IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

IJARSCT
Impact Factor: 6.252

Volume 2, Issue 5, May 2022

Movsic Recommender

Kirti Parde[1], Ankita Sanap[2], Prayag Padwal[3], Saurabh Rathod[4], Nalini Mhetre[5]

Student, Department of Computer Engineering[1,2,3,4]

Professor, Department of Computer Engineering[5]

Sinhgad College of Engineering, Pune, Maharashtra, India[5]

Abstract: *The growing use and users of multimedia and its industry has created a vast amount of unstructured and vague data. For instance, in the data of movies and music, those users generally find overwhelming to deal with this huge data. To simplify this obstacle, we intend on developing a project where users all around the globe can easily access this platform and take advantage of it for their own gratification. We aim at building a movie and music recommender system where users will be able to gain cursory and interesting insights about the movie or/ and music they intend to enjoy. All of this system will be based on an algorithm called Collaborative Filtering Algorithm. In this project, we wanted to develop a platform that is contemporary, and thus the reason behind discarding other standard algorithms like Content Based Filtering is because of its rudimentary factors and several other limitations. Comprehensively studying other similar projects in the past years, we found two flaws that make our project unique: one being that coalescing of movie and music recommender systems has never been done before and the other being the introduction of Collaborative Filtering Algorithm. These two aspects make our work stand out. Here we are using various algorithms, namely, KNN and SVD-WALS. Additionally, we are also implementing an Evaluation Metrics. It is further divided into 2 types, RMSE and MAE. They are used to evaluate the minimum error an algorithm returns and thus find out which algorithm is efficient for our system. Our team is currently researching about deep-learning algorithm, but due to lack of knowledge and experience we are sceptical about it. At backend, we've performed EDA and implemented KNN and SVD and adding some new features. On the other hand, at frontend, we've used HTML, CSS and JavaScript for the UI design. Since one of the most convenient frameworks is Flask, we are implementing it. Deployment is done on Heroku platform. It uses AWS server. Key problem-solving points that we would like to mention in our system are: Sparse Data Issue, Cold start, over fitting data, Scalability with large data. We got highest RMSE of KNN than SVD, further with accuracy of SVD MF the starting RMSE for SVD MF was 0.65, and MAE is 0.50. With optimization we were able to reduce it RMSE to 0.28 and MAE to 0.18. Another foremost factor in this project is its user experience, which we plan to make satisfying, wholesome, informative and fun. We aim at giving the user's at-a-glance view of all the necessary information, like: Actors, similar movies/ music, ratings, direct link to watch the searched result, and more. Thus, reducing all the strenuous efforts.*

Keywords: Collaborative Filtering, cold start, EDA, Flask, KNN, model, movie recommender, music recommender, RMSE, MAE, SVD-WALS, XGBoost

I. INTRODUCTION

With the rise of YouTube, Amazon, Netflix, and many other Internet services over the past few decades, recommendation systems have become increasingly important in our lives. From e-commerce (endorsing items buyers would like) to online advertising (suggesting the right content to customers based on their preferences), recommendation systems are now an inevitable part of our daily commutes.

Collaborative filtering doesn't require features about the items or users to be known. It is suited for a set of different types of items, for example, a supermarket's inventory where items of various categories can be added. In a set of similar items such as that of a bookstore, though, known features like writers and genres can be useful and might benefit from content-based or hybrid approaches.

Collaborative filtering can help recommenders to not overspecialize in a user's profile and recommend items that are completely different from what they have seen before. If you want your recommender to not suggest a pair of sneakers to someone who just bought another similar pair of sneakers, then try to add collaborative filtering to your recommender spell. In general, recommender systems are algorithms designed to suggest relevant items to users (items like movies to watch, text to read, products to buy, or anything else depending on the industry). Recommender systems are very important in some industries because they can generate a lot of revenue if they are efficient or can also differentiate them significantly from the competition.

Objectives of this project are:

- Design a platform that supports cross-content recommendations, thereby addressing the problem of cold start and scarcity. SVD and KNN can be used to overcome this problem.
- This system provides a transparent interface, reducing the frequency of searching for a specific movie/music.
- Finding and getting recommendations is faster and easier.

II. Related Work

In this paper **Error! Reference source not found.**, they have made a data-driven recommendation system that delivers recommendations to customers, such as books, films, etc. Most movie recommendation systems normally base user preferences on similar films. Effort recommendation systems are particularly useful for large customer database collecting organizations. A lot of things could go into a movie recommendation design like the director of the movie, the film genre, or the actors appearing in it. There are two or three characteristics that the systems use to make a recommendation. It has been based on types of content that the user prefers to watch. The distribution method adopted to implement this is focused on genre-based filtering. In this case, the dataset that was used for the project is Movie Lens. Python is the research software used.

Proposed a review on HI2Rec **Error! Reference source not found.**, which integrates multiple information to learn the user's and item's vector representations for top-N recommendation to address the above-mentioned issues. We extract the movie-related information from the Linked Open Data and then leverage the knowledge representation learning approach to embed this information as well as real-world datasets' information of recommender systems to a unified vector space. These vector representations are further calculated to generate a preliminary recommendation list. Finally, we utilize a collaborative filter approach to generate a precision recommendation list.

Proposed **Error! Reference source not found.** an idea of multi-criteria recommender systems (MCRSs) have been developed to improve the accuracy of the RS performance. Additionally, a new source of information represented by the user-generated reviews is incorporated in the recommendation process because of the rich and numerous information included (i.e. review elements) related to the whole item or to a certain feature of the item or the user's preferences. The valuable review elements are extracted using either text mining or sentiment analysis. MCRSs benefit from the review elements of the user-generated reviews in building their criteria forming multi-criteria review based recommender systems.

They **Error! Reference source not found.** represented the overview of Approaches and techniques generated in the Collaborative Filtering based recommendation system **Error! Reference source not found.**. The recommendation system derived into Collaborative Filtering, Content-based, and hybrid-based approaches. This paper classifies collaborative filtering using various approaches like matrix factorization, user-based recommendation, item-based recommendation. This survey also tells the road map for research in this area.

The authors proposed **Error! Reference source not found.Error! Reference source not found.** three novel methods such as collaborative filtering, and artificial neural networks and at last support vector machine to resolve CCS as well ICS problems. Based on the specific deep neural network SADE we can be able to remove the characteristics of products. By using sequential active of users and product characteristics we have the capability to adapt the cold start product ratings with the applications of the state-of-the-art CF model, time SVD++. The proposed system consists of Netflix rating dataset which is used to perform the baseline techniques for rating prediction of cold start items.

Here, they**Error! Reference source not found.** have used the well-known approach named as Collaborative filtering (CF). There are two types of problems mainly available with collaborative filtering. They are complete cold start (CCS) problem and incomplete cold start (ICS) problem. The authors proposed three novel methods such as collaborative filtering, and artificial neural networks and at last support vector machine to resolve CCS as well ICS problems. Based on the specific deep neural network SADE we can be able to remove the characteristics of products. By using sequential active of users and product characteristics we have the capability to adapt the cold start product ratings with the applications of the state-of-the-art CF model, time SVD++. The proposed system consists of Netflix rating dataset which is used to perform the baseline techniques for rating prediction of cold start items.**Error! Reference source not found.** The calculation of two proposed recommendation techniques is compared on ICS items, and it is proved that it will be adaptable method. The proposed method can be able to transfer the products since cold start transfers to non-cold start status. Artificial Neural Network (ANN) is employed here to extract the item content features.

Nowadays, the recommendation system has made finding the things easy that we need. Movie recommendation systems aim at helping movie enthusiasts by suggesting what movie to watch without having to go through the long process of choosing from a large set of movies which group to thousands and millions that is time consuming and confusing. In this article,**Error! Reference source not found.** their aim is to reduce the human effort by suggesting movies based on the user's interests.**Error! Reference source not found.** To handle such problems, we introduced a model combining both content-based and collaborative approach. It will give progressively explicit outcomes compared to different systems that are based on content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box, thus limiting your choice to explore more. Hence, we have focused on a system that resolves these issues.

Rapid development of mobile devices and internet has made possible for us to access different music resources freely. While the Music industry may favour certain types of music more than others, it is important to understand that there isn't a single human culture on earth that has existed without music. In this paper,**Error! Reference source not found.** they have designed, implemented and analysed a song recommendation system. We have used Song Dataset provided to find correlations between users and songs and to learn from the previous listening history of users to provide recommendations for songs which users would prefer to listen most. The dataset contains over ten thousand songs and listeners are recommended the best available songs based on the mood, genre, artist and top charts of that year. With an interactive UI we show the listener the top songs that were played the most and top charts of the year. Listener also have the option to select his/her favourite artist and genres on which songs are recommended to them using the dataset.

III. Proposed System

Data was extracted in CSV format from the IMDB database. he first sets of data we used for visualization creation and the second set of data we used for model creation & training. This system provides a transparent interface, reducing the frequency of searching for a specific movie/music. These CSV files contain: Movie ratings, Movie titles, Actor names, & production companies, Genres and User ratings. Whereas the CSV used to model our Music module is of Spotify which contains Song Title, Genres, Album Name, Artists, and User data.

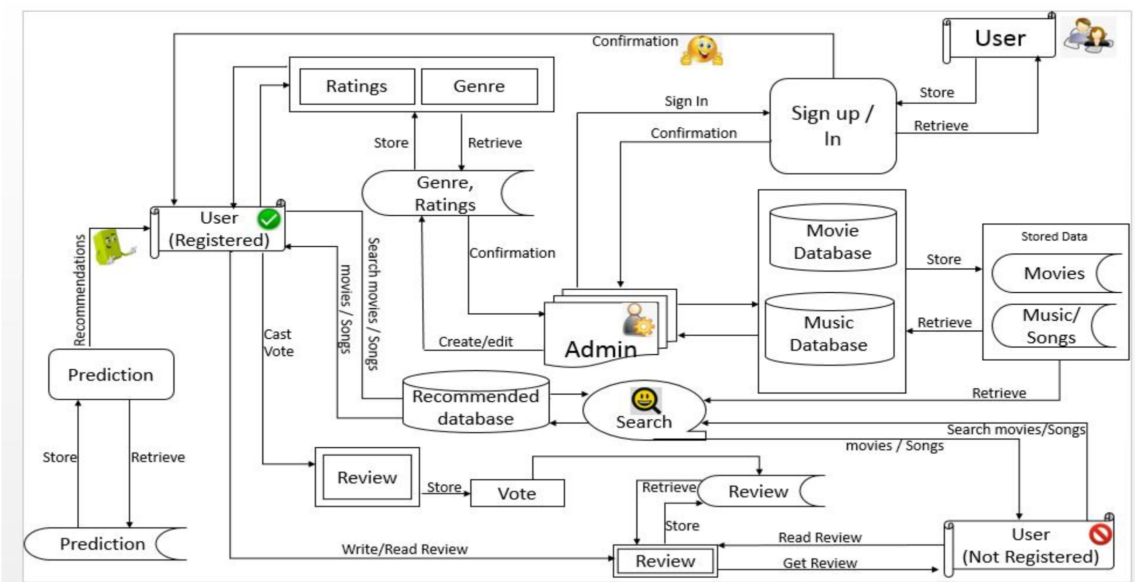

Fig. 1 Proposed System Architecture

Our project consists of three main components – users, recommender system and web application. The general architecture of our system is shown in Fig. 1, which shows the relationship between the various components of our system and how they interact with each other. End users (external component) interact with our system via web interface. The
web interface allows users to search and browse movies and music, read reviews, write reviews, and also get personalized movie and music recommendations.

All information about movies, music, users, reviews and genres are stored in the database. Each time a user submits a rating, it is stored in the ratings database, which assigns it a predicted rating. The recommender system uses the information from users, movies, music and ratings to learn a model to predict user ratings and properties of movies, music, used to create a list of recommended movies and music for all system users. Operating the web is very simple. The website will display a list of movies and music available in the database.

A user can browse them and search for the details of the movie or music he is interested in. For convenience, one can search the film using the search bar provided in the web application.

In order to write reviews and receive recommendations, the user must register on the web application by filling out the registration form. The user needs to fill out some general details for registration.

Once registered, the user can write reviews of the films he has seen. The system analyses the summary part of the rating and automatically converts it into ratings and based on that gives a personalized recommendation to the user. In order to generate good personalized recommendations, the user has to write reviews on a large number of films.

Model accuracy, RMSE and MAE scores will be the main criteria used to evaluate model performance. The proposed system will be subjected to a series of tests including: Dataset: data aspects will be retrieved using the Spotify API for music and the MovieLens25M dataset for movies. We use about 10,000 ratings in our system, with the same amount of user data and actual movies. The data is recorded in CSV (Comma Separated Values) format. This dataset is halved to train and test 80:20 algorithms.

Data Cleansing: A data set is a collection of raw data. It can contain symbols such as numbers, special characters, blank lines, and untagged data. These icons should be removed as they are not important and may affect model performance.

Data pre-processing: After data cleansing, the data is now free of unwanted symbols. This data needs to be converted into the form that can be used to easily extract the features. It involves the following process where the data has been cleaned and is now free of unnecessary symbols. This information needs to be converted into a format that can be easily used to extract features. It includes the following steps:

- To eliminate case ambiguity, each word is converted to lowercase.
- Removed single-letter words.
- Words with digits are deleted.
- Remove punctuation and tokenize data
- Empty tokens are removed.
- Stop words: Stop words are useless words for NLP like “the”, “a”, “an”, “in”. This should be removed.

Lemmatization: The process of converting words to their root forms, such as B. changing the course and studying in one course. However, this has the potential to change the meaning of the words. So, to keep it, to keep the meaning of a token, add a part of speech tag to it.

Feature Selection: - The method of choosing the functions that make contributions the maximum in your prediction variable or output is called feature selection. Certain characteristics of fake information should be extracted, and our classifier should then be taught to anticipate the information. The vital words that seem withinside the information is highlighted here. Classification: - Classifiers are algorithms that classify input relying on the functions it contains. The first classifier needs to learn on characteristics that will appear in a subsequent class. For this analysis, the purpose is to optimize test precision. This is due to the fact we do now no longer need any fake positives in our version and need to make certain that ratings are classified correctly. Additionally, the model is much less sensitive to false negatives, as a wrong rating of two if the actual rating is 3, will now no longer have a significant impact on the interpretability of our model. Therefore, further tune XGBoost and SVM models as those models had the best test precision rankings withinside the vanilla models. SVM is a type set of rules that makes use of supervised gadget getting to know. To separate the classes, the features of each class are transferred to the graph, and a most advantageous aircraft called a hyperplane is drawn between them. This plane is created on the idea of support vectors, one for every class, every of that is closest to a feature point.

IV. METHODOLOGY

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
IJARSCT
Impact Factor: 6.252
Volume 2, Issue 5, May 2022

Movie Module

Our recommendation system works by suggesting movies to the user based on the meta-data information. The similarities between the movies are calculated and then used to make recommendations. In this implementation, when the user searches for a movie the system recommends the top 10 similar movies using our recommendation system. We used a collaborative filtering algorithm for our purpose. The dataset used in our project is from movielens dataset which is of a moderate size.

Music Module

Our recommendation system works by presenting the user with music based on meta-data. The music similarities are calculated and then used to make recommendations. In this approach, when a user searches for music, the system uses our recommendation system to suggest the top 10 comparable songs. For this project, we used a collaborative filtering technique. Instead of a dataset, the Spotify API was used.

Algorithmic Implementation

Collaborative Filtering: Nearest Neighbours- Filters information by using the interactions and data collected by the system from other users. People who agreed in their evaluation of certain movies are likely to agree again in the future. There are two types of collaborative filtering approaches: User-based and Item-based collaborative filtering. User Preference is usually expressed by two categories. Explicit Rating and Implicit Rating.

KNN-

- Each column represents unique userId and each row represents each unique movieId
- Removing Noise from the data
  - To qualify a movie, a minimum of 10 users should have voted a movie.
  - To qualify a user, a minimum of 50 movies should have voted by the user.
- Removing sparsity
  - Applied csr_matrix method to the dataset
- Used KNN algorithm to compute similarity with cosine distance which is very fast and preferred over pearson coefficient. Found similar movies and sort them based on their distance from the input movie

SVD-

Singular Value Decomposition (SVD) is a widely used technique to decompose a matrix into several component matrices, exposing many of the useful and interesting properties of the original matrix. Matrix factorization is a class of collaborative filtering algorithms used in recommender systems. Matrix factorization algorithms work by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices.

The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into 3 matrices. It has a few interesting algebraic properties and conveys crucial geometrical and theoretical insights about linear transformations. The rank of matrix M may be calculated from SVD with the aid of using the wide variety of nonzero singular values. The range of matrix M is the left singular vectors of U corresponding to the non-0 singular values. The null space of matrix M is the right singular vectors of V corresponding to the zeroed singular values.

Curve Fitting Problem: Singular value decomposition may be used to decrease the least square error. It makes use of the pseudo inverse to approximate it. Besides the above application, singular value decomposition and pseudo-inverse also can be utilized in Digital sign processing and picture processing.

Steps involved are:

1. Split Data into Training and Testing sets
2. Fit the model using SVD() and baseline parameters
3. Calculated RMSE and MAE
4. Optimized Model (nested for loop)
5. Refit the Model & got predictions
6. Calculated new RMSE and MAE
7. Used predictions from SVD MF as an input into XGBoost Model

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
IJARSCT
Impact Factor: 6.252
Volume 2, Issue 5, May 2022

V. Results and Discussion

We explored other models beyond KNN, XGBoost, SVD MF that could result in more accurate predictions. Some chunks took very long to run, optimize based on time.

Model Optimization: SVD MF

The starting RMSE for SVD MF was 0.65, and MAE is 0.50. With optimization we were able to reduce it RMSE to 0.28 and MAE to 0.18.

Model Optimization: XGBoost

The starting RMSE for XGBoost was 0.68, and MAPE is 20.6. With optimization we were able to reduce it RMSE to 0.63 and MAPE to 18.7.

GUI Snapshots:

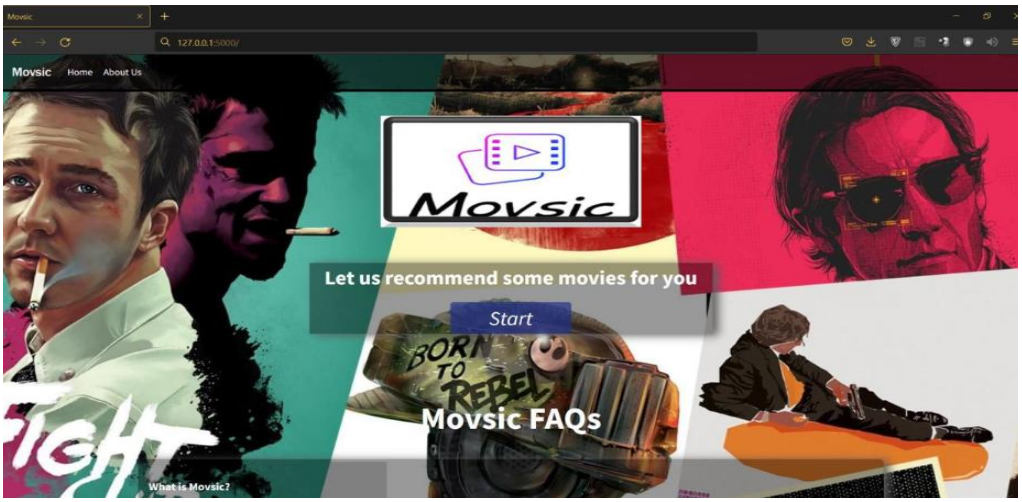

Fig. 02 GUI Snapshot 1-Movsic start page

In Fig. 02, this page allows user to start answering questionnaire for different preferences of his/her choices.

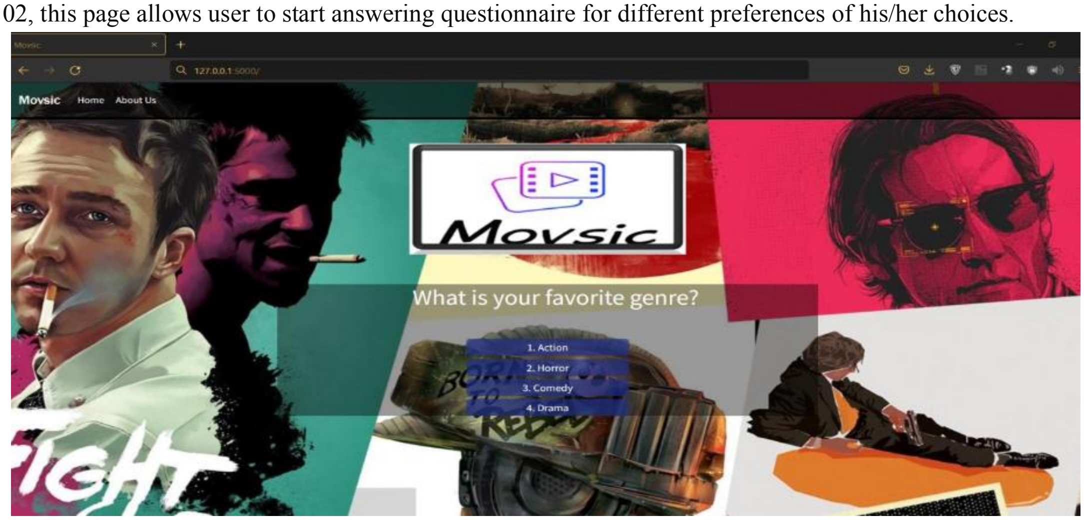

Fig. 03 GUI snapshot 2-Preference Page

In Fig. 03, this webpage is executed when the file is clicked. This webpage shows the output that includes a background image, home page button, about us button and it also displays a logo of our system. It asks the user to give an input of his/her favourite genre that includes action, horror, comedy and drama.

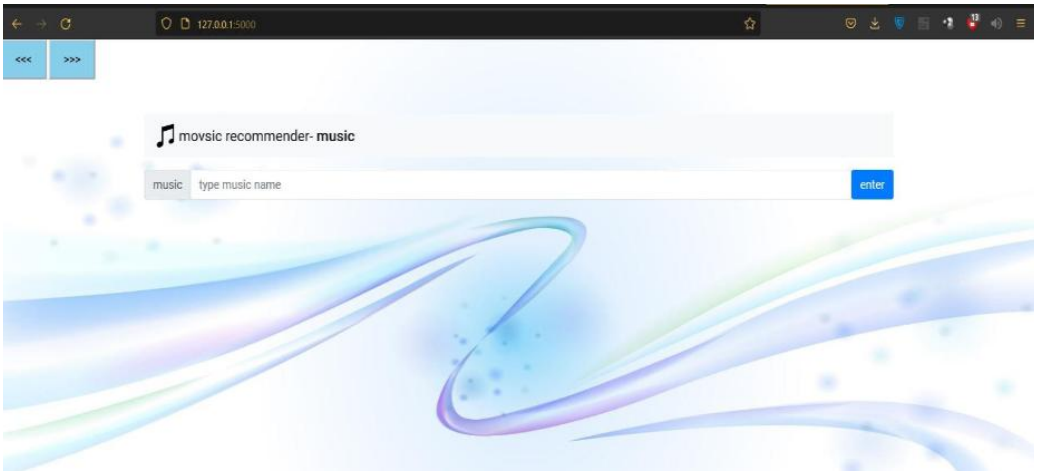

Fig. 04 GUI snapshot 3-Movsic Recommender- Music

In Fig. 04, this webpage is displayed when the movies are recommended to the user according to their type of choices like genre, director and partner preferences. Hence this webpage displays and takes input for the music recommendation that the user is given to input.

This recommendation system is a totally SVD algorithm-based application. It allows the user to get recommendations based on the search results that the user has to input. The application runs on any device, i.e. Chrome, Mozilla, internet explorer etc. and not just on desktop or laptop but also on mobile. All the features incorporated work perfectly fine on the deployment devices with no system crashes if all the libraries and soft wares are installed properly and according to the guidelines. This was run on a local host.

VI. FUTURE WORK

The goal is to enhance the proposed ML model which as a future enhancement in unsupervised getting to know version, it wishes to be seemed constantly to replace the recommendation system model as extra user statistics is collected. While the classification model addresses the cold start problem, the XGBoost model itself nevertheless has an excessive RMSE. Future work may want to have a take a observe which include extra features which can affect a song's popularity. This may be accomplished with the aid of using merging statistics from the Spotify API with data from different song-related APIs, just like the genius API. Including facts from a new API may also assist to enhance the SVD model's matrix sparsity, with the aid of using which include additional information as a proxy for artist ID, along with songwriter IDs and producer IDs. Lastly, the functions of the version may be implemented into LightFM to look if those factors may be covered withinside the recommendation system. Moreover, to enhance the proposed model's type performance, extra data times could be collected, for use for getting to know the type guidelines efficiently. In the destiny, we will put into effect more sophisticated hybrid filtering recommendation models wherein suggestions to the consumer could be furnished in line with their mood.

The proposed recommendation system works on songs and movies in English language. This system can be further extended to recommend songs and movies in Hindi language or other native languages since the users can better express themselves using their native languages. The emotion category can be further improved to consider more complex emotions as hatred, anxiety, jealousy, excitement, etc. And thirdly, a dashboard that will give insights about the movie, its ratings, its category, viewers' liking and disliking will be a part of future scope.

VII. CONCLUSION

Our post-modelling EDA suggests as an alternative unique results than our pre-modelling EDA; that is because of the probabilistic nature of the scores, permitting our model to provide extra specific ratings, and in the end offer extra accurate recommendations. Overall, the model performs very well with a small RMSE of 0.25; we see from the distribution of our errors that the errors are typically distributed. With the combined classification model and the SVD model, the film and song module can't only offer recommendations for current users, however we will avoid the cold-start problem and generate recommendations for brand new users as well. A platform, along with a website, may be built and linked to a trained Machine Learning model. The dataset is trained using SVD techniques and used to anticipate the users input. Movie and Music-Movsic recommendation system will reduce human efforts by searching the huge media collection containing many songs and movies of various genres. Recommendations to the user will be provided according to their mood and choice. Hence this system will provide higher user satisfaction in less time and efforts as they will be automatically provided a recommendation for movies and music.

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr. Nalini Mhetre for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project. We would like to express our gratitude towards Prof M. P. Wankhede, H.O.D. of Computer Engineering, SCOE and Dr. S. D. Lokhende, Principal, SCOE for their kind co-operation and encouragement which helped us in completion of this project.

We would like to express our special gratitude and thanks to all our colleagues for giving us such attention and time. Our thanks and appreciations also go to our friends in developing the project and people who have willingly helped us out with their abilities.

REFERENCES

[1]. Content-Based Movie Recommendation System Using Genre Correlation, International Journal of Research in Science and Communication Vol. 1 Issue 02, Feb-2021

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

[3]. Al-Ghuribi, S.M. and Noah, S.A.M., 2019. Multi-criteria review-based recommender system–the state of the art. IEEE Access, 7, pp.169446-169468.

[4]. Raval, N. and Khedkar, V., 2019. A review paper on Collaborative filtering-based movie recommendation system. Int. J. Sci. Technol. Res, 8, p.12

[5]. Chen, J., Zhao, C. and Chen, L., 2020. Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. Complex & Intelligent Systems, 6(1), pp.147-156.

[6]. Furtado, F. and Singh, A., 2020. Movie recommendation system using machine learning. International journal of research in industrial engineering, 9(1), pp.84-98.

[7]. Kathavate, S., 2021. Music Recommendation System using Content and Collaborative Filtering Methods. International Journal of Engineering Research & Technology (IJERT), 10(02), pp.167-171.

[8]. Rentfrow, Peter J., and Samuel D. Gosling. "The do re mi's of everyday life: the structure and personality correlates of music preferences." Journal of personality and social psychology 84.6 (2003): 1236.

[9]. Celma Herrada, Òscar. Music recommendation and discovery in the long tail. Universitat Pompeu Fabra, 2009.

[10]. Ricci, Francesco, Lior Rokach, and Bracha Shapira. "Introduction to recommender systems handbook." Recommender systems handbook. springer US, 2011. 1-35.

[11]. Lamere, Paul. "Social tagging and music information retrieval." Journal of new music research 37.2 (2008): 101114.