connect all the components and automate the entire data processing and recommendation pipeline.
- **AI Chat Interface:** A real-time chat UI provides the front-end for users to interact with the system and receive recommendations.

2.3 Workflow Automation

The n8n workflow visual from the presentation shows the automated process:

1. **Data Ingestion:** The workflow is triggered, pulling movie data from a source (e.g., a GitHub file).
2. **Embedding and Storage (Upsert):** The movie data is processed, split into manageable chunks (if necessary), and then converted into vector embeddings via the OpenAI API. These embeddings are then "upserted" (updated/inserted) into the Qdrant Vector Store.
3. **User Interaction:** A separate part of the workflow is triggered by a user message in the AI chat.
4. **Semantic Search:** The user's query is sent to an OpenAI Chat Model. The user's request for a recommendation is embedded, and this embedding is used to query the Qdrant database.
5. **Recommendation Delivery:** Qdrant returns the most similar items. The workflow then formats this data and presents the personalized movie suggestions back to the user through the AI agent in the chat.

3. Implementation and Results

3.1 Web Interface

The project was realized as a web application called "Fright Finder," a curated platform for horror movie recommendations. The screenshots demonstrate a professional, dark-themed UI that showcases movie posters, titles, and user ratings.

The interface includes:

- A "Recommended" tag on specific movies, indicating the system's personalized suggestions.
- A search bar for users to find horror movies.
- Filters to narrow down choices.
- A "Setup Backend Integration" button, implying a connection to the n8n workflow.

3.2 Key Features

The final implementation successfully delivered on its objectives, providing the following key features:

- **End-to-end Workflow:** A fully functional and automated movie recommendation pipeline.
- **Advanced Integration:** Seamless integration of OpenAI embeddings with the Qdrant vector store for semantic search.
- **Interactive AI:** A real-time chat interaction system powered by an AI Agent.

- **Personalized Content:** The ability to deliver personalized movie recommendations complete with ratings.

4. Conclusion and Future Scope

4.1 Conclusion

This project successfully built a fully automated, AI-based movie recommender system. By seamlessly combining the strengths of OpenAI, n8n, and vector databases, the system overcomes the limitations of traditional recommenders. The result is a system that achieves personalized, semantic-level recommendations, moving beyond simple keyword matching to understand the true intent and context of user preferences.

4.2 Future Scope

The architecture and pipeline developed for this project are highly flexible. The system is "future-ready" and can be easily adapted to multiple other domains beyond movies. Potential future applications include recommendation engines for music, e-commerce products, news articles, or any other domain where semantic understanding of content is beneficial.

5. References

- OpenAI API Documentation
- n8n Workflow Automation
- Qdrant Vector Database