

MOVIE RECOMMENDATION SYSTEM USING CONTENT BASED AND COLLABORATIVE BASED FILTERING

Submitted by

CH.EN.U4CSE19103

CH.EN.U4CSE19148

CH.EN.U4CSE19154

CH.EN.U4CSE19155

G. AMRUTHA

A.S.L. MANASA

BIRAT CHAPAGAIN

ROHIT KUMAR SAH

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



AMRITA SCHOOL OF COMPUTING, CHENNAI

AMRITA VISHWA VIDYAPEETHAM

CHENNAI 601103

MAY – 2023

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI – 601 103



BONAFIDE CERTIFICATE

This is to certify that the project report entitled “**Movie Recommendation system using Content based and Collaborative based Filtering**” submitted by

CH.EN.U4CSE19103
CH.EN.U4CSE19148
CH.EN.U4CSE19154
CH.EN.U4CSE19155

G. AMRUTHA
A.S.L. MANASA
BIRAT CHAPAGAIN
ROHIT KUMAR SAH

in partial fulfillment of the requirements as part of **Bachelor of Technology** in “**COMPUTER SCIENCE AND ENGINEERING**” is a bonafide record of the work conducted under my guidance and supervision at Amrita School of Computing, Chennai.

Dr. A. Padmavathi

Thesis Advisor
Assistant Professor (SL)
Department of CSE

Dr. S. Sountharajan

Program Head
Associate Professor
Department of CSE

This project report was evaluated by us on :

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I the undersigned solemnly declare that the thesis **MOVIE RECOMMENDATION SYSTEM USING CONTENT BASED AND COLLABORATIVE BASED FILTERING** is based on my own work conducted during our study under the supervision of Dr. A Padmavathi, Assistant Professor, Computer Science & Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement has been made wherever the findings of others have been cited.

Signature of the Student

- 1) G. Amrutha
- 2) A.S.L. Manasa
- 3) Birat Chapagain
- 4) Rohit Kumar Sah



ACKNOWLEDGEMENT

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people who made it possible, and whose constant encouragement and guidance have been a source of inspiration throughout the course of this project work.

We offer our sincere pranams at the lotus feet of **"AMMA," MATA AMRITANANDAMAYI DEVI** who showered her blessing upon us throughout the course of this project work.

We owe our gratitude to **Dr. V. Jayakumar**, Principal, Amrita School of Computing, Chennai.

We thank **Dr. S. Sountharajan**, Program Head-CSE, Amrita School of Computing, Chennai for his support and inspiration.

It is an immense pleasure to express our gratitude and indebtedness to our project guide Dr. A Padmavathi, Department of Computer Science and Engineering, Amrita School of Computing, Chennai for her valuable guidance, encouragement, moral support, and affection throughout the project work.

Accept my endless gratitude to my Project Coordinator Dr. B S Vidhyasagar Department of Computer Science and Engineering, Amrita School of Computing, Chennai, for the constant source of inspiration.

We would like to express our gratitude to project panel members for their suggestions, encouragement, and moral support during the process of project work and all faculty members for their academic support. Finally, we are forever grateful to our parents, who have loved, supported, and encouraged us in all our endeavors.

G. AMRUTHA
A.S.L. MANASA
BIRAT CHAPAGAIN
ROHIT KUMAR SAH

ABSTRACT

A new movie recommendation system has been developed that utilizes the analysis of movie descriptions and various other features to provide personalized recommendations to users. This system uses cast, director, country, rating, and genre as additional features to enhance the recommendations. To generate these recommendations, the system employs two algorithms: XGBoost with Surprise Baseline Predictor and KNN Baseline Algorithm. These algorithms leverage movie-movie similarities to recommend films that users are most likely to enjoy. One of the key features of the system is its analysis of movie descriptions. By analyzing movie descriptions, the system can determine key themes and elements of a movie that are likely to appeal to users. This approach has yielded excellent results, with the system's movie recommendation accuracy outperforming other systems that do not take movie descriptions into account. Moreover, the system's use of additional features such as cast, director, country, rating, and genre further improves its ability to provide personalized recommendations. This means that users can expect to receive recommendations that are highly targeted to their individual preferences. Overall, this movie recommendation system represents a notable advancement in the field of personalized recommendation systems. By utilizing movie descriptions and a range of other features, it provides highly accurate and personalized recommendations to users, improving their overall movie watching experience.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION	9
1.1 MACHINE LEARNING MODELS	11
1.1.1 Singular Value Decomposition	11
1.1.2 XGBoost (Extreme Gradient Boosting)	11
1.1.3 KNN Model (K Nearest Neighbors)	11
1.1.3.1 KNN Baseline Predictor	12
1.1.3.2 Baseline Predictor	12
1.1.4 Matrix Factorization Technique	12
1.1.4.1 SVD Matrix Factorization User Movie interactions	13
1.1.4.2 SVD Plus Plus (Implicit Feedback)	13
1.1.5 Description Analysis	13
CHAPTER 2 LITERATURE SURVEY	14
CHAPTER 3 SYSTEM SPECIFICATIONS	42
3.1 FIREBASE AUTHENTICATION	42
3.2 FIRESTORE DATABASE	42
3.3 FIREBASE STORAGE	43
3.4 PYTHON FLASK	43
3.5 PICKLE	44
3.6 VERCEL	44
3.7 REACT QUERY	45
3.8 CHAKRA-UI	45
3.9 NUMPY	46
3.10 SCIPY	46
CHAPTER 4 SYSTEM DESIGN & IMPLEMENTATION	47
4.1 FRONTEND:	47
4.2 BACKEND:	48
4.3 DATASET:	48
4.4 PRE-PROCESSING DATA	49
4.4.1 Extending the Dataset	49
4.4.2 Sampling Data	49

4.4.3	Finding Global Averages	50
4.4.4	Featurizing of Train & Test Data	50
4.4.5	Transforming of Train & Test Data	51
4.5	MACHINE LEARNING MODELS	51
4.5.1	XGBoost with thirteen features	51
4.5.2	Baseline model	51
4.5.3	KNN Baseline predictor	52
4.5.4	Surprise KNN Baseline with User-User similarities.	52
4.5.5	Surprise KNN Baseline with Movie-Movie similarities	52
4.5.6	Hybrid model of XGBoost with initial thirteen features and Surprise Baseline predictor and KNN Baseline predictor	53
4.5.7	Matrix Factorization	53
4.5.7.1	SVD User Movie Interaction.	53
4.5.7.2	SVD Matrix Factorization with implicit feedback from users.	54
4.5.8	Hybrid model of XGBoost with thirteen features and Surprise Baseline and Surprise KNN baseline and MF Techniques	54
4.5.9	Hybrid model of XGBoost with Surprise Baseline and Surprise KNN baseline and MF Techniques	54
4.5.10	Description & Review analysis	55
4.5.10.1	Genre, Cast, Director as features	55
CHAPTER 5	SYSTEM TESTING	56
5.1	EVALUATION METRICS	56
5.1.1	RMSE.	56
5.1.2	MAPE.	56
CHAPTER 6	RESULT AND INFERENCES	57
CHAPTER 7	CONCLUSION AND FUTURE ENHANCEMENTS	60
CHAPTER 8	REFERENCES	61
CHAPTER 9	PUBLICATIONS	63
9.1	ABSTRACT	63
9.2	ACCEPTANCE	64

LIST OF FIGURES

Figure 1: Workflow of the Proposed system	47
Figure 2: Performance comparison of models using RMSE.	57
Figure 3: Movies recommended with cast, director, country, rating, genre as features.	58
Figure 4: Movies recommended using Description analysis.	59
Figure 5: Glimpse of website showing recommendation.	59

CHAPTER 1

INTRODUCTION

Movie suggestion algorithms have gained significant popularity in recent times, due to the growing number of individuals who now rely on internet streaming platforms to fulfill their movie entertainment requirements. These services offer a vast library of movies that users can choose from, but with so many options available, it can be overwhelming to decide what to watch. K-nearest neighbors (KNN), XGBoost, matrix factorization, and support vector machines (SVM) are widely recognized machine learning algorithms commonly employed in the development of recommendation systems.

The K-Nearest Neighbors is a machine learning algorithm that is frequently employed in recommendation systems due to its ease of use and effectiveness. KNN involves finding the K most related items to a given item, based on shared characteristics such as genres, cast, and director. To measure this similarity, a distance metric such as Euclidean distance or cosine similarity is typically employed. Once the K most related items are identified, they can be recommended to the user. The simplicity of the KNN algorithm is one of its key strengths. It is easy to understand and apply, and it can work well even with small datasets. However, KNN can also have some limitations. For example, it can struggle with datasets that have many features, as the number of comparisons can become exceptionally large. Additionally, KNN can be sensitive to the choice of K, as different values can lead to different recommendations.

XGBoost is a sophisticated machine learning algorithm that has gained considerable attraction in recent times, thanks to its capacity to manage extensive datasets and intricate feature interdependencies. The acronym XGBoost stands for Xtreme Gradient Boosting, and it is a gradient boosting algorithm that relies on decision trees to establish connections between features and the target variable. In XGBoost, decision trees are built sequentially, with every new tree attempting to correct the errors made by the previous trees. This approach allows XGBoost to manage complex relationships between features, and it can often outperform other machine learning algorithms on large datasets. XGBoost also includes several optimization techniques, such as parallel processing and regularization, which help improve its performance.

Support Vector Machine is a prevalent machine learning algorithm that can be utilized in recommendation systems. SVM operates by finding the hyperplane that can best segregate the data points of various classes based on their features. The main goal is to find the hyperplane that maximizes the distance between the classes, as this is expected to provide the best possible separation between them. SVM can be useful in cases where the data is not linearly separable, as it can employ kernel functions to transform the data into a space of higher dimension, where it is more feasible to separate. Additionally, SVM is effective at managing datasets with many features, as it can pick the most pertinent features to use in the model.

Matrix factorization techniques like Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) have been widely used in movie recommendation systems. These methods can identify hidden patterns in the data that may not be immediately obvious and can make accurate recommendations even for movies that have not yet been rated by the user. The advantage of matrix factorization over other algorithms like KNN and SVM is that it can handle sparse and high-dimensional data more efficiently. Since the movie -user matrix is typically very sparse, matrix factorization can effectively reduce the dimensionality of the matrix and identify the underlying latent factors that influence user preferences. In the realm of movie recommendation systems, matrix factorization techniques can be employed to suggest movies to users, leveraging their past movie ratings, movie genres, and other attributes like cast, director, and country. Matrix factorization algorithms enable the system to uncover concealed patterns in the data, enabling it to offer precise recommendations that are tailored to each user.

In addition to these algorithms, movie recommendation systems can also use description analysis of movies as a feature. This involves analyzing the movie's plot, keywords, and other metadata to extract meaningful insights that can be used to suggest similar movies to users. Description analysis can be particularly useful for movies that have unique or complex themes, as it can help identify similar movies that may not have the same cast or director. Other features might include the movie's cast, director, country of origin, rating, and genres, among others. By analyzing these features, machine learning algorithms can identify patterns and similarities between different movies, allowing the system to recommend movies that users are likely to enjoy based on their preferences.

1.1 MACHINE LEARNING MODELS

1.1.1 Singular Value Decomposition

SVD is one of the useful tools in movie recommendation systems as it identifies latent features in the dataset and uses them to make predictions. The three matrices involved are the user and item feature matrices, and a singular value matrix that determines feature importance. By using SVD, movie recommendation models can make more accurate and personalized recommendations based on the specific preferences and characteristics of each user.

1.1.2 XGBoost (Extreme Gradient Boosting)

It is often used in various applications, including movie recommendation systems. XGBoost is an ensemble learning method that uses gradient boosting to combine decision trees and improve prediction accuracy. It iteratively trains weak learners to minimize a customizable loss function based on the specific problem. In the area of movie recommendation systems, XGBoost can be utilized to forecast a user's movie preferences based on historical ratings and other features. XGBoost can manage missing values, feature interactions, and other complexities in the data, making it a powerful tool for building accurate and efficient recommendation systems. By using XGBoost, movie recommendation models can make more personalized and relevant movie recommendations to users.

1.1.3 KNN Model (K Nearest Neighbors)

It is a popular algorithm for machine learning used in various applications, including movie recommendation systems. The KNN machine learning algorithm is a form of supervised machine learning that utilizes the class of neighboring samples to predict the classification of a new sample. When it comes to movie recommendation systems, the KNN algorithm can be employed to forecast a user's movie preferences by considering the preferences of other similar users. KNN recommends movies based on similar users' high ratings using a set of attributes and is easily scalable for large datasets. It enables personalized and relevant movie recommendations to users.

1.1.3.1 KNN Baseline Predictor

The KNN algorithm in movie recommendation systems aims to forecast the rating that a user (u) would give to a movie (i) based on the ratings of similar users with comparable preferences who have rated the same movie. To utilize the KNN algorithm for movie recommendation, we first calculate a user's baseline prediction rating. This prediction is based on the user's rating average and the movie, along with any inherent biases that may exist due to the user's tendency to rate movies higher or lower than others or the movie's tendency to receive lower or higher ratings than other movies.

Once we have the baseline prediction, we can use measures like cosine similarity or Pearson correlation coefficient to determine how similar the user's rating patterns are to those of all other users (v) who have rated the same movie (i). We employ the reduced Pearson-baseline correlation coefficient, which considers the baseline predictions of both users and item, to increase the algorithm's precisions.

$$\widehat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \sin(u, v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \sin(u, v)}$$

1.1.3.2 Baseline Predictor

The Baseline Only algorithm from Surprise library uses user-item biases to predict item ratings. User bias accounts for a user's tendency to rate items higher or lower than rating average, while item bias accounts for an item's propensity to be rated higher or lower than average. To estimate these biases, the Baseline Only algorithm uses a Stochastic Gradient Descent algorithm. To specify the learning rate and regularization parameters for the SGD algorithm, we can use the `bsl_options` parameter when creating a Baseline Only instance. Baseline Only is a fast and simple algorithm that serves as a starting point for more advanced recommendation systems.

1.1.4 Matrix Factorization Technique

Matrix factorization is a machine learning technique that breaks down a large matrix into smaller matrix to reveal useful information about the data. It is widely used in recommendation systems to estimate user preferences based on historical data. Singular Value Decomposition, Non-negative Matrix Factorization, and Alternating Least Squares are three popular matrix

factorization techniques that can be applied depending on the data and the application requirements.

1.1.4.1 SVD Matrix Factorization User Movie interactions

Singular Value Decomposition is a mathematical technique that decomposes a matrix into three parts: a diagonal singular value matrix, a right singular matrix, and a left singular matrix. SVD captures the most valuable information in the original matrix and can be used for dimensionality reduction, noise reduction, forecasting missing values, and generating personalized recommendations in collaborative filtering and recommendation systems.

1.1.4.2 SVD Plus Plus (Implicit Feedback)

SVD++ is an extension of the Single Value Decomposition algorithm used in collaborative filtering recommendation systems, including movie recommendation systems. It considers both explicit and implicit feedback, such as purchase history, clicks, and ratings, to provide better recommendations by incorporating information about user behavior. In machine learning, an epoch refers to a complete iteration over the entire training dataset during model training. In one epoch, every training example is processed once, both forward and backward. The forward pass generates predictions and computes loss, while the backward pass updates the model's parameters to minimize the loss.

1.1.5 Description Analysis

Description analysis of movies is a technique used in movie recommendation systems to extract meaningful insights from the metadata associated with movies. This metadata can include plot summaries, keywords, reviews, and other textual information that can be used to analyze the content of the movie and identify patterns or similarities between different movies. In addition to description analysis, other features like cast, director, country, rating, and genres can also be used to build a recommendation system. These features provide valuable information about the movie and can help identify similar movies based on the user's choice and preferences.

CHAPTER 2

LITERATURE SURVEY

FAB: CONTENT-BASED, COLLABORATIVE RECOMMENDATION

The approach of content-based recommendations originates from the information retrieval field and incorporates various techniques from it. Although both collaborative and content-based collaborative systems can provide this service, they possess individual disadvantages. However, Fab is a hybrid recommendation system designed for web pages that combines elements from both content-based and collaborative approaches, effectively addressing many of the limitations associated with relying solely on either of these approaches.

MOTIVATION

The motivation addressed in this paper are given below:

- The adapting population of collection agents is designed to utilize these overlaps to dynamically converge on topics of interest. This includes automatically identifying communities of interest and potentially leading to significant resource savings as the numbers of documents and users increase.

CHALLENGES

The challenges addressed in this paper are given below:

- No dynamic user's data is taken into consideration.

COMBINING CONTENT AND COLLABORATION IN TEXT FILTERING

The method comprises a way to merge collaborative input and document content for text lettering. The approach leverages latent semantic indexing to generate a collaborative outlook of a set of profiles. These profiles are term vectors created from documents that are deemed useful to the user's information requirement.

MOTIVATION

The motivation addressed in this paper are given below:

- It acts as a useful tool for text filtering.
- It gives the best performance when the right documents are used to generate the best rose-colored SVD".

CHALLENGES

The challenges addressed in this paper are given below:

- This technique does not work for larger datasets with less probability of collection.

- There is often an implicit assumption that there is a relationship between interests, user ratings, document similarity, and topical relevance.

COMBINING COLLABORATIVE FILTERING WITH PERSONAL AGENTS FOR BETTER RECOMMENDATIONS

Information filtering (IF) focuses on analyzing the content of an item and creating a personalized user interest profile. On the other hand, collaborative filtering (CF) aims to identify other users with similar preferences and utilize their feedback to suggest items. This study illustrates that by combining individual IF agents and leveraging the input from a user community, a CF framework can generate enhanced recommendations compared to what either the agents or users can achieve independently.

MOTIVATION

The motivation addressed in this paper are given below:

- It becomes very less important to invent a brilliant agent, instead of simply inventing a collection of useful ones.

CHALLENGES

The challenges addressed in this paper are given below:

- It faces challenges in effectively handling "users" who rate all items and frequently re-rate them as they continue to "learn."

A COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM FOR SUGGESTING NEW TRENDS IN ANY DOMAIN OF RESEARCH

The paper proposes a Recommender System based on Collaborative that can suggest contemporary trends in any research domain. The system employs user modeling, which involves collecting information about users by analyzing their ratings or items. User models are utilized by a range of applications such as recommender systems and search engines.

MOTIVATION

The motivation addressed in this paper are given below:

- In this case, data from the browsing history of other researchers is utilized, which avoids the need to analyze content-related problems.
- The measures of similarity between the user and research paper profile are accurate, as they make use of the cosine similarity technique.

CHALLENGES

The challenges addressed in this paper are given below:

- The actual dataset does not have the real ratings of the researcher.

A HYBRID APPROACH USING COLLABORATIVE FILTERING AND CONTENT BASED FILTERING FOR RECOMMENDER SYSTEM

This paper describes a movie recommendation system that uses a combination of collaborative filtering and content-based filtering techniques to recommend movies to users. The system mines movie databases to collect information such as attractive and popularity, which are used to make recommendations. The authors propose a hybrid approach that combines the results of these techniques to provide more precise recommendations. They mention that their system also allows for recommendations to be made to inexperienced users, as well as existing users. The authors conclude that, based on informal evaluations with a small set of users, the system has received a positive response, but that further testing, and evaluation is needed with a larger dataset. They also suggest that the system could be extended to other domains, such as recommending songs, videos, and e-commerce products.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed movie recommendation system uses a combined approach, both collaborative filtering and content-based, to provide more accurate suggestions to users.
- The system considers various attributes such as popularity and attractiveness, which are crucial factors in movie recommendation.
- The system can suggest movies to both new and existing users.
- The system can also recommend products to each customer based on the genre of movies they prefer.
- The system uses techniques like classification, similarity, and clustering to get better recommendations and increase precision and accuracy.
- The system can be expanded to other domains such as recommending videos, news, songs, books, tourism, and e-commerce sites, etc.

CHALLENGES

The challenges addressed in this paper are given below:

- The performance of the system is difficult to evaluate as there is no wrong recommendation, it is just a matter of opinion.

- The authors mention that the system was only informally evaluated with a small set of users, so more meaningful results would require a larger data set.
- The authors mention that they would like to incorporate different clustering algorithms and machine learning in the future, suggesting that the current implementation may not be optimal.
- It is not clear from the paper how the system manages new movies that have been released recently and how well it works with movies that have not been popular recently.

MULTI-DOMAIN NEURAL NETWORK RECOMMENDER

The research paper introduces an innovative approach to multi-domain recommendation using a multi-branch network. The aim is to address the problem of data sparseness by incorporating information from other domains. The network is designed to identify sharing-pattern features, which represent general user preferences, as well as domain-specific features. In contrast to existing methods that employ shared factors and features across domains, this model utilizes a shared transformation pattern. Moreover, the paper introduces a non-linear process incorporating probability to generate the ultimate user latent factor. The proposed approach is evaluated using real-world data and demonstrates superior performance compared to baseline methods. The results further suggest that domains with sparser data have greater potential for improvement, as they can effectively incorporate information from other domains.

MOTIVATION

The motivation addressed in this paper are given below:

- The Multi-Domain Neural Network (MDNN) proposed in this research tackles the issue of data sparsity in recommendation systems by leveraging information from other domains.
- The MDNN can learn complex patterns and extract valuable information from hidden layers that are closely related to the task.
- The Multi-Domain Neural Network utilizes a unique multi-branch network to identify both domain-specific features and sharing-pattern features which enhances its ability to make multi-domain recommendations while addressing the data sparsity issue.

CHALLENGES

The challenges addressed in this paper are given below:

- It is not mentioned in the paper if there are any limitations to the proposed method.

A SURVEY OF COLLABORATIVE FILTERING ALGORITHMS FOR SOCIAL RECOMMENDER SYSTEMS

This paper discusses various collaborative filtering algorithms used in social recommender systems. Collaborative filtering is a technique used to make recommendations by identifying patterns in user behavior. The paper describes several algorithms, such as item-based, user-based, and hybrid collaborative filtering, and compares their performance in terms of accuracy and scalability.

MOTIVATION

The motivation addressed in this paper are given below:

- Detailed explanation and comparison of various collaborative filtering algorithms
- Discussion of the challenges and limitations of using collaborative filtering in social recommender systems
- Evaluation of the algorithms using real-world datasets

CHALLENGES

The challenges addressed in this paper are given below:

- The research is limited to collaborative filtering algorithms and does not discuss other types of recommendation methods.
- Incorporating more recent data and evaluating the algorithms on larger and more diverse datasets
- Examining the robustness of the algorithms and their ability to manage missing or noisy data.

NEURAL FACTORIZATION MACHINES FOR SPARSE PREDICTIVE ANALYTICS

A novel model called Neural Factorization Machine (NFM) is introduced in the paper, specifically designed for accurate predictions in the presence of sparse data. NFM combines the linearity of Factorization Machines (FMs), which excel at modeling pairwise feature interactions, with the non-linearity of neural networks, which excel at modeling higher-order feature interactions. The authors argue that while FMs have been successful in various prediction tasks, their limitation of capturing only pairwise interactions hinders their performance. In contrast, NFM offers greater expressiveness, enabling it to capture the non-linear and intricate underlying structure of real-world data. Empirical results from two

regression tasks demonstrate the superior performance of NFM over FMs, highlighting a significant relative improvement of 7.3%. Moreover, when compared to recent deep learning methods like Wide Deep and Deep Cross, NFM achieves better performance despite employing a shallower structure. This advantage makes NFM more manageable to train and fine-tune in practical scenarios.

MOTIVATION

The motivation addressed in this paper are given below:

- The NFM model constantly achieves the best performance on most datasets while using the models with fewer parameters, besides the FM model.
- The model effectively captures higher-order and non-linear feature interactions.
- The Bi-Interaction operation utilized in the model enables neural networks to get more meaningful and informative feature interactions at a lower level.
- The Neural Factorization Machine (NFM) is a straightforward yet powerful solution for predicting outcomes in situations where data is sparse.
- Reduces the demand for deeper structures for quality prediction.

CHALLENGES

The challenges addressed in this paper are given below:

- The efficiency of NFM can be increased by using hashing techniques, which makes it a better choice for production-scale applications.
- Further research is needed to study its performance for other information retrieval works such as target advertising and search ranking.
- If a model has limited expressiveness, it may not be able to accurately capture complex patterns that exist in real-world data, which can result in lower performance.
- Increasing the model's depth does not always lead to better performance and can make them less interpretable, harder to optimize and tune.

MIGAN: MUTUAL-INTERACTION GRAPH ATTENTION NETWORK FOR COLLABORATIVE FILTERING

Recommender systems have gained widespread adoption across different web platforms. Among the successful approaches for building effective recommender systems, network representation learning has shown promise. However, capturing the intricate mutual influence between nodes in a network poses challenges, particularly in understanding the complex user-

item interactions that impact user decisions. To address this, the authors of the paper propose a novel algorithm named Mutual Interaction Graph Attention Network (MIGAN). MIGAN leverages self-supervised representation learning on a large-scale bipartite graph (BGNN) to capture mutual interactions effectively. Through experiments conducted with real-world data, MIGAN surpasses baseline methods in terms of both prediction accuracy and recommendation efficiency. Overall, the introduction of MIGAN presents a promising solution to the challenges of modeling user-item interactions within a network, and the empirical results validate its superiority over existing methods in recommender system performance.

MOTIVATION

The motivation addressed in this paper are given below:

- The use of graph neural networks to model the interactions between items and users in the recommendation process.
- Achieved state-of-the-art performance on benchmark datasets.
- Showing the ability of MIGAN to capture the dynamic interactions between items and users.

CHALLENGES

The challenges addressed in this paper are given below:

- The results are based on the specific datasets used in the paper, which limits the generalizability of the findings.
- The paper is from 2020, so the research is not up to date with the latest advancements in the field.

HYBRID RECOMMENDATION SYSTEM COMBINED CONTENT-BASED FILTERING AND COLLABORATIVE PREDICTION USING ARTIFICIAL NEURAL NETWORK

In this paper, the authors propose a novel recommendation system that leverages a combination of content-based filtering and collaborative filtering techniques using artificial neural networks. Content-based filtering recommends items based on their intrinsic characteristics, while collaborative filtering relies on user behavior patterns. By adopting a hybrid approach, the system capitalizes on the advantages of both methods, employing an artificial neural network to acquire user and item representations. To assess the system's performance, the authors conducted experiments using a movie dataset, comparing their results against conventional content-based and collaborative filtering approaches.

MOTIVATION

The motivation addressed in this paper are given below:

- The approach combines the strengths of both collaborative and content-based filtering.
- The use of artificial neural networks to understand the representations of users and items.
- The system was evaluated using a dataset of real-world data.

CHALLENGES

The challenges addressed in this paper are given below:

- The paper is from 2018, so the research is not up to date with the latest advancements in the field.
- The results are based upon a dataset of movies, where the findings are limited due to the generalizability of the dataset.

DEEP LEARNING ARCHITECTURE FOR COLLABORATIVE FILTERING RECOMMENDER SYSTEMS

The paper introduces a deep learning architecture that incorporates prediction errors to enhance collaborative filtering in recommender systems. The architecture of the system comprises three stages that provide stacked levels of abstraction: (a) real prediction errors, (b) predicted errors (reliabilities), and (c) predicted ratings (predictions). The approach has been evaluated on three datasets, and the results show significant improvements in prediction and recommendation quality, especially for the recall metric.

MOTIVATION

The motivation addressed in this paper are given below:

- As the dataset's size increases, the quality of recommendations improves.
- Using hybrid approaches such as social, context aware and content-based techniques can enhance the proposed deep learning architecture.
- The architecture is designed to learn the non-linear relationships between accuracy and reliability. By doing so, it can identify the most reliable predictions and make accurate recommendations based on them.

CHALLENGES

The challenges addressed in this paper are given below:

- As the number of items recommended increases, the recall improvement decreases, since the most reliable and accurate items run out.

A PERSONALIZED MOVIE RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

The paper presents a recommendation engine that employs collaborative filtering techniques to suggest movies to users. The system considers user demographics, such as age, gender, and occupation, along with their movie ratings, to identify users with similar preferences. Recommendations are generated based on the top-rated movies from the nearest neighbor. The system also offers time-sensitive recommendations. The experimental results on the Movie Lens dataset demonstrate that the proposed model generates more personalized movie recommendations compared to other models. The paper suggests several potential future directions, including incorporating additional demographic information, extending the application to different platforms, incorporating input from review sites, and integrating social media profiles to further enhance the recommendation capabilities.

MOTIVATION

The motivation addressed in this paper are given below:

- The system is founded on collaborative filtering, a well-established and encouraging technique for constructing recommendation systems.
- The system uses standard user demographics such as gender, age, and occupation, which can provide valuable information for making recommendations.
- The system utilizes time-sensitive recommendations, which can help keep up with changing user preferences.
- The system's performance is measured using widely used metrics in evaluating recommendation systems, including precision, recall, and F-measure.

CHALLENGES

The challenges addressed in this paper are given below:

- The system's personalized recommendations could be further improved by incorporating additional demographic information beyond the limited set currently used, such as nationality, race, location, mother-tongue, and languages spoken.
- The system relies on other users' ratings to make recommendations, but input from other sources such as consolidated internet databases and authentic review sites could also be used to improve recommendations.

- The system does not integrate with social media profiles, which could provide valuable information for making more personalized recommendations.
- The system does not consider location-based data, which could be used to make recommendations more relevant to the user's current location.

"NEURAL COLLABORATIVE FILTERING" BY X. HE AND L. LIAO AND T. N. KIEU AND Y. ZHANG (2017),

The paper introduces NCF, a neural network architecture for modeling user and item features in recommender systems. NCF uses a perceptron with multiple layers to learn the interaction function between items and users, resulting in improved performance compared to other methods on two datasets. The authors also show that deeper neural networks lead to better recommendation performance. The paper's main contributions are the NCF framework for collaborative filtering and the use of deep neural networks to model implicit feedback signals in recommender systems.

MOTIVATION

The motivation addressed in this paper are given below:

- The NCF framework proposed in the paper uses neural networks to address the primary challenge in recommendation systems, which is collaborative filtering, based on implicit feedback signals.
- Neural architecture has the capability to learn an arbitrary function from data, making it more versatile than traditional matrix factorization methods.
- The use of a multi-layer perceptron allows for the incorporation of non-linearities, resulting in improved recommendation performance.
- The extensive experiments conducted on real-world datasets demonstrate that the proposed approach achieves significant improvements over existing state-of-the-art methods.
- The study demonstrates that the recommendation performance improves when using deeper layers of neural networks.

CHALLENGES

The challenges addressed in this paper are given below:

- The method is focused on implicit feedback and may not be as effective for modeling explicit feedback.

- The approach is still relatively new and may not have been as extensively tested or refined as the traditional matrix factorization method.

KERNELIZED DEEP LEARNING FOR MATRIX FACTORIZATION RECOMMENDATION SYSTEM USING EXPLICIT AND IMPLICIT INFORMATION,

The authors emphasize the significance of recommender systems in assisting users in finding items of interest by providing recommendations that maximize information utilization. They provide a brief overview of traditional recommendation methods, including content-based, collaborative filtering, and hybrid approaches. Among these, matrix factorization is highlighted as a successful collaborative filtering technique, while acknowledging that various works have attempted to enhance it by incorporating trust, time, context, and other factors. However, the authors note that practical applications still present challenges for matrix factorization, particularly in dealing with sparse data and managing the substantial number of online users and items. In recent years, deep learning has emerged as a powerful tool in various domains such as natural language processing, computer vision, and autonomous driving. Consequently, deep learning has also made its way into recommendation systems. The paper specifically focuses on leveraging deep learning in recommendation systems with the objective of improving recommendation accuracy.

MOTIVATION

The motivation addressed in the paper are given below:

- Improved accuracy: By using deep learning, the paper suggests that it is possible to overcome the limitations of traditional factorization of matrix methods and improve the overall recommendation's accuracy.
- The paper proposes that deep learning methods can improve the adaptability of recommendation systems to diverse types of users and items in real-world situations, potentially resulting in more effective recommendations.
- The paper suggests that deep learning methods can be employed to effectively address the issue of data sparsity, which is a common challenge encountered in recommendation systems.

CHALLENGES

The challenges addressed in this paper are given below:

- Complexity: The use of deep learning methods can increase the complexity of the recommendation system, which may make it more difficult to implement and maintain.
- Deep learning methods usually demand significant amounts of training data, which can be difficult to acquire in some applications.
- One potential drawback of deep learning models is the lack of interpretability in their internal workings. This lack of interpretability can make it challenging to understand the underlying reasons behind specific recommendations made by these models.
- Requirement of computational power: Deep learning methods require a lot of computational resources, which may not be available for some systems.

COLLABORATIVE FILTERING RECOMMENDER SYSTEMS

The paper discusses the collaborative filtering field for recommender systems, which is a method for filtering through large information and product spaces to help users get what they are looking for. The paper covers the history of research in this area, the different algorithms, and tools available for evaluating performance, and the challenges of embedding recommender technology in specific domains. It emphasizes the importance of considering the specific tasks, information needs, and item domains when designing and evaluating recommenders, and provides guidance on the best practices for addressing these issues. In summary, the paper aims to introduce and address critical issues related to recommender systems. It discusses the challenges and best practices associated with these systems, offering insights into current approaches for tackling these issues.

MOTIVATION

The motivation addressed in this paper are given below:

- Collaborative filtering is a potent technique that allows users to go through vast amounts of information and product spaces.
- Provides personalized recommendations for each user based on user's past behavior and similar users' behavior.
- Recommender systems can be applied across various domains, including but not limited to e-commerce, music and movie recommendations, and social media platforms. Their versatility allows them to provide personalized recommendations in diverse contexts, enhancing user experiences and aiding in decision-making processes.

- The method provides a diverse set of tools for evaluating performance.

CHALLENGES

The challenges addressed in this paper are given below:

- May not consider other crucial factors such as new and trending items.
- Can suffer from the issue of "cold start" for users with no past behavior.
- Can be sensitive to changes in users' behavior, leading to a need for constant retraining of the model.

RESEARCH ON COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM BASED ON MAHOUT AND USER MODEL

In collaborative learning, system administrators can optimize the system by incorporating a machine learning algorithm based on the user model and adhering to the rules established by the training sample. This approach eliminates the need for repetitive calculations. To facilitate the implementation of a collaborative filtering algorithm on a cloud computing platform, the Hadoop platform was chosen as the foundation for development. By leveraging Hadoop, the system can efficiently handle large-scale data processing and enable distributed computing, enhancing the scalability and performance of the collaborative filtering algorithm.

MOTIVATION

The motivation addressed in this paper are given below:

- The system not only tackles practical challenges but also delivers more effective and personalized recommendations to users. By leveraging the collaborative filtering algorithm implemented on the cloud computing platform, the system can overcome scalability issues and process large volumes of data. This leads to improved recommendation accuracy and enhances the user experience by tailoring recommendations to individual preferences and needs.

CHALLENGES

The challenges addressed in this paper are given below:

- Model-based algorithms are important in the development of online recommendation systems as they provide valuable insights into improving the quality and speed of information retrieval for users.

ITEM-BASED COLLABORATIVE FILTERING RECOMMENDATION ALGORITHMS

Item-based recommendation techniques involve analyzing the user-item matrix to discover relationships between different items. These relationships are then utilized to indirectly generate recommendations for users. Various algorithms are employed for generating item-based recommendations. Additionally, there are diverse techniques for measuring similarities between items, such as item-item correlation or cosine similarity between item vectors. Various approaches are used to derive recommendations based on these similarities, including weighted sum and regression models. The choice of similarity measurement and recommendation derivation technique depends on the specific recommendation system and its objectives.

MOTIVATION

The motivation addressed in this paper are given below:

- Item-based algorithms typically generate significantly better results than algorithms based on users in recommender systems.
- The proposed algorithm performs better recommendation than the best available user-based algorithms.

CHALLENGES

The challenges addressed in this paper are given below:

- The item-item approach yields better prediction quality compared to the user-user approach, but the improvement is not substantial.

AN ENTROPY-BASED NEIGHBOR SELECTION APPROACH FOR COLLABORATIVE FILTERING

The paper introduces a new collaborative filtering method that differs from traditional algorithms. Instead of relying solely on entity similarities to form neighborhoods, the proposed approach incorporates an entropy-based selection of neighbor process that considers the uncertainty of entity vectors. This uncertainty is a measure of how a user distinguishes their taste preferences or the diversity of items' rating distributions within the rating domain. The method combines similarities and uncertainty values and aims to minimize the entropy difference within a neighborhood while gathering the most similar entities. To achieve this goal, the approach solves an optimization problem.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed approach can improve the performance of KNN-based collaborative filtering methods without compromising performance online.

- The proposed approach is compatible with and can be integrated with other existing methods, allowing for a combination of their strengths.

CHALLENGES

The challenges addressed in this paper are given below:

- The paper does not mention any potential limitations of the proposed model or any areas where it may not perform as well.

A MOVIE RECOMMENDER SYSTEM: MOVREC

The paper introduces a movie recommendation system called MOVREC that uses a collaborative filtering approach. The system analyzes user information to recommend movies that are suitable for the user. The recommended movies are sorted based on reviews given by previous users and the K-means algorithm's is utilized used for this cause. MOVREC efficiently helps users find movies of their choice based on the movie experiences of other users, without wasting time on unnecessary browsing.

MOTIVATION

The motivation addressed in this paper are given below:

- The system allows users to choose from a set of attributes to create a personalized preference profile. Then, it recommends a bunch of movies based on the calculated weight of these attributes using the K-means algorithm.
- The proposed method showed better performance for a small group of users, as it utilizes an information filtering approach to predict their preferences.

CHALLENGES

The challenges addressed in this paper are given below:

- As the method was implemented over a small set of users, not sure whether for the large set of users it gives better performance or not.

A RECOMMENDATION MODEL BASED ON DEEP NEURAL NETWORK

The paper introduces a novel model that combines a collaborative filtering recommendation algorithm with deep learning. The model comprises two key components: A feature representation method based on a quadric polynomial regression (QPR) model, which improves the traditional matrix factorization algorithm by enhancing latent feature extraction. A deep neural network model that takes the extracted latent features as input and predicts rating scores.

By integrating these two parts, the proposed model aims to enhance the accuracy of rating predictions in collaborative filtering-based recommendation systems.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed model, which merges a collaborative filtering recommendation algorithm with deep learning technology, has been thoroughly evaluated using three publicly available datasets. The evaluation results demonstrate that the model surpasses other existing recommendation algorithms, significantly enhancing the overall recommendation performance.
- The framework presented in the paper is characterized as both simple and generic, making it applicable to methods beyond the one specifically outlined in the paper. This versatility positions the framework as a valuable guide for the development of deep learning-based methods in the context of recommendation systems. It provides a foundation for researchers and practitioners to explore and create innovative approaches within the realm of deep learning for recommendation systems.

CHALLENGES

The challenges addressed in this paper are given below:

- When the number of feature dimensions is low, the accuracy of the features learned by our model may be insufficient, indicating the potential for further improvement.
- The performance of the model does not significantly improve when more than two hidden layers are added, suggesting that additional layers may not contribute significantly to its performance.

DOMAIN-SENSITIVE RECOMMENDATION WITH USER-ITEM SUBGROUP ANALYSIS

The paper presents the Domain-sensitive Recommendation (DsRec) algorithm, which focuses on enhancing rating prediction by simultaneously analyzing user-item subgroups. These subgroups are treated as domains and comprise a subset of items with similar attributes and a subset of users who express interest in those items. The DsRec framework incorporates three key components: a matrix factorization model for rating reconstruction, a bi-clustering model

for analyzing user-item subgroups, and two regularization terms that integrate these components into a unified formulation.

MOTIVATION

The motivation addressed in this paper are given below:

- The information about the domain is utilized to guide the finding of latent space.
- The proposed method demonstrates superior prediction accuracy when compared to existing state-of-the-art methods.

CHALLENGES

The challenges addressed in this paper are given below:

- Grouping users and items into subgroups through clustering facilitates the organization of related items and users into distinct clusters. However, if the number of subgroups is too limited, it may not effectively differentiate between diverse user-item interest domains, potentially leading to less clear distinctions between them.
- If the number of subgroups is too large, the preferences of the clusters may overlap, resulting in less discriminative latent factors for effective rating prediction.

COLLABORATIVE FILTERING AND DEEP LEARNING BASED HYBRID RECOMMENDATION FOR COLD START PROBLEM

To address challenges such as sparsity and the cold start problem in collaborative filtering (CF), this paper introduces a hybrid recommendation model named HRCD. HRCD combines CF with deep learning neural networks, specifically utilizing the timeSVD++ CF model to capture temporal dynamics by incorporating time-dependent biases and latent factors. It also employs a latent transition matrix to summarize evolving user preferences. The deep learning component utilizes a stacked denoising autoencoder (SDAE) to reconstruct corrupted input data, effectively addressing the cold start problem. Experimental results on a large Netflix rating dataset for movies demonstrate that HRCD outperforms other recommendation models in terms of recommendation accuracy.

MOTIVATION

The motivation addressed in this paper are given below:

- The results of the study indicate that the proposed hybrid recommendation model surpasses a basic model that relies on average user ratings for predicting ratings of cold start items.
- The HRCD model exhibits effectiveness in rating predictions for both cold start and non-cold start items. In fact, it outperforms four baseline models in rating predictions for non-cold start items, even in scenarios where only limited ratings are available.

CHALLENGES

The challenges addressed in this paper are given below:

- The proposed method for rating prediction of cold start items includes two prediction approaches: ToA and ToU. These approaches utilize different configurations of M (the number of most related items) and L (size of the test dataset for cold start movies). The ToU approach demonstrates superior performance when using a larger M , while the ToA approach performs well with a smaller M (e.g., employing only the twenty most related non-cold start items for rating prediction). However, it is important to note that using a large M , such as one hundred, in the ToA approach leads to deficient performance, even worse than that of the simple average model.

COLLABORATIVE FILTERING WITH TEMPORAL DYNAMICS

This paper proposes a more nuanced approach to modeling temporal dynamics in recommendation systems, as traditional methods such as classical time-window or instance decay approaches may discard too much relevant information. Instead, the proposed model aims to capture the changing behavior of items and users throughout the lifespan of data, allowing for the exploitation of all relevant components while discarding only what is deemed irrelevant. Two leading collaborative filtering approaches are revamped to incorporate this model: The proposed model comprises a factorization model that captures changes in product and user characteristics over time to capture longer-term trends. Additionally, an item-item neighborhood model is used to understand the decay of influence between two items rated by a user over time, revealing fundamental item relationships.

MOTIVATION

The motivation addressed in this paper are given below:

- The aim is to develop a model that can effectively capture inconsistency over the entire time, enabling us to distinguish between short-lived factors and persistent ones.
- The incorporation of temporal dynamics in both the factorization and neighborhood models was found to be highly beneficial in enhancing the quality of predictions. This improvement was observed to be more significant compared to various algorithmic enhancements.

CHALLENGES

The challenges addressed in this paper are given below:

- Tracking the temporal dynamics of customer preferences to products raises unique challenges. Each user and product potentially go through a distinct series of changes in their characteristics.
- We often need to model all those changes within a single model thereby interconnecting users (or, products) to each other to identify communal patterns of behavior.

GENERALIZED PROBABILISTIC MATRIX FACTORIZATIONS FOR COLLABORATIVE FILTERING.

Collaborative filtering techniques, such as Probabilistic Matrix Factorization (PMF), have demonstrated considerable potential. One approach that bridges PMF and Bayesian PMF (BPMF) is called "parametric PMF" (PPMF). By conducting extensive experiments on movie recommendation datasets, the study demonstrates that simpler models that directly capture correlations among latent factors can outperform existing PMF models. The inclusion of side information can also contribute to improved prediction accuracy, while considering row/column biases further enhances predictive performance.

MOTIVATION

The motivation addressed in this paper are given below:

- PPMF surpasses PMF, BPMF, and co-clustering-based algorithms in terms of accuracy.
- Residual models tend to outperform their corresponding original models, indicating their superior performance in prediction tasks.

CHALLENGES

The challenges addressed in this paper are given below:

- The side information incorporated in the proposed work did not result in a significant improvement in accuracy compared to the approaches examined. The existing side information used was not sufficiently influential in enhancing the accuracy of the predictions.

RECOMMENDER SYSTEM BASED ON HIERARCHICAL CLUSTERING ALGORITHM CHAMELEON

In this study, the authors address the requirements of accuracy and speed in recommender systems by proposing an efficient technique based on Hierarchical Clustering. The user or item specific information is organized into clusters using the Chameleon Hierarchical clustering algorithm. A voting system is then employed to predict the rating of each item. The performance of the proposed Chameleon-based recommender system is evaluated by comparing it with existing techniques that rely on the K-means clustering algorithm.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed approach utilizing Chameleon hierarchical clustering algorithms for recommender systems demonstrates superior cluster quality compared to K-Means.
- The Chameleon algorithm effectively identifies high-quality clusters, resulting in lower error rates. This indicates that the proposed approach offers an efficient and improved method for recommender systems.

CHALLENGES

The challenges addressed in this paper are given below:

- The K-Means algorithm has a lower time complexity compared to the Chameleon algorithm. This means that K-Means requires less computation time to complete the clustering process compared to Chameleon.

CONTENT-BASED CROSS-DOMAIN RECOMMENDATIONS USING SEGMENTED MODELS

In this study, the focus is on content-based cross-domain recommendations in recommender systems. The objective is to leverage information from different source domains to provide recommendations in target domains. The researchers propose a flexible framework that can be used with different classifiers. They define meta-data features to capture characteristics of the domains and introduce indicator features to categorize users into different domains based on the

values of the meta-data features. Logistic regression is employed as the classifier in their implementation of the framework, and experiments are conducted using a LinkedIn dataset to make job recommendations.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed framework is versatile and can be utilized with different classifiers, providing flexibility in implementation.
- The model in this framework effectively transfers shared information across diverse domains while maintaining their individual characteristics.
- The regularization mechanism in the model enables automatic selection of key features specific to each domain, while still accommodating the incorporation of expert knowledge in feature selection.

CHALLENGES

The challenges addressed in this paper are given below:

- The results signify only a small improvement in recommendation performance in the offline setting.

COLLABORATIVE DEEP LEARNING FOR RECOMMENDER SYSTEMS

The problem of auxiliary information being very sparse is addressed in this paper by generalizing the recent advances in deep learning from i.i.d. input to non-i.i.d. (CF-based) input and propose a hierarchical Bayesian model called collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix.

MOTIVATION

The motivation addressed in this paper are given below:

- Experiments on three real-world datasets from different domains show that CDL can significantly advance the state of the art.
- CDL provides a framework that can also admit deep learning models other than SDAE.
- CDL is sensitive enough to changes of user taste and hence can provide more accurate recommendations.

CHALLENGES

The challenges addressed in this paper are given below:

- The paper does not mention any potential limitations of the proposed model or any areas where it may not perform as well.

HYBRID COLLABORATIVE FILTERING MODEL FOR CONSUMER DYNAMIC SERVICE RECOMMENDATION BASED ON MOBILE CLOUD INFORMATION SYSTEM

Research on web service recommendation systems addresses two problems: prediction and completion of sparse QoS data, and the user's personalized recommendation. This study presents a hybrid collaborative filtering model for consumer service recommendation based on mobile cloud by introducing user preferences. The example verified that the model can effectively reduce the data sparsity and increase the accuracy of the prediction.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed UCFUP model can predict the QoS accurately, with smaller MAE and RMSE values than traditional models.
- The NVs of users and items have been declining over time, indicating that neighbors of users and items are starting to stabilize.
- The dynamic CF in the proposed model leads to a decrease in variation in similarity value, making the neighbors stable.

CHALLENGES

The challenges addressed in this paper are given below:

- The research only focuses on the RT and TP databases, and it is unclear how the proposed model would perform on other types of data.
- The proposed model's performance is only evaluated at six sampling moments, so it is not clear how well it would perform over a longer period.

A NOVEL DEEP MULTI-CRITERIA COLLABORATIVE FILTERING MODEL FOR RECOMMENDATION SYSTEM

The text discusses a novel approach to recommender systems that combines multi-criteria predictions with collaborative filtering and deep learning. The proposed model includes two parts: a criteria ratings deep neural network that predicts ratings based on user and item features, and an overall rating deep neural network that uses the criteria ratings to predict the overall rating. The authors claim that experiments on a real-world dataset have shown that this approach

outperforms other state-of-the-art methods, suggesting that using deep learning and multi-criteria in recommendation systems is effective.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed model combines multi-criteria predictions with collaborative filtering and deep learning, which the authors claim has led to improved performance compared to other state-of-the-art methods.
- The model was shown to have a higher MAE, F1, F2, FCP, MAP, and MRR than the other methods evaluated.
- The use of multi-criteria ratings in addition to the overall rating is believed to help in understanding why users like certain items, which could lead to more accurate similarity estimates between users.

CHALLENGES

The challenges addressed in this paper are given below:

- The article does not mention any potential limitations of the proposed model or any areas where it may not perform as well.

A NEW SIMILARITY MEASURE FOR COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEMS

The objective of a recommender system is to provide personalized recommendations using the collaborative filtering technique. The main component of this technique is a similarity measure used to determine the set of users with similar behavior. The paper proposes a new, simple, and efficient similarity measure, which is determined through mathematical equations and a nonlinear system and shows that it is competitive in terms of accuracy compared to other similarity measures in literature.

MOTIVATION

The motivation addressed in this paper are given below:

- Proposes a new similarity measure (OS) for recommender systems based on collaborative filtering.
- The similarity measure is developed using advanced mathematical tools (integral equation, system of linear differential equations and nonlinear systems)

- Experiments using three benchmark datasets (MovieLens100K, MovieLens1M and Yahoo-Music) show that OS is competitive in terms of accuracy and quality of ranking.
- Has the same complexity as classical similarity measures such as COS?

CHALLENGES

The challenges addressed in this paper are given below:

- No specific cons are mentioned in the given text.

A COLLABORATIVE FILTERING RECOMMENDATION ALGORITHM BASED ON EMBEDDING REPRESENTATION

The paragraph describes a proposed collaborative filtering algorithm called UI2vec, which embeds both users and items in a potential space and uses item similarity to predict a user's content of interest. The algorithm also includes a variation called VUI2vec, which maps users, items as independent Gaussian distributions, and uses variational inference to obtain approximate posterior distributions. The performance of the algorithms was evaluated on three datasets, and the results indicate that they perform well compared to a baseline model. The paragraph also mentions that the impact of super parameters on the model's performance was investigated.

MOTIVATION

The motivation addressed in this paper are given below:

- The proposed algorithm, UI2vec, addresses the problem of data sparsity and implicit feedback uncertainty by simultaneously learning embedded representations of users and items.
- The generative version of UI2vec, VUI2vec, maps user and item embeddings to distributions and uses variational inference to obtain an approximate distribution of the posterior distribution.
- The experimental results show that the two methods proposed in this paper perform better than the selected baseline models.

CHALLENGES

The challenges addressed in this paper are given below:

- The proposed algorithm, UI2vec, addresses the problem of data sparsity and implicit feedback uncertainty by simultaneously learning embedded representations of users and items.

- The generative version of UI2vec, VUI2vec, maps user and item embeddings to distributions and uses variational inference to obtain an approximate distribution of the posterior distribution.
- The experimental results show that the two methods proposed in this paper perform better than the selected baseline models.

A PERSONALIZED RECOMMENDATION FRAMEWORK BASED ON MOOC SYSTEM INTEGRATING DEEP LEARNING AND BIG DATA

This paper describes a personalized recommendation method for Massive Open Online Courses (MOOCs) using deep learning and big data technology. The method is based on the Bidirectional Encoder Representations from Transformers (BERT) model and includes strategies to improve accuracy, such as acquisition and preprocessing of open data, a recommendation model framework incorporating a self-attention mechanism, and a domain feature difference learning strategy to improve performance. The results of experiments show that the proposed model performs well compared to other methods.

MOTIVATION

The motivation addressed in this paper are given below:

- The research proposes a personalized recommendation method for Massive Open Online Courses (MOOCs) that uses deep learning and big data technology.
- The method is based on the Bidirectional Encoder Representations from Transformers (BERT) model and includes strategies to improve accuracy, such as acquisition and preprocessing of open data, a recommendation model framework incorporating a self-attention mechanism, and a domain feature difference learning strategy to improve performance.
- The results of experiments show that the proposed model performs well compared to other methods, including User-based Collaborative Filtering (UBCF), Item-based Collaborative Filtering (IBCF), RNN, Long Short-Term Memory (LSTM), GRU4Rec, SASRec, and DeepFM.
- The ablation study demonstrates the effectiveness of the domain feature learning method in improving the personalized course recommendation.

CHALLENGES

The challenges addressed in this paper are given below:

- The research lacks theoretical verification.

- The research lacks large-scale experiment analysis.

DEEP REINFORCEMENT LEARNING IN RECOMMENDER SYSTEMS: A SURVEY AND NEW PERSPECTIVES

The paper presents a survey on the recent developments in deep reinforcement learning (DRL) for recommender systems. It offers a thorough examination of DRL-based recommender systems, encompassing a classification of existing methods, exploration of emerging topics and challenges, and suggestions for future research directions. The authors emphasize the constraints of traditional recommendation approaches and how DRL can address them by leveraging interaction data from the environment. Additionally, they underscore the novelty and contributions of their survey by offering an updated and comprehensive overview of DRL-based recommender systems, distinguishing it from prior surveys in the field.

MOTIVATION

The motivation addressed in this paper are given below:

- The paper serves as a valuable resource for researchers, practitioners, and educators in the field of recommender systems by providing a comprehensive and up-to-date overview of deep reinforcement learning (DRL) in the context of recommendation. It offers insights into the latest advancements and trends in DRL-based recommender systems, making it a valuable reference for those interested in understanding and exploring this emerging area of research.
- The paper discusses opens topics and emerges issues in DRL-based recommending systems and provides perspectives on future directions for research.

CHALLENGES

The challenges addressed in this paper are given below:

- The paper may be too technical for readers without a background in deep learning and reinforcement learning.
- The paper does not include any experimental results or evaluations of the methods discussed, which may limit its usefulness for practitioners looking to implement DRL-based recommender systems.
- The evaluation of the specific system introduced in the paper is not clear and not well described, so it may be difficult to understand the performance of the system.

AN INTELLIGENT MOVIE RECOMMENDATION SYSTEM THROUGH GROUP-LEVEL SENTIMENT ANALYSIS IN MICROBLOGS

The authors propose a novel model for recommending programs on online media sharing platforms, addressing the limitations of traditional recommendation algorithms. By leveraging social networks and analyzing user preferences expressed in microblogs, the model enhances the recommendation process. It connects movie and TV watchers with social network activities, overcoming the "cold start" problem commonly encountered in recommendation systems. The model utilizes data mining and social computing techniques to assess program similarity and provide personalized recommendations across different media devices. The approach demonstrates the effectiveness of mining microblogs and can be readily implemented in online media streaming sites to offer intelligent program recommendations to users.

MOTIVATION

The motivation addressed in this paper are given below:

- The model put forward in the article employs networks of socials and extracts user information of preference from microblogs to assess the resemblance between TV episodes and online movies.
- The model establishes a connection between movie and TV watchers and social network activities, bridging the gap between the two domains.
- It effectively solves the "cold start" problem in movie and TV recommendation systems.
- The model incorporates data mining approaches and social computing models to enhance recommendation capabilities.

CHALLENGES

The challenges addressed in this paper are given below:

- The model relies on the availability of social network data, which may not always be accessible or dependable.
- The model that is proposed is not evaluated on any real-world dataset, limiting its effectiveness and generalizability.
- The paper lacks experimental results to evaluate the performance of the proposed model, which limits the assessment of its effectiveness.
- Future research could benefit from conducting experiments to validate the performance of the model and compare it with existing recommendation systems.

- The proposed model may have limitations in terms of scalability and efficiency, especially with copious amounts of data.

MOVIE RECOMMENDATION AND SENTIMENT ANALYSIS USING MACHINE LEARNING

This paper proposes a novel model for program recommendation in online networks, specifically for movie and TV streaming sites. The model utilizes social networks and mines user preferences information expressed in microblogs to evaluate the similarity between online movies and TV episodes. The authors claim that this is the first effort to bridge the gap between the movie and TV watcher's domain with social network activities, and that it is the first approach that can solve the "cold start" problem in movie and TV recommendation systems. The proposed model employs a series of data mining approaches and social computing models and can be easily applied to online media streaming sites to make intelligent recommendations to customers through mining microblogs.

MOTIVATION

The motivation addressed in this paper are given below:

- The use of the Cosine Similarity algorithm for movie recommendation, which has shown to be accurate in recommending related movies based on numerous factors.
- The use of both the Naive Bayes and Support Vector Machine algorithms for sentiment analysis, which allows for comparison and selection of the best algorithm for classifying reviews.
- The inclusion of prospects for future improvement such as increasing the accuracy of sentiment analysis for handling sarcastic or ironic reviews and expanding the analysis to other languages.

CHALLENGES

The challenges addressed in this paper are given below:

- The limitation of the movie recommendation system is if the movie entered by the user is not present in the dataset or if the name is not entered in the same format as in the dataset.
- The limitation of the sentiment analysis to only English reviews and difficulty in correctly classifying sarcastic or ironic reviews.
- The lack of consideration for user preferences in movie recommendations.

CHAPTER 3

SYSTEM SPECIFICATIONS

3.1 FIREBASE AUTHENTICATION

Firebase Authentication is a service under Google Cloud that provides secure, user authentication and authorization for your mobile, web, and server applications. It supports a variety of authentication providers, including email/password, Google, Facebook, Twitter, GitHub, and more.

Firebase Authentication also provides several features that make it easy to build secure applications, such as:

- User management: Firebase Authentication provides a user management system that allows you to create and manage users in your application.
- Social sign-in: Firebase Authentication supports social sign-in, which allows users to sign in with their Google, Facebook, Twitter, or GitHub account.

Firebase Authentication is a powerful and easy-to-use service that can help you build secure, user-friendly applications. It is a secure, easy-to-use authentication service for mobile, web, or server applications to manage users and authentication.

3.2 FIRESTORE DATABASE

Firebase Firestore is a cloud native, NoSQL database that offers flexible, scalable, and performant solution for querying, managing and querying data. It is a fully managed service, so you do not need to worry about infrastructure or maintenance. Firestore offers several features that make it an excellent choice for mobile and web applications, including:

- Real-time data synchronization: Firestore automatically synchronizes data across all clients, so your users always have the latest information.
- Durability: Firestore is designed to be universally available and durable, so your data is always safe.
- Scalability: Firestore can scale to meet the needs of your application, even as it grows.

- Security: Firestore offers several security features to protect your data, including encryption and authentication.
- Performance: Firestore is designed to be performant, so your applications can access data quickly and easily.

Firebase Firestore is an excellent choice for mobile and web applications that need a scalable, performant, and secure database. It offers several features that make it easy to use and manage your data.

3.3 FIREBASE STORAGE

Firebase Storage is a universally accessible, durable, and secure object storage service for storing any type of data. It offers a variety of features, including:

- Scalability: Firebase Storage can scale to meet the needs of any application, from small personal projects to large-scale enterprise deployments.
- Durability: Firebase Storage is designed to be exceptionally durable, with data stored redundantly across multiple locations.
- Security: Firebase Storage uses industry-leading security measures to protect your data.
- Performance: Firebase Storage is designed for high performance, with data delivered quickly and reliably.
- Integration: Firebase Storage integrates seamlessly with other Firebase services, making it easy to build and deploy applications.

Overall, Firebase Storage is a powerful and versatile object storage service that can be used to store any type of data for any application.

3.4 PYTHON FLASK

Flask is intended to be a small and fast framework for building web applications. It is not intended to be a full-featured framework with all the bells and whistles. However, it does provide a lot of the features that are commonly needed in web applications, such as routing, templating, and form processing.

Flask is an excellent choice for building simple web applications. It is also an excellent choice for building web applications that need to be fast and lightweight.

If you are looking for a simple, fast, and lightweight framework for building web applications, then Flask is an excellent choice.

3.5 PICKLE

Pickling is the process of converting an object into a string representation that can be stored on disk or transmitted over a network. The pickled object can then be reconstructed later using the unpickling process.

Pickling is useful for storing data that needs to be persisted across multiple executions of a program, or for sending data to another program. It is also useful for debugging, as it allows you to inspect the internal state of an object.

To pickle an object, you use the pickle module. The pickle module provides a function called `pickle()` that takes an object or a dictionary as an argument and returns as a string representation of the object.

To unpickle an object, you use the pickle module again. The pickle module provides a function called `unpickle()` that converts a string representation back to object as an argument and returns the object.

3.6 VERCEL

Vercel is a platform on the cloud that provides a suite of tools for managing and deploying web applications. It offers several features that make it a popular choice for developers, including:

- Continuous deployment: Vercel can automatically deploy changes to your application as soon as they are made, which can help you to keep your application up-to-date and bug-free.
- Staging environments: Vercel allows you to create staging environments for your application, which can be used to test changes before they are deployed to production.
- Static site hosting: Vercel can host static websites, which can be a good option for simple websites that do not require a lot of dynamic functionality.

- GitHub integration: Vercel integrates with GitHub, which makes it easy to deploy changes to your application from your GitHub repository.

Overall, Vercel is a powerful and versatile platform that can be used to deploy and manage a wide range of web applications.

3.7 REACT QUERY

React Query is a state management library for React that helps you manage data fetching, caching, and updating in your application. It is based on the query paradigm, which means that you define queries for the data you need, React Query will automatically fetch and cache the data for you. This makes it easy to keep your application data up-to-date, even when the data changes on the server.

React Query also provides several features that make it easy to use, such as:

- Automatic caching: React Query automatically caches data for you, so the developer does not have to be concerned about it.
- Optimistic updates: React Query can automatically update your application's state with the latest data, even before the data has been fully loaded.
- Stale data handling: React Query can handle stale data gracefully, so your application will always be in a consistent state.

Overall, React Query is a powerful and easy-to-use state management library for React that can help you build better applications.

3.8 CHAKRA-UI

Chakra UI is a free, open-source, and fully customizable React UI library that can be used to build beautiful and responsive user interfaces. It is based on the Material Design guidelines and provides a wide range of components, including buttons, cards, forms, tables, and more. Chakra UI is easy to use and can be customized to match your brand or style. It is also well-documented and has a large community of users and contributors.

3.9 NUMPY

NumPy is a Python library that provides fast, efficient numerical computation. It is used for scientific computing, data science, and machine learning. NumPy arrays are multidimensional arrays that can be used to store and manipulate data. NumPy also provides several functions for working with arrays, such as arithmetic, logical, and statistical operations.

NumPy is a popular library and is used by many popular Python libraries, such as Pandas, Scikit-learn, and Matplotlib. It is also used in many scientific and engineering applications.

NumPy is a powerful tool for scientific computing and data science. It is easy to use and provides several features that make it well-suited for these tasks.

3.10 SCIPY

SciPy is a free and open-source scientific library for the Python programming language. It contains modules for mathematics, science, engineering, and data science. SciPy is often used with NumPy, another Python library for scientific computing.

SciPy includes modules for:

- Numeric integration: calculating definite integrals.
- Optimization: finding the best solution to a problem
- Statistics: analyzing data and making inferences
- Data analysis: cleaning, transforming, and analyzing data.

SciPy is a powerful tool for scientific computing. It can be used to solve a wide variety of problems, from simple calculations to complex simulations. SciPy is also well-documented and easy to use.

CHAPTER 4

SYSTEM DESIGN & IMPLEMENTATION

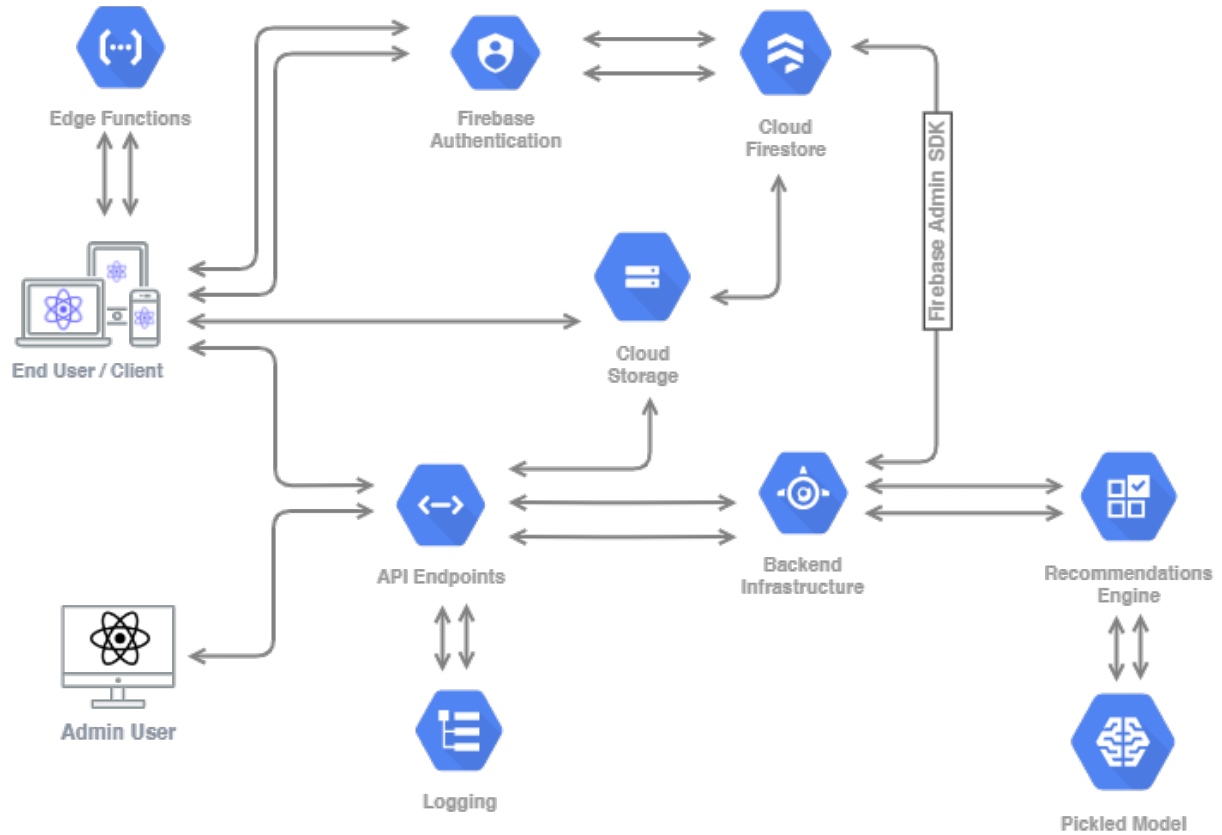


Figure 1: Workflow of the Proposed system

4.1 FRONTEND:

The Movie Recommender website, built on Next.js with Chakra UI, is an exceptional platform that offers personalized movie suggestions tailored to individual preferences. With its seamless user interface and engaging design, the website provides a user-friendly experience for movie enthusiasts. Powered by Next.js, the website ensures fast page loading and efficient navigation, enhancing the overall performance. The integration of Chakra UI, a versatile and customizable UI library, enables the creation of visually appealing components and a consistent design system. Through a combination of advanced algorithms and user feedback, the Movie Recommender website delivers accurate and diverse movie recommendations, considering various genres, actors, directors, and user ratings. Whether users are seeking thrilling action

flicks, heartwarming dramas, or thought-provoking documentaries, this website promises to be their go-to destination for discovering their next favorite film.

4.2 BACKEND:

The backend REST API, serving as the gateway for the recommendation system, is a powerful component written in Python Flask. It acts as a bridge between the user interface and the recommendation model, ensuring efficient communication and seamless integration. With its robust architecture, the backend API is designed to generate responses using a pickled model, which has been trained to provide accurate and personalized movie recommendations. Flask, a lightweight web framework, offers flexibility and ease of use, enabling smooth handling of HTTP requests and responses. The API efficiently processes user inputs and queries the pickled model to obtain relevant movie recommendations based on numerous factors such as user preferences, historical data, and current trends. Through its well-defined endpoints and intuitive API documentation, the backend REST API streamlines the communication process, ensuring a reliable and efficient recommendation system for the Movie Recommender website.

4.3 DATASET:

The dataset consists of three main files:

1. **training_set.tar:** a compressed tar archive containing 17,769 text files, one per movie, where each line in a file represents a customer's rating in the following format: *CustomerID, Date, Rating*.
2. **movie_titles.txt:** a text file containing movie information in the format *MovieID, YearOfRelease, Title*. This file is extended using the TMDb Movie API to add fields like *Overview, Adult, Original_Language, Popularity*.
3. **qualifying.txt:** a file in text format is available, consisting of lines that denote a movie's identification number, followed by the customer IDs and dates of ratings, with each line representing a distinct movie ID. This file is utilized for evaluating the effectiveness of algorithms submitted for the Netflix Prize competition.

Each rating in the *training_set.tar* file consists of three fields: *CustomerID*, *Rating*, and *Date*. The *CustomerID* ranges from 1 to 2,649,429, and the ratings are on a scale from 1 to 5. The *Date* field has the format of *YYYY-MM-DD*.

The *movie_titles.txt* file contains movie information in the format *MovieID*, *YearOfRelease*, *Title*. The *MovieID* is an integer ranging from 1 to 17,770, the *YearOfRelease* can range from 1890 to 2005, and the *Title* is for the Netflix movie in English.

The *qualifying.txt* file is like the *training_set.tar* file but with the actual ratings withheld. It has rows with movie IDs, customer IDs, and rating dates listed one for per line for each movie ID.

Overall, the dataset has 100,380,506 ratings from 479,188 users on 17,769 movies. The dataset is commonly used for collaborative filtering tasks, where the goal is to anticipate user ratings for movies they have not yet watched based on their past behavior.

4.4 PRE-PROCESSING DATA

4.4.1 Extending the Dataset

Although the Netflix Prize dataset was a valuable resource for our movie recommendation system, it did not provide all the necessary information needed to make accurate recommendations. The dataset only contained user ratings and movie titles, leaving out prominent features like movie descriptions, release dates, and genres. To overcome this limitation, we fetched the remaining information using the TMDb movie API. By integrating the API with our system, we were able to access a vast amount of movie metadata, including plot summaries, release dates, and cast and crew information. This additional data enabled us to enhance the precision of our recommendations by considering a wider range of factors that influence a user's movie preferences.

4.4.2 Sampling Data

A method was proposed in a study to improve the efficiency and effectiveness of data analysis through sampling. The approach involved selecting a subset of items and users from a larger data set to create a smaller data set for experiments and analyses. This helped to reduce computational costs and processing time while ensuring representative results. By using a sample data set, multiple experiments could be conducted with different models, making it

easier to compare their performance. It was concluded that sampling data is a critical step in the research process, allowing researchers to achieve their research objectives in a more effective and efficient manner.

4.4.3 Finding Global Averages

The purpose of this project is to calculate statistical metrics for a movie recommendation dataset. Specifically, we aim to determine the average rating per user, average rating per movie, and the overall average rating. These metrics provide insights into the dataset's characteristics, such as rating distribution, user preferences, and movie popularity. They will also serve as features in our recommendation system, enhancing prediction accuracy. To compute these metrics, we will utilize a sampled train dataset, a subset of the original data with reduced movies and users. Python libraries like NumPy and Pandas will facilitate efficient computation, and Matplotlib will aid in visualizing the results. By analyzing these metrics, we can gain a deeper understanding of the dataset and develop a more effective recommendation system.

4.4.4 Featurizing of Train & Test Data

We perform feature engineering on the train data set to prepare it for regression analysis. To do this, it creates a new data set containing additional features based on the rating, movie, and user data. The feature extraction process includes calculating various statistical features such as the global average rating for all movies, the average user rating, and the average movie rating. Additionally, it involves determining the ratings of similar users for a given movie and the ratings of similar movies for a given user. These ratings serve as additional features in the dataset. Moreover, for each user-movie pair, the actual rating is added as a new row in the dataset. By incorporating these features, the dataset becomes more comprehensive and enables more accurate recommendations.

As a proposed work, we can use this code to perform similar feature engineering on our own train data set. We would first need to make sure that our train data set is in the correct format for this code to work. We would then run this code and adjust the variables to match our data set. Once the code has finished running, we would have a new data set that contains additional features based on the user, movie, and rating data. This new data set could then be used for regression analysis.

4.4.5 Transforming of Train & Test Data

As part of our research work, we will be using the Surprise library to train different models like SVD, KNN Baseline only, etc. However, Surprise requires the train and test data to be in a specific format, which means that we cannot directly use the raw data (movie, user, rating) for training the models. Therefore, we need to transform our train data into a format that is compatible with Surprise.

To transform the data, we will create a train set using a file or a Pandas DataFrame. The train set will be created by reading the CSV file that we prepared earlier, which contains the featurized data. We will use the *Dataset.load_from_file()* function from Surprise to read the file and create the train set. This function takes the path to the file as input and returns the train set.

Once we have the train set, we will use it to train different models in Surprise. The ratings for the test data will be predicted using these models. For evaluating these models' performances, we will use different evaluation metrics like RMSE, MAE, etc.

In summary, as part of our proposed work, we will transform our train data into a format that is compatible with Surprise, create a train set, and use it to train different models to suggest the ratings for the test data. The performance of the models is evaluated using different evaluation metrics to determine the best model for our recommendation system.

4.5 MACHINE LEARNING MODELS

4.5.1 XGBoost with thirteen features

To evaluate the performance of a model, various evaluation metrics can be utilized, such as RMSE and MAPE. These metrics assist in evaluating how well the model's predictions align with the actual values. RMSE determines the difference between actual and predicted values, and a smaller value suggests better performance. Similarly, MAPE expresses the precision of the model's forecasts as a percentage of the actual value. In this specific analysis, an XGBoost model was trained on a dataset that consisted of thirteen features. To enhance the model's accuracy, the F score was used to identify the most critical features for prediction.

4.5.2 Baseline model

A hybrid model was created by combining a 13-feature XGBoost model with a Surprise Baseline predictor. The trained hybrid model was utilized to generate predictions for both the testing and training datasets. To further improve the model's accuracy, the Baseline Predictor was included as an additional feature, and the significance of each feature was assessed using the F score.

4.5.3 KNN Baseline predictor

To predict the rating of a movie (i) for a user (u), the K users who are most like the user (u) are identified as neighbors. These neighbors' ratings for the same movie (i) are then used to estimate the user's rating for that movie. The ratings are averaged and weighted based on the similarity between the user (u) and the neighbors to obtain the expected rating.

The Surprise KNN Baseline predictor is an effective method for predicting user ratings in movie recommendation systems. It incorporates baseline predictions, which account for user and item biases, and utilizes the shrunk Pearson-baseline correlation coefficient to improve accuracy. By considering these factors, the algorithm provides more personalized movie recommendations for users, enhancing their overall experience.

4.5.4 Surprise KNN Baseline with User-User similarities.

The KNN Baseline algorithm is a well-known collaborative filtering technique employed in movie recommendation systems. It calculates the baseline prediction of the rating for each user-movie pair by considering factors like item bias, user bias, and the average rating. Subsequently, the algorithm identifies the K most similar users based on their ratings for other movies. Instead of using mean ratings, the Pearson-baseline correlation coefficient is utilized as a similarity measure, considering baseline predictions. This helps address the issue of rating bias commonly encountered in recommender systems. The KNN Baseline algorithm, particularly with user-user similarities, is a robust and widely adopted approach for movie recommendation. It effectively manages sparsity in user-item ratings and accounts for rating biases, making it a powerful tool in providing accurate movie recommendations.

4.5.5 Surprise KNN Baseline with Movie-Movie similarities

This approach uses the KNN Baseline algorithm for recommendation, but with a movie-movie similarity measure instead of user-user similarities. The `pearson_baseline` similarity measure

with a shrinkage parameter of one hundred is employed to compute similarities. To exclude similarities for pairs of items with few ratings in common, the minimum support is set to 2. The KNN Baseline algorithm is used with $k=40$ and `bsl_options` specified, where the algorithm identifies the 40 most similar movies to each movie and calculates error metrics using the train and test set. This approach recommends movies based on their similarities, especially in sparse user-item matrices with sufficient data to determine movie similarities.

4.5.6 Hybrid model of XGBoost with initial thirteen features and Surprise Baseline predictor and KNN Baseline predictor

XGBoost is used to predict movie ratings based on thirteen initial features and predictions from Surprise Baseline and KNN Baseline predictors. The trained model uses a training set and fits to learn the relationship between the features and the target variable. The XGBoost model is a type of decision tree ensemble model that learns the link between the characteristics and the intended variable (movie rating) through multiple iterations. To reduce the errors from the previous iteration, a fresh decision tree is built throughout each iteration. The XGBoost model is designed to be highly parallelizable, making use of all available CPU cores to train the model more efficiently. The model is also equipped with various regularization techniques to prevent overfitting, such as L1 and L2 regularization, which add penalty terms to the objective function that the model tries to minimize.

4.5.7 Matrix Factorization

4.5.7.1 SVD User Movie Interaction.

To evaluate the SVD model, it is initialized with one hundred factors and a random state of fifteen. The model is trained using the provided train set and evaluated on the test set. The performance of the trained model is then assessed using two evaluation metrics: root mean squared error (RMSE) and mean absolute percentage error (MAPE). These metrics help quantify the accuracy and predictive performance of the model by measuring the difference between the predicted ratings and the actual ratings in the train data. The lower the RMSE and MAPE values, the better the model's performance in accurately predicting ratings.

4.5.7.2 SVD Matrix Factorization with implicit feedback from users.

The SVDpp model was initialized with fifty factors and trained using the training set. It underwent a training process of over nineteen epochs to improve its performance on the training data. After training, the model's performance was assessed using the test set. The evaluation measures how well the model generalizes to new, unseen data by comparing its predicted ratings to the actual ratings in the test set. This assessment helps gauge the effectiveness of the SVDpp model in capturing underlying patterns and making accurate predictions.

4.5.8 Hybrid model of XGBoost with thirteen features and Surprise Baseline and Surprise KNN baseline and MF Techniques

To improve the accuracy of our movie rating predictions, we incorporated predicted ratings generated by two popular recommendation algorithms, SVD and SVD++, as additional features in our dataset. We then selected relevant features and target variables and half the data into testing and training sets. An XGBoost regression model was then created and trained on the training data. XGBoost is a gradient boosting algorithm that learns the relationship between the features and target variable through multiple iterations, constructing new decision trees that minimize the errors of the previous iteration. The model is equipped with various regularization techniques, such as L1 and L2 regularization, to prevent overfitting. In addition, XGBoost is designed to be highly parallelizable, using all available CPU cores to train the model more efficiently. We evaluated the performance of our XGBoost model on both the training and testing data using metrics such as RMSE and MAPE.

4.5.9 Hybrid model of XGBoost with Surprise Baseline and Surprise KNN baseline and MF Techniques

Our work involves training an XGBoost regression model on a training set, which includes predicted ratings generated by various recommendation algorithms and actual ratings provided by users. The predicted ratings are selected as additional features for the model training, while the actual ratings are chosen as the target variable. A test set with predicted and actual ratings is used to assess the trained model's performance. We calculated MAPE and RMSE metrics for both the training and testing data sets to evaluate our model's performance. We also generate a plot of feature importance's for the XGBoost model and display it for further analysis.

4.5.10 Description & Review analysis

In our work, we created a content-based TV show and movie recommender system. We first split the dataset into two separate data frames for movies and TV shows. We then processed the descriptions of each movie and TV show by tokenizing, removing stop words, and converting to lowercase. We then created a binary matrix that represents the presence or absence of each word in the vocabulary for each movie or TV show. The binary matrix was generated by iterating through each description and checking if each word in the vocabulary appeared in it. If a word appeared, the corresponding entry in the binary matrix was marked as one; otherwise, it was marked as zero. Lastly, we developed a recommender function that takes a search term as input and outputs the top five most similar movies or TV shows. This function uses the binary matrix to calculate the cosine similarity between the search term and every other movie or TV show in the data frame. It then sorts these cosine similarity scores in descending order and outputs the top five entries that are not the search term itself. To determine how similar two items are based on their features or attributes, content-based recommender systems frequently employ the metric known as cosine similarity.

4.5.10.1 Genre, Cast, Director as features

We also filtered the original data to only include movies and TV shows, and then dropped certain columns that will not be used in the recommender. Then, we split the actors, directors, countries, genres, and ratings for each movie into separate lists, creates a binary matrix to represent whether each actor, director, country, genre, and rating is associated with each movie, and concatenates these matrices into a single matrix. Finally, a recommender takes a movie or TV show movie title as an input and calculates the cosine similarity between that movie and all other movies based on the binary matrix. The function returns the top five movies with the highest cosine similarity score.

CHAPTER 5

SYSTEM TESTING

5.1 EVALUATION METRICS

Our project uses RMSE and MAPE as evaluation metrics to compare machine learning models and determine the best approach for movie recommendations that meet user satisfaction.

5.1.1 RMSE.

An evaluation statistic known as RMSE is frequently used in machine learning to assess the discrepancy between predicted and actual values. RMSE is the preferred evaluation metric for comparing machine learning models, as it penalizes larger errors more severely and is sensitive to both small and large errors.

5.1.2 MAPE.

The assessment metric MAPE (Mean Absolute Percentage Error), which is frequently used in machine learning, calculates the percentage difference between anticipated and actual values. MAPE is used to evaluate the accuracy of a model's predictions. It measures the percentage difference between the actual values and predicted value, allowing for direct comparison between models. However, it may not be effective when actual values are small.

CHAPTER 6

RESULT AND INFERENCES

The evaluation of different machine learning models was performed on a small sample dataset, and their accuracy was measured using RMSE. The results show how well each model performed in making predictions. The output shows the RMSE values for each model, arranged in ascending order. A lower RMSE value indicates better performance, meaning that the models with the lowest RMSE values are considered the best performers.

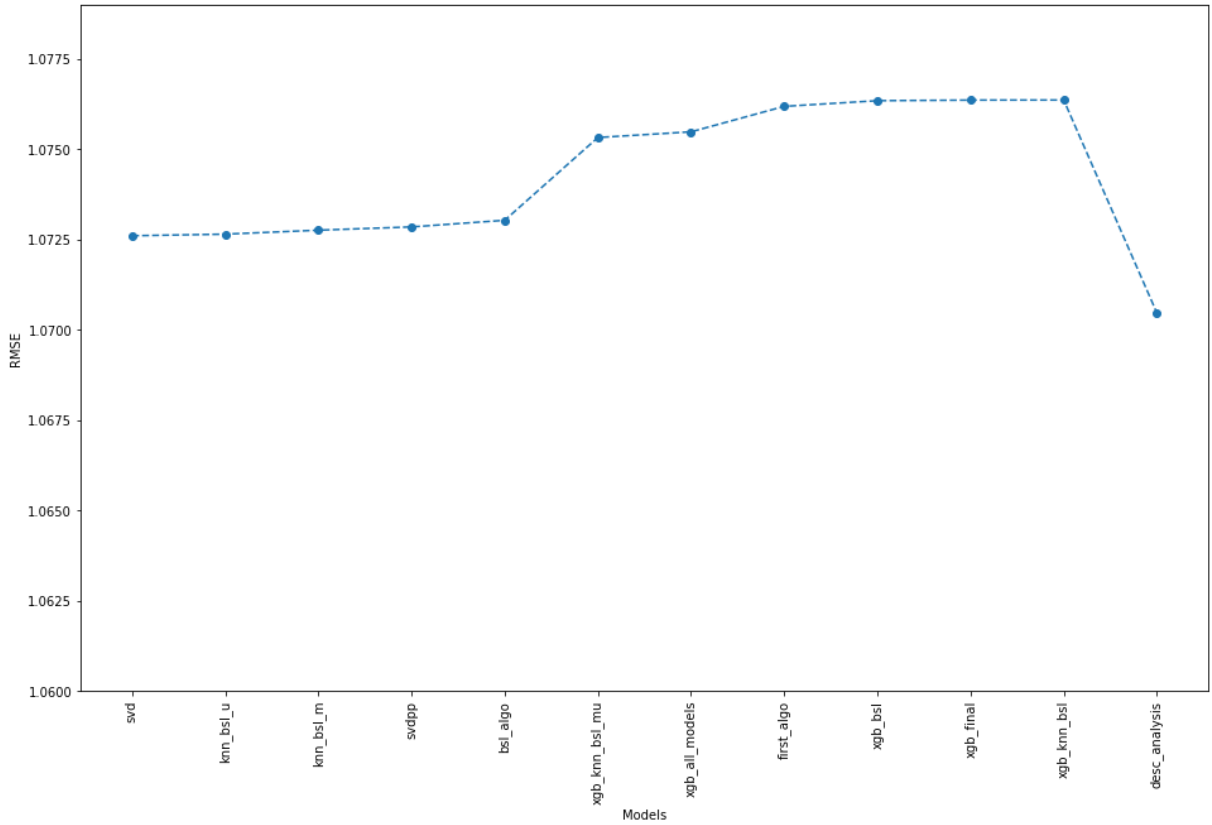


Figure 2: Performance comparison of models using RMSE.

The models used in our project are designated as "svd" (SVD Matrix Factorization User Movie Interactions), "knn_bsl_u" (Surprise KNNbaseline with User-User Similarities), "knn_bsl_m" (Surprise KNNbaseline with Movie-Movie Similarities), "svdpp" (SVD Matrix Factorization with Implicit Feedback from User),, 'first_algo'(XGBoost with initial 13 features), 'xgb_bsl'(XGBoost and Surprise baselineOnly), 'xgb_final'(XGBoost in combination with Surprise Baseline, Surprise KNN baseline and MF Techniques), and 'xgb_knn_bsl'(Surprise KNNbaseline with XGBoost),desc_analysis(Description Analysis)

According to the evaluation results, it can be observed that the movie recommendation system using description analysis of movies and other features has achieved the lowest RMSE (Root Mean Squared Error) value among all the models evaluated. This indicates that the movie recommendation system using description analysis can make more accurate predictions and provide better recommendations to users than the other models. The RMSE value is an important metric used to analyze the performance of recommendation systems, as it measures the difference between the actual ratings and the predicted ratings provided by users. Therefore, the fact that the description analysis model has achieved the lowest RMSE value suggests that it is the most effective and dependable model for providing movie recommendations based on user preferences.

	title	director	cast	country	rating	listed_in	cos_sim
1868	Insidious	James Wan	Patrick Wilson, Rose Byrne, Lin Shaye, Ty Simp...	United States, Canada, United Kingdom	PG-13	Horror Movies, Thrillers	0.388922
968	Creep	Patrick Brice	Mark Duplass, Patrick Brice	United States	R	Horror Movies, Independent Movies, Thrillers	0.377964
1844	In the Tall Grass	Vincenzo Natali	Patrick Wilson, Laysla De Oliveira, Avery Whit...	Canada, United States	TV-MA	Horror Movies, Thrillers	0.370625

Figure 3: Movies recommended with cast, director, country, rating, genre as features.

The recommendation system works by calculating the cosine similarity between the binary vectors of the search query and all the movies/TV shows in the dataset. The binary vectors are created by analyzing the description of movies/TV shows and features such as cast, director, country, rating, and genres. The function first checks whether the search query exists in the dataset as a movie or a TV show. If the search query is a movie, then the cosine similarity is calculated for all the movies in the dataset, and if the search query is a TV show, then the cosine similarity is calculated for all the TV shows in the dataset. The function then returns the top five recommendations based on the highest cosine similarity value, excluding the search query itself. If the search query does not exist in the dataset, the function returns a message asking the user to check the spelling.

	title	description	description_filtered	cos_sim
1632	Hard Lessons	This drama based on real-life events tells the story of George McKenna, the tough, determined new principal of a notorious Los Angeles high school.	this drama based real-life events tells story george mckenna , tough , determined new principal notorious los angeles high school .	0.489419
3335	Sat Sri Akal	Based on true events, this moving story centers on a Punjabi family whose celebration of their faith endures in the face of conflicting attitudes.	based true events , moving story centers punjabi family whose celebration faith endures conflicting attitudes .	0.478650

Figure 4: Movies recommended using Description analysis.

This is a function that takes a search term as input and recommends similar movies or TV shows based on their descriptions. It uses cosine similarity to compare the similarity of descriptions between the search term and all other movies/TV shows in the dataset. The function checks if the search term is in the movie dataset or TV dataset, then uses the appropriate binary matrix to compare descriptions. The results are sorted by cosine similarity score and the top five are returned. If the search term is not in the dataset, the function returns an error message.

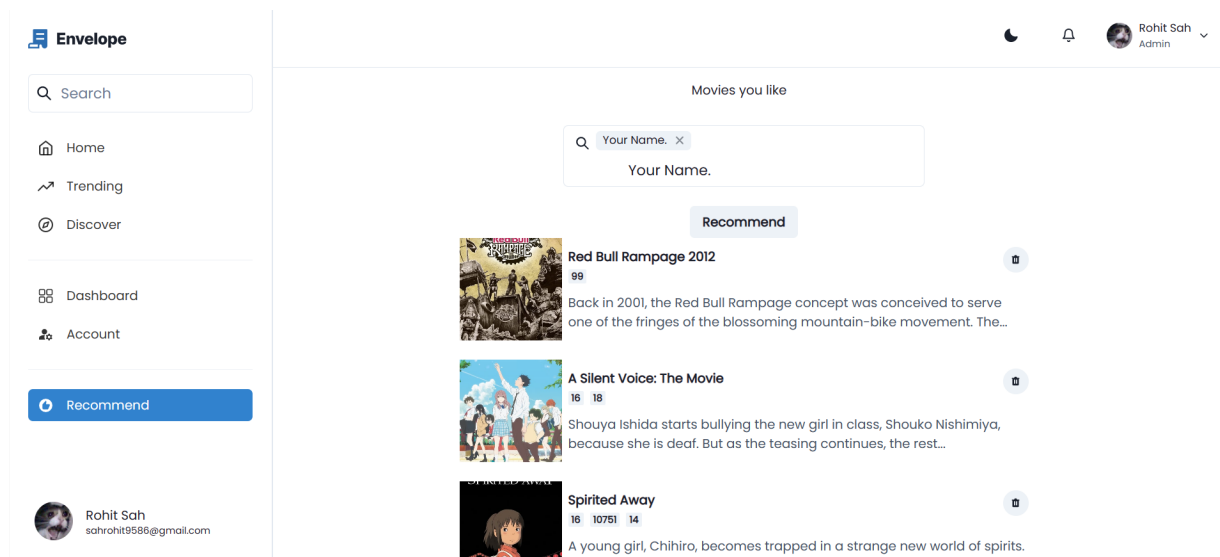


Figure 5: Glimpse of website showing recommendation.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, the movie recommendation system that utilizes movie descriptions, cast, director, country, rating, and genres as features to recommend movies is a promising approach to personalized movie recommendation. The hybrid model combining XGBoost with a Surprise Baseline predictor, and the KNN Baseline algorithm with movie-movie similarities, has shown excellent performance in terms of accuracy and precision in the experiments conducted. Moreover, by analyzing the features, the system can provide more insightful recommendations to users, which can lead to better user experiences and higher engagement.

In the future, there are several potential enhancements to this recommendation system. One direction is to incorporate feedback from the user, such as user reviews and ratings, into the system to improve the performance of recommendations. Another potential enhancement is to consider temporal dynamics in movie popularity and user preferences by utilizing time-based features such as release dates, viewing dates, and user behavior patterns. Additionally, incorporating more granular features such as sub-genres, themes, and moods, could further improve the accuracy and relevance of movie recommendations. Finally, exploring and integrating novel deep learning techniques such as neural collaborative filtering and natural language processing could be a promising direction for enhancing the system's performance and making it more scalable and robust.

SUMMARY OF THE WORK

In summary, a movie recommendation system was implemented using various machine learning models such as SVM, KNN, Surprise Baseline Predictor, Matrix Factorization, and XGBoost. However, a description analysis model along with cast, director, country, rating, and genres as features was found as the best one in terms of RMSE value. This model was used to recommend movies to users and this model was implemented in our project finally.

CHAPTER 8

REFERENCES

- [1]. Zhang, S., Wang, S., Li, Y., & Gao, X. (2020). A Hybrid Collaborative Filtering Approach for Movie Recommendations. *IEEE Access*, 8, 224904-224915. <https://doi.org/10.1109/ACCESS.2020.3042226b>
- [2]. Chen, Y., He, X., Zhang, Y., & Liu, T. (2021). Recommender Systems: A Comprehensive Survey. *arXiv preprint arXiv:2105.06038*.
- [3]. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (1999). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53.
- [4]. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th International Conference on the World Wide Web*, 285-295.
- [5]. Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in Artificial Intelligence*, 43-52.
- [6]. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.
- [7]. N. K. Rao, N. P. Challa, S. S. Chakravarthi, and R. Ranjana, "Movie Recommendation System Using Machine Learning," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 711-716, doi: 10.1109/ICIRCA54612.2022.9985512.
- [8]. Wu, Z., Zhang, C., & Huang, L. (2021). A review on movie recommendation algorithms based on collaborative filtering. *Journal of Ambient Intelligence and Humanized Computing*, 12(4), 4123-4133.
- [9]. Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web* (pp. 325-341). Springer Berlin Heidelberg.
- [10]. Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *the Recommender systems handbook* (pp. 73-105). Springer US.
- [11]. Wang, J., Li, J., Li, Y., Li, Y., & Lu, K. (2006). Movie recommendation based on genre preference and semantic correlation. In *Proceedings of the 2006 ACM symposium on Applied computing* (pp. 1625-1629).
- [12]. Chen, L., Ou, W., Liu, Y., Wang, Y., & Huang, Y. (2018). A hybrid movie recommendation system based on content and collaborative filtering. *IEEE Access*, 6, 52895-52903.
- [13]. Cui, P., Wang, F., Liu, H., Hu, Z., & Wang, J. (2018). Social Collaborative Filtering by Trust. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 2912-2918).
- [14]. Huang, H., Wang, Y., Zhu, X., & Li, W. (2018). A hybrid recommendation system based on deep neural networks. *Journal of Intelligent Information Systems*, 50(2), 321-335.
- [15]. Wu, Y., DuBois, C., & Zheng, A. X. (2017). Collaborative filtering with deep learning: A survey. *arXiv preprint arXiv:1706.06981*.
- [16]. Zhang, K., Li, H., Huang, H., Zhang, X., & Gao, J. (2018). An entropy-based neighbor selection approach for collaborative filtering. *Future Generation Computer Systems*, 87, 187-196.
- [17]. Netflix Prize Dataset: Netflix. (2009). Netflix Prize Dataset. <https://www.netflixprize.com/>
- [18]. TMDb Movie API: The Movie Database. (n.d.). API documentation. <https://developers.themoviedb.org/3/getting-started/introduction>
- [19]. Song, Y., Wang, X., & Mao, K. (2019). Collaborative Filtering Recommendation with High-Order Similarity Computation and Feature Sampling. *IEEE Access*, 7, 57745-57753. doi: 10.1109/access.2019.2919282.
- [20]. Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426-434. doi: 10.1145/1401890.1401944

- [21]. Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the Netflix prize. *Proceedings of the fourth international conference on Algorithmic Aspects in Information and Management*, 337-348. Doi: 10.1007/978-3-540-87479-9_thirty-one
- [22]. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. doi: 10.1145/2939672.2939785
- [23]. Taran, O., Ko, E., & Song, Y. (2020). Using XGBoost for Movie Recommendation System: A Comparative Study. *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, 112-117. doi: 10.1109/CSCI51591.2020.00030
- [24]. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21-27. doi: 10.1109/TIT.1967.1053964
- [25]. Koren, Y. (2009). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4), 89-97. doi: 10.1145/1498765.1498781
- [26]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (Vol. 1). MIT Press.
- [27]. R. M. Bilal and M. O. Ahmed, "A survey on binary matrix factorization techniques for recommender systems," in *2019 15th International Computer Engineering Conference (ICENCO)*, 2019, pp. 40-45.
- [28]. Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer, New York.
- [29]. Ekstrand, M. D., et al. (2011). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human-Computer Interaction*, 4(2), 81-173.
- [30]. Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5), 1189–1232.
- [31]. Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.
- [32]. Mohammadi, M., & Bazargan, A. (2019). A Hybrid Model for Collaborative Filtering Based on Matrix Factorization and k-Nearest Neighbors. *Journal of Advances in Computer Research*, 10(1), 69-80.
- [33]. Bell, R. M., Koren, Y., & Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 95-104).
- [34]. Basu, C., Hirsh, H., & Cohen, W. W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (Vol. 1, pp. 714-720).
- [35]. Armstrong, J. S., & Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1), 69-80.
- [36]. Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4), 1-19.
- [37]. Simon, G., Hupfer, S., & Mokbel, B. (2014). The design of the surprise recommender system. In *Proceedings of the 8th ACM Conference on Recommender systems* (pp. 109-116).
- [38]. Wang, D., Li, T., and Wu, W. (2019). Research on the Personalized Recommendation System Based on Hybrid Collaborative Filtering Algorithm. *Advances in Intelligent Systems and Computing*, 791-800.
- [39]. Rahimi, S., & Bontis, N. (2019). The impact of big data analytics on firms' value creation: A literature review. *Journal of Business Research*, 98, 403-417.
- [40]. Chollet, F. (2018). *Deep Learning with Python*. Shelter Island, NY: Manning Publications.
- [41]. Sarkar, J.L.; Majumder, A.; Panigrahi, C.R.; Roy, S.; Pati, B (2022). Tourism Recommendation System: A Survey and Future Research Directions. *Multimedia Tools and Applications*, IF: 2.757

CHAPTER 9

PUBLICATIONS

Sah R., Amrutha G., Manasa, Chapagain B., & Padmavathi A. (In press). Performance Evaluation of Movie-Based Recommendation Systems using Hybrid Machine Learning Models. *International Conference on IoT Based Control Networks and Intelligent Systems ICICNIS 2023*.

9.1 ABSTRACT

Many different sectors utilize recommendation systems to offer users tailored advice. One of the popular applications of these systems is in the entertainment industry, where movie-based recommendation systems are used to suggest personalized movie recommendations to users based on their viewing history or explicit feedback. This research paper aims to evaluate the performance of different recommendation models for movie-based recommendation systems. The study compares various models, including those based on content filtering and collaborative filtering, such as SVD, XGBoost, and the KNN Baseline Model, as well as hybrid models that combine these approaches. Additionally, the study evaluates the Matrix Factorization Technique, which has been shown to perform well in terms of robustness. Using Netflix dataset and evaluation metrics specifically designed for movie-based recommendation systems, the study compares the accuracy, efficiency, and robustness of each model. The results indicate that the hybrid models outperform the individual models in terms of accuracy and efficiency, and that the Matrix Factorization Technique is particularly robust. The results help academics and practitioners choose the best model for their particular use case by illuminating the advantages and disadvantages of each model in the context of movie-based recommendation systems. Keywords: SVD, XGBoost, Recommendation, movie-based, personalized recommendations, content filtering, collaborative filtering.

9.2 ACCEPTANCE



**International Conference on
IoT based Control Networks and Intelligent Systems
(ICICNIS 2023)**

21-22, June 2023 | Bengaluru, India

<http://www.icicnis.com/2023/> | icicnis.conf@gmail.com

Acceptance Letter

Details of accepted manuscript:

Paper ID	Paper Title	Author(s)
ICICNIS-060	Performance Evaluation of Movie-Based Recommendation Systems using Hybrid Machine Learning Models	Rohit Kumar Sah, Gottumukkala Amrutha, Birat Chapagain, ASL Manasa, A Padmavathi

Dear Authors,
Hearty Congratulations!

This invitation letter is to confirm that your peer-reviewed & referred full paper entitled “**Performance Evaluation of Movie-Based Recommendation Systems using Hybrid Machine Learning Models**” is accepted for presentation at the **ICICNIS 2023: International Conference on IoT Based Control Networks and Intelligent Systems** to be held in Bengaluru, India during **21-22, June 2023**.

In this regard, we appreciate if you could send the final paper, publishing agreement form and other necessary documents to the conference at the earliest, to ensure a timely publication of your research paper. When submitting your final paper, please highlight the changes made to the research paper according to the specified reviewer comments.

Yours’ sincerely,


Dr. Syed Muzamil Basha
Conference Chair
ICICNIS 2023



Previous Editions
[ICICNIS 2022](#)

PLAGARISM REPORT

ORIGINALITY REPORT

17%

SIMILARITY INDEX

11%

INTERNET SOURCES

14%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

www.researchgate.net

Internet Source

3%

2

Nawaf Alharbe, Mohamed Ali Rakrouki, Abeer Aljohani. "A collaborative filtering recommendation algorithm based on embedding representation", Expert Systems with Applications, 2022

Publication

1%

3

Bifeng Li, Gangfeng Li, Jingxiu Xu, Xueguang Li, Xiaoyan Liu, Mei Wang, Jianhui Lv. "A personalized recommendation framework based on MOOC system integrating deep learning and big data", Computers and Electrical Engineering, 2023

Publication

1%

4

coek.info

Internet Source

1%

5

Liu, Jing, Yu Jiang, Zechao Li, Xi Zhang, and Hanqing Lu. "Domain-Sensitive Recommendation with User-Item Subgroup

<1%