MOVIE RECOMMENDER

**PROJECT REPORT**

MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING ALGORITHMS

A PROJECT REPORT Submitted by

* ADITYA SHARMA(21bcef28)
* LOPA KHATOI(21becd68)
* SONIA PADHI(21bceb60)
* ANAND MOHAN MISHRA(21BCEC73)

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to Dr. Asit Patnaik for his invaluable guidance and mentorship throughout the development of our "Personalized Movie Recommendation System" project. His expertise in data analysis and machine learning algorithms proved instrumental in shaping the project's direction and overcoming technical challenges. Dr. Patnaik's unwavering support and insightful feedback fueled our progress, enabling us to design a robust and accurate recommendation system. We are deeply grateful for his commitment to our success and for fostering a collaborative learning environment that nurtured our skills and understanding of the field. We owe this achievement to Dr. Asit Patnaik's dedication and are honored to have benefited from his mentorship.

TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| 1 | ABSTRACT | 1 |
| 2 | INTRODUCTION | 2 |
| 3 | DATASET | 3 |
| 4 | LITERATURE SURVEY | 4-5 |
| 5 | PROPOSED SYSTEM | 6 |
| 6 | ALGORITHMS |  |

ABSTRACT

Everyone loves movies no matter age, gender, race, color, or geographical location. We tend to all in the simplest way be connected to every different via this wonderful medium, nonetheless what is most attention-grabbing is the undeniable fact that however distinctive our selections and combos are in terms of picture show preference. Some individuals like genre-specific movies be it a thriller, romance, or sci-fi. At the same time, others specialize in lead actors and administrators. After we take all the under consideration, it’s astoundingly troublesome to generalize a movie and say that everybody would love it. However, with all that said, it’s still seen that similar movies are liked by a selected part of society. So here’s whether we tend to as information scientists get to play and extract the juice out of all the behavioral patterns of not solely the audience but conjointly from the films themselves.

INTRODUCTION

Gone are the days of aimlessly scrolling through endless movie libraries. This project delves into the captivating domain of personalized movie recommendations, crafting a system that caters to each user's unique cinematic cravings.

We leverage the rich terrain of the Movielens dataset, boasting over 100,000 user ratings across 9,000+ movies. This treasure trove of preferences reveals hidden patterns and trends, forming the bedrock of our recommendation engine.

Our approach utilizes three distinct models, each employing sophisticated techniques to unlock the secrets of user satisfaction:

* Collaborative Filtering: We tap into the collective wisdom of the crowd, analyzing patterns in user-movie interactions. This method identifies movies similar to those enjoyed by others with similar tastes, ensuring a delightful sense of community-driven discovery.
* Content-Based Filtering: We delve into the intrinsic qualities of movies themselves, analyzing textual tags, genres, and other metadata. This model recommends movies that share characteristics with those a user has previously enjoyed, offering a deep dive into their specific cinematic preferences.
* Hybrid Approach: Combining the strengths of both worlds, our hybrid model harnesses the power of collaborative and content-based filtering. Leveraging cosine similarity in this model allows us to identify not only movies enjoyed by similar users but also those that share thematic or stylistic resonance with previously cherished picks.

Through the magic of Truncated SVD, a dimensionality reduction technique, we unlock a lower-dimensional space where movies reside. This "latent matrix" allows for efficient analysis and comparison, paving the way for accurate and personalized recommendations.

Our project promises to revolutionize the way we discover and enjoy movies. Forget tedious browsing and algorithm fatigue – this data-driven engine promises to unveil the perfect flick, tailored specifically to your unique cinematic palate. So, grab your popcorn and settle in, for a personalized journey through the world of film awaits!

DATASET

This code explores movie recommendations on the Movielens dataset, featuring 100k+ ratings and 9k+ movies. Users with at least 55 ratings were chosen, and their preferences were analyzed through textual tags and 5-star ratings. By combining TF-IDF for text and SVD for dimension reduction, the code extracts key information from diverse data.. Finally, it creates a "latent matrix" representing movies in a lower-dimensional space, opening doors for deeper analysis and personalized recommendations. All in all, this code delves into user preferences and movie characteristics to build accurate and insightful movie recommendations.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TITLE | YEAR | AIM | DATASET | METHOD | RESULT | CONCLUSION |
| Movie recommendation System using clustering Algorithm and Pattern recognition network | 2018 | This paper proposed a machine learning approach to recommend movies to users using a K-means clustering algorithm to separate similar users and create a neural network for each cluster. | MovieLens (Publicly Available) has 11 target classes And 12000 users | K Means clustering Neural Network | The system showed 95% accuracy on average in predicting ratings from new user data which can be used to 146 analyze which movie should be recommended to new users. | This proves that our system is a valid one for prediction in the field of movies. This ensures that our system can deal with different types of users with diverse attitudes towards movies. |
| Analysis of Movie Recommendation Systems; with and without considering the low-rated movies | 2020 | To shows that low rated movies are not significant in finding the movie predictions. So it’s suggestable to ignore them while calculating movie predictions | Movie-Lens100k. | collaborative filtering Pearson correlation coefficient | Movie predictions for the user with user-Id 254: it is observed that there is no significant difference between the predictions. The negligible difference between the predictions shows that the effect of removing lowrated movies is negligible and hence can be removed. | This proves that, movies that have never got an above average rating does not have significant contribution in movie recommendations and it’s suggested to ignore such movies |

ALGORITHMS

**Understanding the Code and Algorithms:**

This code builds a movie recommender system using two approaches: content-based filtering and collaborative filtering. It then combines these approaches through hybrid recommendations. Here's a breakdown of the algorithms and their roles:

**1. Content-based Filtering:**

* **TF-IDF (Term Frequency-Inverse Document Frequency**): This algorithm analyzes movie descriptions or metadata to identify keywords and weigh their importance based on frequency and rarity.
* **Truncated Singular Value Decomposition (SVD):** This dimensionality reduction technique compresses the TF-IDF matrix by capturing the most significant information in fewer dimensions (200 in this case). The resulting latent matrix represents each movie as a point in a 200-dimensional space based on its content characteristics.
* **Cosine Similarity:** This metric measures the angle between two points in the latent space. Movies with similar content will have lower angles and thus higher cosine similarity scores.

**2. Collaborative Filtering:**

* **Pivot Table:** This transforms user ratings into a matrix where rows represent movies and columns represent users, with cell values being the specific ratings.
* **SVD:** Again, SVD is used to reduce the dimensionality of the rating matrix, generating a latent matrix where movies are positioned in a 200-dimensional space based on user preferences and how similar their ratings are.
* **Cosine Similarity:** Similar to content-based filtering, movies with similar user-rating patterns will have higher cosine similarity scores in the latent space.

**3. Hybrid Recommendations**:

* This code takes the individual cosine similarity scores from both content and collaborative filtering and simply averages them for each movie. This creates a hybrid similarity score that combines both content-based and user-based preferences.
* Sorting movies by this hybrid score allows for recommendations that consider both the inherent characteristics of the movies and how other users have rated them.
* Overall, the code uses TF-IDF for understanding movie content, SVD for efficient dimensionality reduction and latent space formation, and cosine similarity for measuring movie relationships based on content and user preferences. By combining these algorithms, it creates a hybrid recommendation system that aims to offer personalized suggestions for movie lovers.

FUTURE WORK AND REFERENCES

**Future Work for your Movie Recommender System:**

1. **Deployment and Accessibility:**
2. **Web app or mobile app:**

* Create a user-friendly interface where users can interact with the recommendation system. This could involve searching for movies, browsing recommendations, and providing feedback on suggested titles.

1. **Cloud deployment:**

* Host the recommender system on a cloud platform like AWS or Google Cloud to ensure scalability and reliability. This allows for accommodating larger user bases and potentially integrating with other services.

1. **API integration:**

* Develop an API for the recommender system to allow access from other applications or platforms. This can enable partnerships with online movie streaming services, movie databases, or recommendation platforms to expand reach and user engagement.

1. **Enhanced Functionality and Personalization:**
2. **Dynamic recommendation updates:**

* Consider implementing real-time updates to the recommendations based on new user ratings, movie releases, and trending content. This ensures dynamism and responsiveness to evolving user preferences and market trends.

1. **Content-based filtering refinement:**

* Explore incorporating additional content features like genre, director, cast, and awards to enrich the representation of movies and improve content-based recommendations.

1. **Collaborative filtering optimization:**

* Investigate advanced collaborative filtering techniques like matrix factorization or neighborhood-based approaches for more accurate and personalized recommendations.

1. **User profile development:**

* Implement user profiles that track past interactions, preferences, and ratings. This allows for personalized recommendations tailored to individual tastes and viewing history.