**SAVITRIBAI PHULE PUNEUNIVERSITY**



**A MINI PROJECT REPORT ON**

# Movie Recommendation System

Submitted by

### Name : Manasi.Renuse Roll no :25

**CLASS: TE DIV:B**

### Under the Guidance of

Ms. Pradnya Kasture



**DEPARTMENT OF COMPUTER ENGINEERING RMD SINHGAD SCHOOL OF ENGINEERING** WARJE, PUNE 411058

**2023 - 24**



**DEPARTMENT OF COMPUTER ENGINEERING RMD SINHGAD SCHOOL OF ENGINEERING** WARJE, PUNE 411058

# CERTIFICATE

This is to certify that the project report entitles

**Movie Recommendation System**

*Submitted by*

Name: Manasi.Prakash.Renuse PRN No : 72217955D

is a bonafide work carried out by them under the supervision of Ms. Pradnya Kasture and it is submitted towards the partial fulfillment of the requirement of University of Pune for Third Year.

**(Ms. Pradnya Kasture ) (Dr. Vina M. Lomte)**

Guide Head,

Department of Computer Engineering Department of Computer Engineering

Pune

Date: 16/4/24

**(Dr. V. V. Dixit)**

Principal,

RMD Sinhgad School of Engineering Pune – 58 Place:

II

### Certificate by Guide

This is to certify that Ms. Manasi.Renuse has completed the MINI Project work under my guidance and supervision and that, I have verified the work for its originality in documentation, problem statement, implementation and results presented in the Project. Any reproduction of other necessary work is with the prior permission and has given due ownership and included in the references.

Place: Pune Date: 14/4/2024

Signature of Guide

**(Ms. Pradnya Kasture )**

## ACKNOWLEDGEMENT

It is our pleasure to acknowledge sense of gratitude to all those who helped us in making this seminar. We thank our Mini Project Guide **Ms. Pradnya Kasture** for helping us and providing all necessary information regarding our project. We are also thankful to **Dr. Vina M. Lomte (Head - Department of Computer Engineering)** for providing us the required facilities and helping us while carrying out this seminar work.

Finally we wish to thank all our teachers and friends for their constructive comments, suggestions and criticism and all those directly or indirectly helped us in completing this seminar.

**NAME OF THE STUDENTS**

**Manasi.Renuse**

## ABSTRACT

Movie recommender systems are meant to give suggestions to the users based on the features they love the most. A highly performing movie recommendation will suggest movies that match the similarities with the highest degree of performance. This study conducts a systematic literature review on movie recommender systems. It highlights the filtering criteria in the recommender systems, algorithms implemented in movie recommender systems, the performance measurement criteria, the challenges in implementation, and recommendations for future research. Some of the most popular machine learning algorithms used in movie recommender systems such as *K*-means clustering, principal component analysis, and self-organizing maps with principal component analysis are discussed in detail. Special emphasis is given to research works performed using metaheuristic-based recommendation systems. The research aims to bring to light the advances made in developing the movie recommender systems, and what needs to be performed to reduce the current challenges in implementing the feasible solutions. The article will be helpful to researchers in the broad area of recommender systems as well as practicing data scientists involved in the implementation of such systems.

## CONTENTS

**I Certificate**

1. **Certificate by Guide**
2. **Acknowledgement**
3. **Abstract**
   1. **Introduction** **page no 1**
   2. **Technology Used** **page no 2**
   3. **Filter** **page no 3**
   4. **Algorithm** **page no 4**
   5. **Implementation…………………………………………………………..page no 7**
   6. **Output…** **page no 8**
   7. **Conclusion** **page no 9**

# INTRODUCTION

In the contemporary digital era, the abundance of movies and streaming platforms has pro- vided users with an unparalleled array of entertainment options. However, the sheer volume of available content poses a challenge for users seeking personalized recommendations that align with their individual tastes and preferences. The Movie Recommendation System project ad- dresses this challenge by leveraging advanced machine learning and deep learning techniques to offer tailored movie suggestions.

The primary objective of this project is to enhance the user experience by providing an in- telligent and efficient recommendation system. Traditional methods of browsing and searching for movies often lead to decision fatigue, and users may miss out on hidden gems that match their interests. The recommendation system presented here seeks to alleviate this issue by analyzing user behavior, preferences, and demographic information to generate accurate and personalized movie recommendations.

The project employs collaborative filtering techniques, including both user-based and item- based approaches, to identify similarities between users and movies. Additionally, deep learn- ing models are integrated to capture intricate patterns in user behavior that may not be apparent through traditional methods. This hybrid approach ensures a robust and accurate recommendation system that adapts to individual user preferences.

# TECHNOLOGY USED

1. **Machine Learning Library**

The engine of the recommendation system filters the data via different machine learning al- gorithms, and based on that filtering, it can predicts the most relevant entities to be recom- mended. After studying the previous behaviours of the users, it recommends

* + Machine Learning Library:

1. pandas
2. numpy
3. Difflib
4. AST
5. scikit-learn

pandas numpy difflib AST scikit-learn

1. **Requirement**

Python 3.6

# FILTER

1. **Collaborative Filtering**

Collaborative filtering works by matching the similarities in items and users. It looks at the characteristics of the users and the characteristics of the items the users have watched or searched for before . In general, latent features obtained from rating matrices are looked at. In movie recommender systems, the recommendations are made based on the user information and what other people with similar user information are watching. For example, collaborative filtering in movie recommender systems picks the user demographic characteristics such as age, gender, and ethnicity . Through these features, movie recommendations are made that match other people with similar demographic characteristics and previous user search history. Collaborative filtering suffers from a cold start if the user has not input any information, or the information is too little for any accurate clustering. In these cases, it does not know what to suggest . The accuracy of the suggestion is also limited because people with similar demographic characteristics may not have similar preferences .

1. **Context-Based Filtering**

This filtering technology is an improvement of the collaborative filtering method. It assumes that if person A and person B hold the same opinion on issue X, it is most likely that the same people will hold the same opinion/preference/thinking on a different issue Z. For example, if both people are attracted to Christmas movies from Netflix, it is most likely that they will still like Christmas movies by Showmax. The context-based filtering method recommends items with similar features or characteristics because the applications have just been extended to a different context . It makes the same suggestions though the contexts are different. In most cases, web browsers import bookmarks and other settings when one upgrades from one browser to the next. This represents a change in context, since most of the settings and other items are imported into the new context, and the data available are used in making useful suggestions. Similarly, movie recommender systems may make a similar recommendation based on data from the previous context . It is worth mentioning here about context-aware recommender systems (CARS), where the concept of context is well defined. CARS acclimatize to the exact condition in which the recommended item will be used . In this respect, CARS could avoid recommending a very long film to a user after a stressful day at work or suggest a romantic film if he/she is in the company of his/her partner.

**ALGORITHM**

These are the algorithms that are used in filtering information and data mining so that the desired outcomes can be achieved. It is essential to understand the working of the information filtering methods so that the right algorithm is selected for the specific task in recommender systems .

1. **K-Means Clustering**

This is one of the simplest collaborative filtering approaches that categorizes the users based on their interests . It is common for someone who wants to purchase an item to ask someone who has already purchased the product for their opinion. There is a higher chance that the influence of the current owner will affect the preferences and the tastes of the potentially new owner. Similarly, the algorithm compares the interesting features that can be associated with individuals that are classified to be within a group.

K-means clustering uses interests that are common among the users such as age, gender, movie time, history of the previous movies watched, etc. K-means clustering aims to group the features into clusters that represent the characteristics of the group . If the classification is based on age, the probable Kmeans clustering will use children, teens, youth, and adult clustering methods. If a client falls within any of these age groups, movies are recommended based on what other people within that age group do. If the clustering depends on age, the closer an age is to the centroid age, the better the classification recommendation. The steps in the classification are measuring the similarity between the user and item features, selection of the neighbours, computing the prediction, and suggesting it .

1. **Measurement of the Similarities**

The first step is finding the similarity in the user features that the new user has with the previous system users. The algorithm always has the basic classifications for a beginning, where the user can give inputs and the predictions can be made . Common features in finding the similarities are age, previous history, and geographical locations. Other recommender systems in movie theatres, including the price, the time to watch the movies, etc., are used in coming up with the means (centroids) for clustering. The distance from the centroids can be based on a Pearson correlation, cosine-based similarities, or an adjustment of the cosine-based similarity. The calculation of the similarity may be item-based or user-based.

Itembased computation finds the similarities based on the features in the movies that similar people liked. If it is user-based, the calculation of the centroids is based on the demographic features of the user

.

The computation of the similarities between items or users is shown in the mathematical equations below:

simi,j=∑m (i∩i)(ri,m−ri¯¯¯)(rj,m−rj¯¯¯)∑m (i∩i)(ri,m−ri¯¯¯)2−−−−−−−−−−−−−−−−√×∑m (i∩i)(rj, m−rj¯¯¯)2−−−−−−−−−−−−−−−−√

The equation above computes the correlation between the user and the item; it computes the closeness of the value to the centroid value. It is assumed that the two items i∩j are the correlated features (items or users); the value rj¯¯¯ is the centroid feature, while the value ri is the value of the new user or new feature to be compared through correlation .

1. **Selection of the Neighbours**

There is always a consideration when developing the algorithm. The key metrics are the accuracy to obtain and the running time of the algorithm. To increase the accuracy of an algorithm, a large number of neighbours, which increases the computational time of the algorithm, is required. If a smaller computational time is needed, accuracy will be compromised . To strike a balance, the selection may be threshold-based or use the top-N technique. The threshold technique will run only a specific number (sample number that meets the threshold value) of assessments of the neighbours and predict if that threshold is reached. For example, if the population is 1000, the system will run a prediction from 100 samples and predict out of the 100 samples . In the top-N technique, only the top number of similarities

(N) is run rather than the whole population of neighbours. For example, it will select only the top 10 for suggestions based on the nearest neighbours rather than assessing the whole population .

1. **Prediction Computation**

The computation of the subsequent predictions is based on the closest neighbours found in the system database. The prediction is obtained by the formula below:

predictionu, i=∑n Neighbors(rn,i−rn¯¯¯) simu,n∑n Neighbors | simu,n |+ru¯¯¯

The prediction or the nearest neighbour to the centroid (K-mean) is made. In the equation above, the Kmeans is represented by ru¯¯¯ while the correlation of the other variable on the right-hand side of the equation gives the nearest neighbour, both used in making the suggestion prediction.

# IMPLEMENTATION



**OUTPUT**



**CONCLUSION**

Movie recommender systems have been described and classified. The various types of recommender systems are introduced and discussed. Special emphasis is given to explain in detail the various machine learning and metaheuristic algorithms commonly deployed in movie recommendation research. The various model metrics that summarize the quality of the model are discussed at length. The problems associated with movie recommender systems are also summarized in a structured way and discussed. A total of 77 articles strictly on the area of movie recommender systems are included in the study, and their major conclusions are presented. In addition, 32 other related articles on metaheuristics and recommender systems (not for movies) are also introduced in various sections to present a coherent and meaningful review. One of the limitations of the study is that the Scopus and Web of Science databases were not directly used for selecting the articles for review. In contrast, EBSCO Academic Search Premier, ScienceDirect, IEEE Library, ResearchGate, SpringerLink and the ACM Portal were used for the literature search. Nevertheless, more than 80% of the reviewed papers were found to be indexed in Scopus while more than 60% were available in the Web of Science database.