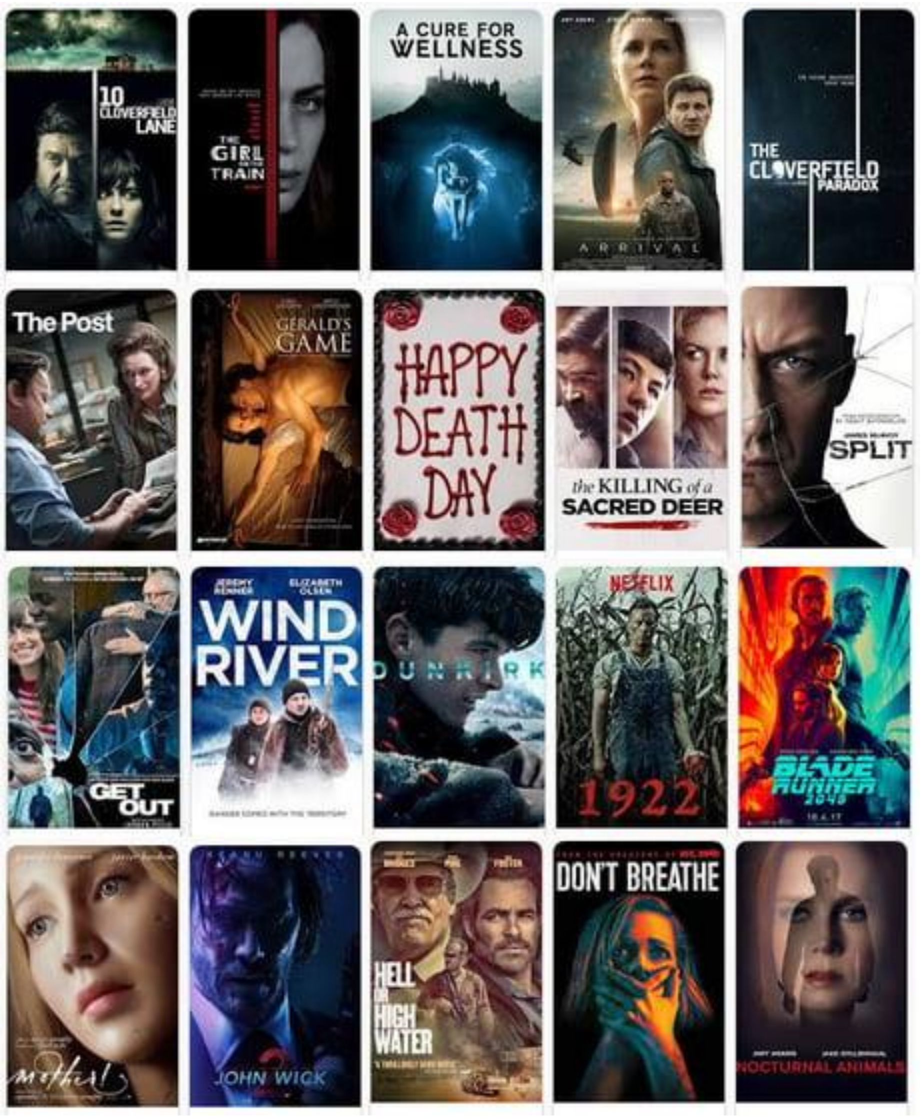

MOVIE RECOMMENDATION SYSTEM

ABSTRACT

This project aims to develop a movie recommendation system utilizing natural language processing techniques, offering personalized suggestions based on user preferences and reviews for an enhanced movie-watching experience.

Spandan Ghosh

Abstract

The movie recommendation system is a crucial application in the field of information retrieval and data analysis. It aims to provide personalized movie suggestions to users based on their preferences and viewing history. In this project, we have developed a movie recommendation system using TF-IDF vectorization and cosine similarity techniques. By leveraging these methods, we can convert textual movie data into numerical vectors and measure the similarity between movies, respectively. This allows us to offer relevant and tailored movie recommendations to users, enhancing their movie-watching experience. In the following sections, we will delve into the data preprocessing steps, the methodology employed, and provide examples of movie suggestions generated by the system.

Data Description:

We'll make use of the tmdb_5000_movie_dataset to construct the recommendation system. The dataset includes information on the roughly 5000 movies that are currently available. The following link will take you to the Kaggle website where you may access this dataset:

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

Two CSV files are included in this dataset:

1. movies.csv and
2. credits.csv.

These two datasets each have 4803 rows.

1. **budget** – budget in US dollar of the movie
2. **genres** – words describing the genre of the movie
3. **homepage** – link to the website of the movies
4. **id** – the unique id of the movie in tmdb database
5. **keywords** – keywords for the movie
6. **original_language** – original language of the movie
7. **original_title** original title of the movie
8. **overview** – brief text description of the movie
9. **popularity** – popularity score of the movie calculated by tmdb
10. **production_companies** – name of the production companies associated with the movie
11. **production_countries** – country of production
12. **release_date** – date of release of the movie
13. **revenue** – revenue in US dollar from the movie
14. **runtime** – total runtime in minutes of the movie
15. **spoken_languages** – language spoken in the movies
16. **status** – whether it is released or not
17. **tagline** – tagline of the movie

18. **title** – title to the movie
19. **vote_average** – average votes of the movie
20. **vote_count** – total votes received by the movie

On the other hand the columns of the tmdb_5000_credits,csv has the following column description –

1. **movie_id** – unique id of the movies assigned by tmdb database
2. **title** – title of the movie
3. **cast** – casts of the movies
4. **crew** – crew of the movie

In the dataset information like genres of the movie, casts, directors, movie keywords etc. are given for each of the movies. We can use this data to get information about the contents of all the movies in the dataset. All this contents that are available in the dataset are suggesting to build up a content-based recommendation system.

Methodology

Before we go into the analysis of the movie dataset we need to pre-process the dataset, so that is useful to process. We have gone through the following stages to pre-process the dataset.

- During the data preprocessing stage, several columns were removed from the dataset, including 'budget', 'id', 'keywords', 'overview', 'release_date', 'title', 'vote_average', 'movie_id', 'cast', and 'crew'. These columns were not relevant for our movie recommendation system.

- As part of our data preprocessing, we have implemented a feature engineering step where we created a new column called 'tags'. This column is generated by combining the information from various sources, including genres, keywords, cast, crew, and overview. By amalgamating these diverse elements, we aim to capture the essence of each movie more comprehensively. The 'tags' column serves as a consolidated representation of the movie, encompassing its genre, key themes, notable cast and crew members, and a concise summary of the plot.

- We have used TF-IDF vectorization method to convert the data into vector form. TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a numerical statistic used to evaluate the importance of a term within a collection of documents. In our case, each movie is treated as a document. TF-IDF vectorization converts textual data into numerical vectors, enabling machine learning algorithms to process and analyse them. This technique assigns higher weights to terms that are more important in a document while downgrading the significance of common terms. The resulting vectors represent the movies in a high-dimensional vector space.

- At last, we have used cosine similarity to suggest similar movies to the used. Cosine similarity is a metric used to measure the similarity between two vectors in a vector space. In the context of our movie recommendation system, we calculate the cosine similarity between the TF-IDF vectors of movies to determine how closely related they are. Higher cosine similarity scores indicate greater similarity between movies.

Sample Outputs

We have implemented the above-mentioned algorithm to build up a content-based recommendation. Let try out inputting some movies and find out the result:

Illustration 1:

```
Input: Avatar
Output:      Titan A.E.
             Small Soldiers
             Independence Day
             Aliens vs Predator: Requiem
             Ender's Game
```

Illustration 2:

```
Input: Iron Man
Output:      Iron Man 2
             Iron Man 3
             Avengers: Age of Ultron
             Captain America: Civil War
             The Avengers
```

Conclusion

The movie recommendation system employs TF-IDF vectorization and cosine similarity to provide accurate movie suggestions based on user preferences. The system analyzes the textual data associated with movies and compares their vectors to find the most relevant recommendations. The presented examples demonstrate the effectiveness of the approach by recommending similar movies to popular films with high similarity scores.