IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁘ Peer-reviewed journal / Refereed journal ⁘ Vol. 12, Issue 10, October 2023

DOI: 10.17148/IJARCCE.2023.121013

Movie Recommendation System Using SVD (Letterboxd)

Aditya Bhardwaj[1], Chirla Rushil Reddy[2], Palak Arora[3]

Department of Information Science and Engineering, BMS College of Engineering, Bengaluru[1-3]

Abstract: The objective of this project is to create a movie recommendation system that utilizes Singular Value Decomposition (SVD) as the core algorithm. SVD is a well-known matrix factorization technique that effectively models user-movie interactions and extracts underlying features from the user-item matrix. By applying SVD, the recommendation system can uncover hidden patterns and similarities in movie preferences, resulting in precise and personalized movie recommendations. The system will utilize user ratings and movie metadata to construct a comprehensive user-item matrix, which will then undergo decomposition using SVD. The resulting low-rank approximation will be used to predict missing ratings and generate top-N movie recommendations for each user. The project will focus on optimizing the SVD algorithm, addressing data sparsity issues, and implementing an efficient recommendation generation process. The aim is to develop a scalable and accurate movie recommendation system that enhances user satisfaction, engagement, and overall movie-watching experiences.

Keywords: Single Value Decomposition, Movie Recommendation

I. INTRODUCTION

Movie recommendation systems have become increasingly popular in the entertainment industry, aiming to provide personalized and tailored movie suggestions to users. These systems leverage advanced algorithms and techniques to analyze user behavior and preferences, enabling accurate and relevant recommendations. One such algorithm is Singular Value Decomposition (SVD), a matrix factorization technique widely used in recommendation systems.

SVD-based movie recommendation systems utilize the concept of latent factors to capture underlying patterns and relationships in user- movie interactions. By decomposing the user-item matrix into lower- dimensional matrices, SVD uncovers latent features that represent user preferences and movie characteristics. These latent factors help in generating accurate predictions and identifying similar movies for personalized recommendations.

The key advantage of SVD is its ability to handle the sparsity of user ratings data, as it can effectively approximate missing values based on the learned latent factors. This makes SVD particularly suitable for recommendation systems operating on large-scale movie datasets with sparse user feedback.

In this project, we will focus on developing a movie recommendation system using SVD. The system will utilize user ratings, movie metadata, and the SVD algorithm to generate personalized movie recommendations for users.

We will explore techniques to optimize the SVD algorithm, handle data sparsity, and enhance the accuracy and scalability of the recommendation system.

Letterboxd is a popular social networking and film discovery platform that allows users to explore, rate, review, and discuss movies. With its intuitive interface and extensive movie database, Letterboxd serves as a comprehensive platform for movie enthusiasts, cinephiles, and casual viewers alike.

The platform enables users to create personalized film diaries, where they can track and log movies they have watched, curate lists of favorite films, and maintain a record of their movie-watching experiences. Additionally, users can rate movies, write reviews, and engage in conversations with other users, fostering a vibrant community of film lovers.

One of the key features of Letterboxd is its powerful search and recommendation system. Users can discover new movies by exploring curated lists, trending films, and personalized recommendations based on their viewing history and preferences. This functionality allows users to expand their cinematic horizons and find movies tailored to their tastes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁞ Peer-reviewed journal / Refereed journal ⁞ Vol. 12, Issue 10, October 2023

DOI: 10.17148/IJARCCE.2023.121013

Letterboxd also serves as a platform for film discussions and interactions. Users can follow other cinephiles, browse their movie diaries, and engage in conversations through comments, likes, and replies. This social aspect of Letterboxd fosters a sense of community and enables film enthusiasts to connect and share their passion for movies.

Moreover, Letterboxd provides valuable insights into the world of cinema through its curated content, including film reviews, editorials, and lists by prominent critics and users. These resources offer a wealth of information and recommendations, further enhancing the movie discovery experience.

Whether users are looking for movie recommendations, a platform to express their opinions, or a community to engage with, Letterboxd offers a comprehensive and engaging experience for film lovers. With its combination of social networking features, personalized recommendations, and a vast movie database, Letterboxd has established itself as a go-to platform for movie enthusiasts seeking to connect, discover, and celebrate the art of cinema.

II. BACKGROUND

Movie recommendation systems have gained significant attention in recent years due to the abundance of movie options and the need to provide personalized suggestions to users. Singular Value Decomposition (SVD) is a widely used technique in recommendation systems that has proven effective in modeling user preferences and item characteristics.

SVD is a matrix factorization method that breaks down a user-item matrix into lower-dimensional matrices, representing latent factors. In the context of movie recommendations, the user-item matrix consists of users' ratings for different movies. By decomposing the matrix, SVD can identify hidden patterns and relationships between users and movies.

The latent factors derived from SVD capture important information about user preferences and movie attributes. For example, a latent factor may represent the level of action in a movie, while another factor may represent the presence of romantic elements. By considering these factors, SVD-based recommendation systems can generate personalized movie recommendations by matching user preferences with movie characteristics.

SVD-based recommendation systems address the sparsity issue commonly found in user-item matrices. In movie recommendation systems, users typically rate only a small subset of available movies, resulting in a sparse matrix. SVD can effectively approximate missing ratings and fill in the gaps by leveraging the learned latent factors.

One of the advantages of using SVD is its interpretability. The latent factors obtained through SVD can provide insights into user preferences and movie attributes. These factors can be used to explain why certain recommendations are made, enhancing transparency and user trust in the recommendation system.

The success of SVD in movie recommendation systems can be attributed to its ability to capture complex relationships and provide accurate recommendations based on user-item interactions. However, there are challenges to consider, such as determining the optimal number of latent factors, handling scalability issues with large datasets, and addressing the cold start problem for new users or movies.

Overall, SVD has been a fundamental technique in the development of movie recommendation systems, offering an effective approach to extract latent factors and provide personalized movie suggestions. Its ability to handle sparsity, interpretability, and recommendation accuracy has made it a popular choice in the field of recommendation systems.

NoSQL (Not Only SQL) databases have emerged as an alternative to traditional relational databases (SQL) for handling large-scale, unstructured, and diverse data. The rise of web applications, social

III. RELATED WORK

The concept of Collaborative filtering is a popular technique used in recommendation systems to provide personalized recommendations based on the collective wisdom of a community of users. It works by analyzing the historical interactions and preferences of users to identify patterns and similarities among their behaviors. Collaborative filtering can be based on either user-based or item-based approaches. User-based collaborative filtering recommends items to a user based on the preferences of similar users, while item-based collaborative filtering suggests items based on the similarity of their characteristics or usage patterns. By leveraging the wisdom of the crowd, collaborative filtering enables the discovery of relevant items that users may not have discovered on their own, enhancing the overall movie recommendation experience.

DOI: 10.17148/IJARCCE.2023.121013

The concept of Content-based filtering is a recommendation technique that focuses on the inherent characteristics and attributes of items, such as movies, to make personalized recommendations. It analyzes the content or features of the items themselves, such as genre, actors, director, and plot keywords, to determine their similarity and relevance to the user's preferences. By understanding the user's past interactions and ratings, content-based filtering identifies items with similar content characteristics and recommends them to the user. This approach is particularly useful when there is limited information about user preferences or when personalized recommendations based on item attributes are desired. Content-based filtering complements collaborative filtering and can provide accurate and relevant recommendations based on item content analysis.

The concept of Hybrid recommendation is an approach that combines multiple recommendation techniques or algorithms to provide more accurate and diverse recommendations. It leverages the strengths of different methods, such Movie ratings are numerical or qualitative assessments given by users to express their subjective evaluation of a movie. Ratings play a vital role in movie recommendation systems as they provide valuable feedback on user preferences and help in determining the relevance and quality of movies for individual users. Ratings can be based on a predefined scale (such as a 1-5 star rating system) or customized criteria. Movie ratings contribute to the generation of personalized recommendations by identifying patterns and similarities in user preferences. They serve as a key input for recommendation algorithms, enabling the system to suggest movies that align with a user's rating history and preferences, ultimately enhancing the overall movie-watching experience.

IV. PROPOSED SYSTEM

The proposed system aims to create a movie recommendation system by leveraging collaborative filtering and Singular Value Decomposition (SVD) using data scraped from users' Letterboxd profiles. The system's workflow involves collecting and processing user ratings, combining them with a sample of ratings from the top 4000 most active users on the site, and using SVD to build a recommender model.

To start, the system collects user ratings from their Letterboxd profile. These ratings are in the form of "star" ratings, which are then converted into numerical ratings ranging from 1 to 10, accounting for half stars. This conversion ensures consistency in the rating scale for all users.

In addition to the user's ratings, a sample of ratings is collected from the top 4000 most active users on Letterboxd. This sample is obtained to broaden the collaborative filtering approach and capture a diverse range of user preferences. It is important to include ratings from highly active users to ensure the recommendations are based on a substantial dataset.

Once the user's ratings and the sample of ratings from the top 4000 users are collected, they are combined to form a comprehensive dataset. This dataset serves as the basis for building the collaborative filtering recommender model using SVD. SVD is a matrix factorization technique that decomposes the user-item matrix into lower-dimensional matrices, representing latent factors. In the context of the movie recommendation system, the user-item matrix consists of users' ratings for different movies. By decomposing the matrix, SVD can identify hidden patterns and relationships between users and movies.

The collaborative filtering model using SVD is trained on the combined dataset, capturing the underlying patterns and relationships between users and movies. This model learns the latent factors that represent user preferences and movie characteristics. It utilizes the ratings from the top 4000 users to make accurate predictions and generate recommendations for each user.

To generate recommendations, the system takes into account all movies in the full dataset that the user has not rated. These unrated movies are run through the collaborative filtering model using SVD to obtain predicted scores. The predicted scores indicate the likelihood of a user enjoying a particular movie based on their historical ratings and the collective behavior of the top 4000 users.

Once the predicted scores are calculated, the system selects the items with the top predicted scores as the recommended movies for the user. These recommendations are based on the collaborative filtering model's analysis of the user's preferences and the similarity of their ratings to those of the top 4000 users.

However, due to constraints in time and computing power, the system imposes a maximum sample size for each user. Users are allowed to select a maximum of 500,000 samples from the ratings of the top 4000 users, even though the full dataset contains over five million ratings. This limitation ensures that the model can be trained and predictions can be generated efficiently within the available resources.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal / Refereed journal ⁜ Vol. 12, Issue 10, October 2023
DOI: 10.17148/IJARCCE.2023.121013

In conclusion, the proposed movie recommendation system utilizes collaborative filtering and SVD to provide personalized recommendations based on users' historical ratings scraped from their Letterboxd profiles. By combining user ratings with a sample of ratings from the top 4000 users, the system captures a broad range of preferences and builds a collaborative filtering model using SVD. The system then predicts scores for unrated movies and recommends the top-scoring items to the user. Although the sample size is limited due to computational constraints, the system aims to deliver accurate and relevant movie recommendations based on user preferences and the behavior of highly active users on Letterboxd.

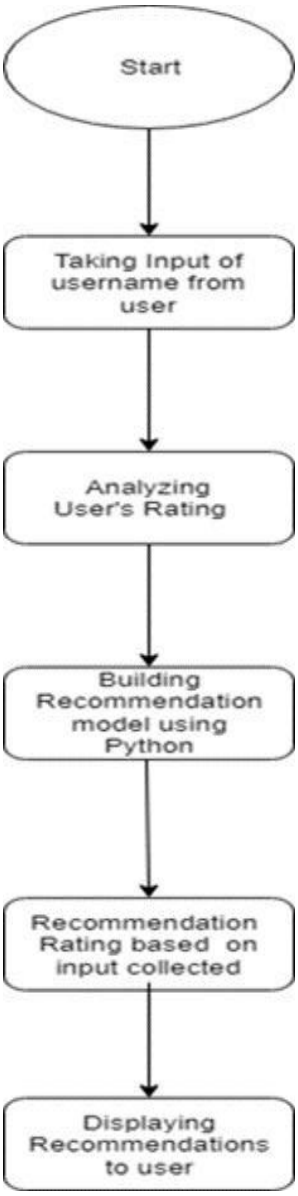

Fig 1. Model Design For Movie Recommendation System

V. METHOD OF EXPERIMENT

The movie recommendation system using Singular Value Decomposition (SVD) involves the following steps:

Data Preparation: Gather a dataset that contains information about movies and user ratings. Typically, this dataset will consist of a user-movie matrix, where each entry represents a rating given by a user to a particular movie.

Matrix Factorization with SVD: Apply SVD to decompose the user-movie matrix into three matrices: U, Σ, and V^T. U represents the user matrix, Σ is a diagonal matrix containing singular values, and V^T represents the movie matrix.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁂ Peer-reviewed journal / Refereed journal ⁂ Vol. 12, Issue 10, October 2023

DOI: 10.17148/IJARCCE.2023.121013

Dimensionality Reduction: Determine the number of dimensions, or latent factors, to consider in the SVD. This step involves selecting a suitable rank for the decomposition.

A higher rank captures more information but may also lead to overfitting. Experimentation and evaluation are required to determine the optimal rank.

Calculation of Latent Factors: Compute the latent factors by multiplying the user matrix U by the square root of the singular value matrix Σ, and the movie matrix V^T by the square root of the singular value matrix Σ.

Similarity Calculation: Measure the similarity between movies based on the latent factors. One common method is to calculate the cosine similarity between the latent factor vectors of different movies.

Recommendation Generation: To recommend movies to a user, start by finding movies they have already rated highly. Identify similar movies based on the calculated similarity measure and recommend the top-rated movies from that set.

Evaluation: Assess the performance of the recommendation system using appropriate evaluation metrics such as precision, recall, or Mean Average Precision (MAP). Split the dataset into training and testing sets to simulate real-world scenarios.
Iteration and Refinement: Fine-tune the recommendation system by adjusting parameters, such as the rank of the SVD, and repeat the evaluation process to measure any improvements.

VI. ANALYSIS

Data Exploration: Examine the dataset to understand its structure, including the number of movies, users, and ratings available. Look for missing values or any data quality issues.

SVD Decomposition: Apply SVD to decompose the user-movie matrix into the U, Σ, and V^T matrices. The rank (number of latent factors) chosen for the decomposition will significantly impact the results.

Dimensionality Reduction: Analyze the explained variance ratio provided by the singular values in the Σ matrix. This information helps determine the optimal rank to capture the most significant information while avoiding overfitting.
Similarity Calculation: Calculate the cosine similarity between movie vectors based on the latent factors. This step enables finding similar movies based on their latent representations.

Recommendation Evaluation: Split the dataset into training and testing sets. Evaluate the recommendation system using metrics such as precision, recall, or Mean Average Precision (MAP). These metrics measure the accuracy and effectiveness of the system in generating relevant recommendations.

VII. METHODOLOGY

Letterboxd is a popular social networking and film review platform that allows users to rate and review movies they have watched. Implementing a movie recommendation system in Letterboxd using Singular Value Decomposition (SVD) can enhance the user experience by suggesting personalized movie recommendations based on their preferences.

SVD can be applied to create a low-rank approximation of the user-movie ratings matrix, where the rows represent users, the columns represent movies, and the entries represent user ratings.By decomposing this matrix into three matrices, U, Σ, and V^T, we can extract latent features and identify patterns in the data.

To implement the recommendation system, we first compute the SVD of the user-movie ratings matrix. The matrix U represents the user preferences or features, and the matrix V represents the movie features. The diagonal matrix Σ contains the singular values, which quantify the importance of each feature.

For a given user, we can project their preferences onto the latent feature space by multiplying their row vector with the matrix U. This projection provides a representation of the user's tastes and preferences based on the underlying features identified by SVD.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁑ Peer-reviewed journal / Refereed journal ⁑ Vol. 12, Issue 10, October 2023

DOI: 10.17148/IJARCCE.2023.121013

VIII. RESULT

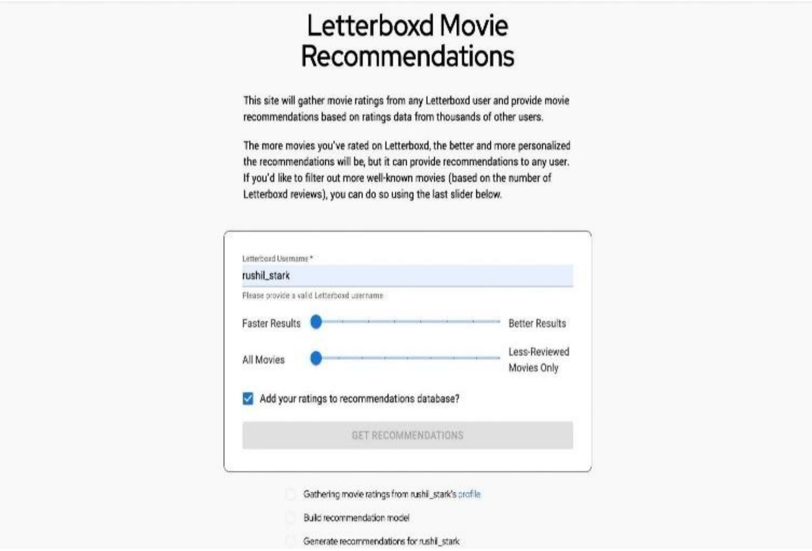

Fig 2. Front End part where user enters the profile id

In the front-end part of the application, users are prompted to enter their Letterboxd profile ID as shown in Fig 2. Fig. 3 shows the Entered Profile id of letterboxd user home page. Once the profile ID is provided, the system retrieves the user's ratings and reviews from the Letterboxd database. The application then compares the user's ratings with the top 200 ratings of the overall user base. By analyzing the user's ratings and comparing them with the top-rated films, the application generates personalized recommendations. The system takes into account factors such as genre preferences, directorial styles, and similar user ratings to provide accurate and relevant movie recommendations.

The recommendation process involves utilizing sophisticated algorithms and machine learning techniques. These techniques enable the system to identify patterns and similarities in user preferences, allowing it to suggest movies that align with the user's taste.

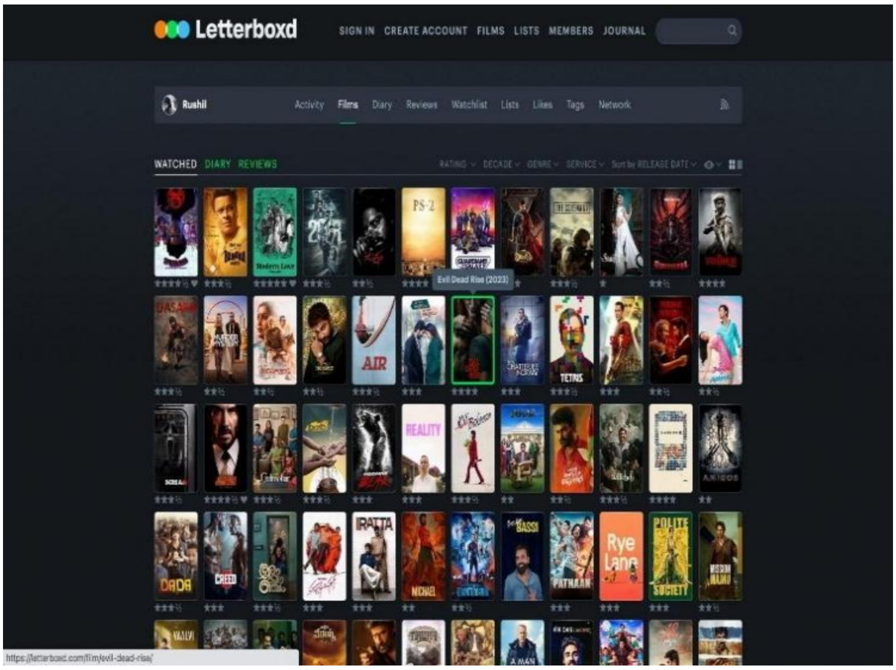

Fig 3. Entered profile id of letterboxd user home page

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed journal / Refereed journal ⁜ Vol. 12, Issue 10, October 2023

DOI: 10.17148/IJARCCE.2023.121013

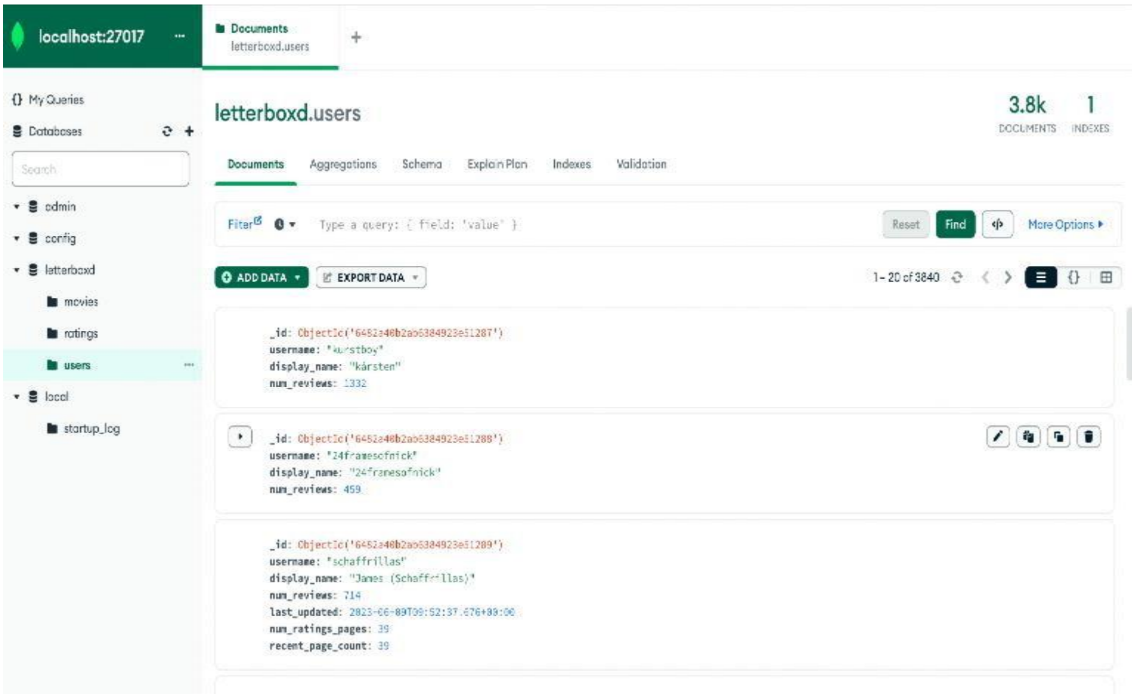

Fig 4.Backend MongoDb having top 100 users rating

Fig 5.Final output recommending various movies

By leveraging the aggregated ratings and reviews from a large user database, the application provides an effective way to discover new movies that are likely to be enjoyed by the user. This approach helps enhance the user experience by tailoring recommendations to individual preferences, leading to more satisfying movie-watching experiences.

Fig. 5 shows final outcome recommending various movies and this recommendation can be further converted to csv data format.

IX. CONCLUSION

In conclusion, the movie recommendation system offers a valuable solution for users seeking personalized movie suggestions. By leveraging advanced algorithms such as collaborative filtering, content-based filtering, and SVD, the system can analyze user preferences, movie attributes, and patterns among users to generate accurate and relevant recommendations.

The system takes into account user ratings, which are collected from sources like Letterboxd, and combines them with a sample of ratings from top users to build a collaborative filtering model using SVD. This allows the system to predict scores for unrated movies and offer recommendations based on the highest predicted scores.

The movie recommendation system enhances the movie- watching experience by providing users with a diverse selection of movies aligned with their tastes and preferences. It assists in discovering new movies, reducing decision-making fatigue, and increasing user engagement and satisfaction.

DOI: 10.17148/IJARCCE.2023.121013

Additionally, the system addresses challenges such as sparsity in user-item matrices and the cold start problem by employing robust algorithms and techniques. It ensures scalability, efficiency, and accurate predictions, even when dealing with large datasets.

Overall, the movie recommendation system holds the potentialto revolutionize the way users explore and discover movies. With its ability to personalize recommendations and cater to individual preferences, the system offers a seamless and enjoyable movie discovery process, fostering a deeper engagement and appreciation for the world of cinema.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to **Dr. B. S. Ragini Narayan, Dr. P. Dayananda Pai, and Dr. S. Muralidhara** for their unwavering support and the provision of necessary infrastructure, which played a crucial role in the successful completion of this Capstone Project Phase-1. Their support and resources were instrumental in our project's progress and ultimate success.

Our deepest appreciation goes **to Dr. Jayarekha P**, the Head of the Department of Information Science and Engineering, along with **Dr. Nalini M K** and **Prof. Harini S**, the Project Coordinators, for their constant guidance and support throughout the project. Their expertise, valuable insights, and unwavering dedication have significantly contributed to shaping our research and ensuring its quality.

Special thanks are extended to **Dr. Shobha T**, our dedicated guide and Assistant Professor in the Department of Information Science and Engineering. We are sincerely grateful for her continuous support, timely guidance, and mentorship. Her expertise and encouragement have been invaluable in navigating the challenges of the project and achieving its objectives.

We would also like to express our gratitude to all the faculty members of the Department of Information Science and Engineering for their contributions, including their insightful views and encouraging ideas. Their support and collaboration have greatly enrichedour project and fostered a conducive learning environment.

REFERENCES

[1]. Thivaharan, S., Srivatsun, G. and Sarathambekai, S., 2020, September. “A survey on python libraries used for social media content scraping”, *IEEE International Conference on Smart Electronics and Communication (ICOSEC),* pp. 361-366, September 2020.

[2]. H. Vogel, P., Klooster, T., Andrikopoulos, V. and Lungu, M., “A low-effort analytics platform for visualizing evolving Flask- based Python web services”, *IEEE Working Conference on Software Visualization (VISSOFT),* pp. 109-113, September 2017.

[3]. Lee, B.H., Dewi, E.K. and Wajdi, M.F., “Data security in cloud computing using AES under HEROKU cloud”, *IEEE 27th wireless and optical communication conference (WOCC),* pp. 1-5, April 2018.

[4]. Chen, S., Tang, X., Wang, H., Zhao, H. and Guo, M., “Towards scalable and reliable in-memory storage system: A case study with Redis”, *IEEE Trustcom/BigDataSE/ ISPA, pp. 1660-1667,* August, 2016.

[5]. Thomas, D.M. and Mathur, S, “Data analysis by web scraping using python”, *IEEE 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA),* pp. 450-454, June, 2019.

[6]. Xiang, D., Lei, D. and Wu, Y., “Data Visualization and Its Implementation of Civil Service Jobs Based on Python”, *IEEE International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp. 283-287, July 2020.

[7]. Mufid, M.R., Basofi, A., Al Rasyid, M.U.H. and Rochimansyah, I.F.,”Design an mvc model using python for flask framework development”, *IEEE International Electronics Symposium (IES)*, pp. 214-219, September, 2019.

[8]. Alcalá-Fdez, J., Alonso, J.M., Castiello, C., Mencar, C. and Soto- Hidalgo, J.M., “Py4JFML: A Python wrapper for using the IEEE Std 1855-2016 through JFML”, *IEEE International Conference on Fuzzy Systems (FUZZ- IEEE),* pp. 1-6, June, 2019.

[9]. Stančin, I. and Jović, A., “An overview and comparison of free Python libraries fordata mining and big data analysis*”, IEEE 42nd International convention on information and communication technology, electronics and microelectronics (MIPRO),* pp. 977-982, May, 2019.

[10]. He, M., Wang, B. and Du, X., “HI2Rec: Exploring knowledge in heterogeneous information for movie recommendation*”, IEEE Access*, 7, pp.30276-30284, 2019.

[11]. Qasem, S.N. and Mohammadzadeh, A., “A deep learned type-2 fuzzy neural network: Singular value decomposition approach”, Applied Soft Computing, 105, p.107244, 2021.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed journal / Refereed journal ⁜ Vol. 12, Issue 10, October 2023
DOI: 10.17148/IJARCCE.2023.121013

[12]. Friedman, J.H. and Tukey, J.W., 1974. A projection pursuit algorithm for exploratory data analysis, *IEEE*.
[13]. Diouf, R., Sarr, E.N., Sall, O., Birregah, B., Bousso, M. and Mbaye, S.N.," Web scraping: state-of-the-art and areas of application", *IEEE International Conference on Big Data (Big Data)*, pp. 6040-6042, December 2019.
[14]. Ongo, G. and Kusuma, G.P., "Hybrid database system of MySQL and MongoDB in web application development", *IEEE International Conference on Information Management and Technology (ICIMTech)*, pp. 256-260, September, 2018.
[15]. Laksono, D., "Testing spatial data deliverance in SQL and NoSQL database using NodeJS fullstack web app", 4th International Conference, August 2018.