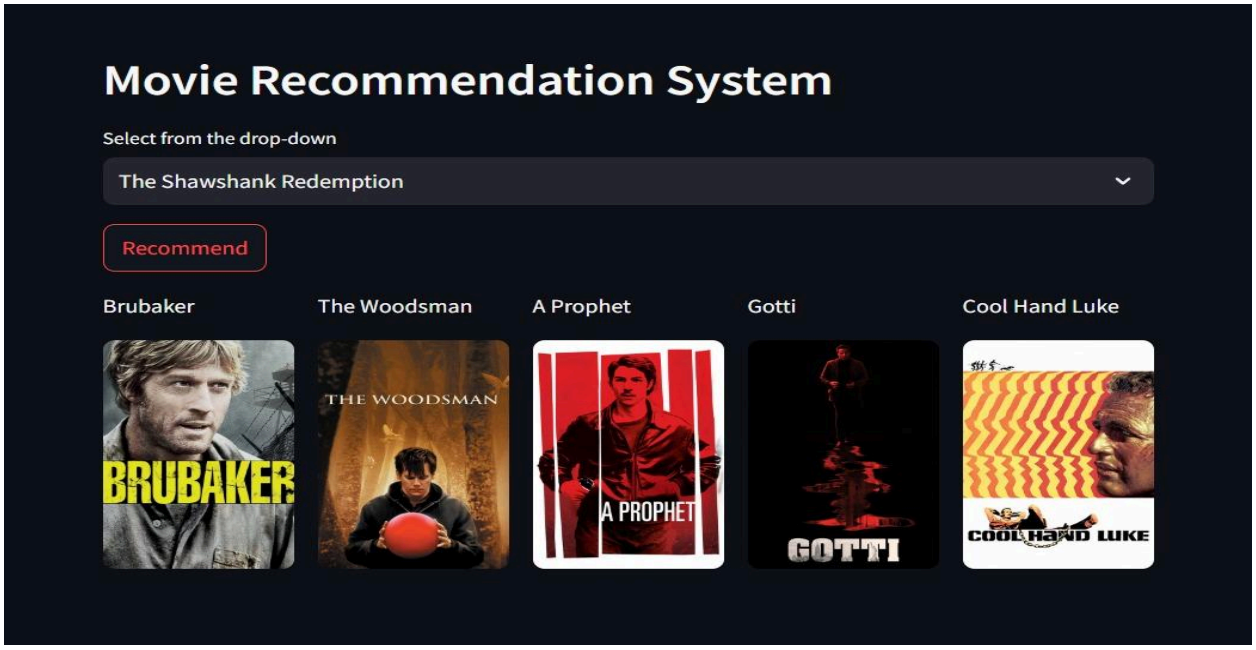
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COGNITIVE COMPUTING
UCS420

PERSONALISED MOVIE RECOMMENDATIONS



SUBMITTED TO - SIR SUKHPAL SINGH
SUBMITTED BY - TEAM IMDB
HARNOOR KAUR(102317138)
RIDHI GUPTA(102317138

1. Introduction & Problem Statement

Movie recommendation systems have become an essential part of online streaming platforms, helping users find content that matches their interests. The goal of this project is to build a movie recommendation system using machine learning techniques, based on similarity scores between movies. By leveraging a dataset containing movie metadata, we aim to provide personalized recommendations to users.

2. Dataset Overview

The dataset used in this project is sourced from TMDb (The Movie Database) and consists of the following attributes:

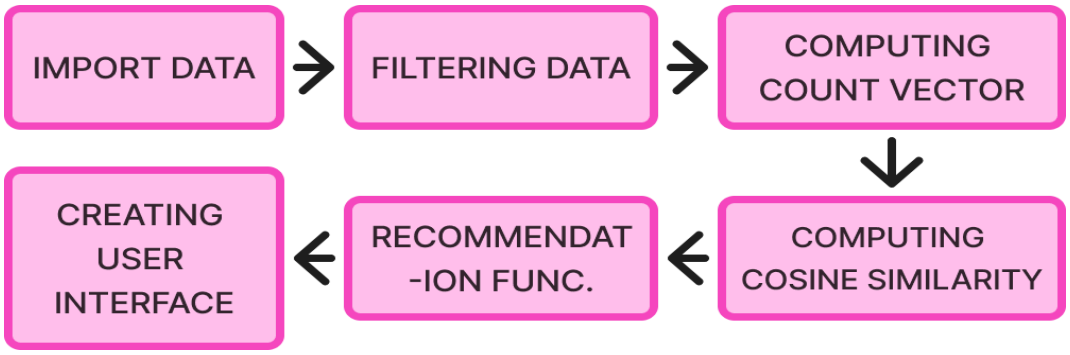
- Movie Title: The name of the movie
- Genres: Categories assigned to the movie
- Popularity: A numerical score indicating the movie's popularity
- Original Language: The primary language of the movie
- Movie ID: A unique identifier used for fetching posters from TMDb
- Other Metadata: Includes cast, crew, and overview details

The dataset contains 10,000 movies, making it suitable for building a robust recommendation system.

3. Technology Stack

- Programming Language: Python
- Libraries:
    - Pandas: For data manipulation and analysis
    - NumPy: For numerical operations
    - Scikit-learn: For similarity calculations
    - Matplotlib & Seaborn: For data visualization
    - Streamlit: For building the web-based user interface
    - Requests: For fetching movie posters using the TMDb API
- Framework: Streamlit for deployment

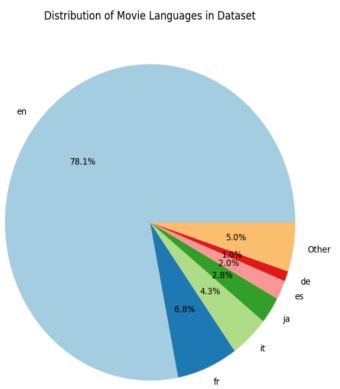
4. High-Level View of process

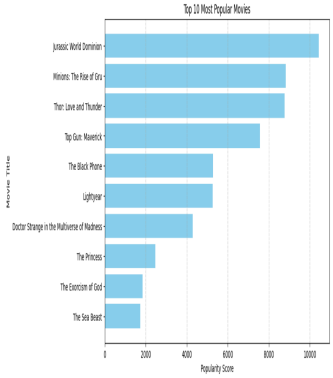


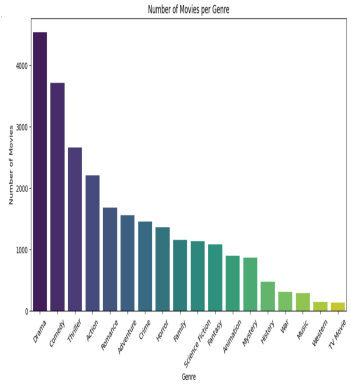
5. ML Model Implementation & Evaluation

- Text Vectorization: Movie descriptions are transformed using TF-IDF or CountVectorizer.
- Similarity Calculation: Cosine similarity is used to measure the closeness between movies.
- Recommendation Function: Based on the selected movie, the system returns the top 5 most similar movies.
- Poster Retrieval: The system fetches corresponding movie posters using the TMDb API.

6. Results & Insights (Visualizations & Metrics)







- Movie Language Distribution: A pie chart showing the distribution of languages across all movies, with minor languages grouped under "Other" for clarity.

- Top 10 Most Popular Movies: A horizontal bar chart displaying movies with the highest popularity scores.
- Genre-Based Analysis:Quantitative graph of genre with the number of movies in the dataset

7. Challenges & Future Improvements

Challenges faced during the project:

- Data Cleaning: Handling missing or inconsistent movie metadata.
- Poster Retrieval Issues: Some movies lacked poster images in the TMDb database.
- Recommendation Quality: Improving accuracy by incorporating additional features like user ratings and collaborative filtering.

Future improvements:

- Implementing hybrid recommendation techniques combining content-based and collaborative filtering.
- Enhancing the user interface with search functionality and filters.
- Integrating real-time user feedback to refine recommendations over time.

8. Conclusion & Learnings

This project successfully demonstrates a content-based movie recommendation system using machine learning and API integration. By analyzing movie metadata, we can provide users with personalized recommendations. The insights gained from this project highlight the importance of data preprocessing, similarity calculations, and API utilization in building an effective recommendation system.

9. References

- TMDb API Documentation: https://developers.themoviedb.org/3/getting-started/introduction
- Scikit-learn Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html
- Github
- Streamlit Documentation: https://docs.streamlit.io/