PROJECT REPORT: MOVIE RECOMMENDATION SYSTEM

Movie recommendation system have become increasingly popular, aiming to alleviate the overwhelming burden of choice in today's vast cinematic landscape. It emerges as a pivotal tool addressing this challenge, enhancing user satisfaction by providing tailored suggestions based on individual preferences. Beyond user gratification, these systems contribute to platform engagement, content discovery and customer loyalty. For content providers effective recommendation algorithms translate into improved user retention, increased monetization and a competitive edge in the crowded streaming market. The following report explores the significance of recommendation through implementation of three distinct approaches: Simple, Content-Based and Collaborative movie recommendation systems.

Dataset:

Dataset Link:

https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

https://files.grouplens.org/datasets/movielens/ml-latest-small.zip

TMDB Dataset: The TMDB 5000 Movie Dataset obtained from Kaggle is a comprehensive collection of movie related information that includes details such as:

- budget: The budget of the movie.
- genres: A stringified list of dictionaries that list out all the genres associated with the movie.
- homepage: The Official Homepage of the move.
- id: The ID of the move.
- original_language: The language in which the movie was originally shot in.
- original_title: The original title of the movie.
- overview: A brief blurb of the movie.
- popularity: The Popularity Score assigned by TMDB.
- production_companies: A stringified list of production companies involved with the making of the movie.
- production_countries: A stringified list of countries where the movie was shot/produced in.
- release_date: Theatrical Release Date of the movie.
- revenue: The total revenue of the movie in dollars.
- runtime: The runtime of the movie in minutes.
- spoken_languages: A stringified list of spoken languages in the film.
- status: The status of the movie (Released, To Be Released, Announced, etc.)
- tagline: The tagline of the movie.
- title: The Official Title of the movie.
- vote_average: The average rating of the movie.
- vote_count: The number of votes by users, as counted by TMDB.
- cast: cast of the movie
- crew: crew of the movie

The data includes metadata about 4803 movies. It serves as a valuable resource for building recommendation systems and conducting analyses in the field of movie data.

MovieLens Dataset: The dataset contains information about 9,742 movies. This information include:

- movieId: Id of the movie

- title: The official title of the movie
- genres: genres associated with the movie
- UserId: Id unique to the user
- rating: rating given by the user to the movie
- timestamp:when a user rated a particular movie

There are 100,836 ratings in the dataset. Each rating corresponds to a user's assessment of a particular movie. Ratings are typically on a numerical scale, such as 1 to 5 stars. This dataset is used for building collaborative filtering models.

Data Preprocessing:

- Relevant columns from ‘meta_df’ including 'id', 'title', 'genres', 'keywords', 'overview', and 'tagline' were selected.
- The 'cast' and 'crew' information was extracted from **credit_df** and merged into the main dataframe using the movie ID ('id').
- Missing values in the 'tagline' column were filled with empty strings to ensure data consistency.
- JSON-formatted columns ('genres', 'keywords', 'cast', 'crew', ’production_companies’ etc) were converted from strings to lists using the literal_eval function.
- Extracted relevant information from the 'genres', 'keywords', 'cast', and 'crew' columns to improve the system's understanding of the data.
- Converted text data to lowercase and removed spaces for consistency and ease of processing.
- A new feature, 'feature', was created by combining relevant textual information ('genres', 'title', 'overview', 'tagline', 'keywords', 'director', 'cast').
- Merged movie and rating data based on the 'movieId' column in the MovieLens dataset.
- Extracted relevant columns ('movieId', 'title', 'genres', 'userId', 'rating') from the merged data.

Exploratory Data Analysis:

- Using wordclouds, we can see the word ‘Man’ is the most commonly used word in movie titles. ‘Love’, ‘Day’, ‘Dead’ are also among the most commonly occurring words.

- In the movie overview, ‘World’ and ‘find’ are most commonly used followed by ‘Life’. It gives us a pretty good idea of the most popular themes in the movies.

- The United States of America is the most popular destination of production for movies given that our dataset largely consists of English movies. UK is also an extremely popular location with the UK, Germany, France and Canada in the top 5.

- Warner Bros is the highest earning production company of all time earning a staggering 49.5 billion dollars. Universal Pictures and Paramount Pictures are the second and the third highest earning companies with 42 billion dollars and 40 billion dollars in revenue respectively.

- TMDB defines 20 different genres for our set of 4803 movies. Drama is the most commonly occurring genre with almost half the movies identifying itself as a drama film. Comedy comes second with 25% of the movies having adequate doses of humour. Other major genres represented in the top 10 are Thriller, Action, Romance, Adventure, Crime, Science Fiction, Horror, and Family.

- Inception and The Dark Knight are the two most voted movies in the top of the chart.

- There is no correlation between population and vote_average, which means both of these features are independent.

Simple Recommendation System:

The Simple Recommendation System employs a straightforward approach to identify top-rated movies based on average ratings and vote counts. The system aims to provide users with a list of recommended movies that have received high ratings and a substantial number of votes, indicating a broad appeal. It also uses popularity to identify most popular movies for recommendation.

For this, the weighted average score(W) for each movie is calculated using a formula that combines average rating and vote counts:

$$W=(Rv+Cm) /(v+m)$$

Where W = Weighted average

R = avg of the movie as a number from 0 to 10 (rating)

V = no. of votes for the movie

M = min votes required to be listed in the Top 250

C = the mean votes across the whole data

This formula ensures that the weighted score considers both the average rating and the vote count, giving a higher score to movies with high ratings and a substantial number of votes. Here we used 75the percentile as our cutoff. That is, for a movie to feature in the charts, it must have more votes than at least 75% of the movies in the list. Therefore, to qualify to considered for the chart, a movie must have at least 737 votes. We get “The Shawshank Redemption”, “The Godfather” and “Fight Club” as our top-rated movies. But based on popularity “Minions”, “interstellar” and “Deadpool” comes top.

Content-Based Recommendation System:

The Content-Based Recommendation System leverages the content or features of movies to generate personalized recommendations. In this approach, movies are described by various attributes such as overviews, genres, and other metadata. The system recommends movies that are similar to those the user has shown interest in.

Based on overview: We employed the widely used TF-IDF (Term Frequency-Inverse Document Frequency) technique to analyse movie overviews. This method

transforms textual data into numerical vectors, highlighting important keywords and phrases while accounting for their overall presence in the dataset. The sigmoid kernel was applied to the TF-IDF matrix to calculate pairwise similarities between movies. This kernel function is suitable for measuring similarity and capturing non-linear relationships. A recommendation function was developed to generate movie recommendations based on the similarity scores. Given a movie title, the system identifies similar movies and presents a list of recommendations.

To validate the effectiveness of the content-based recommendation system, a test case was conducted for the movie "Pirates of the Caribbean: Dead Man's Chest." The system successfully recommended the movies:

```
199     Pirates of the Caribbean: The Curse of the Bla...
17            Pirates of the Caribbean: On Stranger Tides
943                                               Firewall
4364                                   Two Girls and a Guy
2059                                              Moliere
716                                             Ladder 49
1872              Anacondas: The Hunt for the Blood Orchid
4146                                              Highway
1755                                          Joyful Noise
1006                  Indiana Jones and the Last Crusade
Name: title, dtype: object
```

Based on metadata: Instead of relying solely on individual feature like overviews, we will generate a new feature. This feature combines genres, titles, overviews, taglines, keywords, director names, and cast information, that comprehensively defines each movie's identity. Count Vectorization technique is used to converts the "feature" text into a matrix of token counts, essentially quantifying the occurrence of words and phrases within each movie description. Calculated cosine similarity between movies based on the count matrix to quantify their textual similarity. Developed a function, get_recommendations2, which takes a movie title as input and returns a list of top recommended movies based on cosine similarity. Applied the recommendation function to the movie 'Avatar', resulting in a list of recommended movies including 'Man of Steel', 'X-men: Days of Future Past', 'True Lies', and others. These recommendations appear thematically connected to "Avatar," sharing elements of futuristic sci-fi, action-adventure, and fantastical worlds.

```
14                                  Man of steel
46                    X-men: days of future past
282                                   True lies
813                                    Superman
3494    Beastmaster 2: through the portal of time
870                                  Superman ii
2403                                      Aliens
587                                    The abyss
3439                               The terminator
279                    Terminator 2: judgment day
Name: title, dtype: object
```

Collaborative Filtering:

Collaborative recommender systems operate on the premise that past interactions between users and items are valuable indicators for making future recommendations. These interactions are typically organized into a "user-item interactions matrix," where each entry represents a user's interaction (e.g., rating) with a specific item.

Utilizing the popular MovieLens dataset, we aimed to create a personalized recommendation experience based on user preferences. A minimum vote threshold was applied to filter out movies with fewer ratings. This step ensured a more robust recommendation system by focusing on movies with a substantial user base. The collaborative filtering approach using the Nearest Neighbors model with cosine similarity was implemented to find similar movies.This technique identifies movies closest to a user's preferred choice based on user ratings. The model was trained on the filtered dataset to enhance recommendation accuracy. The system generated recommendations for specific movies, showcasing the top recommended movies along with their respective distances. This functionality allows users to discover movies similar to their preferences.

```
Top 10 recommendations for Lion King, The (1994) are :
1.Aladdin (1992),with distance 0.28209064327932476
2.Beauty and the Beast (1991),with distance 0.2916388110805065
3.Mrs. Doubtfire (1993),with distance 0.349909919162831
4.Mask, The (1994),with distance 0.3610848282154827
5.Jurassic Park (1993),with distance 0.38515256256257724
6.Jumanji (1995),with distance 0.4115622741415874
7.Forrest Gump (1994),with distance 0.4135539398338064
8.Babe (1995),with distance 0.423786344732479
9.Home Alone (1990),with distance 0.44147047190650657
10.Snow White and the Seven Dwarfs (1937),with distance 0.4458731949412
9607
```