Movie Recommender System

K Nikhila
21CE31007

Duration: April '24 – May '24

1. Introduction

In the age of digital content, recommender systems have become indispensable for platforms like Netflix and Amazon. This project details the development of a content-based movie recommender system using a dataset from TMDB (The Movie Database). The system's core function is to suggest movies to a user based on the similarity of a movie they have already watched. Unlike collaborative filtering, which relies on user behaviour, this model uses a movie's metadata (genres, keywords, cast, crew) to find other movies that share similar characteristics. The result is a simple yet effective tool for discovering new movies aligned with a user's preferences.

2. Objectives

The primary objectives of this project were to:

- Clean and merge two datasets, one containing movie metadata and the other with credit information (cast and crew).
- Perform feature engineering to create a unified text-based representation of each movie.
- Calculate the similarity between movies using a vectorization technique and cosine similarity.
- Develop a recommendation function that takes a movie title as input and returns a list of similar movies.
- Export the final model and associated data for potential deployment.

3. Scope of Works

The scope of this project involved a detailed analysis of movie data:

- Data Acquisition and Merging: Two datasets, tmdb_5000_movies.csv and tmdb_5000_credits.csv, were loaded. These datasets were merged on the title column to create a single, comprehensive DataFrame.
- Feature Selection: To create a content-based model, only key features relevant to a movie's content were selected. The chosen features included movie_id, title, overview, genres, keywords, cast, and crew.
- Data Cleaning and Preprocessing: The selected features, which were in a string format representing a list of dictionaries, were cleaned and converted into a simple list of strings. This involved removing spaces from names and keywords to prevent them from being treated as separate entities during vectorization.
- Vectorization: A CountVectorizer was used to transform the processed text into a numerical matrix. This process created a vector for each movie, where each dimension corresponds to a word or tag, and the value represents its frequency.

- Similarity Calculation: The cosine similarity metric was applied to the vectorized data to measure the similarity between every pair of movies. Cosine similarity is a particularly effective method for text data as it measures the angle between two vectors, making it robust to variations in document length.
- Recommendation Logic: A Python function was developed to retrieve the index of a given movie, find its most similar movies based on the similarity matrix, and return the top 5 movie titles.

4. Methods Used

The project followed a standard data science workflow. We started by importing the necessary libraries: pandas for data handling, numpy for numerical operations, and ast for safely evaluating string representations of Python data structures.

Data Merging and Feature Engineering:

- The movies and credits DataFrames were merged to combine movie metadata with cast and crew information.
- A new DataFrame was created with the key features: movie_id, title, overview, genres, keywords, cast, and crew.
- Several functions were written to process the string data in genres, keywords, and cast columns to extract only the names and convert them into a list of strings. The crew column was specifically processed to extract the director's name.
- To ensure that multi-word entities like "Science Fiction" and "Johnny Depp" were treated as single tags, spaces were removed from all relevant lists.
- A new tags column was created by concatenating the processed overview, genres, keywords, cast, and crew columns into a single, comprehensive list of strings for each movie.

Model Building:

- The tags column, a list of strings for each movie, was transformed into a matrix of numerical vectors using CountVectorizer. The max_features parameter was set to 5000 to limit the vocabulary size, and stop_words='english' was used to filter out common words that do not add value.
- The resulting matrix of vectors was then used to compute the cosine similarity between all movies, which formed the basis of the recommendation system.

5. Results

The recommender system was tested using the movie "Gandhi". The system successfully identified and recommended a list of movies with similar themes and characteristics, including:

1. Gandhi, My Father
2. The Wind That Shakes the Barley
3. A Passage to India
4. Guiana 1838

5. Ramanujan

This result demonstrates that the model is effective at identifying semantic similarities between movies based on their textual metadata. The recommendations were logical and aligned with the historical drama genre of the input movie, validating the content-based approach.

6. Discussion

The project successfully created a functional and scalable content-based movie recommender system. By using vectorization and cosine similarity, we were able to quantify the similarity between movies and use that score to generate recommendations. The data cleaning and feature engineering steps were crucial in preparing the unstructured text data for the model. The final recommendation function is efficient and provides relevant suggestions, making it a viable solution for a movie-centric application.

7. Conclusion

This project demonstrates the effective use of a content-based filtering approach to build a movie recommender system. The model successfully processes movie metadata to identify and recommend similar films, proving the value of textual analysis in creating personalized user experiences. The final output is a working recommendation engine that can be easily integrated into a larger application.

8. References

Recommender Systems: An Overview, Research Trends, and Future Directions

Link:
https://www.researchgate.net/publication/339172772_Recommender_Systems_An_Overview_Research_Trends_and_Future_Directions

Declaration by the Student

I, K Nikhila, hereby declare that the work presented in this report is an authentic record of my own self-initiated project. This work has been completed in accordance with the guidelines provided for a self-study project. The material has not been submitted to any other university or institution for any academic purpose.

Date: K Nikhila

Place: IIT Kharagpur 21CE31007