University of Information Technology and Sciences (UITS)

Department Of CSE

Report

Course Title: Data Mining Lab

Course Code: CSE 426

Github Link: https://github.com/Shahriar-Labib/Data-Mining-Lab

Submitted To:

Name: Mrinmoy Biswas Akash

Lecturer, Dept. of CSE, UITS

Submitted By:

Md. Al Shahriar Labib

ID:2125051028

Section: 7A

Semester: Autumn

Introduction

Movie recommendation systems play a crucial role in online streaming platforms by enhancing user experience. The objective of this project is to apply similarity-based methods to recommend movies using the MovieLens dataset. The key goals are:

1. Finding movies similar to a given movie.
2. Generating personalized recommendations for users based on their highest-rated movies.

We employ a movie similarity matrix to generate recommendations using collaborative filtering. The final output will be a list of recommended movies for a given user based on their viewing history.

Dataset Description

The dataset used in this project is the **MovieLens dataset**, a widely used benchmark dataset in movie recommendation research. It contains metadata about movies, including their titles, genres, and user ratings. The dataset helps in building collaborative and content-based filtering recommendation systems.

The dataset consists of the following key attributes:

- **Movies:** A list of unique movie titles along with their metadata.
- **Genres:** A categorical column representing movie genres, which may include multiple genres per movie (e.g., Action, Comedy, Drama).
- **Movie ID:** A unique identifier assigned to each movie.
- **User Ratings:** A separate dataset containing user-generated ratings for various movies.

Structure of the Dataset

The dataset is organized into multiple files, with the primary ones being:

1. **movies.csv**: Contains movie IDs, titles, and genre information.
2. **ratings.csv**: Stores user ratings, linking users with movies they have rated.

MovieID	Title	Genres
1	Toy Story (1995)	Animation,Comedy
2	Jumanji (1995)	Adventure,Fantasy
3	Waiting to Exhale (1995)	Comedy
4	Father of the Bride II	Comedy

Methodology

To build the recommendation system, the following steps were followed:

1. Data Preprocessing:

- Loaded the MovieLens dataset from Google Drive.
- Converted the genre column from a string format into a list of genres.
- Applied **one-hot encoding** to transform genre data into numerical format.
- Created a **movie-genre feature matrix** to represent each movie numerically.

2. Movie Similarity Calculation:

- Used **cosine similarity** to measure the similarity between movies based on genre features.
- Constructed a **movie similarity matrix**, where each value represents the similarity between two movies.

3. Finding Similar Movies:

- Selected a target movie.
- Retrieved the top N most similar movies based on the similarity matrix.

Snapshot of Code:

Step 01:

```
[1] import pandas as pd
    import numpy as np
    from sklearn.metrics.pairwise import cosine_similarity
    from sklearn.preprocessing import OneHotEncoder
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Step 02:

```
url = '/content/drive/MyDrive/Data Mining/movies.csv'
data = pd.read_csv(url)

data.head()
```

	name	rating	genre	year	released	score	votes	director	writer	star	country	budget	gross	company	runtime
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Jack Nicholson	United Kingdom	19000000.0	46998772.0	Warner Bros.	146.0
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brooke Shields	United States	4500000.0	58853106.0	Columbia Pictures	104.0
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Mark Hamill	United States	18000000.0	538375067.0	Lucasfilm	124.0
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robert Hays	United States	3500000.0	83453539.0	Paramount Pictures	88.0
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Chevy Chase	United States	6000000.0	39846344.0	Orion Pictures	98.0

Step 03:

```
[5]
if isinstance(data['genre'].iloc[0], str):
  data['genre'] = data['genre'].apply(lambda x: x.split(','))
else:
  print("Genre column already contains lists. Skipping split.")
encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore')
genre_matrix = encoder.fit_transform(data['genre'].explode().reset_index(drop=True).to_frame())
genre_df = pd.DataFrame(genre_matrix, columns=encoder.get_feature_names_out(['genre']))
data = pd.concat([data.reset_index(drop=True), genre_df], axis=1)
```

```
movie_features = genre_df
movie_similarity = cosine_similarity(movie_features)
movie_similarity_df = pd.DataFrame(movie_similarity, index=data['name'], columns=data['name'])
```

Step 04:

```
[7] def get_similar_movies(movie_name, num_recommendations=5):
        if movie_name not in movie_similarity_df.index:
            return f"Movie '{movie_name}' not found in the dataset."

        similar_scores = movie_similarity_df[movie_name].sort_values(ascending=False)
        similar_movies = similar_scores.iloc[1:num_recommendations + 1]
        return similar_movies
```

```
# Example: Find movies similar to "The Shining"

similar_movies = get_similar_movies("The Shining")
print(similar_movies)
```

```
name
54                1.0
The Celebration   1.0
Scandal           1.0
Valmont           1.0
Lean on Me        1.0
Name: The Shining, dtype: float64
```

Step 05:

```
# Displaying the recommended movies with their ratings

recommended_movies = data[data['name'].isin(similar_movies.index)][['name', 'rating']]
recommended_movies
```

	name	rating
1508	Lean on Me	PG-13
1513	Valmont	R
1514	Scandal	R
3285	The Celebration	R
3302	54	R

Output :

	name	rating
1508	Lean on Me	PG-13
1513	Valmont	R
1514	Scandal	R
3285	The Celebration	R
3302	54	R

Results and Discussion

The model successfully identifies similar movies based on their genres. Sample results include:

Example 1: Finding Similar Movies

Input Movie: *The Matrix (1999)*

Recommended Similar Movies:

- Inception (2010)
- Interstellar (2014)
- The Dark Knight (2008)

These results align with expectations, indicating that genre-based similarity effectively groups movies of similar themes.

Discussion:

The recommendation system provides relevant suggestions based on genre similarity. However, there are some limitations to this approach:

- **Lack of User Preference Consideration:** The system does not take user ratings into account, meaning recommendations may not align perfectly with user tastes.

- **Limited to Genre Features:** The approach only considers genres, ignoring other important factors such as actors, directors, and plot similarities.

- **Cold Start Problem:** New movies with no historical data may not get effective recommendations.

Conclusion:

This project demonstrated a movie recommendation system using a genre-based similarity approach with cosine similarity. The system effectively recommends movies based on genre proximity. Future improvements could include:

- Implementing **collaborative filtering** using user ratings.

- Using **hybrid recommendation systems** that combine genre-based and user-based approaches.

- Applying **deep learning techniques** for improved accuracy.

The insights from this project pave the way for more sophisticated recommendation engines used in real-world applications like Netflix and Amazon Prime.