Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Babylon

College of Information Technology

Software Department

Predicting Users' Preferences on Movies and Providing Recommendations Based on a Proposed Hybrid Approach

A Dissertation

Submitted to the Council of the College of Information Technology University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Technology-Software

BY

Sadam Hamdan Ahmed Abdullah

Supervised By

Prof. Dr. Abbas Fadhil Mohammed Ali

2023 A.D. **1445 A.H.**

بسم الله الرحمن الرحيم

ذَٰلِكَ فَضۡلُ ٱللَّهِ يُؤۡتِيهِ مَن يَشَآءُۚ وَٱللَّهُ ذُو

ٱلۡفَضۡلِ ٱلۡعَظِيمِ ٤

صدق الله العلي العظيم

سورة الجمعة : الآية ٤

Declaration

I hereby declare that this dissertation entitled, submitted to University of Babylon as fulfilment of requirements for the degree of Doctor in Information Technology \ Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own.

Signature:

Name: **Sadam Hamdan Ahmed Abdullah**

Date: / / 2023

Supervisor Certification

I certify that the dissertation entitled **"Predicting Users' Preferences on Movies and Providing Recommendations Based on a Proposed Hybrid Approach"** was prepared under my supervision at the department of Software/ College of Information Technology/the University of Babylon as partial fulfillment of the requirements of the degree of Doctor in Information Technology - Software.

Signature:

Supervisor Name: **Prof. Dr. Abbas Fadhil Mohammed Ali**

Date: / / 2023

Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled **"Predicting Users' Preferences on Movies and Providing Recommendations Based on a Proposed Hybrid Approach"** for debate by the examination committee.

Signature:

Asst. Prof. Dr. Sura Zaki Naji

Head of Software Department

Date: / / 2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled **(Predicting Users' Preferences on Movies and Providing Recommendations Based on a Proposed Hybrid Approach)** presented by the student (**Sadam Hamdan Ahmed Abdullah**) and examined him in its content and what is related to it, and that, in our opinion, it is adequate with (Good) standing as a dissertation for the degree of Doctor of philosophy in Information Technology - Software.

Signature:
Name: Dr. Hazeem Baqer Taher
Title: Professor
Date: / / 2023
(**Chairman**)

Signature:
Name: Dr. Mohammed Abdullah Naser
Title: Professor
Date: / / 2023
(**Member**)

Signature:
Name: Dr. Ahmed Habeeb Said
Title: Asst. Professor
Date: / / 2023
(**Member**)

Signature:
Name: Dr. Sura Zaki Naji
Title: Asst. Professor
Date: / / 2023
(**Member**)

Signature:
Name: Dr. Suad Abdullelah Abdull-Hussain
Title: Asst. Professor
Date: / / 2023
(**Member**)

Signature:
Name: Dr. Abbas Fadhil Mohammed
Title: Professor
Date: / / 2023
(**Member and supervisor)**

Approved by the Dean of the College of Information Technology, University of Babylon

Signature:
Name: Dr. Wesam Sameer Bhaya
Title: Professor
Date: / / 2023
(**Dean of Collage of Information Technology)**

Dedication

To my beloved and my idol

… my father

To the one who enlighten my life with her love and patience

… my mother

To my soulmate and all my life

… my wife

To my eyes

… my brothers and sisters

To all my lovely children

Sadam

Acknowledgement

Foremost, I am highly grateful to Allah (God) for His unlimited blessings that continue to flow into my life, and because of You, I made this through against all odds.

To my supervisor, **Prof. Dr. Abbas Fadhil Mohammed** I feel highly indebted to you. I am deeply grateful for your suggestions on this topic, and without your support, comments, and guidance, it would be difficult to finish this work.

To my wonderful mother: words cannot express my appreciation to you, sweetest mother. I wish you could be with me at this moment and see what I've come to. I pray that God will make you happy and heal you from your illness .Thank you for the amount of love you have given us from a young age until now .

Thank you to my sisters for the endless support. My mind cannot write that I love you, and I find in my heart nothing but gratitude for what you have given me throughout my study. Furthermore, many thanks to my wonderful wife, you are supportive and helpful at every stage of this thesis. My son and daughter: I am sorry for being busy and did not spend much long time with you, especially in the last days of this work.

To my family, and my friends: I give you all thanks for your belief in me, your constant support, encouragement, and cooperation at all times.

Special thanks to my colleagues at the central Preparatory School for boys, headed by Mr. Omar Adel, for the great help they gave me.

Last but not least, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and I apologize to them for not being able to mention them by name here, but they are in my heart.

Sadam Hamdan Ahmed

Abstract

With the increasing popularity of digital streaming platforms around the world, a plethora of films are now easily available for viewing. Although there are thousands to choose from, narrowing them down based on personal interest can prove to be a problem. This problem is addressed by movie recommendation systems and offer personalized recommendations to users. The traditional movie recommendation systems depend mostly on single approaches such as collaborative filtering or content-based filtering. The two forms of collaborative filtering are user based collaborative filtering and item based collaborative filtering. However, there are also limitations associated with these techniques of filtering. Collaborative filtering suffers the "cold start" problem for the new users and new contents whereas content based filtering does not offer enough diversified recommendations.

Therefore, this dissertation aims at boosting movie recommendation systems using the mix of collaborative filtering and content based filtering. This suggests that, by analyzing the user's preferences (i.e., their tastes for different movies, characters or film directors) and various attributes of a film or movie, the proposed (i.e., suggested) system will be capable of delivering customized recommendations to the users .This dissertation proposes a hybrid recommendation system based on the strengths of collaborative and content-based filtering. In order to circumvent the issues with every approach, the suggested system merges the potential of collaborative filtering to discover individuals with likable tastes and that of content-based filtering in reviewing movie characteristics. This way both new and old users get a different and appropriate movie recommendation at all times.

The efficacy of the proposed hybrid system is evaluated using two popular datasets: MovieLens 100K and MovieLens 20M. These findings are encouraging as the MovieLens-100K data set has a MAE of 0.79 and an RMSE of 0.92 whereas the MovieLens 20M data set has a MAE of 0.8 These statistics demonstrate how precise the recommendation system is by matching the user preferences and the view history. These results show that it is a highly useful tool for giving precise but varied recommendations that improve user experience and overcome the challenges of a modern culture rich with digital content.

Keywords: recommendation system, collaborative filtering, content-based filtering, machine learning, Deep Learning.

List of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	**Artificial Neural Network**
API	**Application Programming Interface**
CF	**Collaborative Filtering**
CNN	**Convolution Neural Network**
DL	**Deep Learning**
DNN	**Deep Neural Network**
GUI	**Graphical User Interface**
MAE	**Mean Absolute Error**
MAPE	**Mean Absolute Percentage Error**
ML	**Machine Learning**
MSE	**Mean Square Error**
RMSE	**Root Mean Square Error**
RNN	**Recurrent Neural Network**
RS	**Recommendation System**

List of Algorithms

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Background

The entertainment industry has undergone transformative evolution in the era of information abundance and technological advancements. The emergence of digital platforms and streaming services on the Internet has brought about significant changes in our engagement with movies. This shift has led to an overwhelming array of available content, providing viewers with an unprecedented range of films across various genres, time periods, and cultures [1], [2]. However, this multitude of options poses a considerable challenge: guiding users through this vast cinematic landscape and facilitating their exploration of movies that align with their individual preferences.

By harnessing technologies, such as movie recommendation systems, online platforms can assist users in efficiently finding movies tailored to their tastes. These systems utilize algorithmic analysis based on factors, such as user search history and previous viewing choices, to suggest relevant films. The ultimate objective is to enhance reliability by delivering accurate suggestions that closely match what individuals are seeking [3], [4].

The unprecedented expansion of digital platforms, streaming services, and online movie databases has resulted in the immense proliferation of films. As of my most recent update in September 2021, prominent platforms such as Netflix, Amazon Prime Video, Hulu, Disney+, and numerous others offer vast catalogues comprising thousands of movies [5], [6].

Moreover, new titles are constantly being introduced into these platforms. Consequently, users face the overwhelming task of navigating through this enormous collection of content that spans a wide range of genres and styles. This complex landscape poses a significant challenge for individuals seeking to discover films that align with their tastes and preferences.

Movie recommendation systems have emerged as essential tools in digital entertainment [7]. These systems utilize various algorithms and techniques to analyze user behavior, preferences, and historical interactions with movies. Based on this analysis, these recommendation systems can provide personalized movie suggestions based on an individual's specific tastes and preferences. By utilizing insights into user behavior and predicting their interests, these systems assist users in navigating through the vast sea of options available to them, ultimately helping them discover films that align with their unique preferences [8].

The concept of movie recommendations is not just about enhancing user convenience; it has become a crucial aspect of the business model of the modern entertainment industry. Streaming services rely heavily on user engagement and satisfaction, while retaining subscribers. The ability to provide accurate and engaging movie recommendations directly affects user retention, viewing hours, and overall satisfaction. Therefore, the development of effective recommendation systems is not only a matter of user convenience, but also a strategic imperative for digital entertainment platforms [9].

1.2 Problem Statement

The main characteristic of standard movie recommendation methods is that they are mostly based on a single algorithm or data source. As effective as they may be, to a certain extent, these systems are not good at generalizing. Unique preferences among users are a reality, and it is not always possible to apply the one-size-fits-all technique when making recommendations for movies that speak to an individual audience.

Given this limitation, such recommendations may be less effective than they could be, eventually resulting in decreased user involvement and satisfaction.

1. The generalization challenge is twofold. First, there is a need to develop recommendation algorithms that are effective in understanding different and complex individual tastes. The second consideration is that they need to change with customers' tastes over time. Thus, examining alternatives that outstrip a single-algorithm approach is necessary to provide continuous and up-to-date movie recommendations.

2. In contrast, the challenges faced by movie recommendation systems include the complexities of big data that come as well. The digital age has led to vast amounts of data available to researchers through various media, such as user interactions, movie metadata, and social network information. Effective recommendation systems entail the management and harnessing of enormous amounts of information.

3. Big data are a multifaceted challenge. The first issue is to integrate data: how does one collect and use information from various sources in an efficient manner that adheres to data quality measures and consistency guidelines? In addition, large amounts of data pose scaling problems. Managing a vast and rapidly growing user base as well as an expanding movie library for recommendation systems require a practical solution. Third, the 'cold-start' problem further confounds the issue of big data, requiring tactics to generate useful recommendations for users with little or no prior history and newborn movies that have not had massive audience engagement.

1.3 Research Questions

This dissertation attempts to answer the following questions.

1. What measures can be put into place for recommendation systems to adjust to fluctuations in user preferences such that recommended movies are always pertinent and interesting?

2. How can various data sources, such as user interactions, movie metadata, and social network information, be integrated and utilized to improve the recommendation accuracy in movies?

3. How can a recommendation system design be made scalable to accommodate the continuous influx of big data generated by users, growing movie libraries while ensuring high recommendation values?

1.4 Aim and Objectives

This dissertation aims to advance the domain of movie recommendation systems with special consideration for generalization and big data, which present great challenges in modern digital entertainment. Thus, it seeks to improve the accuracy, customization, and scalability of film recommendations aimed at enhancing users' cinematic experiences. To achieve this aim, the following objectives were pursued:

1. Research and advancement of recommendation algorithms and models that are flexible and powerful enough to capture and satisfy a wide range of changing preferences of individual users is key.

2. Investigate adaptive recommendation approaches to maintain the relevance of movie recommendations by constantly adjusting them according to users' changing preferences.

3. Develop efficient data integration methods that incorporate data cleaning techniques to leverage the heterogeneity of multiple data sources, such as user interaction, film metadata, and social media data, to boost the recommendation precision.

4. Design scalable recommendation system architectures and algorithms for effectively dealing with the growing amount of user-generated content and extensive film collection while still providing good recommendations.

1.5 Research Challenges

The development and improvement of movie recommendation systems have some challenges and implications. These challenges can be summarized as follows:

1. Diverse User Preferences: Given that user preferences are diverse by nature, movie recommendation systems must utilize effective algorithms and models that can effectively capture and adapt to these tastes and predispositions. The difficulty lies in mastering personal tastes so that users receive recommendations tailored for different user profiles.

2. Data Integration: The major challenge is combining and effectively using data from various sources, such as user interactions, movie metadata, and social-network-based information. Another essential part of boosting recommender system accuracy is guaranteeing data quality and consistency, as well as integrating a tremendous amount of data into such algorithms.

3. Scalability: However, it produces massive amounts of data that make the digital entertainment ecosystem scalable. A recommendation system should be able to process such an overwhelming amount of information and provide good recommendations.

4. User Satisfaction and Usability: In essence, a movie recommendation system depends on user satisfaction and use. The fundamental problem for the system is that it should provide recommendations that users will value and be user-friendly. It is important to conduct a complete user study to prove the relevance of the developed recommendations and evaluate user satisfaction.

1.6 Dissertation Contribution

The primary contributions of this dissertation can be summarized as follows:

1. This dissertation presents a proposed hybrid recommender system utilizing collaborative filtering and deep learning approaches. The system incorporates both user-based as well as content based recommendations approach hence achieves higher level of accuracy in recommending movies thus outdoing previous systems.

2. This dissertation utilizes several learning algorithms like linear regression, gradient boosting regression, decision tree regression, stochastic gradient descent regression, and Bayesian ridge regression. Every algorithm is carefully applied in order to develop an effective prediction about users' movie preferences and build strong recommendations system for movies.

1.7 Related work

Recently, a multitude of scientists have engaged in extensive data analysis within the online entertainment sector, utilizing a diverse range of methodologies. Numerous studies have been conducted on the primary categories of these procedures, as indicated by scholarly sources[10]. Each methodology has its own advantages, disadvantages, and limitations. Numerous studies have been conducted in this regard.

Abdullah et al. [11] categorization begins with the partitioning of objects into subsets according to previously established criteria. Objects can easily be sorted into designated groups if their classes are defined at the time they are created. "Supervised" classification is a term used to describe this method of organizing the data. An example of a rule-based classifier is the rough set classifier, which provides recommendations for classifying objects according to their social and geographical context. This study examines the function of rough set theory in context recommendation and evaluates the efficacy of the J48, K-nearest neighbor, and decision stump classification methods. In terms of accuracy, we obtained a recall rate of 80% and F1 (F-score) of 0.73.

Al Hassanieh et al. [12] introduced two measures of similarity to handle sparsity in data. By including weight in the Pearson equation, a more robust metric was proposed and named Weighted Pearson's Correlation Coefficient (WPCC). Frequency is given by the second equation. The Correlation Coefficient of Pearson Frequency-weighted Pearson's correlation coefficient (FPCC) was proposed. A frequency weight was added to each variable to create a weighted Pearson’s correlation scale. The MovieLens 100k dataset showed that when only 10% of the users were evaluated, the MAE and RMSE for FPCC and WPCC were approximately 1.002 and 1.31, respectively.

Al-Bashiri et al. [13] proposed two similarity measures as a way to increase the precision of conventional collaborative classification while simultaneously decreasing sparsity. Developed Cosine Correlation (DCC) and Pearson correlation coefficient (DPCC) (DCOS). To solve the similarity problem and discover the connection between seemingly unrelated users, the system suggests items to the user based on an analysis of their preferences. With respect to the MovieLens 100k dataset, the experimental results showed a mean absolute error (MAE) of 0.77, precision (0.015), and recall (0.22).

Wang et al. [14] proposed a new method for making suggestions: the Multi-Factor Context-Aware Recommendation Method using the Improved Random Forest (MCRIRF) algorithm. By employing a technique that randomly selects features from multiple feature subspaces that are categorized according to the significance of the characteristics, the MCRIRF method improves on the random forest algorithm. To further partition and reduce the dimensionality of context characteristics related to people, things, and contexts, MCRIRF uses an improved random forest method. For the 3-dimensional user-item-context recommendation model, the MCRIRF is used to determine the weights. For users with similar contexts, the MCRIRF (Multi-Contextual Recommender with Item Rating Forecasting) algorithm ultimately recommends a set of top-n items with high forecasting ratings. Simulations were run on the LDOS-CoMoDa and Cycle Share datasets, and six different recommendation methodologies were considered for side-by-side evaluation. According to the results of the experiments, the MCRIRF can improve the accuracy anywhere from 2% to 16% in mean absolute error and 2% to 13% in root mean squared error across both datasets. The results of this study suggest that MCRIRF could be used in context-aware recommendation systems.

Demissie and Mogalla [15] proposed a preference extraction method based on context, using three methods. First, we obtained the low-representational dimensions of context-relevant attributes, which do not probe associations between the levels of several categorical variables. Second, the results of the millennial calculations were used as the basis for the mass analysis. Finally, the preferences for recommendations are obtained from the context of a given block. Additionally, we identified six groups of underlying contexts. We used the data from the LDOSCoMoDa movie dataset. An approximate RMSE of 0.240 and an MAE of 0.149 were achieved in the rating prediction using the proposed model.

Singh et al. [16] created a new method called Context-Aware Recommender System. Using a splitting criterion, this system considers both item and user preferences, making it ideal for use in movie recommendation software. In the proposed method, one physical object is first split into two virtual objects, the relative importance of which is established by looking at their context. An updated dataset is produced as a result of this procedure. The user is then split into two digital representations, each with a unique set of context-dependent characteristics. When there is a large gap between two digital entities, users and things are split apart (users). Efficient recommendations are made using a user-based collaborative filtering system. The results obtained by analyzing the LDOS-CoMoDa dataset show that the proposed methodology is effective in terms of several performance metrics. Using the proposed method, the MAE averaged 0.9024 for all values of k.

Valdiviezo-Diazet al. [17] provided a Bayesian model with the predictive power of a matrix factorization model and the added benefit of being easily interpretable.

Item recommendations are generated by leveraging the information of related people and objects, and the proposed methodology incorporates both user- and item-based collaborative filtering techniques. The results of the four datasets showed improvements over several state-of-the-art benchmarks. These experiments improve the prediction accuracy on specific datasets and achieve better overall performance using the Normalized Discounted Cumulative Gain (nDCG) quality metric.

Shuxian and Sen [18] used the Bayes algorithm to check the MovieLens dataset. A user's unique similarity matrix can be developed using comparable analysis applied to the user's scoring data. With 10 recommendations, they get an F1-score of 0.83, accuracy of 0.72, precision of 0.89, recall rate of 0.78, and precision of 0.89. Furthermore, four to six users were randomly selected to participate in data testing.

Kala and Nandhini [19] proposed a Deep Learning model that learns customer similarity directly, taking into account both sequence and item similarity. All information about the item, its environment, and the factors used to determine the ratings were factored into the model. This model employs a distance measure known as Dynamic Temporal Warping (DTW) to accomplish dynamic temporal matching. The model also uses an architecture known as a Two-dimensional Gated Recurrent Unit (2D-GRU). This method will allow for a more accurate and efficient identification of patterns within temporal and geographical settings by effectively addressing the challenge of nonlinearity in the temporal dimension when assessing similarity. The proposed tailored recommendation system architecture is tested on the LDOS-CoMoDa dataset to evaluate its efficacy and practicality. These experimental findings support the efficiency and potential of the proposed design. Including Metrics in Cross-Domain Structures. CCCFNet Cross-

DTW-2DGRU model under scrutiny. The top five recall rates are 0.3586 and 0.5456, respectively.

Zhang et al. [20] proposed an efficient recommendation system that overcomes the temporal complexity that plagues traditional CF approaches, rendering them unsuitable for real-world recommendation systems. The model utilizes user profile information to divide users into smaller and more manageable clusters. The original user-item matrix is shrunk by creating a weighted slope based on the ratings of a virtual opinion leader that represents the entire cluster. The virtual opinion leader-item matrix is used along with the one-VU technique to generate suggestions. As part of the University of Minnesota's Group Lens Research Project, two actual MovieLens datasets will be acquired: MovieLens 100 K and MovieLens 1M. The datasets were used to evaluate the efficacy of this concept. After evaluating the proposed method against other decomposition strategies, it was found to be inferior to singular value decomposition (SVD) and SVD++ in terms of prediction accuracy.

Boppana and Sandhya [21] suggested a crawling-based context-aware recommendation model for mining online service user reviews for pertinent contextual words. Negative, neutral, and positive user feedback are aggregated based on density and are used in various ways. User preferences from each cohort were then fed into a Deep Recurrent Neural Network (DRNN). The proposed model was implemented using the NYC Restaurant Rich dataset after initializing the parameters of the neural network model. The model was tested for accuracy using both confidence and recall, and the results showed that it performed better than deep-learning models by a margin of 99.6%.

Manimurugan and Almutairi [22] developed a video recommender model that utilizes principal component analysis and situational awareness. This study relies on a form of collaborative filtering that considers nearby data sources to fill in gaps. The LDOS-CoMoDa dataset was used to address the information gaps. The accuracy of the recommendations was evaluated by dividing the LDOS-CoMoDa dataset by half (75 percent for training, and 25 percent for testing). The effectiveness of this model was evaluated using the mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), as well as recall, accuracy, and f-measure for analysis (RMSE). The F-measure for the proposed model was 86.90%, precision was 93.80%, and recall was 78.54%.

Jeong and Kim [23] suggested using deep learning and contextual features to generate recommendations. In order to speed up the process of predicting scores based on characteristics, the researcher integrates a neural network with an automatic intelligent encoder. The model considers user features, items, and contexts to predict user preferences. Four different data sources were consulted (CARSKIT, DePaul Movie, InCarMusic, and Restaurant Tijuana). When compared to other recommendation systems using the same dataset with many context dimensions, the researcher demonstrated in his methodology that the system achieved a higher precision by 0.01 0.05. In the low-dimensional dataset, precision estimates varied from 0.03 0.09. There was a 0.06 standard deviation in music, 0.07 in food, and 0.01 in film. However, the proposed method suffers from a slower response time than competing recommendation systems. Although quicker than KNN, it lagged behind the competition.

Sridhar and Dhanasekaran [24] developed a methodology for making film suggestions based on users' Facebook likes and interests. The recommendation system was developed using a hybrid model that combines the Deep Belief

Network (DBN) used for classification with Monarch Butterfly Optimization used for selection. MovieLens and Facebook datasets were used to complete the project. The rating prediction performance of the proposed model was approximately 93.21/92.85/0.732 in terms of Precision, Recall, and MAE.

Saleh et al. [25] used collaborative filtering techniques based on the I Bayes algorithm to determine the teaching method. Before beginning the process of making recommendations, it is necessary to collect datasets that can be completed by administering questionnaires to students and collecting their responses. The provided information will be used as a training set to generate recommendations for pedagogical approaches that can be used in classroom settings. The instructor then provides a response, and the acquired results are used as testing data after the training data collection is complete. The naive Bayes algorithm-based recommendation system achieved an accuracy of 90.91 percent accuracy in determining effective learning strategies for individual students.

Sujatha and Abirami [26] proposed using an ANN to provide contextual suggestions based on user input. Customized recommendations in CARS benefit from ANNs' ability of ANNs to learn events and make decisions based on historical data. This achieves the best possible conditions for a user to purchase. In this analysis, each label set is recast as a multipliable classification (MLC) problem. Experiments show that the proposed ANN performs better than state-of-the-art methods on the Trip Advisor and LDOS-CoMoDa Datasets when compared to the Binary Relevance (BR) instance-based classifier, BR decision tree, and multipliable SVM. Table 1.1 shows the previous studies with essential details for each research.

Table 1.1: A Summary of Related Work

Researcher name	Dataset	Method	challenges
Abdullah et al. (2018)	LDOSCoMoDa	Rough set classifiers	Data Sparsity reduction
Al-Bashir and Salehudin (2018)	MovieLens100K	DPCC and DCOS	To solve the similarity problem with the ability to find the relationship between unrelated users.
Manimurugan and Almutairi	LDOSCoMoDa	PCA and context-aware techniques	Based on neighborhood collaborative filtering to overcome data sparsity
Wang et al. (2018)	LDOSCoMoDa	Improved Random Forest (MCRIRF)	Additions of the dimensions of context characteristics
Demissie and Mogalla (2018)	LDOSCoMoDa	Context-based rating prediction and Context-aware	Explore associations between levels of multiple categorical variables
Singh et al. (2019)	LDOSCoMoDa	Context-Aware Recommender System using CF techniques	Sparsity reduction
Valdiviezo-Diaz et al. (2019)	Yahoo and BookCrossingMovieLens 1M FilmTrust	Bayesian model	Sparsity reduction
Shuxian and	MovieLens and	Deep belief used for categorization	Additions of dimensions of context

Sen(2019)	Facebook	(DBN) Network and Monarch Butterfly Optimization	characteristics
Kala and Nandhini (2020)	LDOSCoMoDa	Deep Learning model	non-linearity in the temporal dimension when assessing similarity
Zhang et al. (2020)	MovieLens	SVD++ Markovian Matrix Process (MFMP)	Scalability and explicit feedback
Boppana and Sandhya (2021)	NYC Restaurant Rich	Web crawling and Deep recurrent neural network	When making suggestions utilizing data from different domains, it's important to take into account the perspectives of those affected.
Jeong and Kim (2022)	CARSKIT, DePaul Movie, InCarMusic, Restaurant Tijuana	context-aware deep learning autoencoder neural network	The proposed method has a slower response time than the existing recommender system.
Sridhar and Dhanasekaran (2023)	movie lens Facebook	deep belief network	Sparsity reduction
Saleh et al. (2023)	LDOSCoMo Da	filtering approaches utilizing the Naïve Bayes algorithm	A significant barrier to developing context-aware recommender

Sujatha and Abirami (2023)	LDOSCoMo Da	achieve contextual recommendations based on user-generated reviews	contextual information is used to create good and intelligent recommender systems

Nonetheless, the studies mentioned above focus mainly on improving recommendation systems through different techniques and methodologies. The studies mentioned in this review are related to different recommendation methods such as rule-based classifiers, correlation-based systems, collaborative filtering, neural networks, and contextual recommendation techniques. The authors looked into ways of improving the accuracy in making recommendations, reducing data sparsity, and enhancing user experiences.

However, several challenges and opportunities emerge from this body of research, which can inform the direction of the proposed thesis.

1. Data sparsity is a major issue in many studies on recommendation systems. Some researchers have suggested using weighted correlation coefficients and context-aware models to overcome this problem and enhance the recommendation accuracy.

2. Several studies have shown that contextual factors are considered by different approaches to create more individualized recommendation models, which are context-aware. Furthermore, it enables recommender systems to implement context-based features, thus enhancing the recommendation accuracy in line with customer needs.

3. Various studies suggest that a combination of multiple methods may be used to take advantage of the strengths of each approach. This can be achieved by further investigating advanced hybrid techniques and their usefulness.

The proposed dissertation builds upon these studies and contributes to the field of recommendation systems by addressing the following:

1. Introduce advanced hybrid recommendation systems based on a combination of classical machine learning and deep learning techniques that can provide more accurate results than existing recommendations.

2. Discover inventive ways to reduce data sparsity and improve product recommendations, especially for unpopular products or individuals with little experience of interaction.

3. Enhance recommendation relevance through user feedback and sentiment analysis and understand preferences.

In conclusion, this dissertation is aimed to tackle issues in prior research on recommendation systems, while leveraging opportunities to enhance their precision, personalization, and customer satisfaction.

1.8 Dissertation Organization

This Dissertation is divided into five sections. Each chapter begins with a brief overview that offers a general impression of it. The remainder of this paper is organized as follows:

- **Chapter 2:** Explains how recommender systems work, and define some of the terms used. The next step is to investigate the reasoning behind and fundamental principles of popular techniques like recommendation algorithms, collaborative filtering strategies, content-based filtering, and others like them. This section outlines the basic concepts that form the basis of recommender systems.

- **Chapter 3** outlines the method and structure used to develop the contextual model for suggestions. The optimal algorithms, models, and data used in the proposed model are discussed in this section. In addition, it explains how the performance of the recommender system is improved by several factors and design decisions. This section delves into the reasoning behind the proposed solution and its implementation.
- **Chapter 4:** The analysis results from experiments conducted with the proposed optimized recommender system are presented and assessed in Chapter three. Measures of efficiency, such as the Mean Squared Error (MSE), Mean Absolute Percent Error (MAPE), and Root Mean Squared Error (RMSE), are discussed at length. This section also includes a detailed discussion and analysis of the experimental data, highlighting the merits, limitations, and potential applications of the proposed strategy. The results are critically examined in this section so that the strengths and scope for improvement of the recommender system can be better understood.
- **Chapter 5:** Examines the extent to which the dissertation stated aims and objectives are accomplished, as well as the broader significance and applicability of the enhanced recommender system. Furthermore, this section provides recommendations for future research and areas of potential improvement in the field of recommender systems. This suggests additional avenues for study and highlights areas where the proposed solution can be enhanced.

CHAPTER TWO

Theoretical background

2.1 Overview

Recently, there has been an increase in population growth leading to various consequences, which has led to an increase in the rate of online shopping, and there is a need to predict user preferences. This chapter provides a theoretical explanation for predicting user needs and requirements, the factors affecting them, and the most important techniques used in the field of electronic e-commerce. An overview of a set of recommendation methods is provided, and a description of the machine and deep learning concepts are presented in this section.

2.2 Social Media Data Analysis

Text mining and sentiment analysis are the two predominant methodologies for analyzing social media data, as illustrated in Figure 2.1 [27]. Social Media Networks with virtual entertainment, or social media, give individuals a stage to collaborate with one another. The notoriety of virtual entertainment destinations, such as Amazon, has expanded unimaginably in recent years. The center of the informal communication experience fixates on the client’s capacity to perform the following activities [28]:

- The Internet provides a platform for individuals to express themselves by sharing personally relevant information on a designated profile webpage.
- The degree of connectivity and interaction between partners and populations is a crucial factor to consider.
- Network extension.

Two types of analyses can be conducted using social media data.

- Content analysis is a significant aspect of the study of social networking sites, such as Amazon, which possess a vast quantity of multimedia content, including text, images, audio, and video. This extensive database has the potential to be utilized in diverse research endeavors.

- The analysis of social networks can be conducted by mapping and measuring the relationships between different entities using linkage data. This approach is well-documented in the literature [29], [30].

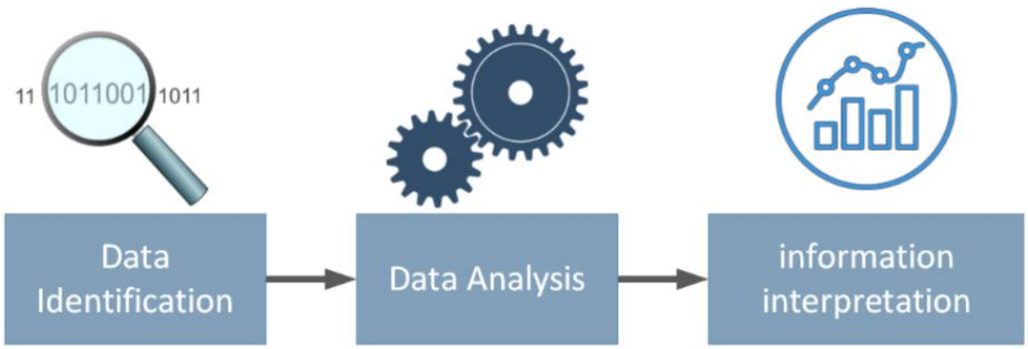

Figure 2.1: Analysis of Social Media Data [27]

2.3 Recommendation System

Systems that provide suggestions on what a user might find beneficial are called recommender systems. The advice provided is intended to aid users in making a variety of decisions, such as what to buy, what music to listen to, and what news to read [31].

Recommendation systems are aimed mostly at people who lack the expertise to critically examine the vast amount of information readily available on the internet. Independent search engines are dedicated to suggestions focused on rating-dependent features [32], [33].

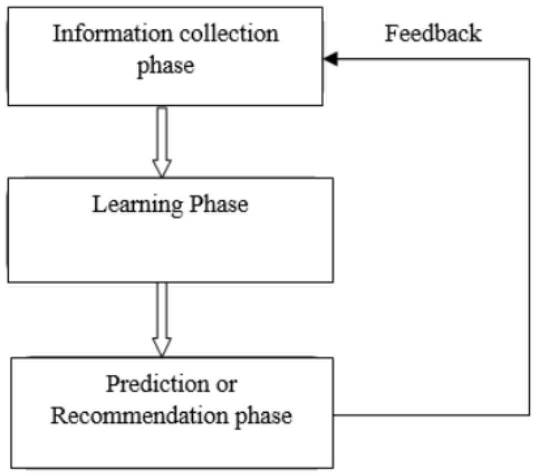

Figure 2.2: Phases of Recommender System [34]

The recommendation system typically operates in three stages [35], [36], as shown in Figure 2.2.

- Modeling stage: Preparing the data was the main focus of this study. In the first of the three possible uses, users and elements are combined in a rating matrix. An element's value is determined by the user's assessment of a given cell of the matrix. Second, a user profile is followed by the generation of a unique vector. Finally, the feature file for that element is compiled into an element-definition file.
- Prediction stage: This phase seeks to determine the degree or categorization of hidden elements described by a function based on the data obtained from the modeling phase.
- Recommendation stage: The prediction phase is continued. Methods are used to aid user decision-making by eliminating irrelevant information and presenting the user with options that are more likely to pique their interests.

2.4 Recommender System Formal Definition

Users of an information domain organize their content into categories. N is typically represented using a rating matrix, which is a standard mathematical notation for the RS. The number of training users is represented as U, where U = {u1, u2,…,un}, and the number of items is denoted as M, where I = {i1, i2,...,im}. We can express the user-item rating matrix as R (N × M), where R_(u_a,l_i) signifies the rating given by user u_a for item I_i.

Consider RF as a rating function that quantifies the degree of utility of a specific product or service (I ∈ I) for a user (u ∈ U). This function is denoted by RF: U×I →R, where R represents the scalar rating. A sparse N × M user-item matrix is shown in Figure 2.3. Here, N refers to the overall number of users, and M represents the total volume of items [37].

	Item1	*Item2*	*Item3*	*Item4*	...	*Item $_M$*
User 1	RF (1,1)=R_{11}	RF (1,2)=R_{12}	RF (1,3)=R_{13}	?	...	RF (1,N)=R_{1M}
User 2	RF (2,1)=R_{21}	RF (2,2)=R_{22}	RF (2,3)=R_{23}	RF (2,4)=R_{24}	...	RF (2,N)=R_{2M}
User 3	RF (3,1)=R_{31}	?	RF (3,3)=R_{33}	?	...	RF (3,N)=R_{3M}
⋮	⋮	⋮	⋮	⋮	...	⋮
User N	RF (N,1)=R_{N1}	?	RF (N,3)=R_{N3}	?	...	RF (N,M)=R_{NM}

Figure 2.3: N×M Sparse User-Item Matrix [38]

2.5 Information Representation in Recommender System

Representation and collection of information are crucial steps in the recommendation process. Accurate recommendations result from locating information that is useful to users. Three methods are commonly used for gathering data.

- Explicit Feedback: Through the system interface, the user is prompted to issue evaluations that will be used to build and improve the model. How well does the RS function in relation to the number of ratings submitted by the consumer? This approach has the drawback of requiring work. Users' reluctance to volunteer for further data is a common problem. User input is more time-consuming, but yields greater trustworthiness and clarity in the final proposal [39].
- Implicit Feedback: Users' clicks, page views, and other interactions with websites, emails, and other media are tracked by Recommender Systems, which then use this data to automatically deduce the users' preferences. Here, the choices made by users while interacting with the system are used to infer user preferences. User convenience comes at the expense of accuracy when using this method [40].
- Hybrid Feedback: As a hybrid system, it improves both safety and efficiency. The best outcomes can be achieved by either using implicit data to verify explicit ratings or only allowing explicit input when the user volunteers [31].

2.6 Recommendation System Categories

Five distinct types of recommendation systems are defined in this section with respect to their development methodologies and graphical representations. Figure 2.4 presents the main categories of recommender systems [41]:

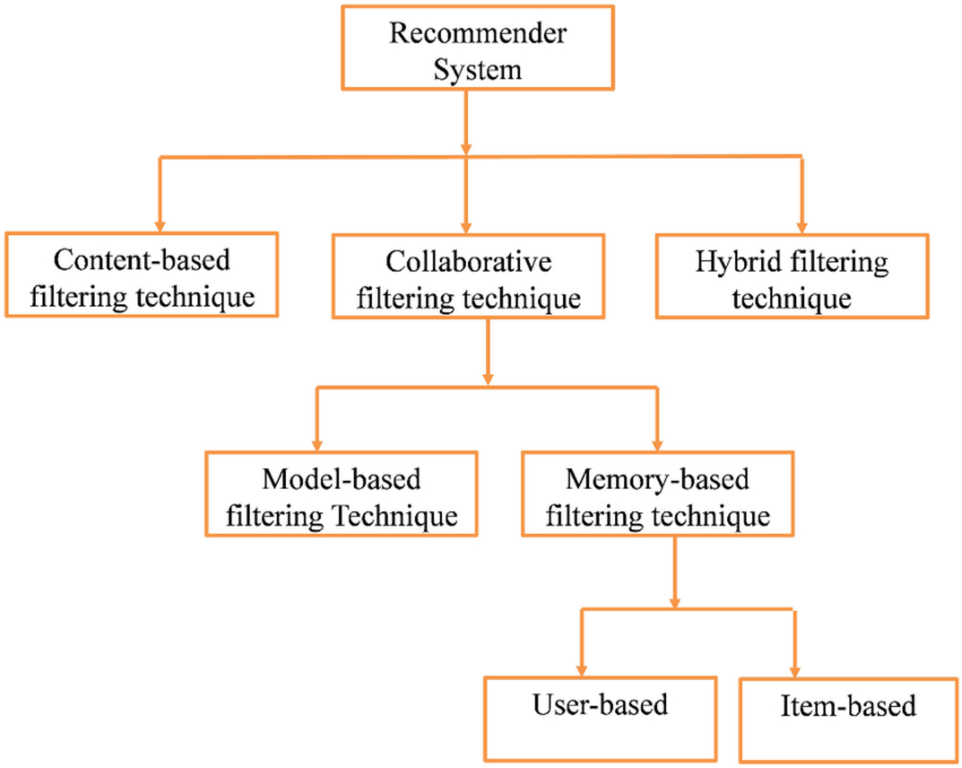

Figure 2.4: Recommendation Systems: A Taxonomy [41]

2.6.1 Collaborative Filtering Techniques

It recommends the most frequently used method. Figure 2.5 shows the collaborative filtering process similar developments are likely to be appreciated by users who share common interests. This technology relies on two basic aspects.

- First, it is a metric for selecting a panel of people whose views will be used as a foundation for a suggestion (nearest neighbors).
- Second, similar opinions are used to predict the ratings of the target user's impact of recommendation [42].

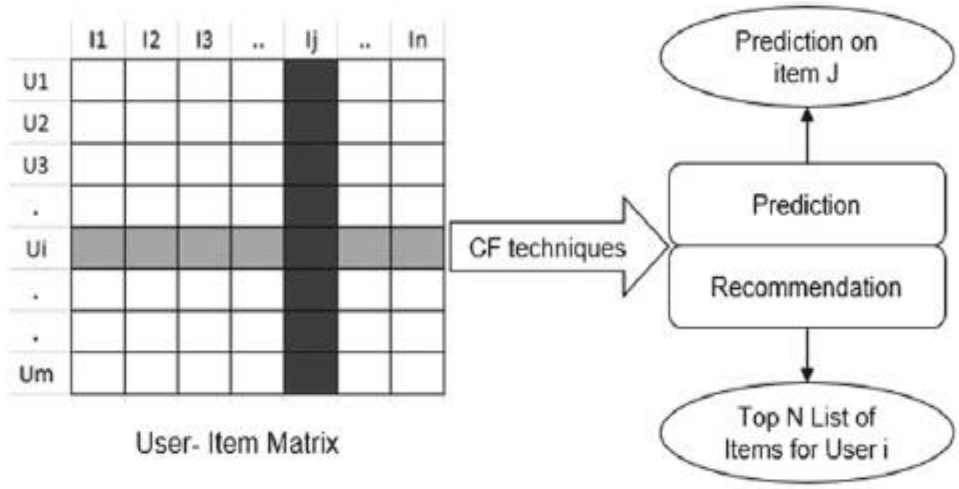

Figure 2.5: Collaborative Filtering Process [43]

Collaborative filtering compiles the opinions of many people into a single set of recommendations. A preference or rating matrix of size M × N can be used to represent the model. Where M is the number of users (U1,U2,U3, . . .,Um), and N is the number of items (I1, I2, I3, . . ., In) rated by the users. Collaborative filtering (CF) algorithms can be divided into two main groups: those that rely on memory and those that rely on models [43].

A. Memory based Collaborative Filtering

Memory-based algorithms typically generate rating forecasts using the entire rating history of users. Because the original rating database is stored in memory, memory-based systems can quickly and easily retrieve it to provide recommendations [44]. The typical flow of a memory-based algorithm is as follows [45].

1. A neighborhood is a group of users or items that share many characteristics with the target user or item, as determined by a comparison with all other users or items in the rating matrix.

2. The KNN technique uses the distance between a user or item of interest and its nearest neighbors to determine which neighborhood is most likely to contain useful information.
3. To make a prediction, the chosen neighborhood was used, and the ratings of its residents were averaged.
4. Top-T is an algorithm that is used to make suggestions. After identifying the k most similar users, user-item matrices were integrated to derive suggestions.

Below is a rundown of the advantages and disadvantages of relying on one's own recollection [46]. The advantages of using this method are as follows:

- Its ease of creation and application.
- It does not have to consider the content of the products suggested.
- The system can easily handle a large number of users and jointly rated items.

The following are the several disadvantages of this method:

- Therefore, user feedback is essential.
- When information is sparse, system performance declines, and new users or objects cannot be added.
- This system has trouble with scaling.

B. User Based Collaborative Filtering

One of the key approaches to collaborative filtering is the computation of the similarity weight between each pair of users, which is the backbone of the recommendation process. The two stages of the user-based collaborative filtering are outlined below.

1. User-Based Similarity

This technique uses customers' ratings of the products they buy together to determine how similar they are. Figure 2.6 depicts the process of determining the relationship between users a and u using a user-based similarity metric. Therefore, the first step in making a recommendation is to determine the similarity between all users and the target user. Equation 2.1 is used to determine the user similarity [38]:

$$P_{a,u} = \frac{\sum_{i=1}^{n}(r_{a,i}-\bar{r}_a)(r_{u,i}-\bar{r}_u)}{\sqrt{\sum_{i=1}^{n}\left(r_{a,i}-\bar{r}_a\right)^2}\sqrt{\sum_{i=1}^{n}(r_{u,i}-\bar{r}_u)^2}} \tag{2.1}$$

However, considering their respective ratings, goods that received high marks from both users and u are shown by, where. The naive user rating of an item is denoted by Ra.i, whereas the average user rating is represented by Ri.

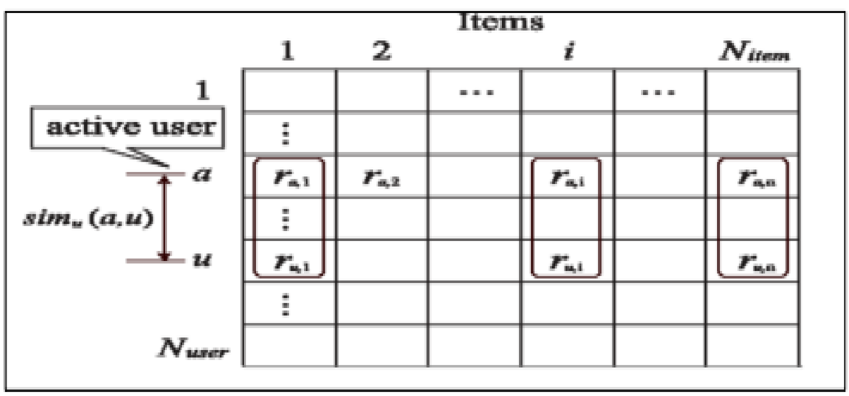

Figure 2.6: Collaborative Filtering Based on User Similarity [47]

2. User-based prediction

Predictions for a target user for a given item z are calculated using an adjusted weighted sum prediction method that involves summing the rating weights for that item a and adding the target user's mean to that sum. To account for users who consistently rated items higher than average, we subtracted the user mean rating ru from the sum of the ratings. Equation 2.2 issued for user-based forecasting is [43]:

$$\boldsymbol{predict(a,i) = \bar{r}_a + \frac{\sum_{u \in U} Sim(a,u).(r_{u,i-}\bar{r}_u)}{\sum_{u \in U} |Sim(a,u)|}} \tag{2.2}$$

r_a is the stand-in for an average user. The resemblance between two users is quantified by a statistic called "Sim (a,u).” The user gave this item a rating of $r_{u,\,i:}$

3. Item-Based Similarity Computation

The similarity between the target element and elements previously obtained by the target user was calculated using this method, and the most similar elements were determined.

4. Item based similarity

This method uses a calculated similarity to rank the user-obtained results in order of resemblance to the desired result and then returns the results with the highest similarity. Equation 2.3 can be used to determine which elements are most chemically related to one another [43]:

$$sim(b,i)\,\frac{\sum_{u \in U}(r_{u,b}-\bar{r}_b)(r_{u,i-}\bar{r}_i)}{\sum_{u \in U}(r_{u,b}-\bar{r}_b)\sqrt{\sum_{u \in U}(r_{u,b}-\bar{r}_b)(r_{u,i-}\bar{r}_i)}} \tag{2.3}$$

Here, $u \in U$, U represents the specified evaluation of the co-rated user for individual items *b* and *naïve*. Figure 2.7 shows the similarity calculation based on the items [68].

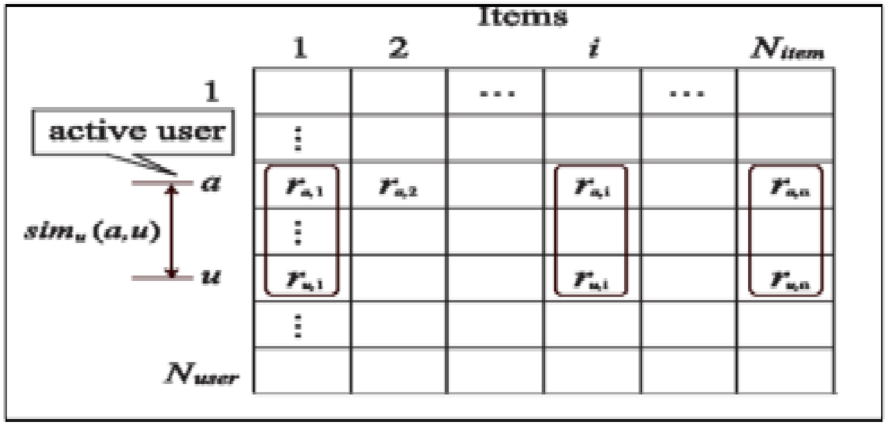

Figure 2.7: Differences Between Items Filtering using Group Input [48]

5. Item-Based Prediction Computation

Predicting a target user's (a) rating of item (b), represented as P(a, b), can be performed with a straightforward weighted sum formula (b). Equation 2.4 is a commonly used formula for forecasting [43].

$$predict(a,b) = \bar{r}_a + \frac{\sum_{i\in I} Sim(i,b).(\bar{r}_{a,})}{\sum_{i\in I}|Sim(i,b)|} \quad (2.4)$$

C. Model-Based Filtering

Model-based solutions apply machine learning and statistical methodologies to comprehend the model's underlying factors to predict a user's choice of a product [49], [50]. For collaborative filtering tasks involving test data or real-world data, models (such as machine learning or data mining methods) can be used to train the system to recognize complex patterns in training data, and then use these models to produce accurate predictions. Researchers have investigated model-based alternatives to traditional CF algorithms, such as Bayesian models, clustering models, and dependency networks, to address these issues.

Classification methods can be used as CF models if the user ratings are categorical, whereas regression models and SVD approaches are better suited for numerical ratings [51].

Model-based collaborative filtering recommender systems provide various benefits over memory-based methods.

1. Scalability: Most models generated by model-based algorithms are substantially smaller than the actual dataset to minimize runtime complexity and increase efficiency. Therefore, they are better equipped to handle sparsity and scalability concerns [52].
2. Prediction speed: Model-based solutions are more likely to be efficient because it takes far less time to query a model than to query an entire dataset [46].
3. Accuracy: Compared to memory-based systems, which are simple to create but sometimes lack precision, model-based solutions, especially when utilized as latent factor models, are more accurate [46].

2.6.2 Content Based Technique

The process of recommending items to a user through content-based filtering involves matching the user's preferences with the attributes of the items. Typically, content-based filtering involves a tripartite process, although the specific means of acquiring these steps may vary across different systems [51].

- Analyzing the content: This particular stage pertains to the representation of the substance of the items. The process involves extracting pertinent information and utilizing it as input for the subsequent step.

- Learning the profile: The subsequent stage involves utilizing the data obtained from the preceding step to generate a user profile.

- Filtering: This process identifies items that align with the given profile and provides recommendations accordingly. Figure 2.8 present a simplified chart illustrating the content based filtering.

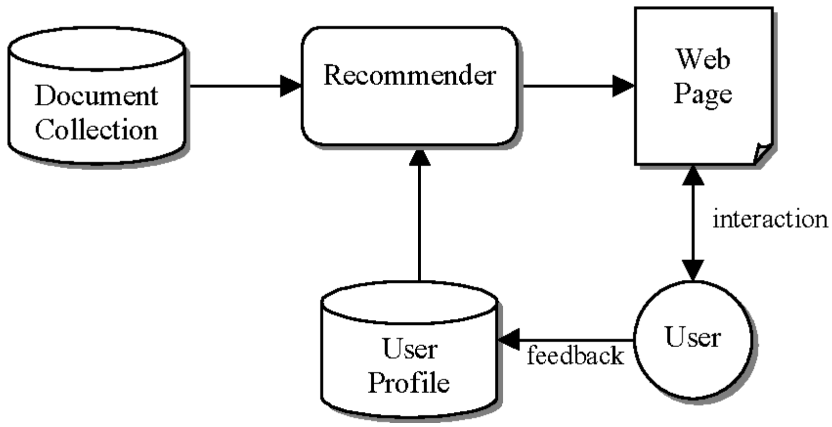

Figure 2.8: Demonstration for the Content Based Filtering [54].

2.7 Machine Learning

The use of machine learning is important for the development and advancement of recommendation systems. Computers are trained to enhance their performance by providing example data or previous user interactions. Recommendation systems employ machine learning, whereby the parameters are optimized using training data as well as historical user behavior. The system develops the capability of predicting consumers' future preferences, and it is possible to generate relevant insights from the data. Machine Learning in Recommendation Systems is based on Statistical Theory, which helps create Mathematical Models.

There are two types of predictive models that enable the system to forecast user preferences, and descriptive models that explore information in data and can meet the second purpose. In machine learning, the recommendation system and its efficiency are important considerations. Training is characterized by large datasets, and efficient algorithms are required to optimize the model parameters. Furthermore, after training the model, an optimal representation and inference algorithm for efficiency must be developed. Sometimes, the efficiency of the learned or derived algorithm in terms of complexity in time or space is as essential as its predictive power. In the context of recommendation systems, machine-learning approaches can be categorized into three main types: Supervised, Unsupervised, And Semi-Supervised Methods.

1. Supervised: For supervised systems, a predetermined class or label is provided for each item that appears in the training set of user–item interactions. User preferences, such as user ratings or purchase history, can also be labelled for use in training a model of such a recommendation. Supervised machine learning algorithms, such as linear regression, gradient boosting regression, decision tree regression, stochastic gradient descent regression, and Bayesian ridge regression, are common for recommendation systems.
2. Unsupervised Recommendation Systems: Unlike supervised methods, unsupervised recommendation systems do not depend on any external knowledge or preset labels for user-item interactions. On the contrary, they categorize users and items based on attributes only while ignoring the content or behavior of the items and users. In this manner, the system can identify trends and cluster-related users or items.

3. Semi-supervised Recommendation Systems: Supervised and unsupervised paradigms were combined using a semi-supervised approach. They may rely on unsupervised methods for some instances of users or items, and use limited labeled data in other cases. This may prove particularly useful when few labeled data are available. Supervised, unsupervised, and semi-supervised learning are the three most important subclasses of ML, as shown in Figure 2.9.

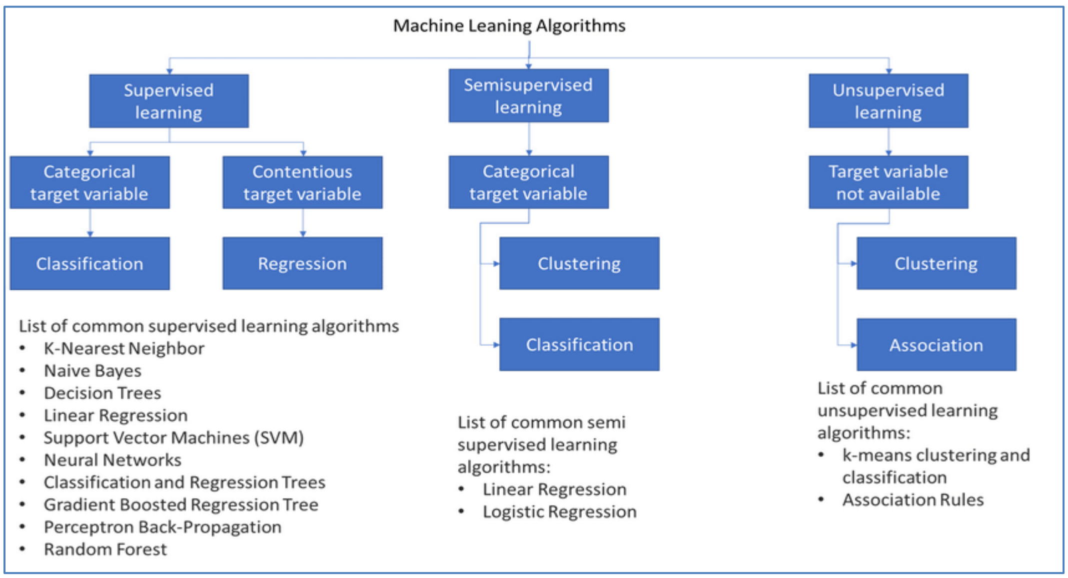

Figure 2.9: Three Main Categories of Machine Learning Methods [53]

2.8 Regression Algorithms

Numerous regression and prediction algorithms exist in the field of machine learning and are briefly outlined in this section.

2.8.1 Linear Regression

If a statistical approach that can be used for both binary and multiclass classification is needed, look no further than the logistic regression model. This classifier uses probability scores as anticipated values of a dichotomous dependent variable to evaluate the relationship between the dependent variable and one or more (categorical or continuous) independent variables. The independent variables need not be normally distributed, linearly related, or of equal variance within each group [54], [55]. Algorithm 2.1 presents the mechanism of the linear regression technique.

Algorithm 2.1: Linear Regression [55]

Input: Number of Data Points (n)
Arrays of x and y values: x[] and y[] of size n
Output: Intercept (J) and Slope (K) of the linear regression line
Begin
Step 1: Initialization:
    Initialize sum_x, sum_x2, sum_y, and sum_xy to 0.
Step 2: Computation of the Sums:
    For i = 1 to n:
        sum_x = sum_x + x[i]
        sum_x2 = sum_x2 + x[i] * x[i]
        sum_y = sum_y + y[i]
        sum_xy = sum_xy + x[i] * y[i]
Step 3: Calculation of Slope (K) and Intercept (J):
    Calculate the slope (K):
    K = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x * sum_x)
    Calculate the intercept (J):
    J = (sum_y - K * sum_x) / n
Step 4: Prediction:
    For each data point x[i]:

Calculate the corresponding y value using the linear regression equation: y = J + K * x[i] **End**

This approach to classifying data uses a weight vector to define and optimize the log-odds in favor of one of the classes. The sigmoid-shaped logistic function assigns a value between 0 and 1 to every weighted feature vector. This number represents the likelihood that a given instance belongs to a specific category. The probability of categorization is on the vertical axis, and the x-value is horizontal [54], [55]. LR can assess the likelihood of an occurrence by fitting the data to a logistic curve, as shown in Figure 2.10.

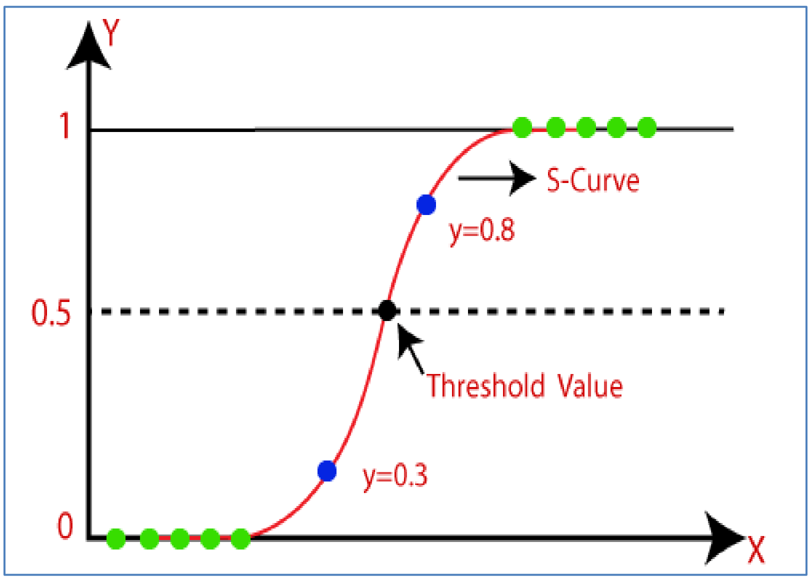

Figure 2.10: Best Fit Line for a Linear Regression Model [56]

2.8.2 Stochastic Gradient Descent Regression

This is an effective form of facilitation. The SGD updates are presented in a simplified form as in Equation 2.5:

$$\theta^{(t+1)}=\theta^{(t)}-\alpha_t \nabla l_i(\theta^{(t)}) \quad (2.5)$$

Where: **ϴ** is the parameter updates, t is the iteration, α is the step size or learning size. In this case, the index I changes at each iteration. In practice, samples are frequently jumbled before going through them. Note that the path can be easily extended to exploit the gradient of many samples, that is, $\sum_{j=1}^{b} \nabla\, l_{i+j}\left(\theta^{(t)}\right)$, Small-batch Gradient Descent is the proper term for this procedure. Mini-batching allows quicker matrix operations and parallelization with more stable convergence [57]. Algorithm 2.2 shows the mechanism of this technique.

Algorithm 2.2: Stochastic Gradient Descent Regression [57]
Input : features set D **Output: accuracy evaluation matrices R** **Begin** Step 1: initialize the observation weights w_i= 1/N, ί = 1,2,….,N. Step 2: For m=1 to M: Fit a classifier $G_m(x)$ to the training data using weights w_i. Step 3: compute $err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$. Step 4: compute $a_m = \log((1-err_m) / err_m)$. Step 5: set $w_i \leftarrow w_i \cdot \exp[a_m \cdot I(y_i \neq G_m(x_i))]$, i= 1,2,…,N. Step 6: output $G(x) = \text{sign}\left[\sum_{m=1}^{M} a_m G_m(x)\right]$. End

When the volume of step becomes less pursuant to $\sum_t \alpha_t^2 < \infty$ *and* $\sum_t \alpha_t = \infty$, *e.g.*, $\alpha_t = \mathcal{O}(1/t)$, Moreover, SGD commonly converges to a regional minimum and even a universal minimum for a convex objective work when conditions are mild. Under regular circumstances, gradient descent can achieve linear convergence, and Newton's approach can achieve quadratic convergence. This means that gradient descent must be used if the desired mission precision is to be $\mathcal{O}\ (log\ (1/\in))$ repetitions, and Newton's way occupies fewer. While, SGD occupies $\mathcal{O}\ (1/\in)$ repetitions, There's a good chance that's twice as bad as a gradient decline. Nonetheless, when n is sufficiently enough, assuming that the time complexity of computing the gradient of one specimen is a constant C, the overall time complexity of SGD equals is $\mathcal{O}(C/\in)$,that is weaker than that of gradient descent ,$\mathcal{O}(nC\log(1/\in))$ [57]. Figure 2.11 presents the flow diagram of gradient boosting machine learning method.

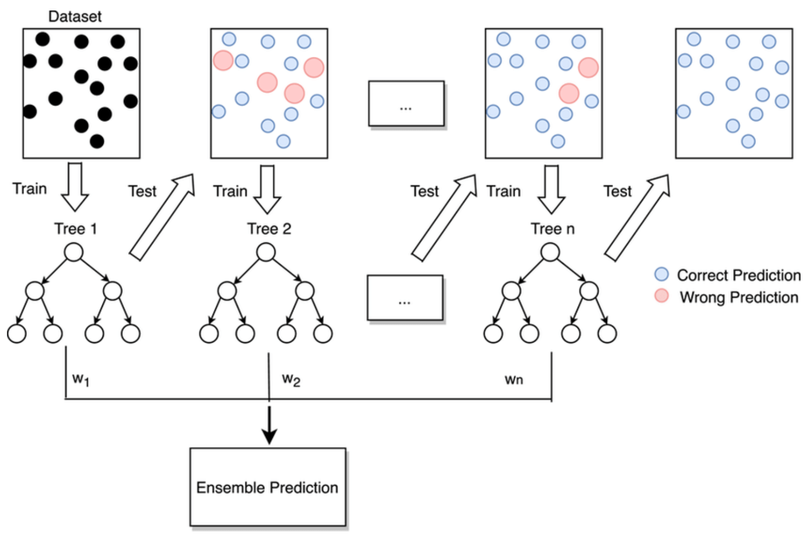

Figure 2.11: Flow Diagram of Gradient Boosting Machine Learning Method [86].

2.8.3 Decision Tree Regression

Many different types of signals can be classified using decision tree algorithms. There is one root node in the decision tree with no outgoing edges, test nodes with outgoing edges, and terminal nodes with one incoming edge and no outgoing edges. Each node and its child node have their own testing attributes and values. Each leaf node is matched with the best possible destination based on its label [58], [59]. Algorithm 2.3 refers to the mechanism of this method.

Algorithm 2.3: Decision Tree Regression [58]
Input : features set **D**
Output: accuracy evaluation matrices **R**
Begin **Step 1: Initialize an empty dictionary R to store the decision tree.** **Step 2: If D is considered "pure" or if other stopping criteria are satisfied, eliminate all attributes that belong to D.** **Step 3: Impurity Function Criteria:** **Obtain the impurity function criteria by performing a split on a variable, "a."** **Step 4: Attribute Selection:** **Compute the best attribute a_best based on the aforementioned criteria.** **Step 5: Create Decision Node:** **Create a decision node that evaluates the optimal choice a_best as the root of the decision tree.** **Step 6: Split Dataset:** **Derive subsets DV from the dataset D based on the attribute a_best.** **Step 7: Recursion: For each subset Dv, do the following:** **Recursively apply the Decision Tree Regression algorithm (J48) to subset Dv to create the decision tree Rv.** **Connect Rv to the corresponding branch of the decision tree R.** **Step 8: Output:** **Return the constructed decision tree R.** **End**

Decision trees use a recursive top-down hierarchy to organize the training data. Using a discrete function of the input attribute values, the algorithm decides how best to partition the training set at each iteration. Each node further split the training set into smaller subsets after selecting the most appropriate split. The tree continues to expand until certain halting conditions are met [58], [59]. Figure 2.12 shows the schematic of decision tree algorithm.

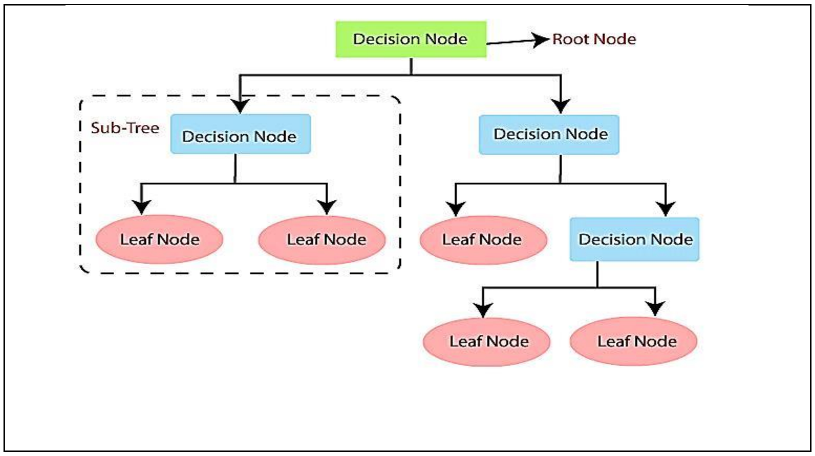

Figure 2.12: Schematic of Decision Tree Algorithm Technique [60]

2.8.4 Gradient-Boosting Regression

The application of gradient-boosting regression in the field of ensemble learning allows for making very good prediction about customers' preferences when it comes down to movies recommendations. In this approach, the technique proposed is a variation of learning base ensemble that aims at combining several base learners towards reducing prediction mistakes especially through the utilization of decision trees in an addition model and while minimizing the loss function by the means of the Gradient Descent. This method results in the creation of the Gradient Boosting Tree (GBT), which can be defined as the sum of regression trees as in Equation 2.6:

GBT=∑i=1ton(Ti) (2.6)

This is done through a regression tree per Ti. These trees are constructed sequentially, with each new decision tree (Ti+1) estimated using Equation 2.7 [61]:

$$Ti+1=Ti-\eta\cdot\nabla L(Ti) \quad (2.7)$$

Here, η' (learning rate) is used for denominator, while the gradient of the loss function L' (∇L(Ti)). The idea of the latter minimization using steepest descent approach is iterationally to minimize the loss function. It uses an iterative method to build serial of models that are progressively perfected in order to give accurate movie advises [61], [62]. Algorithm 2.4 refers to the mechanism of this method.

- Initialization: Observation Weights Initialization: Set initial observation weights (w_i) as 1/N to represent an equal contribution of all observations at the initial stage.
- Boosting Process: Iterative Fitting: It runs for M iterations. The training consists of several rounds involving a new weak learner G(x)_n to each. Each G(x)_n takes into account the observation weights. Represented by these learners, are the movie suggestion models which can take into account user's likes.
- Error Calculation: Compute the weighted error for each round. This mistake estimates the fraction of misclassified user tastes with respect to the weights of observations. The model reflects where its prediction is inaccurate.
- Weight Assignment: Find the weight (a_m) for the latest iteration G_m. The weights are calculated as err_m and form an evaluation tool, proving that good functioning of a model is essential for prediction.
- Updating Observation Weights: For the next round, update the observation weights (w_i). Weights have been recalculated for such observations where the model has committed some errors, thus being more urged for correction.

Algorithm 2.4: Gradient Boosting Regression [62]
Input : features set D
Output: accuracy evaluation matrices R
Begin
Step 1: The first vector of parameters is randomly selected.
Step 2: Continue the phases below till a minimum is reached approximately:
Step 3: Examples are mixed in the training set randomly
For i = 1 to n
$$\boldsymbol{w} = \mathbf{w} - \boldsymbol{\eta}\nabla\boldsymbol{\varrho}_{\boldsymbol{i}}(\boldsymbol{w})$$
Here, Q (w) refers to the experimental risk while Qi (w) refers to the value of the loss function.
End.

The final movie recommendation model G(x) emerges at the end of M iterations after accumulating the weighted weak learners. The ensemble model gives precise predictions on user's taste in films. Figure 2.13 shows the flow diagram of gradient boosting method.

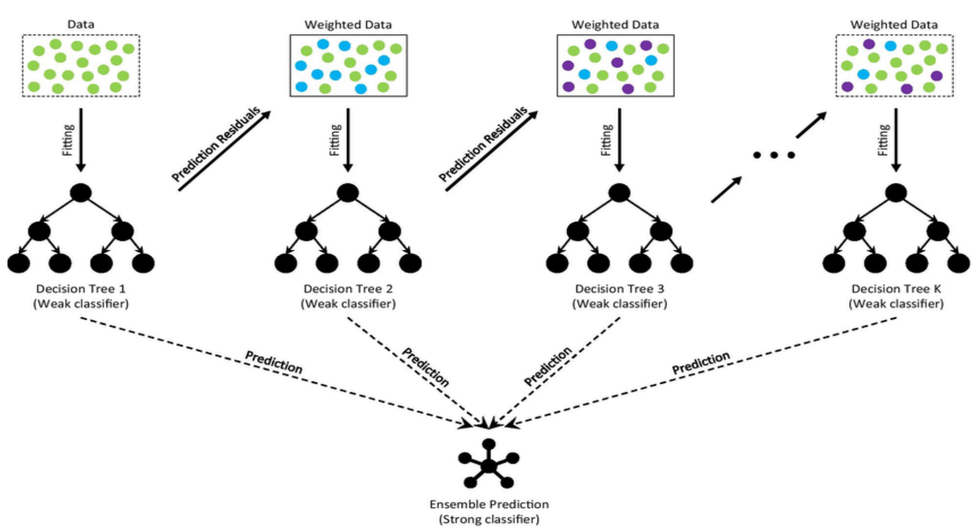

Figure 2.13: Flow Diagram of Gradient Boosting Machine Learning Method [63]

2.8.5 Bayesian Ridge Regression

Bayes' theorem and the "naive" assumption of conditional independence between any pair of features, given the value of the class variable, form the basis of the Naive Bayes family of supervised learning algorithms. As they are simple to construct and do not require complex iterative parameter estimates, naive Bayesian models might be useful in medical research and text classification challenges. Despite its apparent lack of complexity, the Naive Bayesian classifier has found widespread application because it consistently outperforms its more complex counterparts [64]. Algorithm 2.5 refers to the mechanism of this method.

Using Bayes' theorem, we can determine the posterior probability is. P (j | i) from P(j), P(i), and P (i | j). The Naive Bayes classifier assumes that the weight assigned to each predictor (x) does not change the impact of x on a particular class (c). This presumption is known as "class conditional independence.”. Equation 2.8 using to calculate Bayesian Ridge Algorithm.

$$P(j|i,\ldots,i\_n)=(P(j)P(i\_1,\ldots,i\_n|j))/(P(i\_1,\ldots,i\_n)) \qquad (2.8)$$

where P(I 1,...,I nj) is the probability of the predictor given class, P(I 1,...,I n) is the prior probability of the predictor, P(j) is the prior probability of the class, and P(ji,...,I n) is the final probability of the conditional probability [65], [66].

Algorithm 2.5: Bayesian Ridge Regression [66]
Input : features set D
Output: accuracy evaluation matrices R
Begin
Step 1: Calculate Class Probabilities:

For each class c in category [c], do the following:

Compute the probability of class c based on the training data.

Step 2: Compute Log-Priority:

Calculate log-priority [c] for each class c using the formula: log(Number of Categories divided by the number of documents).

Step 3: Compute Document Probabilities:

Compute the probability for each document in the dataset using the Bayesian Ridge algorithm.

Step 4: Return Results:

Return the calculated Log-prior and Log probability for each document.

End

2.9 Deep Learning (DL)

To enable robots to learn new information on their own, researchers have resorted to a new subject known as "deep learning," which tries to develop theories and algorithms that mimic human neural networks. Deep learning is a type of machine learning that was first developed as an artificial intelligence (AI) technique to replicate how humans learn in a certain domain [67].Because each layer of a deep learning algorithm builds on the one below, the phrase "hierarchical learning" defines how these algorithms are structured. Most existing machine-learning algorithms are expected to have a linear structure. Deep learning was introduced in 2007 [68]. Deep learning is a cutting-edge machine-learning technique that bridges the gap between traditional ML and artificial intelligence. Deep learning has numerous applications, including but not limited to object detection, speech recognition, and medicine [69].

Deep learning has been developed to overcome the limitations of human intelligence in AI problem-solving. Deep neural networks outperform shallow ML methods in most applications involving the processing of text, images, video, speech, and audio data, making DL particularly effective in many domains with enormous, high-dimensional data. The rapid advancement of ML algorithms and processing is one reason for the appeal of deep learning [70], [71].

2.9.1. Deep Neural Networks (DNN)

A deep neural network (DNN) is an artificial neural network with multiple hidden layers. MLP is the most common artificial neural network (ANN) structure in DNN. The neurons in a neural network are linked and collaborate across multiple layers. Because the number of weights in a DNN is enormous, sometimes hundreds or millions, training the samples necessitates considerable investment in computational labor and data sources. Convolutional neural networks (CNN) and recurrent neural networks (RNN) are two of the most well-known and powerful deep-learning network topologies. Numerous improvements over the last decade have made it possible to train such networks [72], [73]. The multilayered organization of the nervous system reflects its complexity, as shown in Figure 2.14.

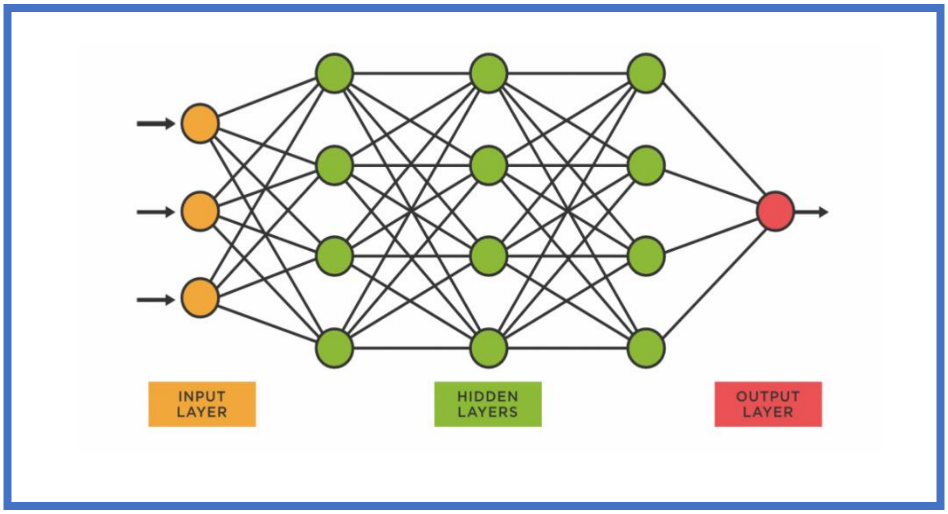

Figure 2.14: Structure of a Deep Neural Network [74]

2.9.2. Types of Learning Algorithms

Deep learning is critical to human survival. It has had a significant impact on various sectors, including illness detection, precision medicine, and voice recognition. Feature extraction from large datasets, for example, DL can help overcome the constraints of shallow networks. Deep neural networks (DNNs) perform their tasks by utilizing complicated algorithms and designs comprising numerous (deep) layers of modules [75], [76]. Some of the most popular deep-learning algorithms are shown in Figure 2.15.

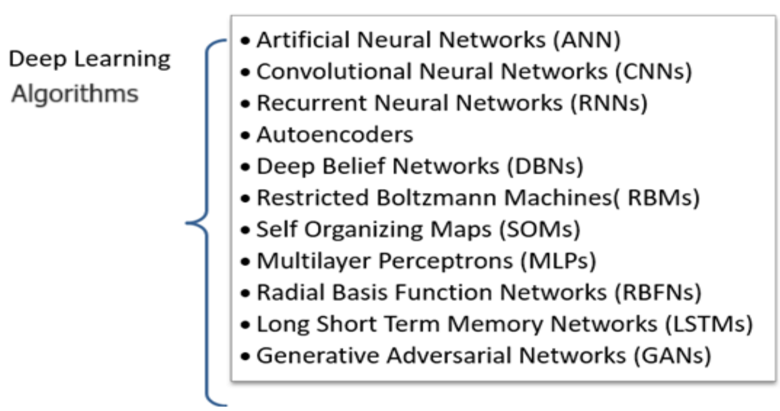

Figure 2.15: Deep Learning Algorithms [77]

These methods are crucial in deep learning because they can be applied to almost any data type, and require a large amount of data and computer power to solve a variety of challenging tasks.

2.9.3 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a typical deep neural network used in machine vision. A Convolutional Neural Network (CNN) is a deep learning technology that attempts to imitate how the human brain operates [78]. It belongs to the feedforward network class of artificial neural networks. Convolutional neural network (CNNs) networks are similar to multilayer networks (perceptron's) because they can merge several networks with local connections into a single unified network. In addition to increased accuracy in automated diagnostic systems, CNN also offer promise in the domain of disease prediction. CNNs have gained popularity in the AI community owing to their tremendous data-processing capacity [79]. In a convolutional neural network (CNN), the output of each layer is used as the input for the next layer. The network consisted of an input layer and an output layer. The hidden layers of the network are located between the input and output nodes. Every "layer" consists of only one "activation function.” Overall, the CNN model improved prediction accuracy by emphasizing more complex interactions between genes and the target set [80].

A. Components of CNN Architecture

Convolutional neural networks can be trained to mimic human cognitive ability. The ability to predict future outcomes is a significant advantage of this network. A typical CNN has two main parts: feature extraction and classification [81]:

1) Feature Extractor

The initial steps in the data processing pipeline for a CNN are feature extraction and the generation of a feature map. In CNN, each of the filters has a distinct purpose. Therefore, many different types of feature maps were created, each of which represents a different group of filters. The output of the feature extraction method is a low-dimensional feature vector assigned to a classifier. The feature extractor comprises multiple layers (multiple convolution layers with optional pooling layers). To create feature maps, a convolution layer is used to combine the input and filter before being reduced in the pooling layer. To extract increasingly complex features, the system is iterated by feeding the output feature maps back into the system as the input feature maps. Finally, a low-dimensional feature vector was created by flattening the reduced-dimensional feature maps [82].

2) Classifier

The best features from each of the retrieved feature maps are then combined into a single low-dimensional feature vector to train a classifier. The classifier reports the likelihood that an input belongs to a specific class. To achieve this, a classifier composed of one or more completely linked layers was used [82].

B. CNN Architecture

A CNN's input layer, which reflects the model's input (the features that were intentionally picked), can be any size, without affecting the network as a whole. When processing gene expression data using a convolutional neural network (CNN), the input layer is typically an (n m) × (m) matrix, where n is the sample size and m is the number of features [83], [84]. Figure 2.16

illustrates several layers that constitute a neural network. Specifically, they were as follows:

1. Convolutional layer.
2. Pooling Layer.
3. Fully Connected Layer.
4. Non-Linearity Layers.
5. Dropout Layer.
6. Adam Optimization Algorithm.

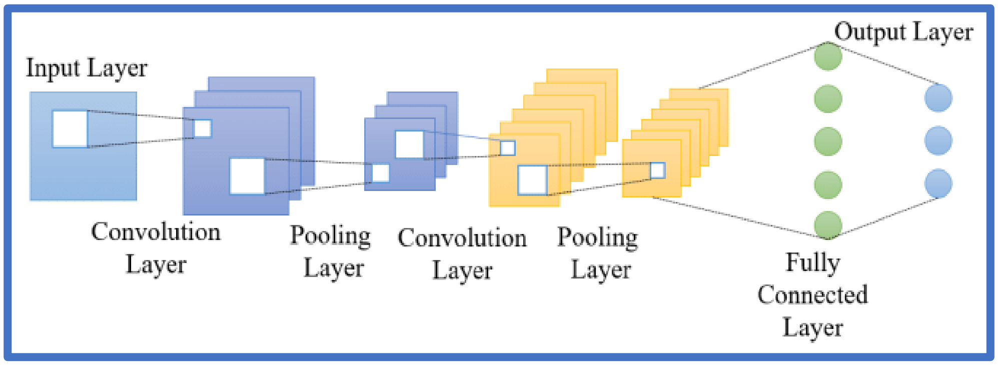

Figure 2.16: Simplified Representation of a Convolutional Neural Network [85]

1) Convolutional Layer

The convolution layer, which can handle high-dimensional data, is the backbone of a convolutional neural network (CNN). The first layer in a convolutional neural network is the convolution layer, which partially communicates with the second layer (pooling layer). The pooling layer, as shown in Figure 2.17, may be fed information from a 3 × 3 window of input neurons that incrementally travel over the data from top left to bottom right at regular intervals (the "stride" number, which is generally 1). When the kernel hits the end of a column, it shifts down one cell until it reaches the end of a row, and so on, until all information has been recorded [86].

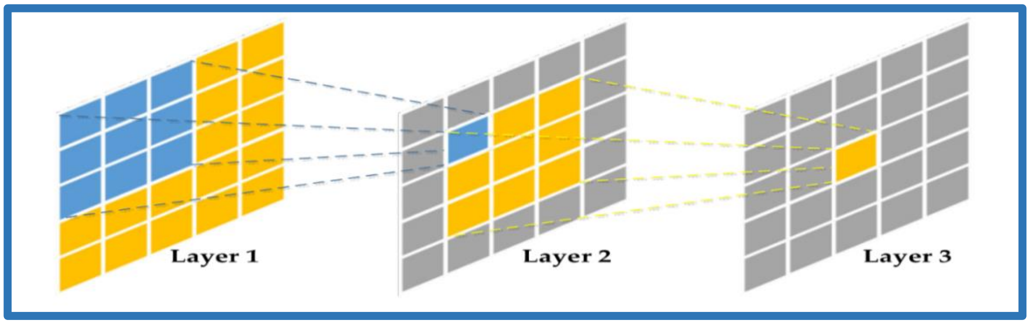

Figure 2.17: Schematic of Receptive Field in CNNs [86]

The receptive field is a small window region created from the input data. To extract features, a tiny section of the input data is convolved with a shared weight window, called a kernel or filter. Figure 2.18 depicts the convolution process.

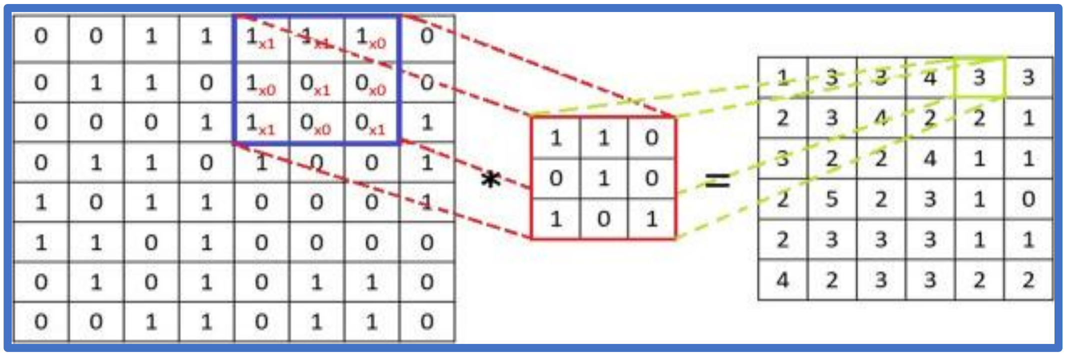

Figure 2.18: Convolution Layer [87]

Many distinct filters have been applied to a single entry. The activation maps are integrated into the convolutional layer to produce a single output file that serves as the input data for the succeeding layer. The values in the filter matrix are appropriately represented by default weights. Each filter must have individual values for these parameters for its output matrices to have distinguishing characteristics or features [87].

Convolutional neural networks (CNNs) rely significantly on these hyper parameters.

a) Number of filters: Filters can be used, and there are many to choose from, all of which have different dimensions.

b) Filter size: An essential CNN hyper parameter is filter size. It establishes the region of the input data examined by the filter (receptive field). Careful consideration must be given to the filter size selection to ensure that significant patterns are captured without the filter outgrowing the input data.

c) Stride: The local receptive field of the filter is created by concurrently shifting a certain number of cells. A single cell can move in both the horizontal and vertical directions simultaneously. When the stride is too short, there is overlap, and vice versa.

d) Padding: Padding is a concept incorporated into the CNN architecture to enhance accuracy. Padding was used to control the shrinkage of the output of the convolutional layer.

The feature map of the convolutional layer was substantially smaller than that of the input image. The resulting feature map prioritizes information closer to the center of the map over data near the map's boundaries. Blank rows and columns are placed at the borders of the image to protect the feature map from decreasing to a practical size. The following equations are used to calculate the size of the final feature map: The relationships between the feature map size, kernel size, and stride are defined in Equations 2.9 and 2.10 [88].

$$W_{nx} = W_{n-1x} - F_{nx} S_{nx} + 1 \qquad (2.9)$$

$$W_{ny} = W_{n-1y} - F_{ny} S_{ny} + 1 \qquad (2.10)$$

where (S_{nx}, S_{ny}) is the stride size, (F_x, F_y) is the kernel size, and (W_{nx}, W_{ny}) is the output feature map size. The layer index is denoted as n.

2) Pooling Layer

CNN's output of a CNN is the result of a sequence of layers that mix convolutional processing with pooling. The major function of this layer is to provide a reduced-dimensional output by compressing the input dimensions while retaining the most relevant details. In this layer, maximum and average pooling are utilized to reduce the dimensionality. If maximum or average pooling [89] is used, the pooling layer receives an input feature map and pools its features into no overlapping blocks, each of which returns a single value. Figure 2.19 depicts the best potential pooling.

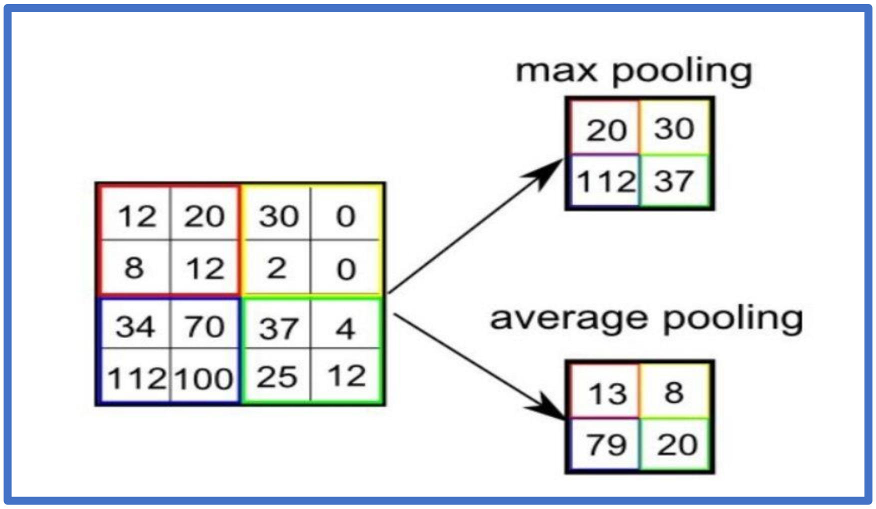

Figure 2.19: Max-Pooling Operation with a 4X4 Block Size [89]

3) Fully Connected Layer (classification layer)

The last layer of the network is a completely connected layer (FC), also known as a dense layer. The layer is said to be fully linked when all neurons in the layer below are connected to all neurons in the layer above. The generated feature map must be flattened into a feature vector to complete the connection between the output and previous layers. The final CNN layer utilized either a softmax or sigmoid activation function to classify the learned data, and the number of neurons in the output layer was proportional to the number of classes. These activation algorithms improve the multiclass and binary class classification performances, respectively [90]. The connection between the finished feature maps and a completely connected layer is shown graphically in Figure 2.20.

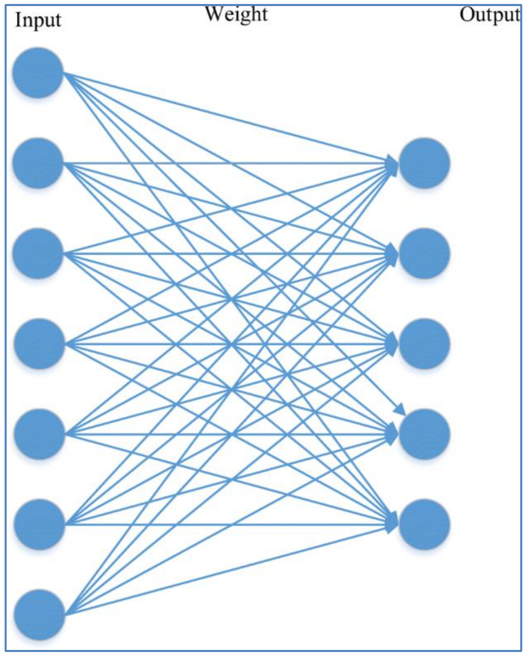

Figure 2.20: Connection Between Convolution Layer and Fully Connected Layer [9 1]

4) Non-Linearity Layers

To introduce nonlinearity into the activation map, nonlinear layers are typically placed immediately after the convolutional layer. There is a vast range of possible nonlinear operations; some of the most common ones are outlined below [91].

- Sigmoid: is the mathematical form for the sigmoid nonlinearity. A real number is then transformed into a floating point value. Equation 2.11 explain that.

$$f(x) = 1/1 + \exp(-x) \tag{2.11}$$

- Rectified Linear Unit (ReLU): This function is determined using the ReLU maximization formula f(x) =max (0, x). That is, the threshold was initially set to zero before activation was performed.

- In contrast to a flat ReLU, the slope of an activation function, known as a Leaky ReLU, is slightly negative. In Equation 2.12, the formal definition of the Leaky ReLU is as follows:

$$f(x)_{LeakyReLU} = \begin{Bmatrix} x & if\ X > 0 \\ mx & if\ x \leq 0 \end{Bmatrix} \tag{2.12}$$

- The softmax function operates on an input vector by exponentiation each element and then normalizing the resulting values by dividing them by the sum of all exponential values. This normalization step ensures that the output values fall within a range of 0 to 1 while also ensuring that their collective sum equals 1, thereby rendering them suitable for probabilistic representation. Mathematically, one can express the formula for computing the softmax function, as in Equation (2.13):

$$Softmax(x_i) = \frac{exp(\ x_i)}{\Sigma_j\ exp\left(x_j\right)} \tag{2.13}$$

where x_i refers to an individual input value that corresponds to a specific class. $exp(x_i)$ represents the result obtained by applying an exponential function to x_i. The term $sum(exp(x_j))$ denotes the aggregation over all classes and measures their respective exponential magnitudes.

5) Dropout Layer

Overfitting of the training dataset occurred when all features were added to the fully connected layer. Overfitting occurs when a model performs exceptionally well on training data but fails to generalize to new data. To address this issue, a dropout layer was used during training to arbitrarily prune the network and delete neurons and their connections [92]. Figure 2.21 depicts a dropout rate.

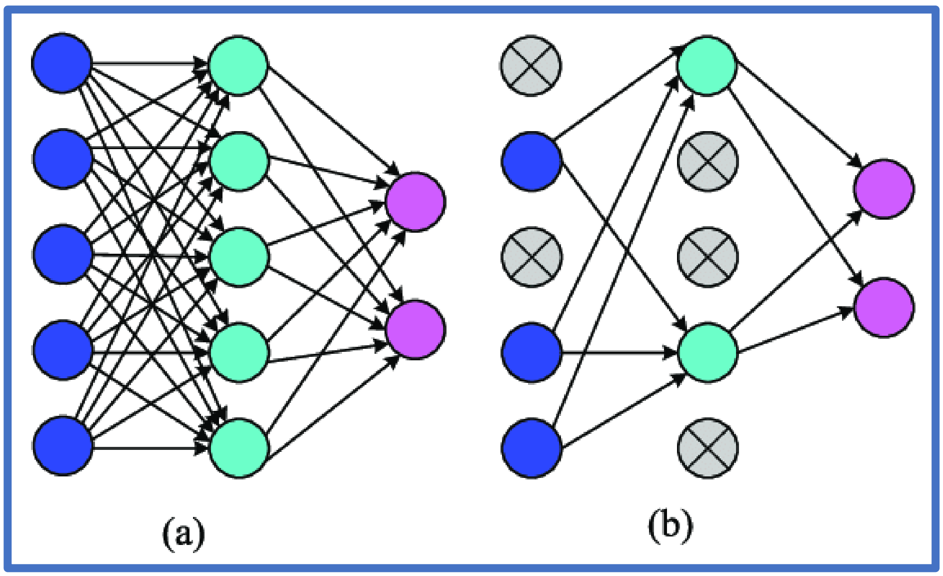

Figure 2.21: Schematic Comparison of (a) a Regular Neural Network and (b) a Neural Network Trained with Dropout [93]

6) Adam Optimization Algorithm

In contrast to stochastic gradient descent, Adam can iteratively change the network weights using only training data. AdaGrad and RMS Prop are Adam's go-to tools used for solving sparse gradients in noisy environments. Adam automatically adjusts the rate of learning for each parameter to speed up the convergence and improve the performance. The adaptive learning rates for each parameter were calculated by averaging the prior gradient and squared values.

These equations allow Adam to fine-tune the learning rate of each parameter by estimating the gradient's momentum and variance over time. Last update iteration [94], [95] can be calculated using Equation 2.14:

$$\theta = \theta - \alpha * m / (\sqrt{v} + \varepsilon) \quad (2.14)$$

where the first and second moments are m and v, respectively, is the learning rate, and is a small constant to prevent division by zero. By dynamically modifying the learning rate of each parameter in response to its prior gradients, the Adam optimizer blends momentum-based optimization with adaptive learning rates to enhance the training of deep-learning models [96].

2.10 Performance Evaluation

The datasets were then subjected to evaluation based on the following criteria.

A. Mean Squared Error (MSE)

Mean Squared Deviation (MSD) is a widely recognized metric used to determine the average of the squares of errors in an estimation procedure for an unobserved variable. Given that the error is squared, MSE always takes non-negative values, and values that are closer to zero are better.

In relation with the shape of the function's graph the MSE is the second moment of the error, making thus possible to incorporate both the variance of the estimator (i.e. how widely are the estimates form one data sample to another spread) and its bias denotes the term "error," more specifically the difference between the mean estimated value and the actual value [97], [98]. MSE is calculated using Equation 2.15.

$$MSE = [\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2]^{1/2} \qquad (2.15)$$

Where X_i denote to the original sample data, and Y_i refered to the processed one.

B. Mean Absolute Error (MAE)

This method quantifies the difference or divergence between two continuous variables. Given that X and Y are paired variables representing the same underlying phenomenon, Instances of Y versus X can be illustrated by contrasting prophecy versus observation, posterior time versus initial time, and gauge approach versus ersatz measurement approach.

The present study concerns a scatter plot comprising n data points, where each point i is characterized by its coordinates (xi, yi). The mean absolute error (MAE) is defined as the arithmetic mean of the absolute vertical deviations of all data points from the Y=X line, which is commonly known as the one-to-one line [98], [99]. Furthermore, it can be stated that MAE refers to the average horizontal deviation between the Y=X line and each individual point. MAE is computed based on Equation 2.16.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \qquad (2.16)$$

Given a set of n model errors, is computed as (ei 5, i = 1, 2... n).

C. Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) is a statistical measure utilized to determine the average deviation between the predicted and observed values in a regression analysis [98], [100]. The aforementioned value is computed as in Equation 2.17.

$$MAPE = (1/n) * \Sigma(|Oi - Pi|/Oi * 100 \qquad (2.17)$$

where:

Σ The symbol in question is a sophisticated mathematical notation that represents the operation of addition, commonly known as "sum".

Pi The value predicted for the ith observation is referred to as

Oi value for The ith observation.

n is the sample size.

D. Root Mean Squared Percentage Error (RMSE)

The prevalent practice in the fields of meteorology, air quality, and climate research is to employ conventional statistical metrics to evaluate the efficacy of models as in Equation 2.18.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \qquad (2.18)$$

Given a set of n model error samples, denoted by (ei. 5, i = 1, 2... n) [98].

2.11 Evaluation of the Recommender System

The evaluation of the hybrid recommender system is crucial to assess its efficacy in offering accurate and satisfying movie recommendations. These cosine similarity metric provide an in-depth assessment of the system's prediction ability by quantifying the difference between the projected and actual scores.

Cosine similarity, a cornerstone of collaborative filtering, describes the degree to which two vectors in a multidimensional space have the same direction. For hybrid recommendation engine, this can be stated creatively as in Equation 2.19 [11], [75]:

$$cosin(\overline{R_u}, \overline{R_v}) = \frac{\sum_{i=1}^{n}(R_{u,i} * R_{v,i})}{\sqrt{\sum_{i=1}^{n}\left(R_{u,i}\right)^2}\sqrt{\sum_{i=1}^{n}(R_{v,i})^2}} \quad (2.19)$$

Where Rw and Rv are the ith component of vectors R, The dimension of the feature space, denoted by n, is described. Collaborative filtering is summed up in this formula, which calculates the cosine of the angle between two user preference vectors. Cosine similarity scores near 1 indicate strongly aligned preferences, which makes for a solid basis for group suggestions.

These ratings and scores capture the symphony of our recommender system's efficacy as we traverse the cinematic world with the help of numerical precision and computational flair. The dedication to giving each user not just films, but an individually designed cinematic experience, is demonstrated by the intricate dance of algorithms and measurements.

Chapter Three

The Proposed System

3.1 Overview

This chapter presents the carefully developed system and its execution. This chapter, reveal the proposed movie recommendation system's key components, each designed to improve user experience and prediction accuracy. This dissertation focuses on creating a recommendation engine that accounts for users' interests and movie interactions to propose films. It used a hybrid strategy, integrating advanced deep learning with well-established regression algorithms. This combination propose films that match users' preferences. System architecture is crucial to attaining these goals. The complex structure manages the whole suggestion process. It uses a revolutionary hybrid deep learning method to reliably anticipate user preferences based on movie viewing history. This chapter goes through data regression to predict movie ratings using machine learning techniques including Linear Regression, Gradient Boosting Regression, Decision Tree Regression, Stochastic Gradient Descent, and Bayesian Ridge Regression. Also offers a description of hybrid deep learning model that integrates layers using a computational graph. Complex patterns from user-movie interactions are extracted by this approach to make precise movie recommendations. Also discuss key performance measures including MSE, MAE, MAPE, and RMSE. This foundation prepares for extensive analysis of each component and method in later parts.

3.2 The Proposed System Architecture

The proposed movie recommendation system rests on the proposed system architecture that encompasses a multitude of components and has been carefully structured to ensure smooth and efficient functioning. The fundamental essence of this architecture is a cutting-edge hybrid deep learning method which equips our model to forecast preferences with great precision. This intricate architecture operates in two distinct but interconnected phases: Personalization plays two critical roles here; that is, data preprocessing, and data regression towards final recommendations based on personal preferences.

Figure 3.1 shows the proposed system architecture. The structure highlights, the main factors and stages of implementation for the decision making mechanism. They also include data preprocessing for handling of the missing value and one-hot encoding of the categorical features and normalization of the numerical features. Movie rating predictions are then generated using Linear Regression, Gradient Boosting Regression, Decision Tree Regression, Stochastic Gradient Descent Regression, and Bayesian Ridge Regression algorithms of machine learning in the regression phase. The system architecture has been augmented by DL techniques. It provides deep learning components, like embedding layer, densely connected layers and dropout layers, to capture the sophisticated patterns in users-movie connections. The developed DL model is aimed at accurate movie recommendation considering multifarious relations among user traits and preferred films. Further, it considers the measuring of the performance metrics such as MSE, MAE, MAPE, and RMSE for the purpose of evaluating the system’s efficacy.|

Lastly, as shown in figure 3.1, it captures the recommend phase that entails movie recommendations made and submitted to the users.

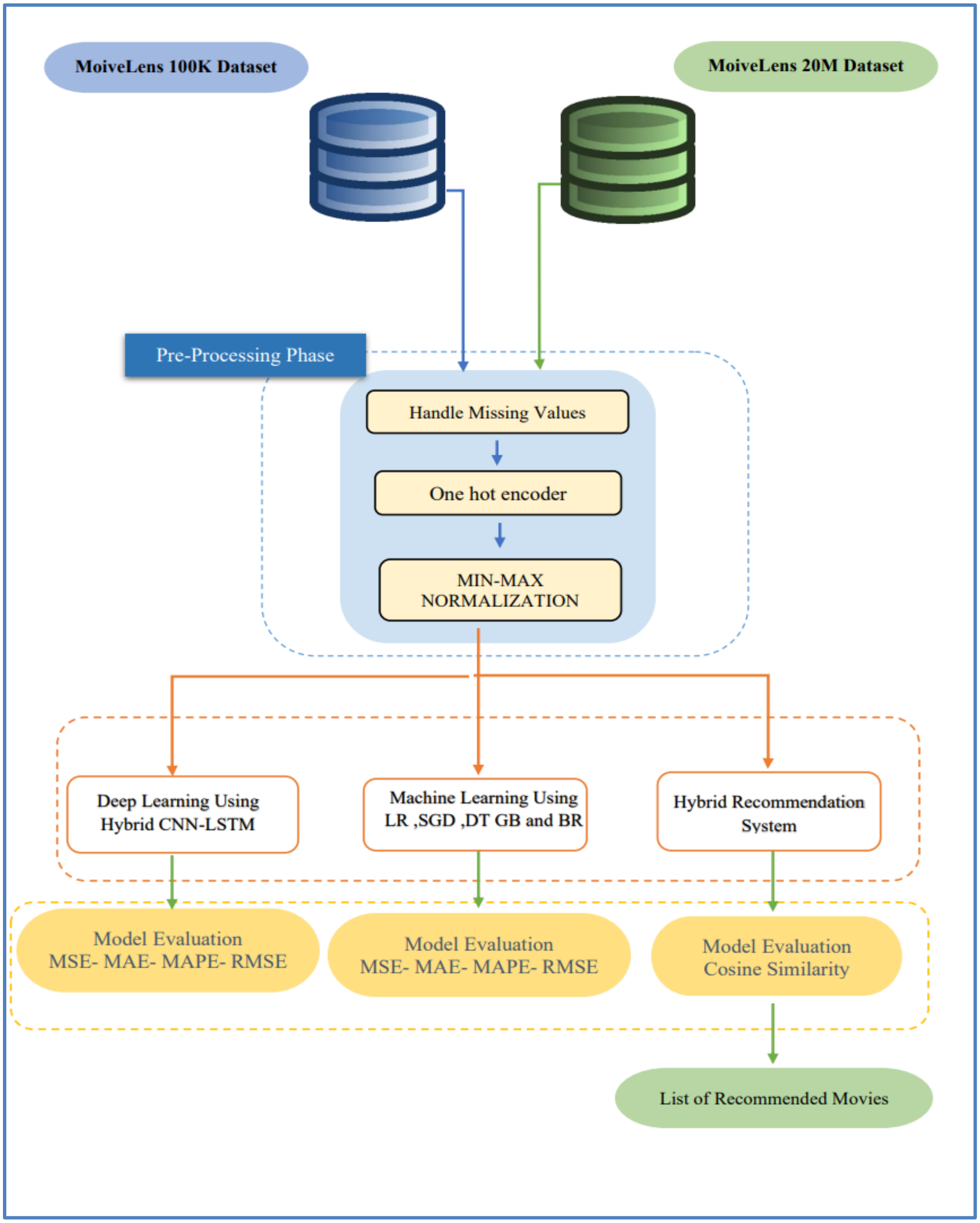

Figure 3.1: The Proposed System Structure

3.2.1. Description of the Dataset

The research conducted in this study is dependent on two primary datasets, specifically Movielens20M and Movielens100K. These datasets are of utmost importance as they are crucial for both training and evaluating the performance of our movie recommendation system. The datasets serve as the fundamental basis for the development of our algorithms, guaranteeing that our system produces movie recommendations that are both significant and pertinent.

- The dataset used in this study is the Movielens20M dataset. The dataset provided is a valuable resource including a large amount of user-movie interactions, encompassing numerous data points that facilitate the acquisition of insights and the provision of precise suggestions by our system. The system incorporates a diverse array of user preferences, so enabling the algorithms to acquire knowledge and adjust with high efficacy.
- The dataset used in this study is the Movielens100K dataset. In comparison to its more expansive alternative, Movielens100K provides a smaller but still significant dataset, including 100,000 instances of user-movie interactions. The platform functions as an initial experimental environment for the advancement and enhancement of our algorithms. The reduced dataset proves to be quite advantageous for the purposes of early model training and assessment.

Both datasets have a same structural framework and encompass the subsequent essential columns:

- The movie identifier. The collection includes a distinctive code for each movie, enabling reliable differentiation across movies.

- A Comprehensive Analysis The title of a film serves as a crucial characteristic that offers valuable insights about the substance of the movies. Content-based recommendations rely on a crucial element that allows our system to effectively pair users with films that are in line with their preferences.
- The subject matter of this discussion pertains to several categories or classifications of artistic expression, sometimes referred to as genres. Films are classified according to their genres, and this characteristic forms the basis for content-based recommendations. The analysis of user preferences aids in the identification of favorite genres inside our system, hence enhancing the accuracy of our recommendations.

The datasets, containing extensive and detailed user-movie interactions, play a crucial role in sustaining our recommendation engine. By using these datasets, we guarantee that our system is guided by data, allowing it to consistently improve its suggestions and better the overall user experience. By using both the comprehensive Movielens20M dataset and the concise Movielens100K dataset, our recommendation system is able to accommodate a wide range of user preferences, encompassing both general and specialized interests. This approach enables us to strike a harmonious equilibrium between the accuracy of recommendations and the efficiency of the system.

3.2.2 Data Preprocessing Phase

To get the most out of the Movielens20M and Movielens100K datasets, we go through an exhaustive step of data preprocessing. The next important phase is the preprocessing stage; these guarantee consistency of sets, readiness to train our recommendation system.

1. Handling Missing Values

In this respect, the first step of data preprocessing is carefully filling in the existing empty values within the datasets as shown in Algorithm 3.1. Lack of complete data is quite common, which could affect the reliability and effectiveness of the system building. However, such challenge is resolved by using suitable imputation techniques that 'fill up' missing data and keep the dataset intact. This ensures that the algorithms will work with complete and dependable information.

Algorithm 3.1: Handling Missing Values
Input: Dataset **Output:** Dataset with missing values filled **Begin** **Step 1:** Initializing the dataset containing missing values. **Step 2:** For each column of the dataset. For each row in the column: • If the value is missing (e.g., NaN or null): • Use an appropriate imputation method to replace the missing field value. • Record the value and update in the dataset. **Step 3:** Impute the dataset with missing values. **End**

2. One-Hot Encoder

The one-hot encoding format is used to convert categorical data like movie genres. Machine learning algorithms cannot use categorical variables, as they are. Then convert them into binary vectors representing the presence of each category by using a one hot encoding format as shown in Algorithm 3.2. By converting the numerical data into appropriate categorized types like text or labels, that makes the data machine learning ready for recommendation.

Algorithm 3.2: One-Hot Encoder
Input: Categorical data column (categorical_column) **Output:** One-hot encoded binary vectors **Begin** **Step 1:** Instantiate a one-hot encoding dictionary (one_hot_dict). **Step 2:** • Create a function that will group each unique category under the categorical_column. • Build an array of zeros in all categories and one in the relevant category. • Place the binary vector in the one_hot_dict using the category as the key. **Step 3: For each row of the dataset.** **For the categorical_column in the row:** Use the appropriate binary vector from the one_hot_dict in place of the categorical value. **Step 4:** Retain the one-hot encoding dataset. **End**

3.Min-Max Normalization

Firstly, Min-Max normalization is an important aspect of data preprocessing where numerical features are standardized. Here, this normalization technique converts the numerical data into a range of zero to one. By standardizing the data, the machine learning algorithms will be more effective in the way they undergo training on the varied scale features. We equalize the scales of all numerical features so that every attribute contributes similarly to the training course.

Equation 3.1 shows the calculation of normalization. This is described farther in Algorithm 3.3.

$$X` = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

Where X is the attribute data, Min(x), Max(x) are the minimum and maximum absolute of X respectively. X' is the new value of each entry in data.

Algorithm 3.3: Min-Max Normalization
Input: Numerical data column (numerical_column)
Output: Normalized numerical data
Begin
Step 1: Set the numerical_column minimum value as X_min and the numerical_column maximum value as X_max.
Step 2: Foreach value X in numerical_column:
Calculate the normalized value (X_norm) using the formula:
X_norm = ((X – X_min)/(X_max – X_min)).
Step 3: Finally, submit a dataset that has been normalized in terms of numerical values.
End

3.2.3 Regression Algorithms Phase

To achieve this goal, we use a range of regression algorithms, all differing in their style of prediction of users' tastes. To be effective, such a system should base its provision of movie recommendations on individual preferences (see Algorithm 3.4).

Algorithm 3.4: Predicting the Rating using ML

Input:

- User preferences data (UserFeatures)
- Movie data (Movies)
- Movie ratings data (Ratings)

Output:

- Trained recommendation model

Begin

Step 1: Load MovieLens20M dataset or MovieLens100K dataset

Step 2: Data Preprocessing Phase

For each dataset:

- Handle missing values: Appropriate methods should be used to fill in missing data points.
- One-Hot Encoding: Turn categorical data such as movie genres to numbers with the help of the one hot encoding algorithm.
- Apply Min-max normalization for all numerical features.

Step 3: Regression Phase

For each user:

Implement various regression algorithms:

- Apply Linear Regression using algorithm 2.1
- Apply Gradient Boosting Regression using algorithm 2.2
- Apply Decision Tree Regression using algorithm 2.3
- Apply Stochastic Gradient Descent Regression using algorithm 2.4
- Apply Bayesian Ridge Regression using algorithm 2.5

Step 4: Evaluation

Use of performance evaluation metrics like MSE, MAE, MAPE, and RMSE.

End

A. The Linear Regression Algorithm

Within the movie recommendation system, the Linear Regression technique is used to make predictions by considering a wide range of user and movie-specific factors. By analyzing user interactions with them, including user evaluations, Linear Regression provides an easy yet effective way of prediction, taking into account many movie factors such as genres and release years. As shown in Algorithm 2.1, this requires the creation of a regression equation that best explains the discrepancies between observed and anticipated scores. This method allows the system to accurately forecast user preferences based on their behavior as it relates to a wide variety of movie characteristics.

To better cater to a user's likes, linear regression can utilize to determine which characteristics have an impact on their movie-viewing habits and preferences. The model analyses data such as movie genres, release dates, and user ratings to discover trends in users' actions. Using these characteristics, a linear equation is formulated, and its coefficients are learned through training by aiming for the lowest possible prediction error. Once the Linear Regression model is trained, it can be used to predict a user's rating of a movie based on their previous experience with movies that are comparable to the one being rated and the features those movies have in common. In order to make useful movie suggestions, the model draws on the associations it has learnt during training to assign ratings to films that viewers may trust. This method improves the movie recommendation system's ability to provide individualized recommendations, which in turn boosts user happiness and enjoyment.

B. The Gradient Boosting Algorithm

Gradient Boosting Regression, a powerful ensemble learning technique, improves prediction accuracy, making it ideal for movie recommendation systems. This method improves predictions and corrects earlier errors. This approach is ideal for capturing minor dependencies in user-movie collaborative relationships, generating an effective prediction engine that learns from several weak learners and becomes sensitive to movie rating details. Gradient Boosting Predicting movie ratings based on user preferences is a regression machine learning task. A powerful and accurate ensemble model is created by integrating many weak regression model predictions. This ensemble technique captures intricate user-movie relationships, which is necessary for personalized and accurate recommendations.

Initial models, usually simplistic, make data-driven predictions. Since these projections are based on a simple model, they are often inaccurate. Gradient Boosting Regression then finds these flaws by comparing initial predictions to user movie evaluations. Gradient Boosting Regression develops weak models to fix earlier faults in subsequent iterations. These new models seek to reveal overlooked user-movie interaction patterns and nuances. The algorithm improves its predictions by learning from prior models' faults. This process repeats until a certain number of weak models or accuracy is reached. The final prediction combines all weak model predictions, with more accurate models having a bigger impact. This boosted ensemble of models greatly improves movie rating predictions as seen in Algorithm 2.2. Gradient Boosting Regression identifies user preferences and small changes in movie recommendation systems. It can capture complicated movie rating patterns including user tastes and the impact of genres, actors, and directors on movie choices.

C. The Decision Tree Algorithm

Decision Tree Regression is a powerful non-linear model that helps movie recommendation systems comprehend and predict movie ratings. This approach uses decision trees, hierarchical structures that divide data by qualities or features. A tree-like structure that decides on each attribute predicts user preferences and movie ratings. By capturing non-linear data patterns, this method allows for precise and personalized movie suggestions.

The decision tree regression model iteratively divides the dataset into attributes or features. The feature with the greatest prediction error reduction is chosen. You can halt this process by reaching a tree depth or establishing nodes with a certain number of data points. Decision trees have predictions at each leaf node. In a movie recommendation system, leaf nodes represent projected ratings. To assess user interest in a film, the model considers user preferences, genres, and past interactions as explained in Algorithm 2.3. Decision Tree Regression captures complex, non-linear user-movie preferences relationships well. It can reveal intricate trends like users who like action films with a given actor or romantic comedies from a certain decade. This granularity allows the algorithm to make highly personalized movie suggestions that match each user's tastes.

D. The Stochastic Gradient Descent Algorithm

The movie recommendation system makes use of the optimization method known as stochastic gradient descent (SGD), which is typically used in regression methods. When it comes to recommending movies to others, frequently deal with massive datasets that include countless user-movie interactions. This necessitates hypothesizing a user's potential reaction to and rating of a film.

SGD is used to perform iterative parameter optimization on the model. With SGD, rather than processing the full dataset at once, just a tiny, randomly selected fraction is processed at each iteration. The key idea is to tweak the model's parameters after analyzing certain data, improving the model's accuracy in small increments. This method works well with huge datasets, such as those seen in movie recommendation systems. SGD increases productivity and responsiveness to user preferences shifts by processing less data in each iteration. The iterative nature of SGD improves the process's efficiency and speed, letting the recommendation system respond rapidly to fresh data and changing user preferences as explained in Algorithm 2.4. Recommendations are always up-to-date and reflective of each user's preferences since the model is always being fine-tuned based on a stream of user interactions. In conclusion, SGD is an important optimization technique that improves our movie recommendation system by iteratively improving the model's predictions based on a smaller and smaller sample of the data.

E. The Bayesian Ridge Algorithm

When it is used within the framework of a Movie Recommendation System, the Bayesian Ridge Regression method significantly contributes to improving the predicted accuracy of movie ratings. It's a method that helps us see the whole scope of rating uncertainty, which ultimately leads to more exact suggestions. Here's more on how Bayesian Ridge Regression fits into the bigger picture. In order to address and quantify the uncertainties included in the movie ratings data, Bayesian Ridge Regression is based on the principles of Bayesian approaches. Movie preferences are very personal and impacted by a wide range of circumstances; as a result, it is crucial for a movie recommendation system to account for noise in the data. In addition to taking this noise into account, Bayesian Ridge Regression also quantifies the forecast uncertainty.

Bayesian Ridge Regression's basic principle is to model ratings in a probabilistic way, rather than only as point estimates. It takes into account the fact that a user's choice may shift depending on their current state of mind, making it unlikely that a single rating can accurately represent that preference. The program attempts to capture the user's ambiguity in their preferences. To estimate the model's parameters, Bayesian Ridge Regression makes use of Bayesian statistics. The best-fit parameters and their associated uncertainties are included in this estimate. Rather than giving a single numerical rating, it gives a range of possibilities based on a probabilistic method. Taking into account the subjectivity of individual tastes, this distribution shows the possible range of ratings a user may provide to a film as shown in Algorithm 2.5.

The use of such a variety of regression algorithms enables us to handle the complexities of user-movie interaction with our recommendation system. Every algorithm has its own virtues which, when combined with others, increase overall accuracy and effectiveness of movie suggestions. Thus, it takes into account the user’s personal preference in suggesting movie suggestions for a delightful cinema time.

3.2.4 The Deep Learning Model

The study also proposes a deep learning method to complement traditional linear regression equations. The use of this innovative model is based on a computational graph with multiple layers which allows extracting detailed patterns from user-movie interactions. As a result, very accurate movie recommendations are made for each person. Algorithm 3.5 shows the prediction based on the proposed DL technique.

Algorithm 3.5: Predicting the Rating using DL

Input:

- User preferences data (UserFeatures)
- Movie data (Movies)
- Movie ratings data (Ratings)

Output:

- Trained recommendation model

Begin

Step 1: Load MovieLens20M dataset or MovieLens100K dataset

Step 2: Data Preprocessing Phase, For each dataset:

- Handle missing values: Appropriate methods should be used to fill in missing data points.
- One-Hot Encoding: Turn categorical data such as movie genres to numbers with the help of the one hot encoding algorithm.
- Apply Min-max normalization for all numerical features.

Step 3: The Deep Learning Model

For each user in the dataset:

- Utilizing a deep learning model to develop user and movie embedding.
  - Input Layers: Acceptance of user and movie IDs through two input layers.
  - Embedding Layers: The input IDs are mapped into dense vector (embedding) of 64-dimensional that encapsulates similarities between users and items.
  - Flatten Layers: Prepare the embedding vectors for subsequent processing by flattening them.
  - Dropout Layers: Randomly discard some neurons in the flatten and dense layer to avoid overfitting when training.
  - Dot Layer: It calculates the dot product of user and movie embeddings capturing the affinity between them.
  - Dense Layers: Dense layers of interconnections, on top of that, continue processing and extracting of the interactions and patterns from data.
  - Output Layer: The last one gives a prediction on a user-movie pair's movie rating.
- Evaluate user-movie inner product.
- Prevent overfitting through the use of dropout layers
- Evaluate a final predicted movie grade.

Save the predicted score into ‘user_movie_ratings’.

Step 4: Evaluation

Use of performance evaluation metrics like MSE, MAE, MAPE, and RMSE.

End

The model is a hybrid model, which means it takes into account both simplistic and sophisticated connections between user characteristics and movie preferences (simple and complex relationships between user features and movie preferences). The algorithm offers a healthy mix of the benefits from deep-learning models and conventional regression approaches. Figure 3.3 shows the deep learning layers within the proposed system architecture, key components include:

- Embedding Layers: In this way, it turns raw user and movie IDs into dense vectors. These are features that capture latent influential factors for users’ preferences and movie characteristics. These are crucial in getting an insight of user-movie interaction and predicting accurately.
- Flatten Layers: Our prediction lies in feature representation produced using these flattened vectors obtained from the last two embedding layers. This process involves flattening these vectors to bring them into a form convenient for further processing.
- Dot Product Layer: In terms of interaction between user and movie embedding, the dot product layer is crucial. These embedding embody the affinity or relationship that is existing to users when it comes to movies. This calculation constitutes basic understandings of how close a particular movie gets to a person’s taste.
- Dense Layers: The interactions are then processed through several dense layers and further distill the intricate pattern from the data. The existence of

these layers enables the model to capture intricate correlations between a user and various films watched or a non- linear feature in user-movie interaction. These are vital for obtaining good predictions.

- Dropout Layers: The dropout layers are essential part of our architecture because they provide good protection against overfitting. Dropout layers also help us improve the generalization of our recommendation by dropping out temporary a number of Neuron during training. The model is able to generate good forecasts for new datasets.
- Output Layer: Finally, the output layer completes the puzzle. It represents the final prediction of an ideal score for a user with respect to each and every movie. As such, this forms an integral part of our recommendations and what the end user is able to view as movie recommendations. Figure 3.2 shows the DL model.

Layer	Output Shape	Param #	Connected to
input_1	(None, 1)	0	-
input_2	(None, 1)	0	-
embedding_1	(None, 1, 64)	39040	input_1
embedding_2	(None, 1, 64)	622336	input_2
flatten_1	(None, 64)	0	embedding_1
flatten_2	(None, 64)	0	embedding_2
dropout_1	(None, 64)	0	flatten_1
dropout_2	(None, 64)	0	flatten_2
dot_1	(None, 1)	0	dropout_1, dropout_2
dense_1	(None, 96)	192	dot_1
dropout_3	(None, 96)	0	dense_1
dense_2	(None, 1)	97	dropout_3

Figure 3.2: The Deep Learning Model

The architecture is set up using advanced deep learning libraries and carefully designed for high-performance of the system. This base forms a sound basis for offering personalized movie recommendations matching user's taste. In particular, the deep learning model fills this gap by combining traditional regressions with modern DL that results into an effective recommender system.

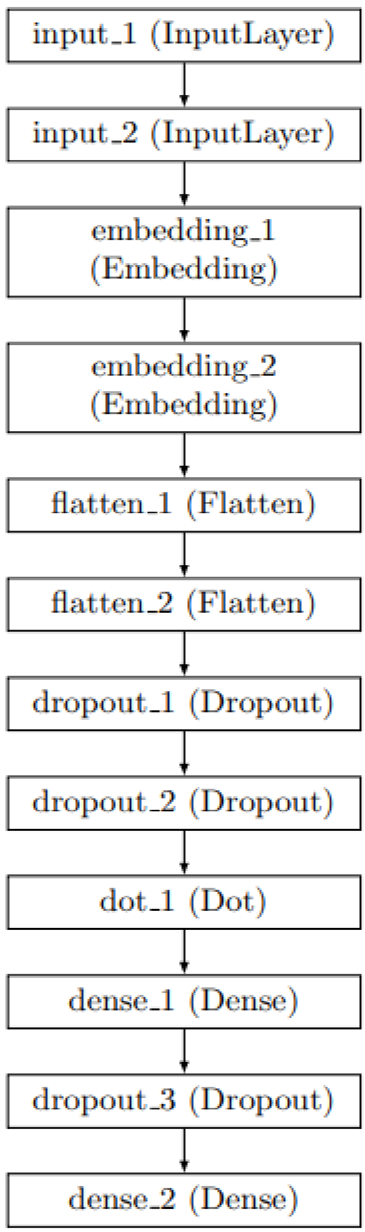

Figure 3.3: The Deep Learning Model Layers

3.2.5 Hybrid Recommendation System

For each user, there is a hybrid recommendation made via content-based filtering and collaborative filtering so as to yield relevant movie suggestions. To begin with, the system collects movie rating data from other users which is based on the history of ratings on a collaborating filter model. At this level, the system identifies users with common viewing tastes for movies they liked and recommends such movies. The system also uses content-based filtering where it looks at movie features which include genres, actors, and director to come up with movies that match the users previous preference. Hybrid approach combines recommendations from collaborative and content based filtering approaches providing for good variety of movies which are suitable according to individual preference and movie watching history as shown in Figure 3.4. Combination of these techniques enables delivery of accurate and various recommendations on movies leading to rewarding user service experience.

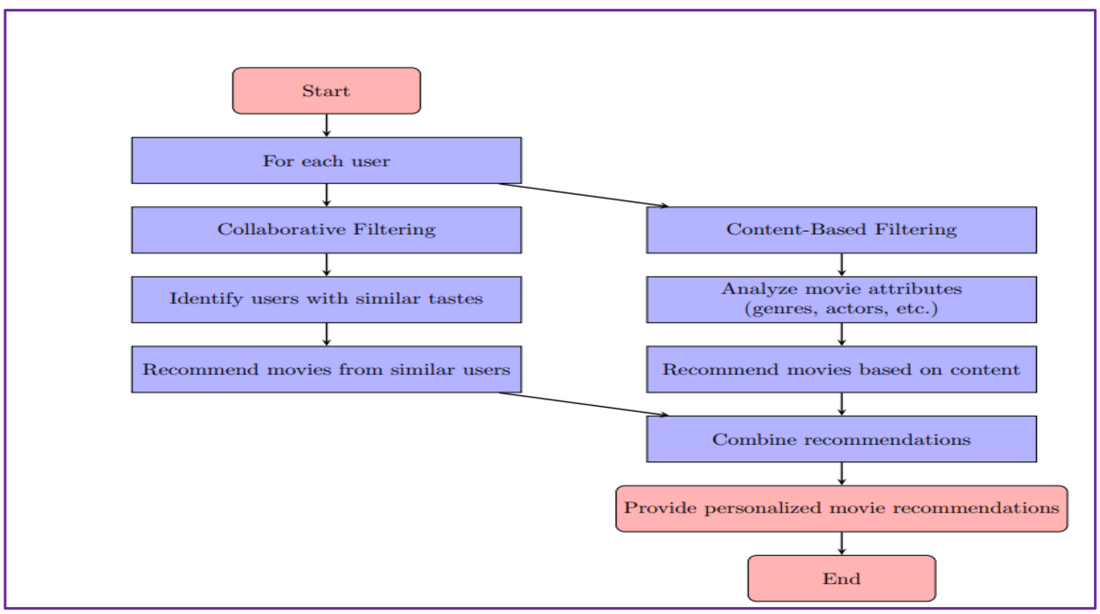

Figure 3.4: Hybrid Recommendation System

The Hybrid Recommendation System in Algorithm 3.6 provides the target_user with tailored film suggestions by combining collaborative filtering with content-based filtering. Collaborative filtering finds users with similar likes and recommends films that have been popular among them. On the other hand, content-based filtering takes into account the target_user's tastes when making recommendations by analyzing movie attributes and determining content similarity scores. The hybrid method takes into account both sets of suggestions and returns a well-rounded list of the top N films the user should watch. This method ensures a complete and accurate list of film recommendations based on user actions and content characteristics. Algorithm 3.6 shows the proposed hybrid recommendation system.

Algorithm 3.6: Hybrid Recommendation System

Input:

User ID (target_user)

User preferences data (UserFeatures)

Movie data (Movies)

Movie ratings data (Ratings)

Output: Hybrid recommender system based on collaborative and content filtering.

Begin

Step 1: Collaborative Filtering

- This involves collaborative filtering where the system looks at the earlier ratings and tastes of the past in order to pick similar users to the target user.
- Make collaborative filtering recommendations involving movies liked by other users.

Step 2: Content-Based Filtering

- Analyze the characteristics of movies like genre, actor, director etc, and do a content-based filtering.
- Determine similarity of movie characteristics with regard to specific_user preferences.

- Constructing content-based recommendations by means of content similarity.
- Describe four leadership styles and how they are used differently by leaders.

Step 3: Combine Recommendations

- Create a hybrid approach, involving both collaborative and content based techniques for obtaining recommended movie suggestions.
- Make sure a balanced list is created from both methods by ensuring their recommendations.

Step 4: Sort and Filter Recommendations

- Arrange the hybrid suggestions according to a weighted score or some kind of different criteria, for example, predicted ratings.
- Use the filter process to bring out just N-most movies having the highest ratings.

Step 5: Output Recommendations

Present the resulting top-N personalized movie suggestions for the target_user based on using a hybrid method of collaborative and content-based filtering.

End

Additionally, the work gives particular consideration to improving prediction reliability through the examination of essential performance measures such as Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). They are crucial measures that appraise overall system efficiency, indicating what is functioning or not.

Chapter Four

Results and Analysis

4.1 Overview

In this significant chapter, we commence our exploration of the results derived from our comprehensive efforts in constructing a resilient movie recommendation system through the utilization of a hybrid methodology. In this study, we extensively explore the implementation of our proposed system, providing comprehensive insights into the complex intricacies of its operating environment. Furthermore, we present the outcomes of each phase of the system, encompassing data preprocessing, the utilization of machine learning algorithms, and the introduction of a deep learning approach.

4.2 System Implementation

The portrayal of the implementation environment holds significant significance as it establishes the basis for evaluating the anticipated behavior and efficacy of our recommendation system. The implementation of our suggested system will be thoroughly described, accompanied by charts, in order to offer a clear comprehension of its operation.

The programming language selected for the implementation of the suggested system was Python version 3.6. The operations of our system were conducted using an HP laptop that was equipped with a sixth-generation Core i7 CPU, 16 GB of RAM, and an NVIDIA GTX 6G graphics card. The laptop is equipped with a solid-state drive and utilizes the 64-bit version of the Windows 10 operating system. The recommendation system incorporates a visually intuitive graphical interface that effectively presents the pictures and information produced after the execution of each function.

4.3 Implementation Results of the Proposed System Phases

This section explores the core of our recommendation system, which utilizes a combination of conventional regression methods and deep learning methodologies. Our objective is to clarify the procedures used and present a thorough summary of the outcomes achieved throughout each stage of our system's progression. The main stages encompass data preparation, the use of machine learning techniques, and the deep learning methodology.

4.3.1 Results of the Linear Regression Algorithm

Linear regression algorithms are considered one of the simplest and most popular machine learning techniques for performing data regression tasks. In this section, we present the results obtained from applying the linear regression algorithm to predict movie ratings. The purpose is to evaluate the algorithm's performance and compare it with the results from the proposed deep learning model to assess its effectiveness in providing accurate predictions for the MovieLens datasets.

The results of the linear regression algorithm for both the MovieLens 100K and MovieLens 20M datasets are presented in Table 4.1 and Figure 4.1. These results are based on key evaluation metrics, including Mean Absolute Error (MAE), Mean Square Error (MSE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE).

Table 4.1: Linear Regression Algorithm Prediction Results				
	MAE	MSE	MSPE	RMSE
Movie Lens 100k	1.04	0.81	0.37	1.02
Movie Lens 20M	1.09	0.83	0.37	1.04

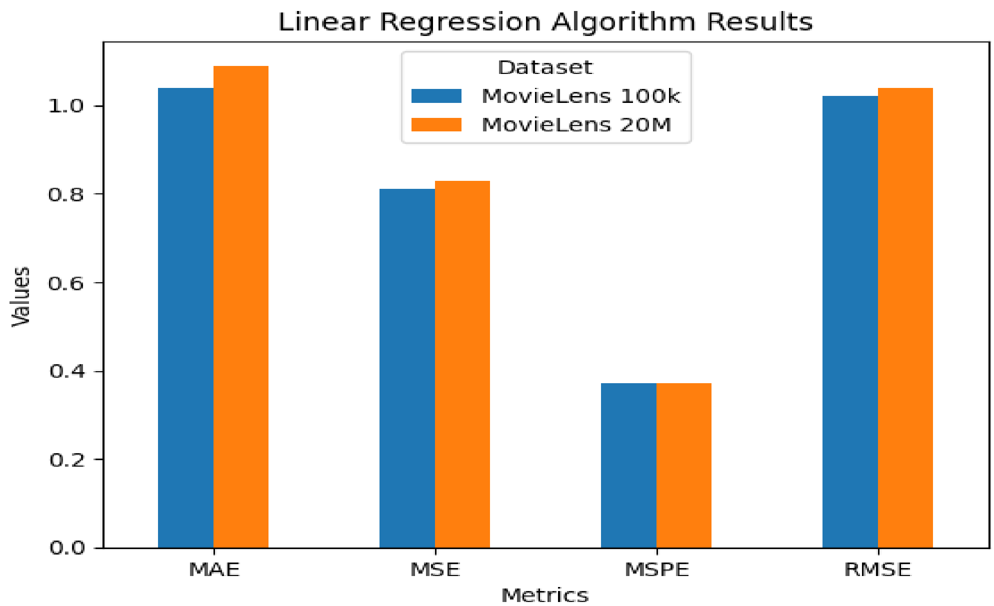

Figure 4.1: Linear Regression Algorithm Prediction Results

The results indicate that the linear regression algorithm performs well in predicting movie ratings for both datasets. The Mean Absolute Error (MAE) values of approximately 1.04 and 1.09 for the MovieLens 100K and MovieLens 20M datasets, respectively, suggest that the algorithm's predictions are, on average, off by approximately one rating point. The Mean Square Error (MSE) values of 0.81 and 0.83 for the respective datasets indicate the average squared difference between predicted and actual ratings, further underlining the algorithm's effectiveness. Additionally, the Mean Squared Percentage Error (MSPE) values of 0.37 signify the algorithm's relatively low percentage error in predicting ratings. The Root Mean Squared Error (RMSE) values of 1.02 and 1.04 reflect the standard deviation of prediction errors. These low RMSE values reinforce the algorithm's accuracy in predicting movie ratings.

Overall, the linear regression algorithm shows promise in providing accurate movie recommendations. These results serve as a strong foundation for comparison with the deep learning model in the subsequent sections, where we aim to evaluate whether the proposed hybrid approach can outperform this traditional regression method.

4.3.2 Results of the Gradient Boosting Algorithm

The Gradient Boosting Regression method, which is a robust ensemble learning technique, plays a crucial role in our recommendation system. The use of numerous models is employed in order to enhance the precision of predictions. The outcomes of its implementation are displayed in Table 4.2, demonstrating its ability to make accurate predictions on both the MovieLens 100K and MovieLens 20M datasets. Table 4.2 and Figure 4.2 show the prediction results obtained from the Gradient Boosting Algorithm.

Table 4.2: Gradient Boosting Algorithm Prediction Results

	MSE	MAE	MSPE	RMSE
Movie Lens 100k	0.92	0.75	0.34	0.96
Movie Lens 20M	1.02	0.8	0.36	1.01

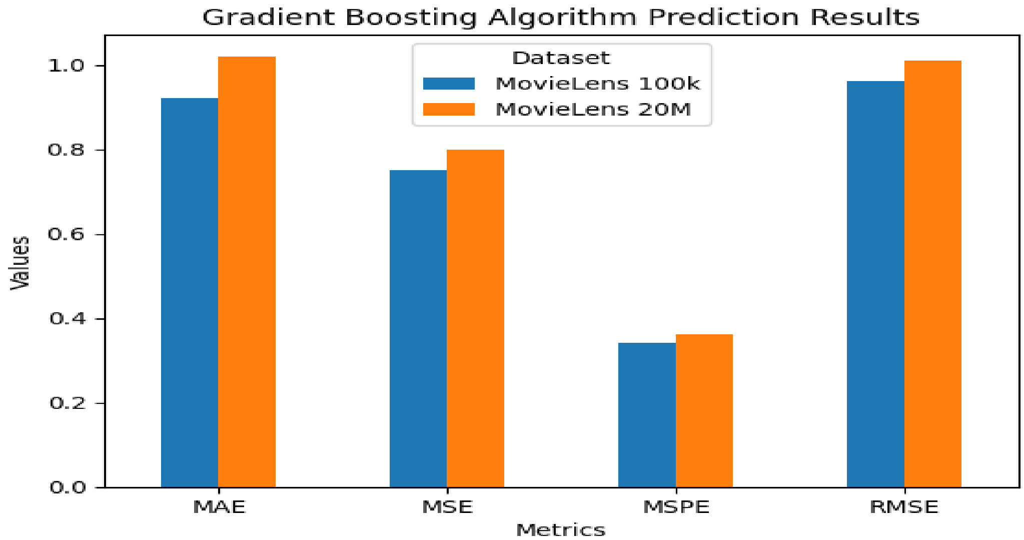

Figure 4.2: Results Obtained by the Gradient Boosting Algorithm

The dataset was evaluated using four performance metrics: mean squared error (MSE), mean absolute error (MAE), mean squared percentage error (MSPE), and root mean squared error (RMSE). The Movie Lens 100K dataset has correlation coefficients of 0.92, 0.75, 0.34, and 0.96.The dataset used in this study is the Movie Lens 20M dataset. The values provided are 1.02, 0.8, 0.36, and 1.01. The Mean Square Error (MSE) value of 0.92 is observed for the MovieLens 100K dataset, which signifies the average squared discrepancy between the anticipated and actual ratings. In contrast, the MovieLens 20M dataset has a mean squared error (MSE) value of 1.02. A smaller mean squared error (MSE) indicates superior predictive accuracy, implying that the Gradient Boosting algorithm exhibits more accuracy in forecasting ratings for the MovieLens 100K dataset.

The Mean Absolute Error (MAE) of the MovieLens 100K dataset is 0.75, indicating the average absolute discrepancies between the projected and actual ratings. The Mean Absolute Error (MAE) for the MovieLens 20M dataset is 0.8. Once again, it can be shown that a decrease in Mean Absolute Error (MAE) corresponds to an increase in the accuracy of forecasts. The findings suggest that the Gradient Boosting technique has a modest superiority in terms of Mean Absolute Error (MAE) when applied to the MovieLens 100K dataset. The Mean Absolute Percentage Error (MAPE) is a relative metric, with values of 0.34 and 0.36 seen for the MovieLens 100K and MovieLens 20M datasets, respectively. A decreased Mean Absolute Percentage Error (MAPE) signifies a reduced proportion of inaccuracies in the forecasting outcomes. The Root Mean Squared Error (RMSE) is a statistical metric that quantifies the level of error present in predictions by considering both their size and direction. The accuracy score for the MovieLens 100K dataset is 0.96, whereas for the MovieLens 20M dataset it is 1.01. Once again, it can be observed that a smaller root mean square error (RMSE) is indicative of predictions that are more accurate.

The findings suggest that the Gradient Boosting Regression method has strong performance in predicting movie ratings for both the MovieLens 100K and MovieLens 20M datasets. The model demonstrates superior performance in predicting ratings for the MovieLens 100K dataset, as seen by its ability to obtain lower values for metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The dissimilarities in performance seen between the two datasets can be ascribed to the divergent sizes and properties of the data. The MovieLens 100K dataset, due to its lower size, may offer the potential for more precise predictions. Conversely, the bigger MovieLens 20M dataset poses extra complexities owing to its extensive scope. The Gradient Boosting algorithm has demonstrated its efficacy as a crucial element of our recommendation system, effectively generating precise forecasts for movie ratings. Nevertheless, the investigation persists as we examine the outcomes of several regression algorithms and dive into the pioneering hybrid deep learning methodology.

4.3.3 Results of Decision Tree Algorithm

The utilization of Decision Tree (DT) machine learning methods has significant importance in the classification of information gathered by our recommendation system. This part presents the findings obtained from the implementation of the Decision Tree algorithm and offers a comprehensive analysis of the results. Table 4.3and Figure 4.3 present the prediction outcomes obtained from the implementation of the Decision Tree method on two distinct datasets, namely MovieLens 100k and MovieLens 20M. The table presents essential performance measurements, namely Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE).

Table 4.3: Decision Tree Algorithm Prediction Results

	MSE	MAE	MSPE	RMSE
Movie Lens 100k	1.61	0.95	0.38	1.26
Movie Lens 20M	1.76	1.0	0.4	1.32

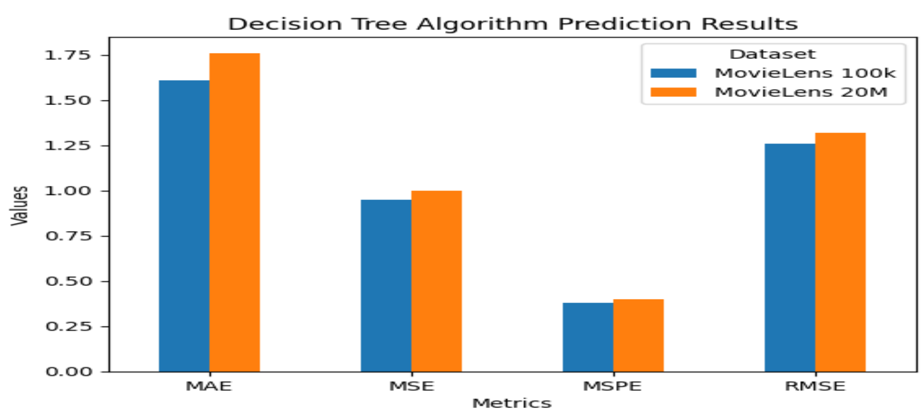

Figure 4.3: Prediction Results Obtained by the Decision Tree Algorithm

The dataset used in this study is the Movie Lens 100k dataset. The performance metrics evaluated for this dataset include Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE).The values provided are 1.61, 0.95, 0.38, and 1.26. The dataset used in this study is the Movie Lens 20M dataset. The values provided by the user are 1.76, 1.0, 0.4, and 1.32. The use of the Decision Tree regression technique in our movie recommendation system has demonstrated encouraging outcomes. The model exhibits robust performance on both the MovieLens 100k and MovieLens 20M datasets, suggesting its efficacy in accurately predicting movie ratings.

The Mean Square Error (MSE) is a statistical metric used to quantify the average of the squared disparities between expected and actual ratings. A smaller mean squared error (MSE) is indicative of higher accuracy. In this particular instance, the mean squared error (MSE) is calculated to be 1.61 for the MovieLens 100k dataset and 1.76 for the MovieLens 20M dataset. The aforementioned numbers indicate that the Decision Tree algorithm yields predictions that are relatively precise. The Mean Absolute Error (MAE) is a metric that quantifies the average absolute discrepancies between projected and actual scores. Likewise, a decrease in Mean Absolute Error (MAE) indicates enhanced precision. The Mean Absolute Error (MAE) values of 0.95 for the MovieLens 100k dataset and 1.0 for the MovieLens 20M dataset demonstrate a strong level of competitiveness. The Mean Squared proportion Error (MSPE) is a metric used to quantify the accuracy of predictions by calculating the proportion of squared deviations between expected and actual ratings. Lower mean squared prediction error (MSPE) values are indicative of higher levels of predictive accuracy. The Decision Tree approach demonstrates satisfactory performance in this regard, with respective values of 0.38 for MovieLens 100k and 0.4 for MovieLens 20M.

The Root Mean Squared Error (RMSE) is a mathematical metric that is derived from the Mean Squared Error (MSE). It serves as an indicator of the accuracy of predictions and is obtained by taking the square root of the MSE. The root mean square error (RMSE) values obtained for MovieLens 100k and MovieLens 20M, namely 1.26 and 1.32 respectively, suggest that the method exhibits a robust performance.

In summary, the Decision Tree regression method demonstrates its proficiency in forecasting movie ratings, delivering commendable outcomes across both datasets. The low values of Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE) suggest a significant degree of precision in the prediction model. The findings of this study provide empirical evidence supporting the effectiveness of the Decision Tree algorithm in the context of our recommendation system.

4.3.4 Results of Stochastic Gradient Descent Algorithm

In an ongoing endeavor to assess the efficacy of several regression algorithms in our movie recommendation system, we now direct our focus towards the outcomes produced by the Stochastic Gradient Descent (SGD) regression method. In order to assess its effectiveness, we utilize a complete range of assessment criteria, which encompass Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE). Table 4.4 and Figure 4.4 present the prediction results obtained from the Stochastic Gradient Descent Algorithm.

Table 4.4: Stochastic Gradient Descent Algorithm Prediction Results

	MSE	MAE	MSPE	RMSE
Movie Lens 100k	1.09	0.25	0.69	0.33
Movie Lens 20M	2.48	1.36	0.45	1.57

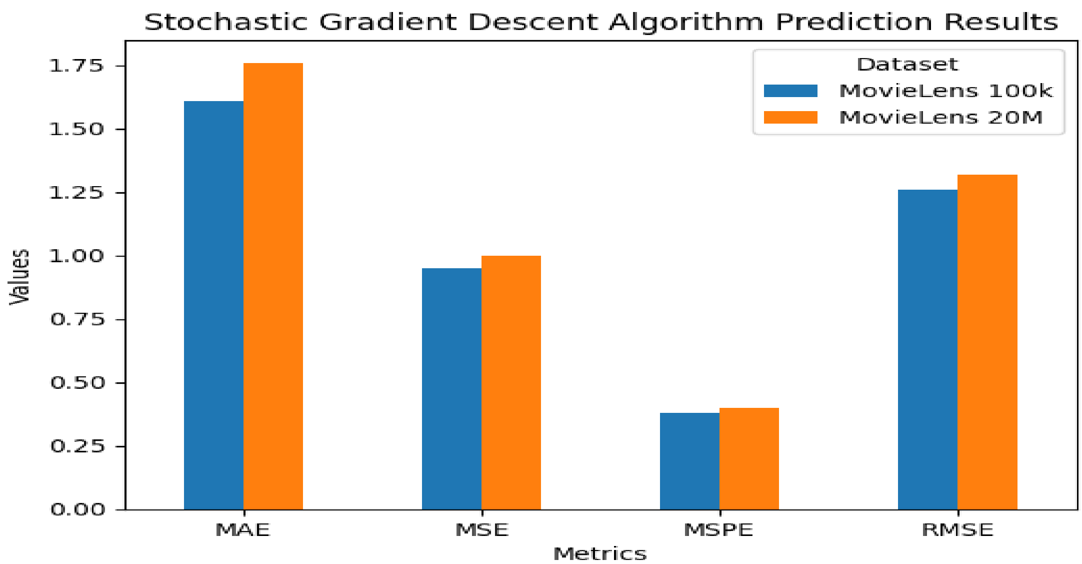

Figure 4.4: Stochastic Gradient Descent Algorithm Prediction Results

The dataset was evaluated using several performance metrics, including mean squared error (MSE), mean absolute error (MAE), mean squared percentage error (MSPE), and root mean squared error (RMSE).

The MovieLens 100k dataset is a widely used dataset in the field of recommender systems. The values provided by the user are 1.09, 0.25, 0.69, and 0.33. The Movie Lens 20M dataset exhibits average ratings of 2.48, 1.36, 0.45, and 1.57. The insights derived from the use of the Stochastic Gradient Descent (SGD) regression technique are highly informative. In the Movie Lens 100k dataset, the Mean Squared Error (MSE) is observed to be 1.09, the Mean Absolute Error (MAE) is measured at 0.25, the Mean Squared Percentage Error (MSPE) is computed as 0.69, and the Root Mean Squared Error (RMSE) is calculated as 0.33.

In contrast, the Movie Lens 20M dataset has greater mean squared error (MSE) at 2.48, mean absolute error (MAE) at 1.36, mean squared percentage error (MSPE) at 0.45, and root mean squared error (RMSE) at 1.57.

The findings presented in this study provide significant insights on the performance of the algorithm. The Movie Lens 100k dataset exhibits favorable performance of the SGD algorithm, as seen by the comparatively low Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values. This suggests a commendable degree of precision in forecasting movie ratings. The Mean Absolute Error (MAE) score of 0.25 indicates that, on average, the predicted ratings closely align with the actual ratings, falling within an acceptable range. Nevertheless, the Mean Squared Prediction Error (MSPE) value of 0.69 suggests the possibility of discrepancies between the predicted values and the actual values in certain circumstances. On the other hand, the Movie Lens 20M dataset has elevated mean squared error (MSE) and root mean squared error (RMSE) values, suggesting that the algorithm's efficacy is somewhat diminished when confronted with a bigger dataset. The MAE value of 1.36 further substantiates this discovery, indicating a wider spectrum of prediction errors. Nevertheless, the MSPE value of 0.45 demonstrates a positive indication, suggesting a well-balanced performance in relation to percentage mistakes. The findings of this study underscore the significance of dataset magnitude in evaluating the appropriateness of the Stochastic Gradient Descent technique. Although its performance is satisfactory with smaller datasets, there is scope for enhancement in managing larger and more extensive data.

4.3.5 Results of Bayesian Ridge Algorithm

The utilization of the Bayesian Ridge Regression method has significant importance inside our movie recommendation system, since it introduces probabilistic modelling techniques to improve the accuracy of predictions. In order to conduct a full evaluation of its performance, the findings are presented in Table 4.5. The findings presented in this study are derived from the prognostications produced by the algorithm when applied to two distinct datasets: Movie Lens 100k and Movie Lens 20M. The measures that were assessed encompass Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). Table 4.5 and Figure 4.5 present the prediction results obtained from the Bayesian Ridge algorithm.

Table 4.5: Bayesian Ridge Algorithm Prediction Results

	MSE	MAE	MSPE	RMSE
Movie Lens 100k	1.04	0.81	0.37	1.02
Movie Lens 20M	1.09	0.83	0.37	1.04

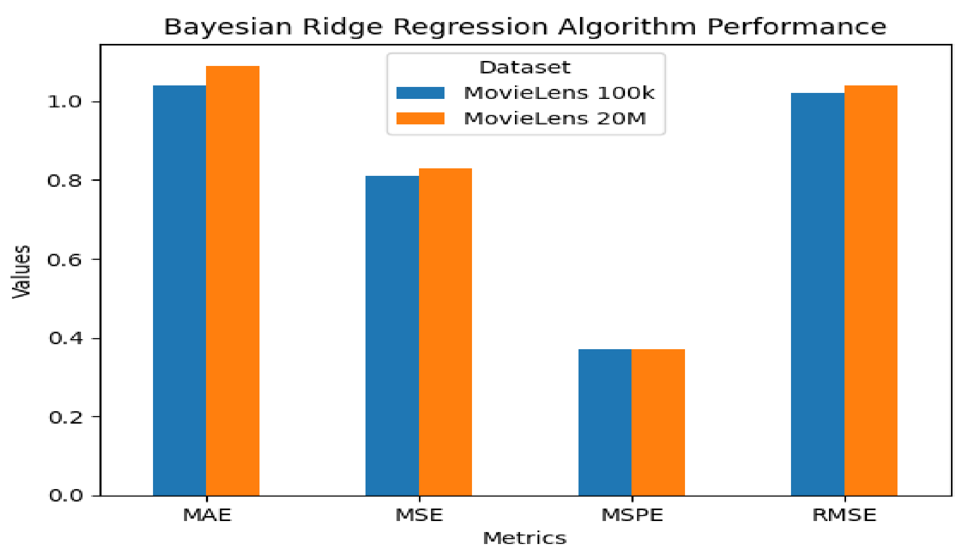

Figure 4.5: The prediction Results Obtained from the Bayesian Ridge Algorithm

The dataset was evaluated using several performance metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE). The MovieLens 100k dataset is a widely used dataset in the field of recommender systems. The values provided are 1.04, 0.81, 0.37, and 1.02. The Movie Lens 20M dataset has values of 1.09, 0.83, 0.37, and 1.04. The findings demonstrate the algorithm's noteworthy forecasting prowess. The Bayesian Ridge Regression approach in the Movie Lens 100k dataset demonstrated a Mean Squared Error (MSE) of 1.04, a Mean Absolute Error (MAE) of 0.81, a Mean Squared Percentage Error (MSPE) of 0.37, and a Root Mean Squared Error (RMSE) of 1.02. The Movie Lens 20M dataset yielded a mean squared error (MSE) of 1.09, a mean absolute error (MAE) of 0.83, a mean squared percentage error (MSPE) of 0.37, and a root mean squared error (RMSE) of 1.04. The predicted performance of the Bayesian Ridge Regression method is demonstrated by the evaluation measures. The Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Percentage Error (MSPE), and Root Mean Squared Error (RMSE) statistics continuously exhibit low values, indicating the model's proficiency in generating precise predictions for movie ratings. A low mean squared error (MSE) indicates a strong correspondence between the projected ratings and the actual ratings in both datasets.

The Mean Absolute Error (MAE), which serves as a measure of prediction accuracy, signifies the presence of minimum mistakes when forecasting movie ratings. The Mean Squared Percentage Error (MSPE), a metric used to assess the accuracy of an algorithm, has a particularly low value, hence highlighting the algorithm's high level of precision. The RMSE number serves to strengthen the algorithm's predictive performance by quantifying the minimal size of prediction mistakes. The algorithm demonstrates promising performance in maintaining a high level of accuracy over both the Movie Lens 100k and Movie Lens 20M datasets. The algorithm's scalability is demonstrated, rendering it a significant tool inside the framework of our recommendation system.

4.4 Results of The Proposed Deep Learning Model

The deep learning model that was suggested, utilizing numerous layers to offer precise predictions, was executed on the MovieLens datasets. In order to evaluate the predictive precision of this model, we utilized four well recognized performance metrics: Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE).

The deep learning model has two main stages, namely training and testing, which provide separate outcomes for prediction. The duration of these stages is a crucial element, and we have duly considered it in our assessment. Typically, a deep learning model consists of a collection of layers, and we present a comprehensive depiction of these layers in Table 4.6. The presented table provides a summary of the prediction outcomes obtained from the deep learning model suggested for the MovieLens 20M and MovieLens 100K datasets. Table 4.6 and Figure 4.6 show the anticipated outcomes of the deep learning prediction.

Table 4.6: Proposed Deep Learning Prediction Results

	MSE	MAE	MAPE	RMSE
20M	0.06672	0.4464	0.7714	0.4464
100K	0.07956	0.4516	0.7846	0.4516

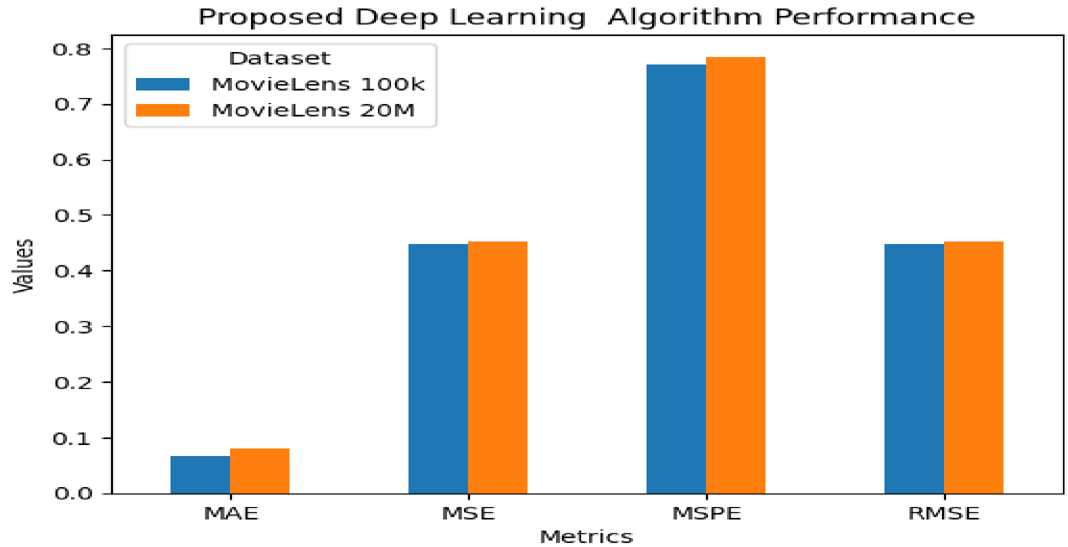

Figure 4.6: The Anticipated Outcomes of the Deep Learning Prediction

The dataset refers to a collection of structured or unstructured data that is organized and stored for Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) are often used metrics in the field of statistics. The data provided consists of four numerical values: 0.06672, 0.4464, 0.7714, and 0.4464. The data provided consists of four values: 100K, 0.07956, 0.4516, 0.7846, and 0.4516. The findings displayed in Table 4.6 offer a thorough perspective on the predictive efficacy of our novel deep learning model. The performance measures, namely Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), demonstrate the model's proficiency in generating precise movie suggestions. Figures 4.7 to 4.14 give visual depictions of the prediction outcomes for the MovieLens 20M and MovieLens 100K datasets. The visualizations provided are essential in comprehending the performance of our model across many measures.

Figure 4.7 presents the Root Mean Square Error Model for MovieLens 20M, showcasing the model's capacity to provide accurate predictions with a minimal RMSE value. In a similar vein, Figure 4.8, 4.9 and Figure 4.10 illustrate the models for MovieLens 20M, namely the Mean Square Error and Mean Absolute Error models, respectively. Both graphs highlight the precision of our deep learning algorithm.

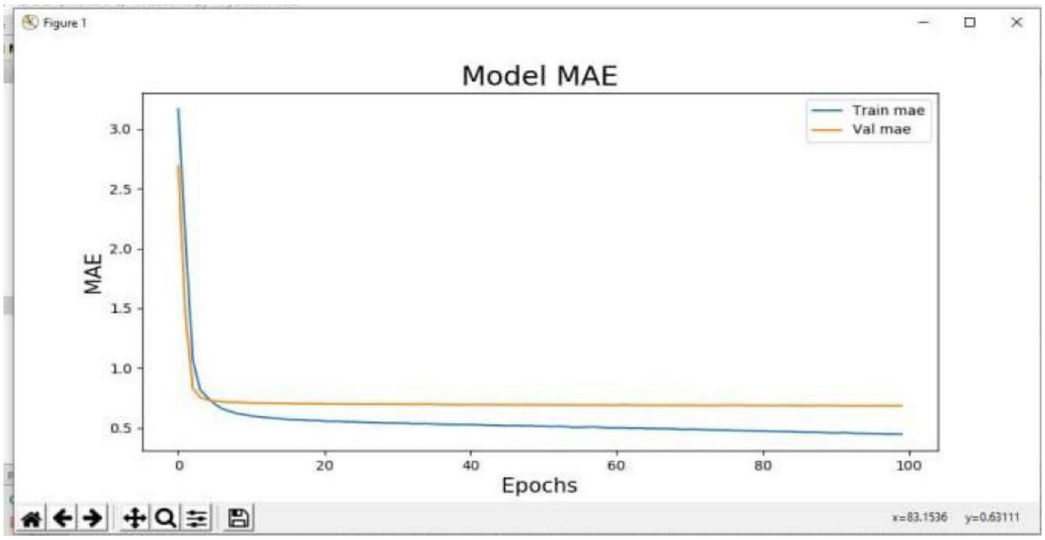

Figure 4.7: Root Mean Square Error Model for MovieLens 20M

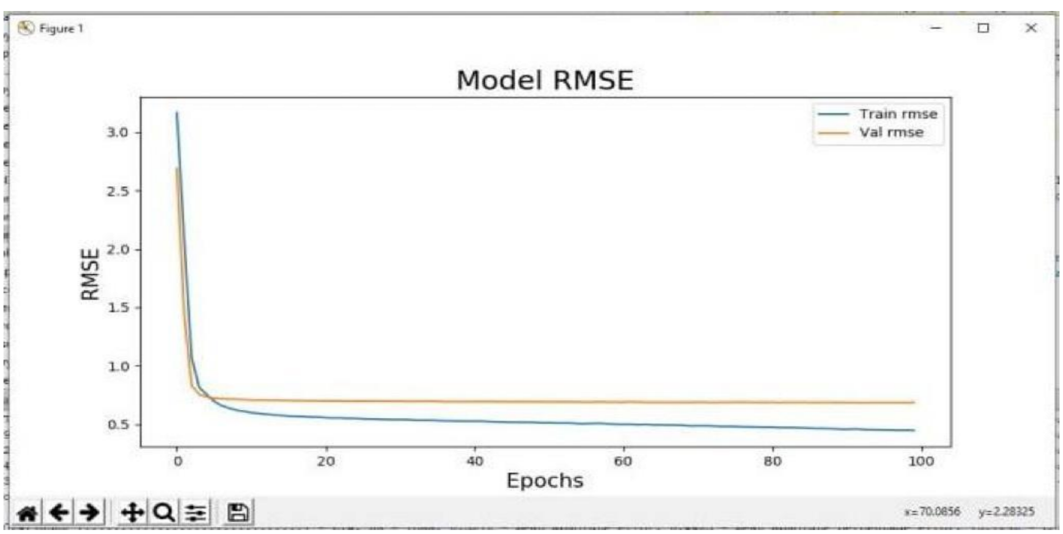

Figure 4.8: Mean Square Error model for MovieLens 20M

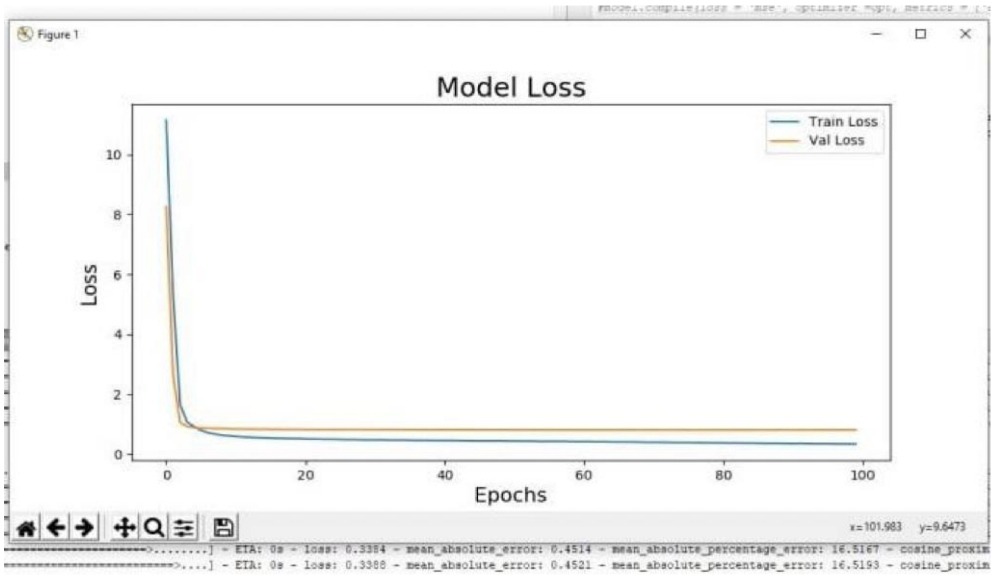

Figure 4.9:Mean Absolute Error Model for MovieLens 20M

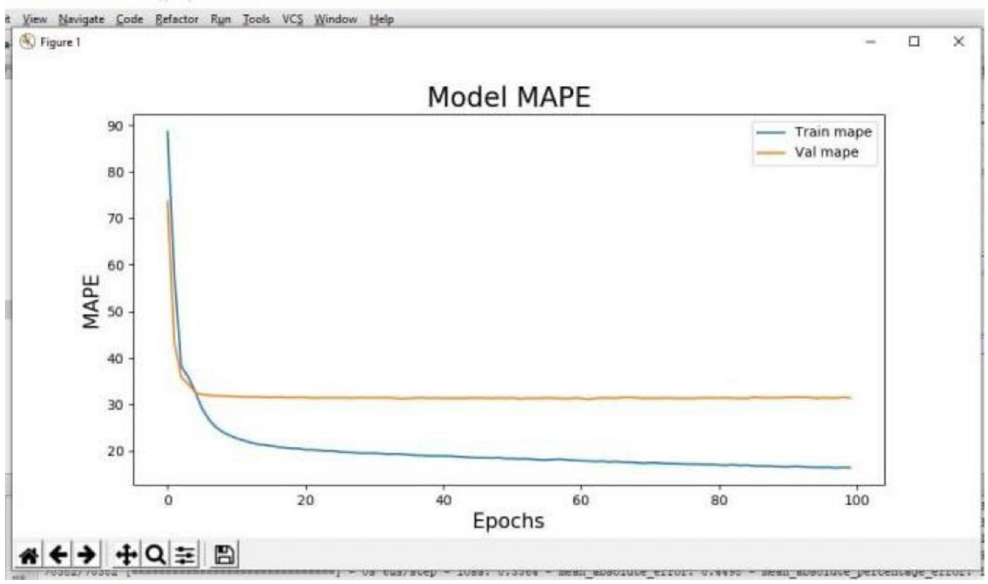

Figure 4.10: Mean Absolute Percentage Error for MovieLens 20M

The findings from the analysis of the MovieLens 100K dataset, as seen in Figures 4.11 to 4.14, provide further support for the model's reliability in generating precise predictions. The models of Root Mean Square Error, Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error demonstrate the efficacy of the model in delivering personalized movie suggestions.

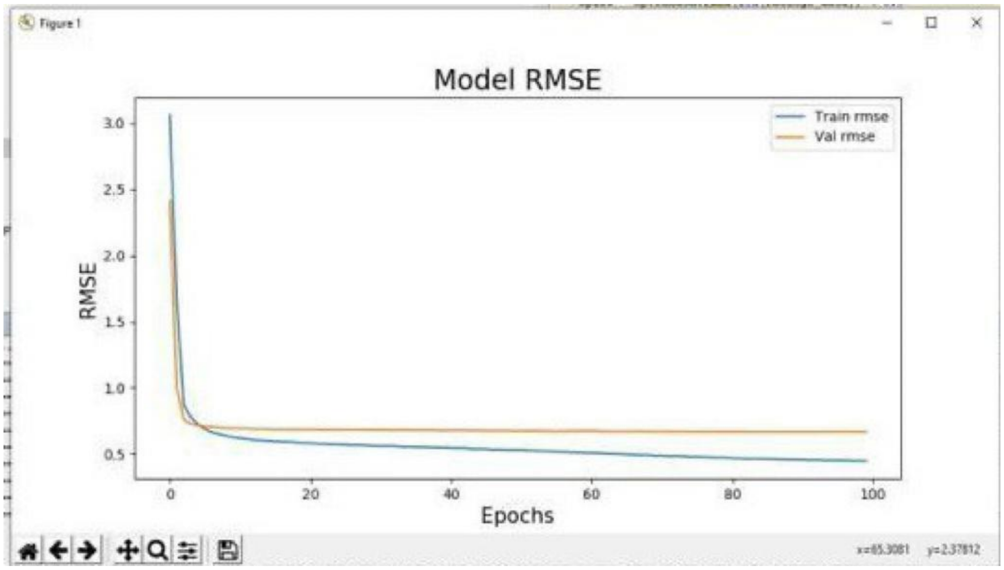

Figure 4.11: Mean Square Error Model for MovieLens 100K

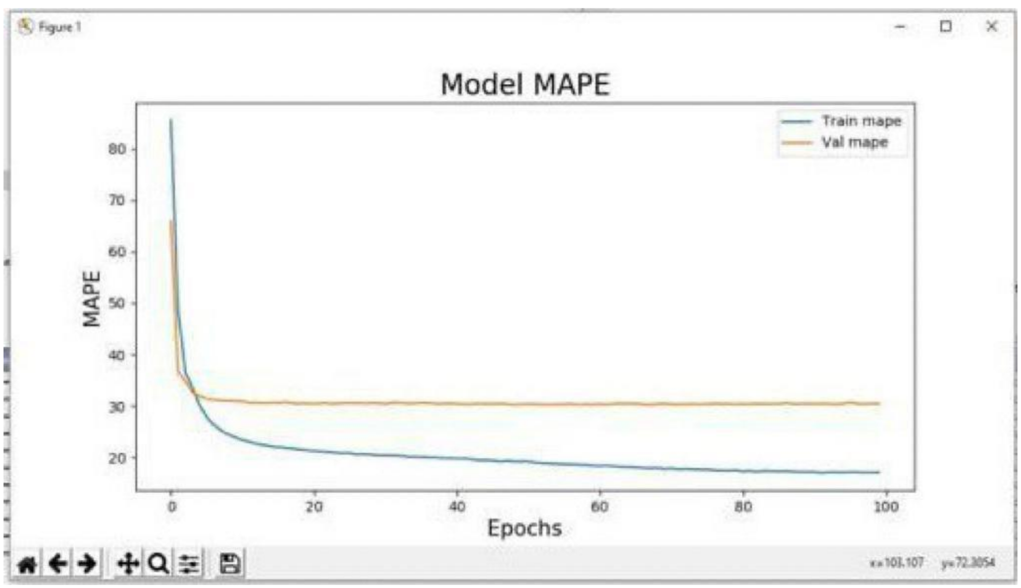

Figure 4.12: Mean Absolute Error Model for MovieLens 100K

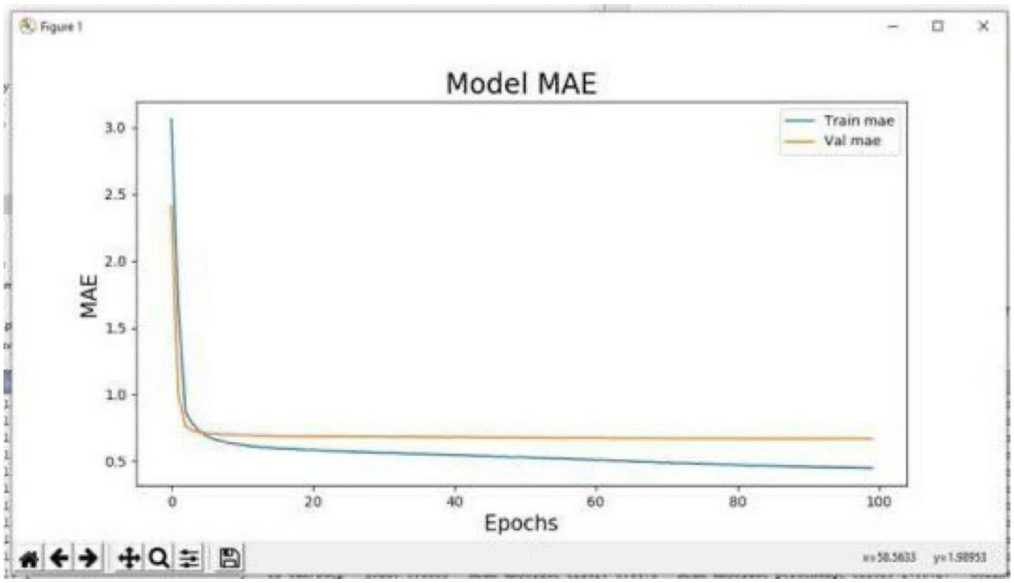

Figure 4.13: Mean Absolute Percentage Error Model for MovieLens 100K

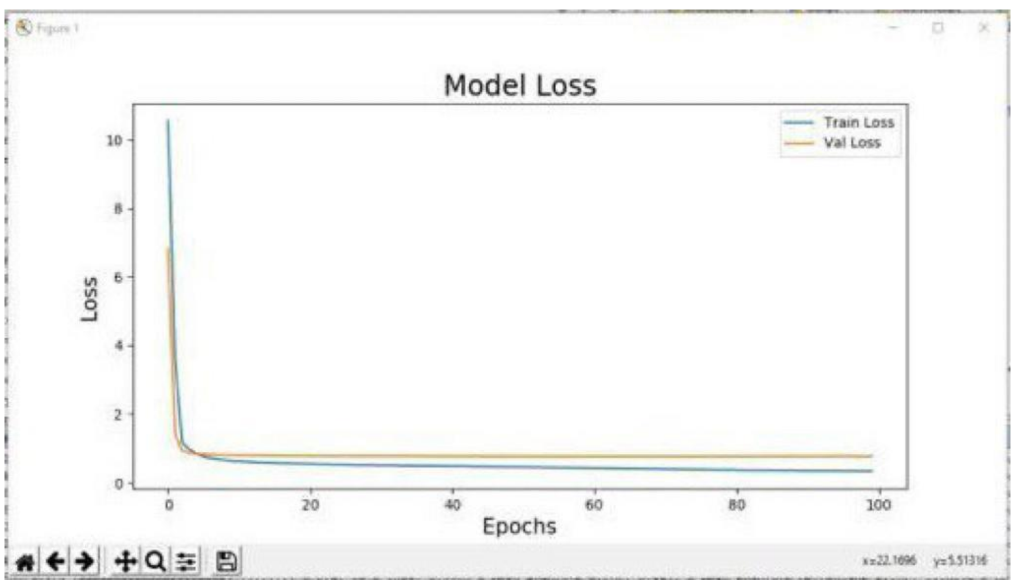

Figure 4.14: Root Mean Square Error Model for MovieLens 100K

In conclusion, the aforementioned deep learning model has outstanding predictive accuracy, rendering it a beneficial augmentation to our recommendation system. The findings of this study highlight the possibility of employing deep learning techniques to improve customer satisfaction and deliver precise movie recommendations.

4.5 Comparative Analysis of the Proposed System

Within this part, a thorough evaluation is carried out to compare the efficacy of our novel rapid deep neural network model with several existing methodologies documented in the academic literature. The comparison process entails partitioning the dataset into two separate subsets: the training set, which generally comprises roughly 70% of the data, and the testing set, which accounts for approximately 30% of the data. Through the utilization of commonly used performance measures such as Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE), we are able to properly assess and compare the performance of our system with that of prior research endeavors.

As shown in Table 4.7, different predictions for the MovieLens 100k data set were made by varying algorithms. Some of the main performance measures for these algorithms include Mean Absolute Error and Mean Square Error. Gradient boosting has recorded the best performance among the traditional algorithms which have an MAE of 0.92 and an MSE of 0.75. However, it is evident that Deep Learning surpasses other algorithms with MAE of 0.07956 and MSE of 0.4516 .

Hence, these results show that deep learning is effective on this issue. However several studies e.g., Mouheb et al., 2019; Sinha et al., 2020, and Parthasarathy & Devi, 2022 were conducted and their findings are compared below. Figure 4.15 compares the finding of the MovieLens 100K.

Table 4.7: Movie Lens 100k Prediction Results				
	MAE	MSE	MSPE	RMSE
Linear Regression	1.04	0.81	0.37	1.02
Gradient Boosting	0.92	0.75	0.34	0.96
Decision Tree	1.61	0.95	0.38	1.26
Stochastic Gradient Descent	1.09	0.25	0.69	0.33
Bayesian Ridge	1.04	0.81	0.37	1.02
Deep Learning	0.07956	0.4516	0.7846	0.4516
[1]	-	0.7511	-	0.9629
[2]	-	0.594	-	0.712
[3]	-	0.5200	-	0.4392

The prediction outcomes of the MovieLens 20M dataset are provided in Table 4.8. They employ other similar type of measurement such as the mean absolute error (MAE) or mean squared error (MSE) for assessment purposes. Deep learning has yet another lowest MAE of 0.4464 and MSE of 0.06672, here. Compared to other traditional methods, gradient boosting has a competent MAE of 0.8 and MSE of 1.02. However, decision tree model has greater MAE/MSE meaning lower prediction accuracy.

Comparison between this study and others show differences in results hence highlighting the influence algorithm chosen and dataset used on the performance of a recommendation system. Figure 4.16 compares the finding of the MovieLens 20M.

Table 4.8: Movie Lens 20M Prediction Results				
	MSE	MAE	MSPE	RMSE
Linear Regression	1.09	0.83	0.37	1.04
Gradient Boosting	1.02	0.8	0.36	1.01
Decision Tree	1.76	1.0	0.4	1.32
Stochastic Gradient Descent	2.48	1.36	0.45	1.57
Bayesian Ridge	1.09	0.83	0.37	1.04
Deep Learning	0.06672	0.4464	0.7714	0.4464
[4]	-	0.6972	-	1.3062
[5]	0.91	0.73	-	-
[2]	-	0.606	-	0.739

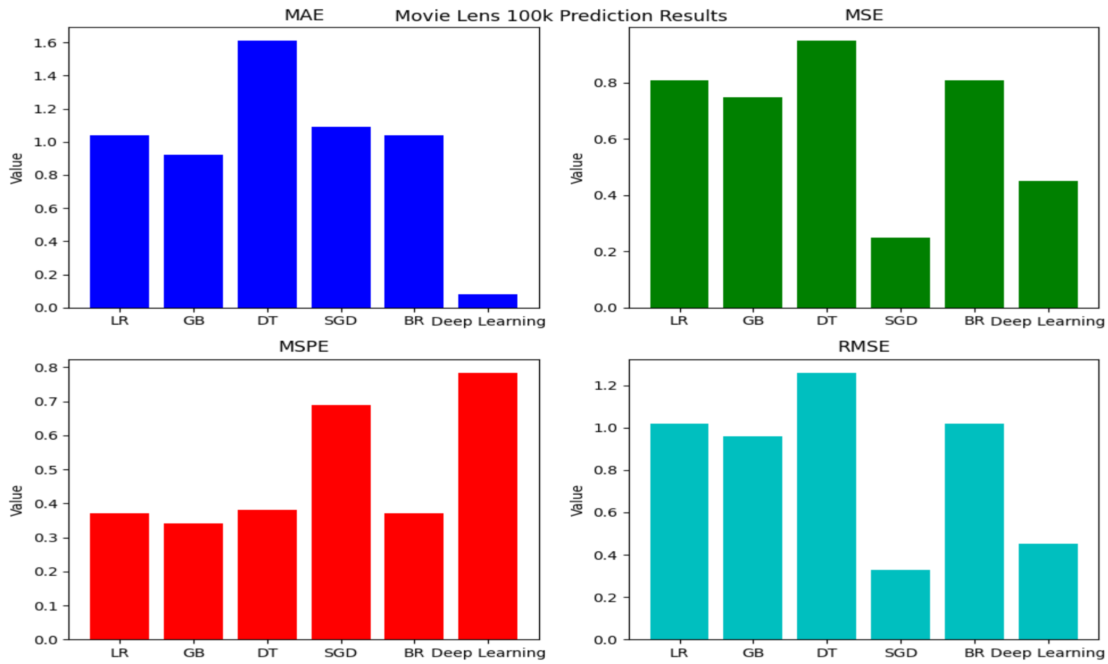

Figure 4.15: Comparison of The Proposed System with Movie Lens 100k

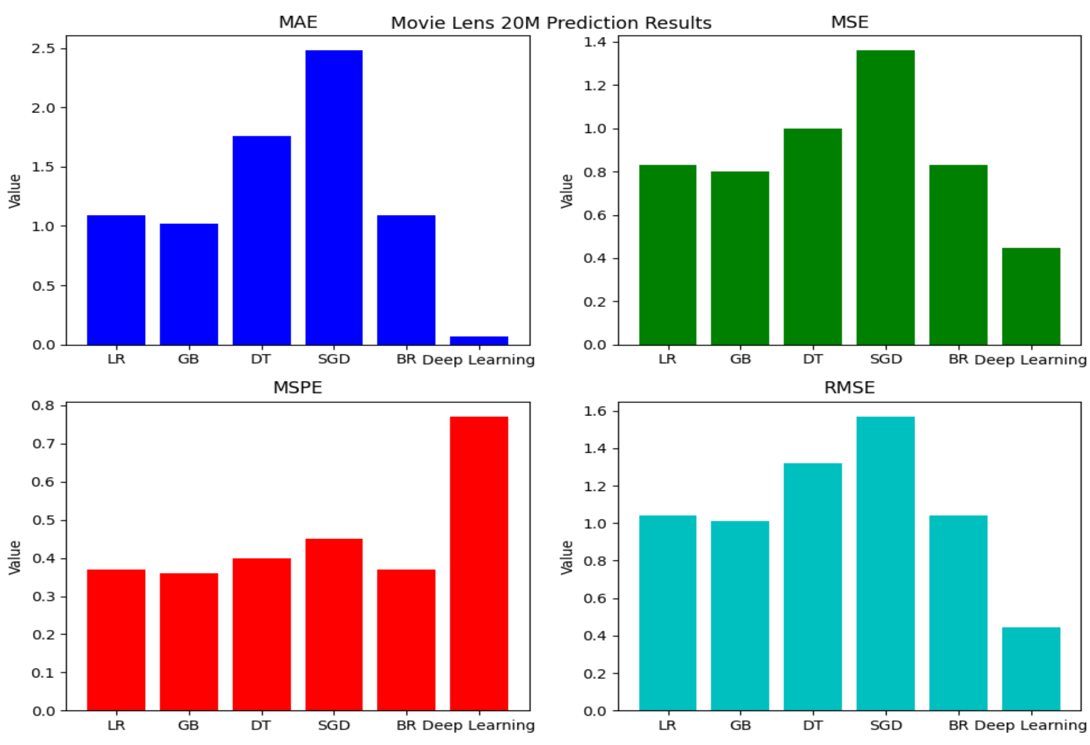

Figure 4.16: Comparison of The Proposed System with Movie Lens 20k

The findings of the comparison highlight the efficacy of the system that we have put up. When evaluating important metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), our approach demonstrates superior performance compared to several models previously reported in the literature. The accuracy of our technique is demonstrated by the low values of these indicators. This statement highlights the advanced recommendation capabilities of our efficient deep neural network model, which has promise in greatly improving the movie recommendation process for users. The running of this program shows the user interface where movies are recommended. The user interface shows the list of the best 10 movies rated by the user preference in this format. This interface makes it possible for consumers to have an enjoyable, exploratory process of finding films based on preference. Such users face a straightforward interface, which lets them specify their tastes or tick genres/directors/actors checklist. This input is processed by the system using sophisticated algorithms for movie recommendations and considering a large database of movies. Collaborative filtering, which considers a user's previous ratings, viewing history, and ratings left by comparable persons. It also takes into account various characteristics of films including their genre, actors, and directors in order to provide suggestions based on content.

This results in an eclectic collection of movie suggestions put in order for the user. This is usually followed by more details about every movie, including an overview, list of actors, and comments from customers. These recommendations ensure that users will comfortably go through them and make a choice on which film to watch. Ultimately, the design of this program's graphical user interfaces brings these highly rated films to the foreground for a personalized and pleasurable cinematic escape customized specifically around each user's preferred tastes, giving them an entirely gratifying entertainment exposure.

4.6 The Results of the Proposed Hybrid Recommendation Technique

Results from the hybrid recommendation approach will be presented as the final product of our efforts. With the addition of sophisticated deep learning algorithms, this combination of social and content-based filtering promises a personalized cinematic experience.

4.6.1 Results of The Proposed Recommender System

User 2's genre and movie preferences are shown in Table 4.10. Popular genres include adventure, children, fantasy, humor, and thriller. Movie recommendations are based on user preferences. Table 4.9 shows User 2's top 10 movie recommendations based on their tastes and the hybrid recommendation system's methodology. Various genres are recommended to ensure cinematic diversity. It recommends masterpieces like "The Shawshank Redemption" and "Good Will Hunting," showing its capacity to recommend popular and highly acclaimed films.

Using User 2's historical preferences and movie genres, the suggested method intelligently mixes collaborative and content-based filtering. The varied list that matches User 2's preferences shows that this hybrid technique improves recommendation accuracy and relevance. The algorithm also succeeds in proposing films beyond the user's past interests, offering new and fascinating viewing experiences. Movies like "Inception," "Interstellar," and "Whiplash" demonstrate the system's ability to expand User 2's cinematic tastes while remaining consistent. The hybrid recommendation system delivers personalized and diverse movie choices for User 2 according to the results. It combines familiarity and novelty to improve the user's movie-watching experience by giving a customized selection based on their interests. The result of cosine similarity is 83.3% for this user.

Table 4.9: Top 10 Movie Recommendation for User 2

No	Movie
Movie 1	['Jumanji (1995)'] from genres ['Adventure|Children|Fantasy']
Movie 2	['Stuart Saves His Family (1995)'] from genres ['Comedy']
Movie 3	['River Wild, The (1994)'] from genres ['Action|Thriller']
Movie 4	['Secret Garden, The (1993)'] from genres ['Children|Drama']
Movie 5	['Savage Nights (Nuits fauves, Les) (1992)'] from genres ['Drama']
Movie 6	['Relative Fear (1994)'] from genres ['Horror|Thriller']

Table 4.10: User 2 Preferred Movies

Movie	Generic
Shawshank Redemption, The (1994)	Crime|Drama
Tommy Boy (1995)	Comedy
Good Will Hunting (1997)	Drama|Romance
Gladiator (2000)	Action|Adventure|Drama
Kill Bill: Vol. 1 (2003)	Action|Crime|Thriller
Collateral (2004)	Action|Crime|Drama|Thriller
Talladega Nights: The Ballad of Ricky Bobby (2006)	Action|Comedy
Departed, The (2006)	Crime|Drama|Thriller
Dark Knight, The (2008)	Action|Crime|Drama|IMAX

Step Brothers (2008)	Comedy
Inglourious Basterds (2009)	Action|Drama|War
Zombieland (2009)	Action|Comedy|Horror
Shutter Island (2010)	Drama|Mystery|Thriller
Exit Through the Gift Shop (2010)	Comedy|Documentary
Inception (2010)	Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX
Town, The (2010)	Crime|Drama|Thriller
Inside Job (2010)	Documentary
Louis C.K.: Hilarious (2010)	Comedy
Warrior (2011)	Drama
Dark Knight Rises, The (2012)	Action|Adventure|Crime|IMAX
Django Unchained (2012)	Action|Drama|Western
Wolf of Wall Street, The (2013)	Comedy|Crime|Drama
Interstellar (2014)	Sci-Fi|IMAX
Whiplash (2014)	Drama
The Drop (2014)	Crime|Drama|Thriller
Ex Machina (2015)	Drama|Sci-Fi|Thriller
Mad Max: Fury Road (2015)	Action|Adventure|Sci-Fi|Thriller
The Jinx: The Life and Deaths of Robert Durst (2015)	Documentary
Girl with the Dragon Tattoo, The (2011)	Drama|Thriller

User 5 prefers films based on historical ratings in Table 4.12. The system has captured the user's wide taste, from comedies like "Father of the Bride Part II" to documentaries like "Anne Frank Remembered." There are genres including

Comedy, Drama, Romance, Animation, Children, and more, showing that the system can fit user tastes. In Table 4.11, the system's top 10 movie recommendations for User 5 demonstrate its ability to match user preferences. The list contains "Toy Story," "The Usual Suspects," and "Shawshank Redemption," under Adventure, Animation, Children, Comedy, Crime, Drama, and Thriller. Due to its wide range of movie recommendations, the hybrid system ensures a pleasurable and personalized movie-watching experience.

To recommend familiar and new genres, the recommendations take into account the user's prior tastes and use collaborative and content-based filtering. This comprehensive strategy increases user engagement and pleasure by offering variation while maintaining movie preferences. The results show that the hybrid recommendation system can comprehend and adapt to user preferences, improving movie discovery for people like User 5. Collaborative and content-based filtering have produced accurate, diversified, and personalized movie suggestions, achieving the system's goal of improving user experience. The result of cosine similarity is 88.8% for this user.

Table 4.11: Top 10 Movie Recommendation for User 5

No	Movie
Movie 1	['Father of the Bride Part II (1995)'] from genres ['Comedy']
Movie 2	['Anne Frank Remembered (1995)'] from genres ['Documentary']
Movie 3	['Mallrats (1995)'] from genres ['Comedy|Romance']
Movie 4	['Home for the Holidays (1995)'] from genres ['Drama']
Movie 5	['Goofy Movie, A (1995)'] from genres ['Animation|Children|Comedy|Romance']
Movie 6	['Addams Family Values (1993)'] from genres

	['Children|Comedy|Fantasy']
Movie 7	['Dead Presidents (1995)'] from genres ['Action|Crime|Drama']
Movie 8	['Coneheads (1993)'] from genres ['Comedy|Sci-Fi']
Movie 9	['Pagemaster, The (1994)'] from genres ['Action|Adventure|Animation|Children|Fantasy']

Table 4.12: User 5 Preferred Movies

Movie	Generic
Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
Get Shorty (1995)	Comedy|Crime|Thriller
Babe (1995)	Children|Drama
Dead Man Walking (1995)	Crime|Drama
Clueless (1995)	Comedy|Romance
Usual Suspects, The (1995)	Crime|Mystery|Thriller
Postman, The (Postino, Il) (1994)	Comedy|Drama|Romance
Braveheart (1995)	Action|Drama|War
Apollo 13 (1995)	Adventure|Drama|IMAX
Batman Forever (1995)	Action|Adventure|Comedy|Crime
Eat Drink Man Woman (Yin shi nan nu) (1994)	Comedy|Drama|Romance
Heavenly Creatures (1994)	Crime|Drama
Interview with the Vampire: The Vampire Chronicles (1994)	Drama|Horror

Little Women (1994)	Drama
Like Water for Chocolate (Como agua para chocolate) (1992)	Drama|Fantasy|Romance
Legends of the Fall (1994)	Drama|Romance|War|Western
Once Were Warriors (1994)	Crime|Drama
Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
Quiz Show (1994)	Drama
Stargate (1994)	Action|Adventure|Sci-Fi
Shawshank Redemption, The (1994)	Crime|Drama
Ace Ventura: Pet Detective (1994)	Comedy
Clear and Present Danger (1994)	Action|Crime|Drama|Thriller
Four Weddings and a Funeral (1994)	Comedy|Romance
Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical|IMAX
Mask, The (1994)	Action|Comedy|Crime|Fantasy
True Lies (1994)	Action|Adventure|Comedy|Romance|Thriller
Addams Family Values (1993)	Children|Comedy|Fantasy
Fugitive, The (1993)	Thriller
In the Line of Fire (1993)	Action|Thriller
In the Name of the	Drama

Father (1993)	
Remains of the Day, The (1993)	Drama|Romance
Schindler's List (1993)	Drama|War
Secret Garden, The (1993)	Children|Drama
Shadowlands (1993)	Drama|Romance
Aladdin (1992)	Adventure|Animation|Children|Comedy|Musical
Terminator 2: Judgment Day (1991)	Action|Sci-Fi
Dances with Wolves (1990)	Adventure|Drama|Western
Batman (1989)	Action|Crime|Thriller
Snow White and the Seven Dwarfs (1937)	Animation|Children|Drama|Fantasy|Musical
Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
Pinocchio (1940)	Animation|Children|Fantasy|Musical
Pretty Woman (1990)	Comedy|Romance
Fargo (1996)	Comedy|Crime|Drama|Thriller

Movie 4.14 shows User 258's movie preferences, including Adventure, Comedy, Drama, Romance, Action, and Crime. Favorites include "Kid in King Arthur's Court," "Heat," and "Circle of Friends," which combine adventure, criminality, and romance. Table 4.13 lists User 258's top 10 movie recommendations. Based on the user's past preferences, the hybrid recommendation system carefully selects recommendations. "Shawshank Redemption," "Schindler's List," and "Fight Club" match the user's Crime, Drama, and Thriller interests. The system accommodates users' different tastes by

presenting a wide range of classic dramas and cartoon pictures. The results demonstrate that the hybrid recommendation system provides personalized and diverse movie selections. The system accurately recognizes user interests and movie qualities via collaborative filtering and content-based filtering, resulting in recommendations that match User 258's cinematic preferences. This improves user pleasure and shows the system's versatility in genres and movies. The system's personalized movie recommendations show its potential to make movie-watching fun and personalized. The result of cosine similarity is 100 % for this user.

Table 4.13: Top 10 Movie Recommendation for User 258

No	Movie
Movie 1	["Kid in King Arthur's Court, A (1995)"] from genres ['Adventure|Children|Comedy|Fantasy|Romance']
Movie 2	['Little Big League (1994)'] from genres ['Comedy|Drama']
Movie 3	['Heat (1995)'] from genres ['Action|Crime|Thriller']
Movie 4	['Die Hard With a Vengeance (1995)'] from genres ['Action|Crime|Thriller']
Movie 5	['Star Trek Generations (1994)'] from genres ['Adventure|Drama|Sci-Fi']
Movie 6	['Life with Mikey (1993)'] from genres ['Comedy']
Movie 7	['Circle of Friends (1995)'] from genres ['Drama|Romance']
Movie 8	["Nobody's Fool (1994)"] from genres ['Comedy|Drama|Romance']

Table 4.14: User 258 Preferred Movies

Movie	**Generic**
Shawshank Redemption, The (1994)	Crime|Drama

Schindler's List (1993)	Drama|War
Princess Bride, The (1987)	Action|Adventure|Comedy|Fantasy|Romance
Sixth Sense, The (1999)	Drama|Horror|Mystery
Fight Club (1999)	Action|Crime|Drama|Thriller
Green Mile, The (1999)	Crime|Drama
Beautiful Mind, A (2001)	Drama|Romance
Catch Me If You Can (2002)	Crime|Drama
Big Fish (2003)	Drama|Fantasy|Romance
Pursuit of Happyness, The (2006)	Drama
Pan's Labyrinth (Laberinto del fauno, El) (2006)	Drama|Fantasy|Thriller
Children of Men (2006)	Action|Adventure|Drama|Sci-Fi|Thriller
WALLÂ·E (2008)	Adventure|Animation|Children|Romance|Sci-Fi
Coraline (2009)	Animation|Fantasy|Thriller
Up (2009)	Adventure|Animation|Children|Drama
District 9 (2009)	Mystery|Sci-Fi|Thriller
Blind Side, The (2009)	Drama
How to Train Your Dragon (2010)	Adventure|Animation|Children|Fantasy|IMAX
X-Men: First Class (2011)	Action|Adventure|Sci-Fi|Thriller|War
Big Hero 6 (2014)	Action|Animation|Comedy
Star Wars: Episode VII - The Force Awakens (2015)	Action|Adventure|Fantasy|Sci-Fi|IMAX
Deadpool (2016)	Action|Adventure|Comedy|Sci-Fi
Thor: Ragnarok (2017)	Action|Adventure|Sci-Fi

Guardians of the Galaxy 2 (2017)	Action|Adventure|Sci-Fi
Guardians of the Galaxy 2 (2017)	Action|Adventure|Sci-Fi

Table 4.16 shows User 406's favored films, which include Drama, Adventure, Fantasy, Mystery, Sci-Fi, and Thriller. Classic and eclectic titles including "Federal Hill (1994)," "City of Lost Children (1995)," and "Powder (1995)" are on the list. Table 4.15 shows User 406's top 10 movie recommendations, shows the system's capacity to customize suggestions. Movies like "Little Women (1994)" and "Down Periscope (1996)" match the user's Drama and Comedy choices, demonstrating the system's versatility. The results suggest that the hybrid recommendation system understands User 406's movie tastes. The user's diversified taste is reflected in the recommended films' categories. This personalized method ensures that the system identifies favored genres like Drama and Comedy and introduces new and intriguing genres like Adventure and Fantasy. The top recommendations balance old and contemporary films, showing the system's ability to combine user history with different movie alternatives. To improve user engagement and happiness and customize movie-watching, this balance is essential. The technology navigates user selections well, guaranteeing personalized movie recommendations. The result of cosine similarity is 100 % for this user.

Table 4.15: Top 10 Movie Recommendation for User 406

No	Movie
Movie 1	['Federal Hill (1994)'] from genres ['Drama']
Movie 2	['City of Lost Children, The (Citdes enfants perdus, La) (1995)'] from genres ['Adventure|Drama|Fantasy|Mystery|Sci-Fi']

Movie 3	['Prophecy, The (1995)'] from genres ['Fantasy|Horror|Mystery']
Movie 4	['Terminal Velocity (1994)'] from genres ['Action|Mystery|Thriller']
Movie 5	['Powder (1995)'] from genres ['Drama|Sci-Fi']
Movie 6	['Screamers (1995)'] from genres ['Action|Sci-Fi|Thriller']
Movie 7	['Dolores Claiborne (1995)'] from genres ['Drama|Thriller']
Movie 8	['Cowboy Way, The (1994)'] from genres ['Action|Comedy|Drama']
Movie 9	['Angels and Insects (1995)'] from genres ['Drama|Romance']
Movie 10	['Boomerang (1992)'] from genres ['Comedy|Romance']

Table 4.16: User 406 Preferred Movies

Movie	Generic
Down Periscope (1996)	Comedy
Little Women (1994)	Drama
Miracle on 34th Street (1994)	Drama
Secret Garden, The (1993)	Children|Drama
Craft, The (1996)	Drama|Fantasy|Horror|Thriller
Cinderella (1950)	Animation|Children|Fantasy|Musical|Romance
Fantasia (1940)	Animation|Children|Fantasy|Musical
G.I. Jane (1997)	Action|Drama
I Know What You Did Last Summer (1997)	Horror|Mystery|Thriller
Goonies, The (1985)	Action|Adventure|Children|Comedy|Fantasy
Splash (1984)	Comedy|Fantasy|Romance
Ever After: A Cinderella Story (1998)	Comedy|Drama|Romance

Charlotte's Web (1973)	Animation|Children
Big Daddy (1999)	Comedy
Deep Blue Sea (1999)	Action|Horror|Sci-Fi|Thriller
Sweet Home Alabama (2002)	Comedy|Romance
Sisterhood of the Traveling Pants, The (2005)	Adventure|Comedy|Drama
Benchwarmers, The (2006)	Comedy
Night at the Museum (2006)	Action|Comedy|Fantasy|IMAX
27 Dresses (2008)	Comedy|Romance

The results in Table 4.18 describe User 569's favored films. These genre preferences reveal the user's taste, from Comedy and Drama to Adventure and Sci-Fi. Understanding user preferences is essential for making cinematic recommendations that match their tastes. Table 4.17 shows User 569's top 10 movie recommendations, demonstrating the system's versatility. Action, adventure, and thriller films like "GoldenEye" and "Usual Suspects" may appeal to those who appreciate fast-paced, suspenseful stories. The inclusion of "Pulp Fiction" and "Dances with Wolves" shows the system's ability to recommend popular and highly acclaimed films to a wide range of user tastes.

Collaborative filtering recommended "Star Trek: Generations," a movie User 569 liked. This shows the algorithm can identify a user's preferences and suggest related stuff. The top recommendations include a variety of genres to give users a diverse and entertaining movie-watching experience. Overall, the hybrid recommendation system captures User 569's movie tastes and provides different and relevant recommendations. The results demonstrate the system's ability to recognize user preferences and recommend a mix of popular, classic, and unknown films. These results demonstrate that the hybrid method improves user satisfaction and cinematic experiences. The result of cosine similarity is 100 % for this user.

Table 4.17: Top 10 Movie Recommendation for User 569

No	Movie
Movie 1	['Little Big League (1994)'] from genres ['Comedy|Drama']
Movie 2	['StarTrekGenerations (1994)'] from genres ['Adventure|Drama|Sci-Fi']
Movie 3	['Clean Slate (1994)'] from genres ['Comedy']
Movie 4	['Far From Home The Adventures of Yellow Dog (1995)'] from genres ['Adventure|Children']
Movie 5	['Blue Chips (1994)'] from genres ['Drama']

Table 4.18: User 569 Preferred Movies

Movie	Generic
GoldenEye (1995)	Action|Adventure|Thriller
Usual Suspects, The (1995)	Crime|Mystery|Thriller
Batman Forever (1995)	Action|Adventure|Comedy|Crime
Die Hard: With a Vengeance (1995)	Action|Crime|Thriller
Net, The (1995)	Action|Crime|Thriller
Dumb & Dumber (Dumb and Dumber) (1994)	Adventure|Comedy
Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
Stargate (1994)	Action|Adventure|Sci-Fi
Star Trek:	Adventure|Drama|Sci-Fi

Generations (1994)	
Ace Ventura: Pet Detective (1994)	Comedy
Clear and Present Danger (1994)	Action|Crime|Drama|Thriller
Forrest Gump (1994)	Comedy|Drama|Romance|War
Speed (1994)	Action|Romance|Thriller
True Lies (1994)	Action|Adventure|Comedy|Romance|Thriller
Cliffhanger (1993)	Action|Adventure|Thriller
Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller
Aladdin (1992)	Adventure|Animation|Children|Comedy|Musical
Dances with Wolves (1990)	Adventure|Drama|Western
Batman (1989)	Action|Crime|Thriller
Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX

4.6.2 Evaluating the Proposed Recommender System

In previous sections, discussed the structure of the Recommendation System. This paragraph, shall look at GUI which is crucial in allowing users to interact with system functionality. The recommendation system has an easy to use web app called GUI. The system has unique functionalities on each page that allow seamless retrieval of data through the associated application programming interfaces (API). Users can find here numerous movie propositions in this part of the application. Users are able to narrow the set of recommended stories with different user-defined filters. Such filters include release-year and any other factors such as duration, genre, or a combination of these features. After selecting a preferred movie recommendation, users are directed into another window that has comprehensive information regarding the selected movie as illustrated in Figure 4.17.

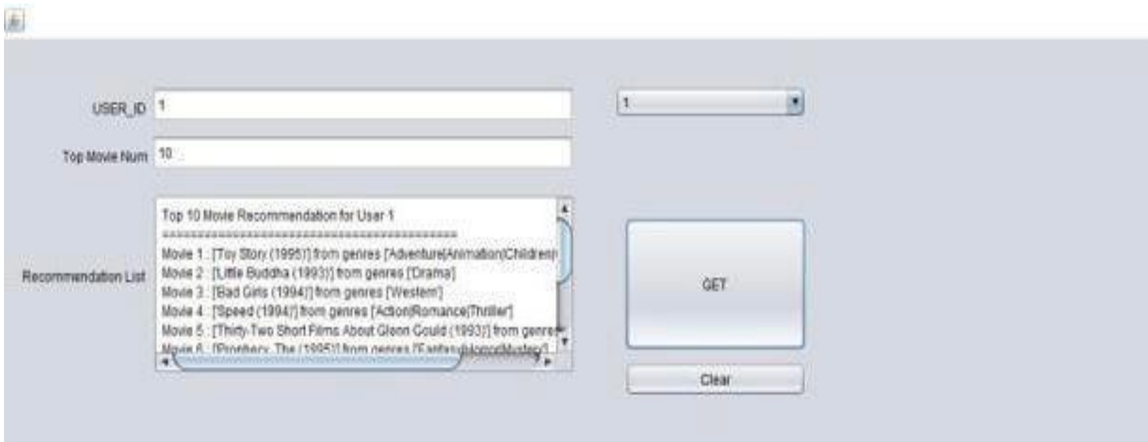

Figure 4.17: Recommendation Page

During its second development stage, the system was enriched with personal estimated scores. This aspect appreciates the fact that numerous customers opt for independent research before settling on specific products. Personalized estimated ratings allow for better decision making by the user as it is aligned with specific interests and needs of the user. Intuitive design of this user interface along with its extensive features creates unparalleled conditions for reaching personalized movie recommendations; such a tool can hardly be found while accessing personalized movies.

The GUI integrates smoothly with the back-end of the recommendation system hence enabling users to browse, filter as well as pick movies which are close to their personal preferences and requirements. In Table 4.19, there is a summary of the cosine similarity results based on the percentage of similarity between each user's preferred genres.

Table 4.19: Cosine Similarity Results

No	Cosine Similarity
User 2	83.3 %
User 5	88.8 %
User 258	100 %
User 406	100 %
User 569	100 %

Chapter Five

Conclusion and Future Work

5.1 Introduction

This chapter encompasses the culmination of the endeavors in constructing an advanced movie recommendation system that effectively utilizes conventional machine learning methods as well as deep learning methodologies. The deployment and thorough evaluation of our system have yielded significant insights and identified potential areas for future investigation.

5.2 Conclusion

The study conducted has resulted in the development of a movie recommendation system that demonstrates exceptional capabilities in accurately predicting user interests and delivering personalized movie recommendations. The proposed have accomplished the following:

- A robust system architecture was developed to seamlessly incorporate data preprocessing, classical machine learning methods, and a new deep learning model.
- The study made use of two essential datasets, namely Movielens20M and Movielens100K, to enable thorough testing and assessment.
- The data preparation step was executed, incorporating strategies to address missing values, employing one-hot encoding, and performing Min-Max normalization.
- The study utilized a range of regression techniques, such as Linear Regression, Gradient Boosting Regression, Decision Tree Regression, Stochastic Gradient Descent, and Bayesian Ridge Regression.

- .The study proposed an innovative hybrid deep learning model that demonstrated substantial enhancements in prediction accuracy. This was demonstrated via the use of superior performance indicators such as mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE).
- The performance of the proposed system surpasses that of previous models, therefore confirming its potential to provide accurate movie suggestions. We have meticulously delineated every stage, commencing from data preparation through hybrid deep learning, offering valuable insights into their importance in the overall efficacy of the system.

5.3 Future Work

In this section, we discuss potential avenues for future research and development in the field. As the research draws to a close, numerous potential areas for further investigation become apparent.

- Additional investigation into sophisticated deep learning architectures, such as recurrent neural networks (RNNs) and transformer-based models, may possibly yield heightened precision in recommendation systems.
- The inclusion of other data sources, such as social media activity and user demographics, has the potential to improve the system's capacity to accurately record user preferences.
- The objective of this study is to design and implement mechanisms that enable real-time recommendations. Additionally, the potential application of reinforcement learning techniques for the development of dynamic recommendation methods will be investigated.

- The study aims to explore approaches that enhance the transparency and interpretability of suggestions, with a specific focus on mitigating concerns related to fairness and bias in the recommendation process.

- Scalability is a crucial factor in practical applications as it involves the capacity of the system to effectively manage greater datasets and accommodate growing user populations without compromising performance.

In summary, the study has effectively demonstrated a robust hybrid methodology for the development of a complete movie recommendation system. The findings illustrate the efficacy of the system in delivering users with movie suggestions that exhibit a high level of accuracy. The future work delineated in this paper presents opportunities for more innovation and enhancements, so assuring the ongoing evolution and global benefits of the recommendation systems area.

References

[1] E. YALÇIN, "Effects of neighborhood-based collaborative filtering parameters on their blockbuster bias performances," *Sakarya University Journal of Computer and Information Sciences*, vol. 5, no. 2, 2022, doi: 10.35377/saucis...1065794.

[2] M. Schedl and C. Bauer, "An analysis of global and regional mainstreamness for personalized music recommender systems," *Journal of Mobile Multimedia*, vol. 14, no. 1, 2018, doi: 10.13052/jmm1550-4646.1415.

[3] L. Duan, W. Wang, and B. Han, "A hybrid recommendation system based on fuzzy c-means clustering and supervised learning," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 7. 2021. doi: 10.3837/tiis.2021.07.006.

[4] C. Liu, "Personalized Recommendation Algorithm for Movie Data Combining Rating Matrix and User Subjective Preference," *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/2970514.

[5] J. Li, J. Song, and C. Wang, "The Business Negotiation Between Apple, Netflix and Samsung: An Interest-based Analysis," in *Proceedings of the 2021 6th International Conference on Modern Management and Education Technology（MMET 2021）*, 2021. doi: 10.2991/assehr.k.211011.020.

[6] T. Qing-Ji, W. Hao, W. Cong, and G. Qi, "A personalized hybrid recommendation strategy based on user behaviors and its application," in *2017 International Conference on Security, Pattern Analysis, and Cybernetics, SPAC 2017*, 2018. doi: 10.1109/SPAC.2017.8304272.

[7] B.-B. Cui, "Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm," *ITM Web of Conferences*, vol. 12, 2017, doi: 10.1051/itmconf/20171204008.

[8] B. Zhang, M. Zhu, M. Yu, D. Pu, and G. Feng, "Extreme Residual Connected Convolution-Based Collaborative Filtering for Document Context-Aware Rating Prediction," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2981088.

[9] J. Monsalve-Pulido, J. Aguilar, E. Montoya, and C. Salazar, "Autonomous recommender system architecture for virtual learning environments," *Applied Computing and Informatics*, 2020, doi: 10.1016/j.aci.2020.03.001.

[10] V. Vellaichamy and V. Kalimuthu, "Hybrid collaborative movie recommender system using clustering and bat optimization," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 5, pp. 38–47, 2017, doi: 10.22266/ijies2017.1031.05.

[11] E. F. Abdullah, G. Al-Sultany, and H. N. Nawaf, "Rough set based context suggestions," *J Theor Appl Inf Technol*, vol. 96, no. 21, 2018.

[12] L. Al Hassanieh, C. A. Jaoudeh, J. B. Abdo, and J. Demerjian, "Similarity measures for collaborative filtering recommender systems," in *2018 IEEE Middle East and North Africa Communications Conference, MENACOMM 2018*, 2018. doi: 10.1109/MENACOMM.2018.8371003.

[13] H. Al-bashiri, M. A. Abdulgabber, A. Romli, and N. B. Salehudin, "A developed collaborative filtering similarity method to improve the accuracy of recommendations under data sparsity," *International Journal of Advanced*

Computer Science and Applications, vol. 9, no. 4, 2018, doi: 10.14569/IJACSA.2018.090423.

[14] X. Li, Z. Wang, L. Wang, R. Hu, and Q. Zhu, “A multi-dimensional context-aware recommendation approach based on improved random forest algorithm,” *IEEE Access*, vol. 6, pp. 45071–45085, Aug. 2018, doi: 10.1109/ACCESS.2018.2865436.

[15] S. Demissie Seifu and S. Mogalla, “INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING A Clustering-based Context-Aware Recommender System through Extracting Relevant Contextual Features and Exploring Latent Preferences,” vol. 6.

[16] M. Singh, H. Sahu, and N. Sharma, “A Personalized Context-Aware Recommender System Based on User-Item Preferences,” in *Advances in Intelligent Systems and Computing*, vol. 839, Springer Verlag, 2019, pp. 357–374. doi: 10.1007/978-981-13-1274-8_28.

[17] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, and R. Lara-Cabrera, “A Collaborative Filtering Approach Based on Naïve Bayes Classifier,” *IEEE Access*, vol. 7, pp. 108581–108592, 2019, doi: 10.1109/ACCESS.2019.2933048.

[18] L. Shuxian and F. Sen, “Design and Implementation of Movie Recommendation System Based on Naive Bayes,” in *Journal of Physics: Conference Series*, 2019. doi: 10.1088/1742-6596/1345/4/042042.

[19] K. U. Kala and M. Nandhini, “Gated recurrent unit architecture for context-aware recommendations with improved similarity measures,” *KSII Transactions on*

Internet and Information Systems, vol. 14, no. 2, pp. 538–561, 2020, doi: 10.3837/tiis.2020.02.004.

[20] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," *Tsinghua Sci Technol*, vol. 25, no. 2, 2020, doi: 10.26599/TST.2018.9010118.

[21] V. Boppana and P. Sandhya, "Web crawling based context aware recommender system using optimized deep recurrent neural network," *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00534-7.

[22] S. Manimurugan and S. Almutairi, "A user-based video recommendation approach using CAC filtering, PCA with LDOS-CoMoDa," *Journal of Supercomputing*, vol. 78, no. 7, pp. 9377–9391, May 2022, doi: 10.1007/s11227-021-04213-5.

[23] S. Y. Jeong and Y. K. Kim, "Deep Learning-Based Context-Aware Recommender System Considering Contextual Features," *Applied Sciences (Switzerland)*, vol. 12, no. 1, Jan. 2022, doi: 10.3390/app12010045.

[24] S. Sridhar, D. Dhanasekaran, and G. C. P. Latha, "Content-Based Movie Recommendation System Using MBO with DBN," *Intelligent Automation and Soft Computing*, vol. 35, no. 3, 2023, doi: 10.32604/iasc.2023.030361.

[25] A. Saleh, N. Dharshinni, D. Perangin-Angin, F. Azmi, and M. I. Sarif, "Implementation of Recommendation Systems in Determining Learning Strategies Using the Naïve Bayes Classifier Algorithm," *Sinkron*, vol. 8, no. 1, pp. 256–267, Jan. 2023, doi: 10.33395/sinkron.v8i1.11954.

[26] R. Sujatha and T. Abirami, "Improving Recommendation for Effective Personalization in Context-Aware Data Using Novel Neural Network," *Computer Systems Science and Engineering*, vol. 46, no. 2, pp. 1775–1787, 2023, doi: 10.32604/csse.2023.031552.

[27] A. Britzolakis, H. Kondylakis, and N. Papadakis, "A Review on Lexicon-Based and Machine Learning Political Sentiment Analysis Using Tweets," Int J Semant Comput, vol. 14, no. 4, 2020, doi: 10.1142/S1793351X20300010.

[28] A. Gangwar and T. Mehta, "Sentiment Analysis of Political Tweets for Israel Using Machine Learning," in Springer Proceedings in Mathematics and Statistics, 2023. doi: 10.1007/978-3-031-15175-0_15.

[29] U. Can and B. Alatas, "A new direction in social network analysis: Online social network analysis problems and applications," Physica A: Statistical Mechanics and its Applications, vol. 535. 2019. doi: 10.1016/j.physa.2019.122372.

[30] S. Bayrakdar, I. Yucedag, M. Simsek, and I. A. Dogru, "Semantic analysis on social networks: A survey," International Journal of Communication Systems, vol. 33, no. 11, 2020, doi: 10.1002/dac.4424.

[31] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook: Third Edition. Springer US, 2022. doi: 10.1007/978-1-0716-2197-4.

[32] S. Ilarri, R. Trillo-Lado, and R. Hermoso, "Datasets for context-aware recommender systems: Current context and possible directions," in Proceedings - IEEE 34th International Conference on Data Engineering Workshops, ICDEW 2018, 2018. doi: 10.1109/ICDEW.2018.00011.

[33] J. Zhao, F. Zhuang, X. Ao, Q. He, H. Jiang, and L. Ma, “Survey of Collaborative Filtering Recommender Systems,” Journal of Cyber Security, vol. 6, no. 5. 2021. doi: 10.19363/J.cnki.cn10-1380/tn.2021.09.02.

[34] M. V. Kumar and P. N. V. S. P. Kumar, “A study on different phases and various recommendation system techniques,” International Journal of Recent Technology and Engineering, vol. 7, no. 5, 2019.

[35] Y. Cui, “Intelligent Recommendation System Based on Mathematical Modeling in Personalized Data Mining,” Math Probl Eng, vol. 2021, 2021, doi: 10.1155/2021/6672036.

[36] D. Jin et al., “A Survey on Fairness-aware Recommender Systems,” ArXiv, 2023.

[37] R. Chen, Q. Hua, Y. S. Chang, B. Wang, L. Zhang, and X. Kong, “A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks,” IEEE Access, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2877208.

[38] A. Sheshasaayee and P. Muniyandi, “Recommender Systems: An Introduction and Overview,” The International journal of analytical and experiment modal analysis, vol. XII, no. I, 2020.

[39] A. Ziani et al., “Recommender System Through Sentiment Analysis,” 2nd International Conference on Automatic Control, Telecom- munications and Signals, 2017.

[40] Y. Hu, Z. Qiu, and X. Wu, “Denoising Neural Network for News Recommendation with Positive and Negative Implicit Feedback,” in Findings of

the Association for Computational Linguistics: NAACL 2022 - Findings, 2022. doi: 10.18653/v1/2022.findings-naacl.178.

[41] P. Kumar, V. Kumar, and R. S. Thakur, “A new approach for rating prediction system using collaborative filtering,” Iran Journal of Computer Science, vol. 2, no. 2, 2019, doi: 10.1007/s42044-018-00028-5.

[42] N. F. AL-Bakri and S. H. Hashim, “A modified similarity measure for improving accuracy of user-based collaborative filtering,” Iraqi Journal of Science, vol. 59, no. 2, 2018, doi: 10.24996/IJS.2018.59.2B.15.

[43] N. F. Al-Bakri and S. H. Hashim, “Reducing Data Sparsity in Recommender Systems,” Journal of Al-Nahrain University Science, vol. 21, no. 2, pp. 138–147, Jun. 2018, doi: 10.22401/JNUS.21.2.20.

[44] K. Abhishek, S. Kulkarni, V. K. N. Archana, and P. Kumar, “A Review on Personalized Information Recommendation System Using Collaborative Filtering,” International Journal of Computer Science and Information Technologies, vol. 2, no. 3, 2011.

[45] V. K. Sejwal, M. Abulaish, and Jahiruddin, “Crecsys: A context-based recommender system using collaborative filtering and lod,” IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3020005.

[46] F. Ricci, B. Shapira, and L. Rokach, “Recommender systems: Introduction and challenges,” in Recommender Systems Handbook, Second Edition, 2015. doi: 10.1007/978-1-4899-7637-6_1.

[47] A. Yamashita, H. Kawamura, and K. Suzuki, “Adaptive fusion method for user-based and item-based collaborative filtering,” in Advances in Complex Systems, 2011. doi: 10.1142/S0219525911003001.

[48] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, “Evaluating Collaborative Filtering Recommender Algorithms: A Survey,” IEEE Access, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2883742.

[49] A. Gharahighehi, K. Pliakos, and C. Vens, “Recommender systems in the real estate market—a survey,” Applied Sciences (Switzerland), vol. 11, no. 16. 2021. doi: 10.3390/app11167502.

[50] S. Dara, C. R. Chowdary, and C. Kumar, “A survey on group recommender systems,” J Intell Inf Syst, vol. 54, no. 2, 2020, doi: 10.1007/s10844-018-0542-3.

[51] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” Advances in Artificial Intelligence, vol. 2009, 2009, doi: 10.1155/2009/421425.

[52] G. Yao and L. Cai, “User-Based and Item-Based Collaborative Filtering Recommendation Algorithms Design.”

[53] A. Aldahiri, B. Alrashed, and W. Hussain, “Trends in Using IoT with Machine Learning in Health Prediction System,” Forecasting, vol. 3, no. 1, 2021, doi: 10.3390/forecast3010012.

[54] K. Yeturu, “Machine learning algorithms, applications, and practices in data science,” in Handbook of Statistics, vol. 43, 2020. doi: 10.1016/bs.host.2020.01.002.

[55] Y. Ren, "Python Machine Learning : Machine Learning and Deep Learning With Python ," International Journal of Knowledge-Based Organizations, vol. 11, no. 1, 2021.

[56] J. Bowlee, "Logistic Regression for Machine Learning," Machine Learning Mastery, 2016.

[57] V. Gavrishchaka, O. Senyukova, and M. Koepke, "Synergy of physics-based reasoning and machine learning in biomedical applications: Towards unlimited deep learning with limited data," Advances in Physics: X, vol. 4, no. 1. 2019. doi: 10.1080/23746149.2019.1582361.

[58] C. Bulac and A. Bulac, "Decision Trees," in Advanced Solutions in Power Systems: HVDC, FACTS, and AI Techniques, 2016. doi: 10.1002/9781119175391.ch18.

[59] Y. Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Arch Psychiatry, vol. 27, no. 2, 2015, doi: 10.11919/j.issn.1002-0829.215044.

[60] A. Rani, N. Kumar, J. Kumar, and N. K. Sinha, "Machine learning for soil moisture assessment," in Deep Learning for Sustainable Agriculture, 2022. doi: 10.1016/B978-0-323-85214-2.00001-X.

[61] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments," Energies (Basel), vol. 14, no. 16, 2021, doi: 10.3390/en14165196.

[62] K. Gao, H. Chen, X. Zhang, X. K. Ren, J. Chen, and X. Chen, "A novel material removal prediction method based on acoustic sensing and ensemble XGBoost learning algorithm for robotic belt grinding of Inconel 718," International Journal of Advanced Manufacturing Technology, vol. 105, no. 1–4, 2019, doi: 10.1007/s00170-019-04170-7.

[63] M. Rajesh, S. Anishka, P. S. Viksit, S. Arohi, and S. Rehana, "Improving Short-range Reservoir Inflow Forecasts with Machine Learning Model Combination," Water Resources Management, vol. 37, no. 1, 2023, doi: 10.1007/s11269-022-03356-1.

[64] S. I. Ayon, M. M. Islam, and M. R. Hossain, "Coronary Artery Heart Disease Prediction: A Comparative Study of Computational Intelligence Techniques," IETE J Res, vol. 68, no. 4, 2022, doi: 10.1080/03772063.2020.1713916.

[65] L. M. Sinaga, Sawaluddin, and S. Suwilo, "Analysis of classification and Naïve Bayes algorithm k-nearest neighbor in data mining," in IOP Conference Series: Materials Science and Engineering, 2020. doi: 10.1088/1757-899X/725/1/012106.

[66] B. Krithiga, P. Sabari, I. Jayasri, and I. Anjali, "Early detection of coronary heart disease by using naive bayes algorithm," in Journal of Physics: Conference Series, 2021. doi: 10.1088/1742-6596/1717/1/012040.

[67] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning-Based Text Classification," ACM Computing Surveys, vol. 54, no. 3. 2021. doi: 10.1145/3439726.

[68] Y. Zhao, Q. Chen, W. Cao, W. Jiang, and G. Gui, "Deep Learning Based Couple-like Cooperative Computing Method for IoT-based Intelligent Surveillance Systems," in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2019. doi: 10.1109/PIMRC.2019.8904229.

[69] H. Greenspan, B. Van Ginneken, and R. M. Summers, "Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique," IEEE Transactions on Medical Imaging, vol. 35, no. 5. 2016. doi: 10.1109/TMI.2016.2553401.

[70] F. J. Díaz-Pernas, M. Martínez-Zarzuela, D. González-Ortega, and M. Antón-Rodríguez, "A deep learning approach for brain tumor classification and segmentation using a multiscale convolutional neural network," Healthcare (Switzerland), vol. 9, no. 2, Feb. 2021, doi: 10.3390/healthcare9020153.

[71] F. Alrasheedi, X. Zhong, and P. C. Huang, "Padding Module: Learning the Padding in Deep Neural Networks," IEEE Access, vol. 11, pp. 7348–7357, 2023, doi: 10.1109/ACCESS.2023.3238315.

[72] C. Edwin Singh and S. Maria Celestin Vigila, "WOA-DNN for Intelligent Intrusion Detection and Classification in MANET Services," Intelligent Automation and Soft Computing, vol. 35, no. 2, pp. 1737–1751, 2023, doi: 10.32604/iasc.2023.028022.

[73] S. A. Alazawi and M. N. Al Salam, "Evaluation of LMT and DNN Algorithms in Software Defect Prediction for Open-Source Software," in Advances in Intelligent Systems and Computing, 2021. doi: 10.1007/978-981-15-7527-3_19.

[74] J. Baek and Y. Choi, “Deep neural network for predicting ore production by truck-haulage systems in open-pit mines,” Applied Sciences (Switzerland), vol. 10, no. 5, 2020, doi: 10.3390/app10051657.

[75] V. Sharma, M. Gupta, A. K. Pandey, D. Mishra, and A. Kumar, “A Review of Deep Learning-based Human Activity Recognition on Benchmark Video Datasets,” Applied Artificial Intelligence, vol. 36, no. 1, 2022, doi: 10.1080/08839514.2022.2093705.

[76] J. Sui, M. X. Liu, J. H. Lee, J. Zhang, and V. Calhoun, “Deep learning methods and applications in neuroimaging,” Journal of Neuroscience Methods, vol. 339. 2020. doi: 10.1016/j.jneumeth.2020.108718.

[77] D. Jung and Y. Choi, “Systematic review of machine learning applications in mining: Exploration, exploitation, and reclamation,” Minerals, vol. 11, no. 2. 2021. doi: 10.3390/min11020148.

[78] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” IEEE Trans Neural Netw Learn Syst, pp. 1–21, Jun. 2021, doi: 10.1109/tnnls.2021.3084827.

[79] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” Mech Syst Signal Process, vol. 151, 2021, doi: 10.1016/j.ymssp.2020.107398.

[80] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, “Evolving Deep Convolutional Neural Networks for Image Classification,” IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, 2020, doi: 10.1109/TEVC.2019.2916183.

[81] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” Artif Intell Rev, vol. 53, no. 8, 2020, doi: 10.1007/s10462-020-09825-6.

[82] R. A. Pratiwi, S. Nurmaini, D. P. Rini, M. N. Rachmatullah, and A. Darmawahyuni, “Deep ensemble learning for skin lesions classification with convolutional neural network,” IAES International Journal of Artificial Intelligence, vol. 10, no. 3, 2021, doi: 10.11591/ijai.v10.i3.pp563-570.

[83] M. A. Gómez-Guzmán et al., “Classifying Brain Tumors on Magnetic Resonance Imaging by Using Convolutional Neural Networks,” Electronics (Switzerland), vol. 12, no. 4, 2023, doi: 10.3390/electronics12040955.

[84] V. H. Phung and E. J. Rhee, “A High-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” Applied Sciences (Switzerland), vol. 9, no. 21, 2019, doi: 10.3390/app9214500.

[85] H. Gu, Y. Wang, S. Hong, and G. Gui, “Blind channel identification aided generalized automatic modulation recognition based on deep learning,” IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2934354.

[86] P. Xu, Z. Guo, L. Liang, and X. Xu, “MSF-net: Multi-scale feature learning network for classification of surface defects of multifarious sizes,” Sensors, vol. 21, no. 15, 2021, doi: 10.3390/s21155125.

[87] S. A. Singh, T. G. Meitei, and S. Majumder, “Short PCG classification based on deep learning,” in Deep Learning Techniques for Biomedical and Health Informatics, 2020. doi: 10.1016/B978-0-12-819061-6.00006-9.

[88] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved handwritten digit recognition using convolutional neural networks (Cnn),” Sensors (Switzerland), vol. 20, no. 12, 2020, doi: 10.3390/s20123344.

[89] R. D. Rakshit, D. R. Kisku, P. Gupta, and J. K. Sing, “Cross-resolution face identification using deep-convolutional neural network,” Multimed Tools Appl, vol. 80, no. 14, 2021, doi: 10.1007/s11042-021-10745-y.

[90] A. Kost, W. A. Altabey, M. Noori, and T. Awad, “Applying neural networks for tire pressure monitoring systems,” SDHM Structural Durability and Health Monitoring, vol. 13, no. 3, 2019, doi: 10.32604/sdhm.2019.07025.

[91] S. H. Wang, J. Hong, and M. Yang, “Sensorineural hearing loss identification via nine-layer convolutional neural network with batch normalization and dropout,” Multimed Tools Appl, vol. 79, no. 21–22, 2020, doi: 10.1007/s11042-018-6798-3.

[92] X. Zhang, Y. Wang, N. Zhang, D. Xu, and B. Chen, “Research on scene classification method of high-resolution remote sensing images based on RFPNet,” Applied Sciences (Switzerland), vol. 9, no. 10, 2019, doi: 10.3390/app9102028.

[93] Z. Chen, Q. Xue, Y. Wu, S. Shen, Y. Zhang, and J. Shen, “Capacity prediction and validation of lithium-ion batteries based on long short-term memory recurrent neural network,” IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3025766.

[94] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on ADAM," Applied Sciences (Switzerland), vol. 10, no. 3, 2020, doi: 10.3390/app10031073.

[95] W. E. L. Ilboudo, T. Kobayashi, and K. Sugimoto, "Robust Stochastic Gradient Descent with Student-t Distribution Based First-Order Momentum," IEEE Trans Neural Netw Learn Syst, vol. 33, no. 3, 2022, doi: 10.1109/TNNLS.2020.3041755.

[96] R. Tiwari, "Stabilizing the training of deep neural networks using Adam optimization and gradient clipping," INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, vol. 07, no. 01, 2023, doi: 10.55041/ijsrem17594.

[97] T. O. Hodson, T. M. Over, and S. S. Foks, "Mean Squared Error, Deconstructed," J Adv Model Earth Syst, vol. 13, no. 12, 2021, doi: 10.1029/2021MS002681.

[98] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput Sci, vol. 7, 2021, doi: 10.7717/PEERJ-CS.623.

[99] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," Geoscientific Model Development, vol. 15, no. 14. 2022. doi: 10.5194/gmd-15-5481-2022.

[100] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean Absolute Percentage Error for regression models," Neurocomputing, vol. 192, 2016, doi:

الخلاصة

مع زيادة شعبية منصات البث الرقمي في جميع أنحاء العالم ، أصبحت مجموعة كبيرة من الأفلام متاحة بسهولة للعرض. على الرغم من أن هناك الآلاف للاختيار من بينها ، فإن تضييقهم بناءً على الاهتمام الشخصي يمكن أن يكون مشكلة. تتم معالجة هذه المشكلة من خلال أنظمة توصية الأفلام وتقديم توصيات مخصصة للمستخدمين. تعتمد أنظمة التوصية بالأفلام التقليدية في الغالب على أساليب فردية مثل التصفية التعاونية أو التصفية القائمة على المحتوى. شكلا التصفية التعاونية هما التصفية التعاونية القائمة على المستخدم والتصفية التعاونية القائمة على العناصر. ومع ذلك، هناك أيضًا قيود مرتبطة بتقنيات التصفية هذه. تعاني التصفية التعاونية من مشكلة (البداية الباردة) للمستخدمين الجدد والمحتويات الجديدة، في حين أن التصفية القائمة على المحتوى لا تقدم توصيات متنوعة كافية. لذلك، تهدف هذه الأطروحة إلى تعزيز أنظمة التوصية بالأفلام باستخدام مزيج من التصفية التعاونية والتصفية القائمة على المحتوى. يشير هذا إلى أنه من خلال تحليل تفضيلات المستخدم (أي أذواقه لمختلف الأفلام أو الشخصيات أو مخرجي الأفلام) والسمات المختلفة للفيلم أو الفيلم، سيكون النظام المقترح (أي المقترح) قادرًا على تقديم توصيات مخصصة للمستخدمين . تقترح هذه الأطروحة نظام توصية هجين يعتمد على نقاط القوة في التصفية التعاونية والمبنية على المحتوى. من أجل التحايل على المشكلات في كل نهج، يدمج النظام المقترح إمكانية التصفية التعاونية لاكتشاف الأفراد ذوي الأذواق المحبوبة وإمكانية التصفية القائمة على المحتوى في مراجعة خصائص الفيلم. بهذه الطريقة يحصل كل من المستخدمين الجدد والقدامى على توصيات مختلفة ومناسبة للأفلام في جميع الأوقات. يتم تقييم فعالية النظام الهجين المقترح باستخدام مجموعتي بيانات شائعتين: K١٠٠MovieLens و M٢٠MovieLens. هذه النتائج مشجعة حيث أن مجموعة بيانات -K١٠٠MovieLens تحتوي على MAE يبلغ ٠.٧٩ وRMSE يبلغ ٠.٩٢ بينما تحتوي مجموعة بيانات M٢٠MovieLens على MAE يبلغ ٠.٨. توضح هذه الإحصائيات مدى دقة نظام التوصية من خلال مطابقة تفضيلات المستخدم وسجل العرض. . تظهر هذه النتائج أنها أداة مفيدة للغاية لتقديم توصيات دقيقة ومتنوعة تعمل على تحسين تجربة المستخدم والتغلب على تحديات الثقافة الحديثة الغنية بالمحتوى الرقمي.

الكلمات المفتاحية: نظام التوصيات، التصفية التعاونية، التصفية المبنية على المحتوى، التعلم الآلي، التعلم العميق.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرمجيات

التنبؤ بتفضيلات المستخدمين للأفلام وتزويد التوصيات بالاعتماد على طريقة هجينة مقترحة

أطروحة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات جامعة بابل استكمالاً لمتطلبات نيل شهادة الدكتوراه فلسفة في تكنولوجيا المعلومات / البرمجيات

الطالب

صدام حمدان أحمد عبدالله

بإشراف

الاستاذ الدكتور عباس فاضل محمد علي

٢٠٢٣ ميلادي ١٤٤٥ هجري