



Data Mining Project : Movie Recommendation System

By :

MEZAHY Mohamed
OULDRHILA Oussama
LAKHIRI Rim

Supervised by :

Pr. EL ASRI Ikram

Table des matières

1	Introduction	2
2	Data Collection	2
3	Problem Statement	2
4	Methodology	3
	4.1 Data Pre-processing	3
	4.2 Recommendation Techniques	3
5	System Architecture	3
6	Evaluation and Results	3
7	Improvements	4
8	Conclusion	4

1 Introduction

The aim of this project is to build a personalized movie recommendation system using machine learning and data mining techniques. The system collects user preferences including favorite genres, actors, directors, and recently watched movies to generate accurate and dynamic movie suggestions. It utilizes a MySQL database populated with real-world data fetched from The Movie Database (TMDB) API. The recommendation engine is deployed via a Flask web application.

2 Data Collection

The dataset is sourced from the TMDB API, which provides rich movie metadata. A Python script (**populate_db.py**) was developed to fetch :

- Movie title
- Genre(s)
- Director
- Cast
- Overview

The data is stored in a MySQL database with the following schema :

- ID (custom auto-incremented)
- Title
- Genre
- Director
- Actors
- Overview

A data cleaning process ensures consistency by handling missing values and normalizing text fields.

3 Problem Statement

Users are often overwhelmed by the vast array of content available. Our goal is to implement a content based recommendation model that :

- Suggests similar movies based on watched ones.
- Learns user preferences

With this approach, the user can easily find a movie that suits them and reduce the time spent looking for a movie to watch.

4 Methodology

4.1 Data Pre-processing

- Text normalization (lowercasing, punctuation removal).
- Creation of a **combined_features** column using genre, director, actors, and overview.
- TF-IDF vectorization to transform textual data.
- Cosine similarity to find related movies.

4.2 Recommendation Techniques

A. Content-Based Filtering :

Using the TF-IDF matrix and cosine similarity : For a given movie title, recommend the most similar movies based on combined features.

B. Preference-Based Filtering : User's favorite genres, actors, and directors.

C. Machine Learning Enhancement (Future Work) : Planned extension to a collaborative filtering model using user watch history and implicit feedback through user interactions (likes, views, etc.)

5 System Architecture

The application is built with the following components :

- **Frontend :** HTML/CSS/JS with movie search and recommendation UI
- **Backend :** Flask REST API
- **Database :** MySQL storing movie metadata
- **Future Deployment :** Dockerized backend with CORS (Cross Origin Resource Sharing) support
- **Future CI/CD Pipeline :** GitHub Actions or GitLab CI for auto-deployments, with staging and production environments

6 Evaluation and Results

- We validated the functions using realistic scenarios for each feature.
- We Tested the quality evaluation by testing known movies and their suggestions but also by adding a score for each recommendation.

7 Improvements

- One of the main improvements is to push the ML aspect even further by adding a model that learns the preferences of the user over time.
- Adding user login and history tracking.
- Incorporate posters and rating for each recommendation.
- Incorporating a full CI/CD pipeline.

8 Conclusion

This project demonstrates a practical implementation of a personalized recommendation system using data mining techniques. The recommendation model provides a strong baseline and is scalable for future extensions. With the addition of machine learning and user feedback loops, this system has a none negligible potential to be a marketable solution for various usages.

NOTE :

- No image was provided to this report since a video explaining and showcasing the project will be provided.
- The code repo can be accessed trthough this link : <https://github.com/mezlin/Movie-Recommendation-App.git>