**Movie and tv show chatbot**

**SCOPE**

The goal of the project is to develop a chatbot specializing in movies and TV shows, providing users with detailed movie information using the TMDb API. This chatbot is an interactive assistant capable of answering user queries and recommending movies or TV shows. The business impact has to do with its potential to enhance user engagement relating to entertainment, reducing manual searches for movie details.

Currently, similar solutions exist as part of large language models like ChatGPT. But this implementation is focused on movie and tv show queries. Without machine learning, the task would involve manually searching for movies on platforms like IMDB or TMDb. The chatbot automates this by getting data in real time.

Performance will be measured by the chatbot’s ability to correctly retrieve movie data via API calls, handle unsupported queries, and provide fallback solutions. Stakeholders include students, developers, and people who are interested in entertainment.

Milestones include:

* TMDb API integration
* Functional chatbot deployment
* User testing and refinement

Resources include a TMDb API key, Python, and libraries such as `gradio` and `tmdbv3api`. Personnel: 1 developer.

**METRICS**

Success will be measured by:

- Correct retrieval of movie details in over 80% of test cases.

- Fallback responses for unsupported or complex queries.

- Real time response latency below 3 seconds for API queries.

Software metrics include API response accuracy, system uptime, and error handling.

**DATA**

The chatbot fetches data via the TMDb API, which provides structured information about movies, TV shows, and actors. Data is structured as JSON responses. Which ensures consistent formatting. No additional data collection will be needed.

Ethical considerations include respecting TMDb’s API usage policies. Privacy is not a concern since no user data is stored or shared.

Data cleaning is minimal, since TMDb ensures high-quality structured data. Query preprocessing (for example, removing unnecessary phrases like “Tell me about”) improves API compatibility.

**MODELING**

The solution does not involve traditional machine learning but integrates API-based automation to get movie and tv details. The base approach is the direct querying of TMDb. It is a success if it retrieves valid results for specific titles. Improvements focus on on robust query cleaning to improve match rates.

Mistake handling includes debugging attribute errors and expanding query preprocessing for greater API compatibility.

**DEPLOYMENT**

The chatbot is deployed locally using Gradio, allowing users to access the interface via a web browser. Movie and tv details are returned in real time. Post deployment, the system requires occasional monitoring for API updates or quota issues. Improvements could include integrating more APIs or local storage for faster responses.

**REFERENCES**

- TMDb API documentation

- Python libraries: `tmdbv3api`, `gradio`

- Gradio library documentation

- OpenAI’s ChatGPT platform for fallback queries