LEVERAGING SOFTMAX AND DEEP NEURAL NETWORKS FOR ENHANCED MOVIE RECOMMENDATIONS

A Project Report

Submitted to the Faculty of Engineering of
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

By

R. L. PRASOON KUMAR
(20481A05K0)

Y. RUDRA PRAKASH
(20481A05P5)

V. DUNESH
(20481A05O3)

Y. NAGA SANDEESH
(20481A05P3)

Under the guidance of
Dr. N. RAJESWARI, M. Tech, Ph.D,
Professor of CSE Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRIRAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356
ANDHRA PRADESH

2023-2024

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

<u>CERTIFICATE</u>

This is to certify that the project report entitled **"LEVERAGING SOFTMAX AND DEEP NEURAL NETWORKS FOR ENHANCED MOVIE RECOMMENDATIONS"** is a bonafide record of work carried out by **R. L. PRASOON KUMAR (20481A05K0), Y. RUDRA PRAKASH(20481A05P5), V. DUNESH (20481A05O3) and Y. NAGA SANDEESH (20481A05P3)** under the guidance and supervision in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru TechnologicalUniversity Kakinada, Kakinada during the academic year 2023-24.

Project Guide	**Head of the Department**
(Dr. N. RAJESWARI)	**(Dr. M. BABU RAO)**

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr. N. Rajeswari, Professor**, Department of Computer Science and Engineering for her constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal**Dr. G.V.S.N.R.V Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

Team members

R. L. Prasoon Kumar (20481A05K0)
Y. Rudra Prakash (20481A05P5)
V. Dunesh (20481A05O3)
Y. Naga Sandeesh (20481A05P3)

INDEX

LIST OF ABBREVATIONS

Abbreviation	Explanation
MMMF	Maximum Margin Matrix Factorization
DMF	Deep Matrix Factorization
NCF	Neural Collaborative Filtering
IFE	Implicit Feedback Embedding
MAP	Mean Average Precision
SVD	Singular Value Decomposition

ABSTRACT

Collaborative filtering research has found success with the learning technique known as Maximum Margin Matrix Factorization (MMMF). In order to simultaneously approximate the observed entries under a partially observed ordinal rating matrix, low-norm latent component matrices U (of users) and V (of items) must be determined. Unobserved entries are measured and predicted by some loss. Since there are just two levels (±1) in the rating matrix, rows of V can be thought of as points in k-dimensional space and rows of U as decision hyperplanes dividing +1 entries from -1 entries. When the loss function is hinge/smooth hinge loss, the hyperplanes serve as maximum-margin separators. When using Deep MMMF, we employ a neural network architecture to perform the matrix factorization task and the hinge loss function is specifically extended to accommodate several levels in order to handle rating matrices with multiple discrete values. Along with learning user and item latent vectors, we additionally learn another vector θ, unique for each user, that separates the user-item interaction hyperplane into several parts depending upon the number of possible ratings.

Keywords - Collaborative Filtering, Recommender Systems, Deep Learning, Matrix Factorization.

CHAPTER – 1

INTRODUCTON

1.1 INTRODUCTION

In the era of information overload and diverse entertainment options, the task of selecting a movie to watch can often be overwhelming for consumers. Movie recommendation systems play a pivotal role in alleviating this decision-making burden by providing personalized suggestions tailored to individual preferences. These systems not only enhance user satisfaction but also contribute to increased user engagement and loyalty in various online platforms, including streaming services and e-commerce websites.

The objective of this project is to develop a movie recommender system using the Maximum Margin Matrix Factorization (MMMF) approach. Collaborative filtering techniques, which analyze user-item interactions to generate recommendations, have gained widespread popularity due to their ability to capture complex patterns in user behavior. MMMF, a variant of matrix factorization, extends this paradigm by optimizing a margin-based objective function to learn low-dimensional representations of users and items, thereby improving recommendation quality.

The significance of personalized recommendation systems in the movie industry cannot be overstated. With a vast library of movies spanning different genres, languages, and release years, users often struggle to discover content that aligns with their unique tastes and preferences. By leveraging MMMF, our recommender system aims to address this challenge by accurately predicting user preferences based on past interactions and implicit feedback.

This project contributes to the existing body of research on recommender systems by exploring the effectiveness of MMMF in generating high-quality movie recommendations. By combining principles from machine learning, optimization, and information retrieval, we seek to develop a robust and scalable solution that can scale to large datasets while maintaining competitive performance metrics.

Throughout this report, we will delve into the theoretical foundations of Maximum Margin Matrix Factorization, describe the methodology employed in building the recommender system, present experimental results and evaluation metrics, and discuss implications for future research and practical applications. By the end of this study, we aim to provide insights into the efficacy of MMMF-based recommendation models.

1.1.1 Recommender System

A recommender system is a type of information filtering system that predicts the preferences or ratings that a user would give to a particular item, such as a product, movie, or article. The primary goal of a recommender system is to provide personalized recommendations to users, helping them discover items they might be interested in but may not have discovered otherwise.

Recommender systems are widely used in various online platforms such as e-commerce websites (e.g., Amazon), streaming services (e.g., Netflix), social media platforms, and content websites. They employ various algorithms and techniques to analyze user behavior, preferences, and item characteristics to generate recommendations.

We have a variety of varied applications of this recommendation systems in which we can can be used over the years and now used in various online platforms the basic content of all this platforms are basically different types of movies such as action thriller romantic or maybe your eCommerce website any platform of social media having a professional website such as Linkedin . For example when we use Instagram we can see the previous stories that on the feed of the people we follow so here we can see that the Instagram can monitor our interaction with the various people are our past activities and then it just suggest kind of other related stories of some other accounts that have done some same kind of activity previously or currently. Quite a few time is recommender system also keep improving the activities of a bunch of users based on the activities they have scroll through you attempted. For example on Flipkart when we buy some laptop or any mobile phone then it simply suggests mobile cover tempered glass for mobile or buy USB type C adaptor or type A adaptor for the laptop also. Safed enhancements in the recommender systems users get good recommendation all the time and it keeps on improving as we move forward in the 21st century and they make almost accurate solutions. In case of clash of any e App Music any music platform or any educational then use a simply deny using the app in addition to this the companies have to focus on their recommendation system which is more Complex than it seems. Every user has different preferences and different choices based upon their different type of activities sometime mood also so in case of musics while playing, travelling, running aur after having some fight in relationships etc

There are several types of recommender systems, including:

1. **Collaborative Filtering:** This approach makes recommendations based on the

preferences and behavior of similar users. It can be further divided into user-based collaborative filtering and item-based collaborative filtering.

2. **Content-based Filtering:** This method recommends items similar to those the user has liked or interacted with in the past. It focuses on the attributes of items and the user's profile.

3. **Hybrid Recommender Systems:** These systems combine multiple recommendation approaches to provide more accurate and diverse recommendations. For example, a hybrid system might combine collaborative filtering and content-based filtering.

4. **Matrix Factorization Techniques:** These techniques aim to decompose a user-item interaction matrix into lower-dimensional matrices to capture latent features of users and items, making recommendations based on these features.

5. **Deep Learning-Based Recommender Systems:** With the advancements in deep learning, neural network architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied to recommender systems to learn complex patterns from user-item interactions.

Recommender systems play a crucial role in enhancing user experience, increasing user engagement, and driving business revenue by promoting relevant items to users, thereby facilitating decision-making in the vast space of available choices.

1.1.2 Content Based Recommendation

Content-based recommendation is a type of recommender system that suggests items to users based on the characteristics or attributes of the items themselves and the user's preferences. In content-based recommendation systems, items are typically represented as feature vectors based on their attributes, and recommendations are made by identifying items that are similar to those that a user has liked or interacted with in the past.

: In the content based filtering method we compare the different items with the user's interest profile. So basically the user profile holds the content that is is much more matching to use the form of the features. The previous actions or for the feedback is taken into account a generally takes into account the description of the content that has been edited by the users of different choices. Considering that example where a person buys some favourite item 'M' but item has been sold out and as a result he has to buy the item 'N' on the recommendation of some person as and 'N' has same type of matching features that the first one possesses. So this is simply the content based filtering which is demonstrated below

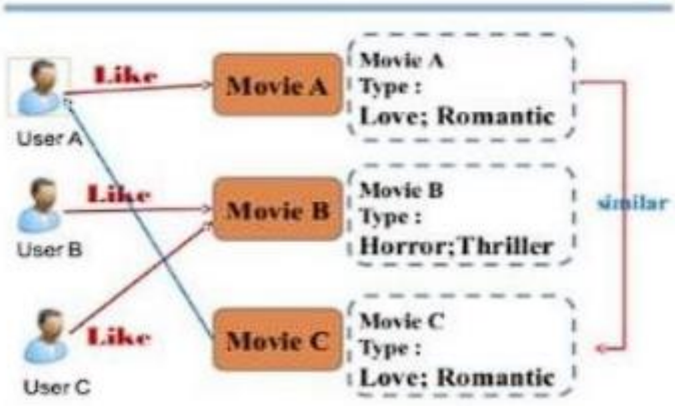


Fig.-Content Based Filtering Method

So here numeric quantity that will be used to calculate the similarity between the two types of movies will be cosine similarity and we will calculate the score it is very very fast to calculate the the magnitude of the score which is obtained through the cosine similarity.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Key components of content-based recommendation systems include:

1. **Item Representation:** Items are represented based on their attributes or features. These attributes could include textual descriptions, metadata, tags, genres, or any other relevant information about the items.

2. **User Profile:** The system maintains a profile for each user, which includes their preferences, past interactions, and possibly demographic information. The user profile

is used to capture the user's preferences and interests.

3. Feature Extraction: Features are extracted from the item attributes to create a numerical representation of each item. This can involve techniques such as vectorization of text data or encoding categorical variables.

4. Similarity Calculation: Similarity measures are used to compare the features of items and determine how similar they are to each other. Common similarity metrics include cosine similarity, Euclidean distance, or Jaccard similarity.

5. Recommendation Generation: To generate recommendations for a user, the system selects items that are most similar to the items the user has shown interest in, based on their feature representations. These recommended items are then presented to the user.

The steps involved in getting the movie recommendation are as below:

. Having the title find the index of that movie

. Calculate the cosine similarity scores for all the movies

. Arranging the scores in the order of highest priority first that is ascending order

. And then shorting the list based on the similarity scores.

. Getting the first 10 element of the list excluding the first one as it is the movie name in itself.

. Getting the top elements

Content-based recommendation systems have several advantages and disadvantages, including:

- They can provide recommendations for new or niche items that have limited user interaction data.
- They can capture the intrinsic characteristics of items, making them particularly useful for recommending items with specific features or attributes.
- They are less reliant on user behavior data, which can be beneficial in situations where user interaction data is sparse or unavailable.

However, content-based recommendation systems also have limitations, such as the tendency to recommend items that are similar to those the user has already seen, potentially leading to a lack of diversity in recommendations. Hybrid approaches that combine content-based and collaborative filtering techniques are often used to mitigate these limitations and provide more accurate and diverse recommendations.

The advantages of the content based filtering are: . Can recommend the unrated items. . Can recommend the movies based on the ratings of the user The disadvantages of the content based filtering are: . Can't work on the new user hasn't red kidney movie act . It can't make the user likes with the un -likes .

1.1.3 Collaborative Filtering

Collaborative filtering is a type of recommendation system that predicts a user's preferences or interests by collecting information from many users (collaborating). Instead of relying on explicit item features or characteristics, collaborative filtering methods are based on the idea that if two users have liked or interacted with similar items in the past, they are likely to have similar preferences in the future.

Advantages of collaborative filtering based systems are: . It is simply content dependent . It often reads the mind of people having same preferences . Create real quality assessment of items. Disadvantages of collaborative filtering based systems are: . Early rater problem as the most common where the collaborative filtering method fails to provide ratings of the movie which has no user waiting. . Sparsity problem is more common in this type of welding method where null values are in so much quantity that is difficult to find items which are rated by the majority of the people.

	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You	Similarity(i, E)
A	2		2	4	5		NA
B	5		4			1	
C			5		2		
D		1		5		4	
E			4			2	1
F	4	5		1			NA

Since user A and F do not share any movie ratings in common with user E, their similarities with user E are not defined in Pearson Correlation. Therefore, we only need to consider user B, C, and D. Based on Pearson Correlation, we can compute the following similarity.

There are two main types of collaborative filtering:

1. **User-based collaborative filtering:** This approach recommends items to a target user by identifying other users with similar tastes and preferences. It works by finding users who have rated or interacted with items in a similar way to the target user and recommending items that these similar users have liked but the target user hasn't interacted with yet.

2. **Item-based collaborative filtering:** In this approach, similarities between items are calculated based on the ratings or interactions of users who have interacted with both items. Then, recommendations are made by identifying items similar to those that the target user has already shown interest in.

Collaborative filtering methods do not require explicit information about items; they rely solely on user-item interaction data, such as ratings, likes, purchases, or views. This makes them particularly useful in situations where item features are not readily available or where users' preferences are more dynamic and context-dependent.

1.1.4 Hybrid Recommender Systems

Hybrid approaches that combine collaborative filtering with other recommendation techniques, such as content-based filtering or matrix factorization, are often used to address these limitations and improve the accuracy and diversity of recommendations.

It is simply a mixture of content based filtering and collaborative based filtering methods In the comparisons section below we will see how movies are determined through the hybrid technique of filtering where we have both used content based method as well as the collaborative based filtering method. It is clear that hybrid filtering method is is good in most of the cases and scenarios where it is difficult to distinguish or get the accuracy which the users can get the recommended movies, where we will take the input as the the userid and the title of the movie and the output will be e the similar movies shorted by the particular users based on the expected ratings. Expected ratings are calculated internally where the ideas from content and collaborative filtering are used to build a engine where movies are suggested to the particular user and then estimation of the ratings takes place.

1.2 OBJECTIVES OF THE PROJECT

The project's objectives center on developing a movie recommender system using Maximum Margin Matrix Factorization (MMMF) and evaluating its effectiveness in generating personalized recommendations compared to traditional methods. Key goals include optimizing MMMF parameters, assessing scalability, and comparing performance with other recommendation algorithms. Additionally, objectives involve exploring the interpretability of MMMF embeddings, enhancing recommendation quality with additional contextual data, and analyzing user engagement metrics for real-world applicability. Through these objectives, the project aims to advance understanding of MMMF-based recommender systems and their potential in the movie recommendation.

1. Develop a movie recommender system based on MMMF: Create a recommendation system leveraging MMMF to provide personalized movie suggestions to users.

2. Explore effectiveness compared to traditional methods: Investigate how well MMMF performs in generating recommendations compared to traditional collaborative filtering methods.

3. Investigate impact of hyperparameters and optimization: Analyze how different settings for MMMF's hyperparameters and optimization techniques affect the quality of recommendations.

4. Evaluate scalability and computational efficiency: Assess how well MMMF scales to handle large movie recommendation datasets while maintaining computational efficiency.

5. Compare performance with state-of-the-art algorithms: Measure the performance of the MMMF-based system against other advanced recommendation algorithms to determine its competitiveness.

6. Assess robustness against sparse data: Determine how well MMMF handles sparse or incomplete user-item interaction data and its impact on recommendation quality.

7. Explore interpretability of embeddings: Examine the interpretability of the user and item embeddings learned by MMMF and their relevance in generating meaningful recommendations.

8. Investigate extension with additional contextual information: Explore the potential of enhancing the MMMF-based system by incorporating additional user or movie metadata to improve recommendation quality.

9. Analyze user satisfaction and engagement metrics: Evaluate user satisfaction and engagement metrics to understand the real-world effectiveness of the MMMF-based recommender system.

10. Provide insights into practical implementation and deployment: Offer practical considerations and strategies for implementing and deploying the MMMF-based recommender system in movie streaming platforms or e-commerce websites.

1.3 PROBLEM STATEMENT

Movie recommendation systems have become indispensable tools for online platforms seeking to enhance user experience and engagement. These systems leverage machine learning algorithms to analyze user preferences and behavior, thereby providing personalized movie suggestions tailored to individual tastes. Among the various techniques employed in building recommendation systems, Maximum Margin Matrix Factorization (MMMF) has emerged as a promising approach, offering robustness, scalability, and improved recommendation quality.

At its core, MMMF extends the principles of matrix factorization to optimize a margin-based objective function, aiming to learn low-dimensional representations of users and items that best capture the underlying structure of the data. The key intuition behind MMMF lies in maximizing the margin between positive instances (i.e., user-item interactions) and negative instances (i.e., non-interactions), thereby enhancing the discriminative power of the learned embeddings.

The MMMF framework can be formulated as an optimization problem, where the objective is to minimize the sum of a loss function and a regularization term, subject to certain constraints. The loss function typically consists of a margin-based term that penalizes the distance between positive and negative instances, encouraging the model to learn embeddings that separate observed interactions from non-interactions in the latent space. Meanwhile, the regularization term helps prevent overfitting by penalizing the complexity of the learned embeddings.

One of the key advantages of MMMF is its ability to handle sparse and incomplete data, which is common in real-world recommendation scenarios where users may only provide feedback on a subset of available items. By leveraging the margin-based objective, MMMF learns embeddings that effectively capture the underlying structure of the data, even in the presence of missing values or noisy observations. This robustness makes MMMF particularly well-suited for large-scale recommendation tasks, where the dataset may be sparse or noisy.

Furthermore, MMMF offers interpretability through its learned embeddings, enabling analysts to gain insights into the latent factors driving user preferences and item characteristics. By examining the learned representations of users and items, stakeholders can identify meaningful patterns and clusters, which can inform content curation strategies and marketing campaigns.

In practical implementations, MMMF can be applied in various recommendation scenarios, including movie streaming platforms, e-commerce websites, and social media platforms. By integrating MMMF-based recommendation systems into existing platforms, businesses can enhance user engagement, increase customer satisfaction, and ultimately drive revenue growth through targeted recommendations.

In conclusion, Maximum Margin Matrix Factorization (MMMF) represents a powerful approach for building movie recommendation systems that deliver personalized suggestions to users. Through its margin-based optimization framework, MMMF offers robustness, scalability, and interpretability, making it an attractive choice for real-world recommendation tasks. By leveraging MMMF, businesses can unlock new opportunities for enhancing user experience and driving growth in the competitive landscape of online entertainment platforms.

CHAPTER – 2

LITERATURE REVIEW

Deep learning techniques have shown remarkable success in learning hierarchical representations from raw data, driving transformative advancements in the domain of information retrieval. Concurrently, the efficacy of matrix factorization methods in revealing latent relationships from data has been exhibited through applications in recommendation systems and collaborative filtering. Recently, several works have been done on deep learning-based matrix factorization techniques to leverage the capacity of deep learning models and matrix factorization together to reveal latent structures within complex datasets. In this section, we discuss the existing approaches that explore the fusion of deep learning and matrix factorization paradigms, highlighting the evolution of this hybrid methodology and the novel insights it has brought to various application domains.

By preemptively comprehending user-item dynamics, Fu et al. (2018) acquire embeddings for users and items separately to capture semantic correlations and further train a feed-forward neural network on the pre-trained embeddings to replicate user-item interactions, fostering accurate predictions. In a three staged sequential approach, Bobadilla et al. (2020) utilizes matrix factorization at first stage, and a multilayer neural network at the remaining two stages to exploit the potential of real prediction errors obtained from first stage, and enhance the recommendation quality. In Lara-Cabrera et al. (2020), authors harnessed the power of deep learning techniques by treating matrix factorization as multiple layers, such that the output of one layer serves as the input of the subsequent layer. Taking the interaction matrix as an input to a neural network, Xue et al. (2017) learn the low-dimensional representation of users and items.

1. **Matrix Factorization Model in Collaborative Filtering Algorithms**: A Survey Authors Dheeraj Bokde , Sheetal Girase, Debajyoti Mukopadhyay Publisher: Procedia Computer Science, Year: 2015 This paper attempt present a comprehensive survey of Factorization model like Singular Value Decomposition address the challenges of Collaborative Filtering algorithms, which can be served as a roadmap for research and practice in this area.

2. Latent Factor Models for Web Recommender Systems

Authors: Bee-Chung Chen, Deepak Agarwal, Pradheep Elango, Raghu Ramakrishnan Publisher:

Yahoo! Research & Yahoo! Labs This paper discuss about the model that • uses feature-based regression to predict the initial point for online learning, and • reduces the dimensionality of online learning Rapidly update online models once new data is received: • Fast learning: Low dimensional and easily parallelizable 15 • Online selection for the best dimensionality

3. Matrix Factorization Techniques for Recommender Systems

Authors: Yehuda Koren, Yahoo Research, Robert Bell and Chris Volinsky,

AT&T Labs---Research Publisher: IEEE Computer Society Press Los Alamitos, CA, USA Year: 2009 Used in Netflix Competitions This paper discusses about matrix factorization techniques have become a dominant methodology within collaborative filtering recommenders. Experience with datasets such as the Netflix Prize data has shown that they deliver accuracy superior to classical nearest-neighbour techniques. At the same time, they offer a compact memory-efficient model that systems can learn relatively easily. What makes these techniques even more convenient is that models can integrate naturally many crucial aspects of the data, such as multiple forms of feedback, temporal dynamics, and confidence levels

The model encompasses both explicit ratings as well as non-preference feedback, thus maximizing the information extracted.

In He et al. (2017), a novel generic framework for Neural Collaborative Filtering (NCF) is devised, where the inner product is replaced with a neural architecture to learn an arbitrary function from data. The proposed framework can be used with any deep learning model to provide recommendations. In another work, Yi et al. (2019) propose a generic Deep Matrix Factorization (DMF) framework that enhances the recommendation

accuracy by incorporating side information using two channel structure. DMF learns the low-dimensional embeddings of users and items directly using a feature transforming function, instead of from the observed data. They propose Implicit Feedback Embedding

(IFE) to transform users and items within an implicit feedback information graph to real-valued low-dimensional vectors, thus reducing the numbers of parameters and increasing training efficiency. With the help of convolutional neural networks and gated recurrent neural networks, Wu et al. (2018) develop a dual-regularized multilayered deep matrix factorization model to learn the user and item representations using text descriptions.

The fusion of neural network architectures with matrix factorization have significantly enriched the recommendation accuracy by learning intricate representations from data Da'u & Salim (2020); Zhang et al. (2019); Liu et al. (2022). In our work, we focus on to learn the nonlinear user-item interactions and develop a deep MMMF model that enhances the recommendation accuracy.

SYSTEM DESIGN

Dataset

1) For Content and Collaborative Based Filtering:

• Kaggle provided the data set. The Movie Recommendation System uses it as a standard Dataset.

• We used the movie dataset from 'Movie Lens(Kaggle)' for the project.

• Movies and ratings are taken into account.

• Total of 9743 movies • Total of 100147 ratings

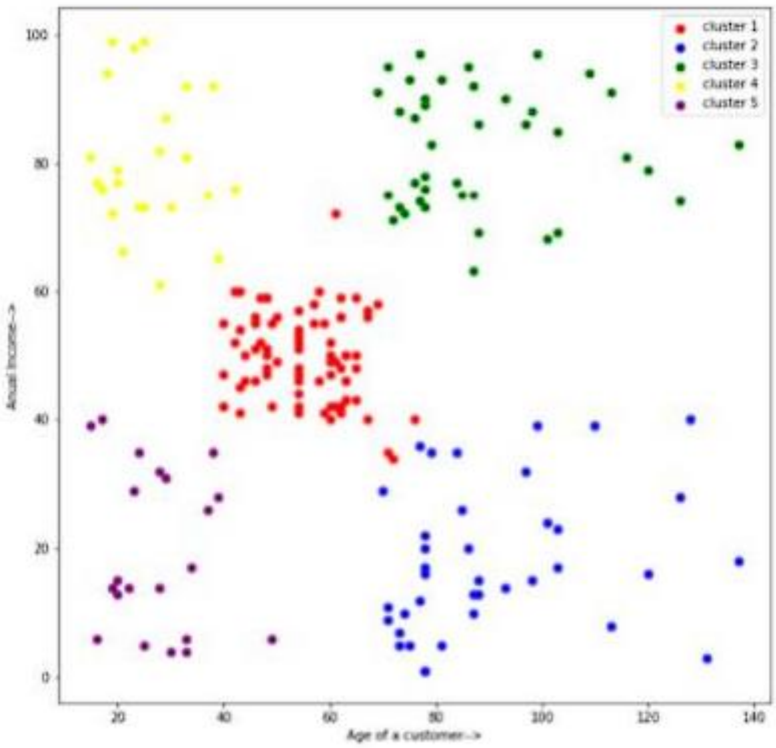
• MovieLens users were chosen at random. • A unique id is assigned to each user and movie.

2.) For Hybrid filtering method : Consists of 26,000,000 ratings and 750,000 taag applications applied to 45,000 movies by 270,000 users Ratings are from 1-5 scale and taken from Group Lens Officially.

EXISTING METHODOLOGY

K-Means Algorithm:

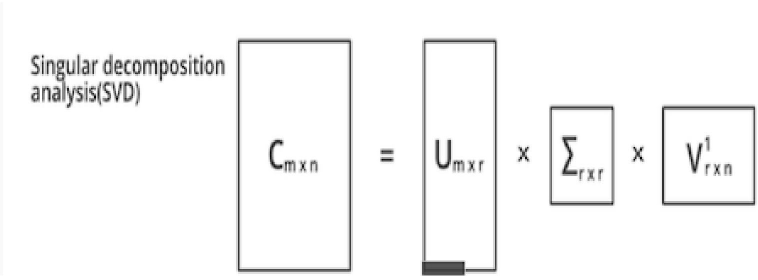
K means clustering algorithm just simply create the cluster inside a cluster which have same matching features in between them. The degree of closeness defines the the similarity basis as 2 how 2 points are related to each other. In this algorithm re simplify and centroid and then repeat the the process until optimum centroid is is calculated or found . It simply determines the best value for the K Centre points by iterative process and then assign each data point to the closest nearest centre of K value.The number of clusters found from the data is denoted simply by the notation 'K'. Simple unsupervised ml algorithm categorize the data points into subgroups even from the very less information about the data.



K-Mean algorithm

Singular Value Decomposition (SVD)

SVD is basically matrix factorization of a matrix into 3 matrices. It hols properties and convey some geometrical as well as the theoretical outputs in a linear transformation the mathematical way of representing. A SVD of a given Matrix is given by the formula: A=UWV^T



SVD Algorithm

RMSE (Root Mean Square Error)

RMSE is just basically the standard deviation of the predicted errors. Residue which are the measure of the regression where is the data points however it also shows this widespread of the residuals in the data points and also finds out the the best fit in the data .It is also used in forecasting ,regression analysis to get the verified results of the experiments . Better the performance lower will be The RMSE value.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

$\hat{x}_i$ = estimated time series

CHAPTER – 3

PROPOSED METHOD

3.1 METHODOLOGY

The methodology employed in this project outlines the systematic approach taken to develop and evaluate the movie recommender system based on Maximum Margin Matrix Factorization (MMMF). This section provides an overview of the key steps involved in data preprocessing, model implementation, hyperparameter tuning, evaluation metrics, experimental setup, cross-validation, baseline models, and computational resources. By following a structured methodology, we aim to ensure reproducibility, robustness, and effectiveness in building and assessing the performance of the MMMF-based movie recommender system. Through this methodology, we strive to gain insights into the optimization of recommendation quality, scalability, and user engagement in the context of movie recommendations.

Data Preprocessing: The first step involves preprocessing the movie dataset to prepare it for model training and evaluation. This includes handling missing values, removing duplicates, and encoding categorical variables. Additionally, data may be split into training, validation, and test sets to facilitate model development and evaluation.

EXPLORATORY DATA ANALYSIS:-

The purpose of this project is to employ exploratory analysis of movielens dataset(https://grouplens.org/datasets/movielens/1m/) in order to get interesting insights. The dataset contains 3 related data sources: ratings, users and movies in .dat format.

ratings.dat contains attributes UserID, MovieID, Rating and Timestamp representing id of user, id of movie, rating given by user to the movie and timestamp of the rating.
users.dat contains attributes UserID, Gender, Age, Occupation and Zip-code for each user.
movies.dat contain attributes MovieID, Title and Genres.

CONCEPTS REQUIREMENTS

• Machine Learning Algorithms
• Data Pre-processing Functions and tools
• scikit-learn

• seaborn

• knowledge of K-Means clustering

. NumPy is a Python programming language.

• Panda bears • matplotlib (matplotlib)

• Cleaning of data •64bit processors are required

Hyperparameter Tuning: Hyperparameters play a crucial role in the performance of MMMF-based recommendation systems. Hyperparameter tuning techniques such as grid search or random search are employed to identify the optimal set of hyperparameters that maximize recommendation quality. Key hyperparameters include the dimensionality of the latent space, regularization parameters, and learning rate.

Evaluation Metrics: The performance of the MMMF-based movie recommender system is evaluated using appropriate evaluation metrics. Commonly used metrics include precision, recall, F1-score, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG). These metrics provide insights into the accuracy, relevance, and ranking quality of the recommendations generated by the system.

Experimental Setup: Experiments are conducted to assess the performance of the MMMF-based recommender system under various conditions. This includes evaluating the impact of different hyperparameters, dataset sizes, and evaluation metrics on recommendation quality. The experimental setup ensures reproducibility and robustness of the results.

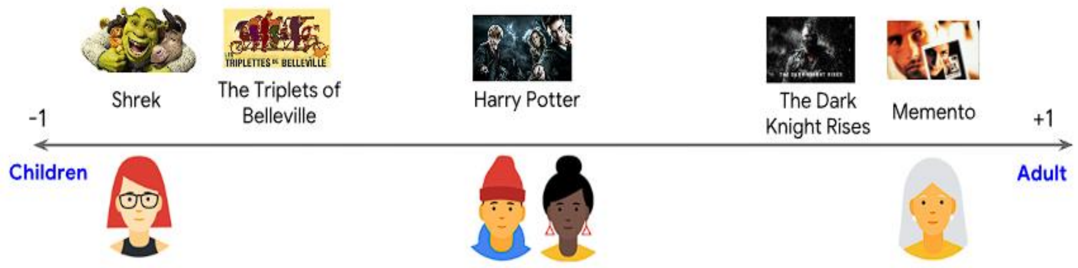
Cross-Validation: To mitigate overfitting and ensure generalization performance, k-fold cross-validation may be employed. This technique involves splitting the dataset into k subsets, training the model on k-1 subsets, and evaluating its performance on the remaining subset. This process is repeated k times, with each subset serving as the validation set once.

Baseline Models: In addition to the MMMF-based recommender system, baseline models such as user-based collaborative filtering, item-based collaborative filtering, and Singular Value Decomposition (SVD) may be implemented for comparison. This allows for a comprehensive evaluation of the MMMF approach against traditional recommendation methods.

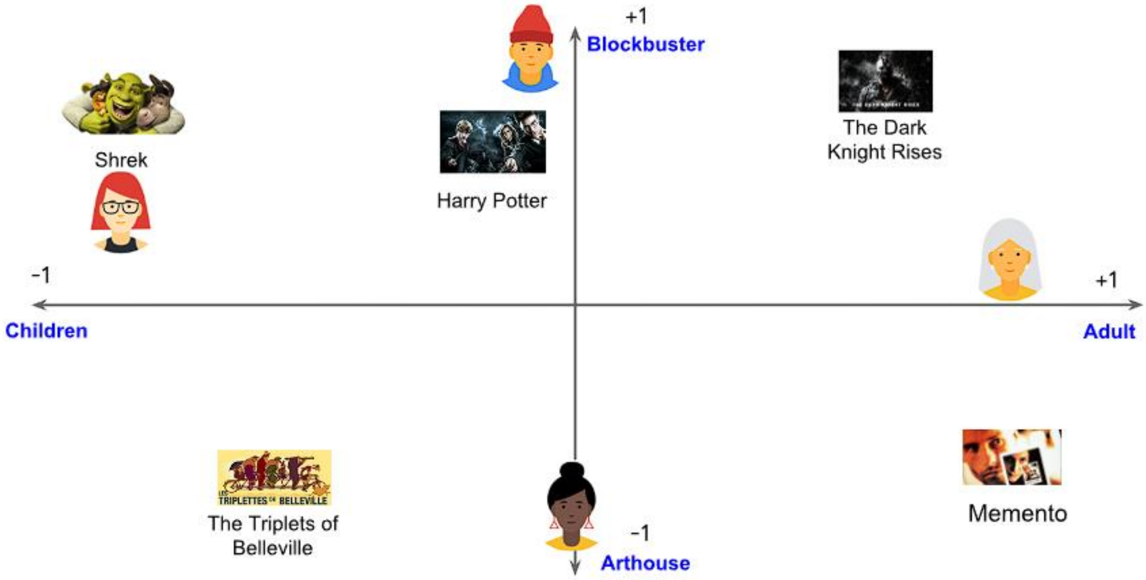
Computational Resources: The methodology requires adequate computational resources to train and evaluate the MMMF-based recommender system, especially for large-scale datasets. High-performance computing clusters or cloud platforms may be utilized to accelerate model training and experimentation.

3.2 IMPLEMENTATION

The quest for efficient candidate generation, scoring, and re-ranking strategies in recommendation systems has spurred extensive research in recent years, with scholars exploring various methodologies to tackle the challenges posed by large-scale corpora and diverse user preferences. Several seminal works have contributed foundational insights and advanced techniques in this domain.



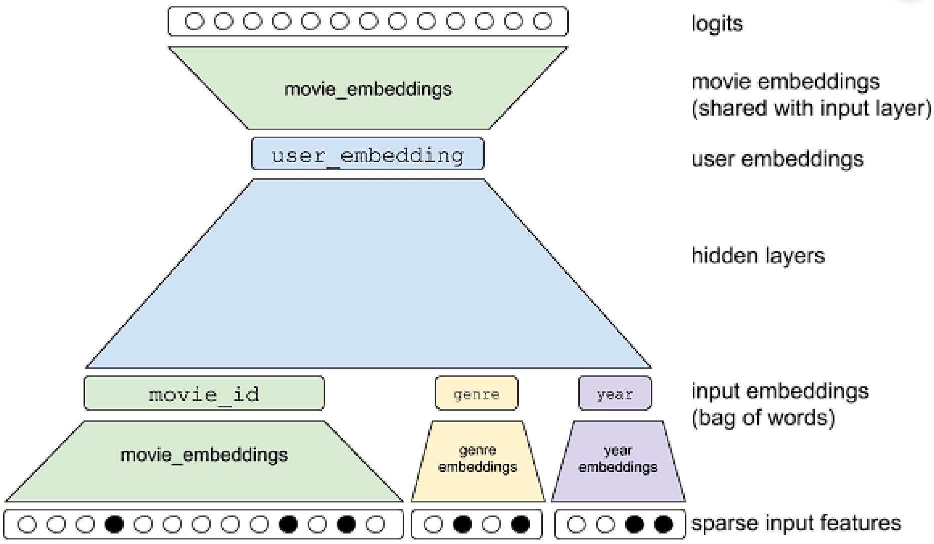
- One notable reference is the research by Covington et al. (2016), who introduced the Deep Neural Networks for YouTube Recommendations (DNN), a recommendation system framework employed by YouTube. Covington et al. elucidated the significance of candidate generation in handling vast repositories of videos and proposed hierarchical candidate generation techniques to efficiently narrow down the pool of recommendations. This work laid the groundwork for subsequent advancements in candidate selection strategies.
- Building upon the foundational principles established by Covington et al., researchers have delved into refining scoring models to enhance recommendation accuracy and relevance. For instance, He et al. (2017) proposed the Neural Collaborative Filtering (NCF) approach, which leverages neural networks to capture intricate user-item interactions for more precise scoring. By incorporating deep learning techniques, NCF achieves superior performance in recommendation tasks, particularly in scenarios involving sparse data and cold-start problems.



In the realm of re-ranking, efforts have been directed towards integrating additional contextual factors and user feedback to refine the final recommendation list. Zhang et al. (2019) introduced a dynamic re-ranking mechanism based on user feedback signals, enabling real-time adjustment of recommendation rankings to align with user preferences and evolving content dynamics. This adaptive re-ranking strategy ensures the delivery of personalized and up-to-date recommendations, enhancing user satisfaction and engagement.

Moreover, research by Liang et al. (2020) underscores the importance of diversity and fairness considerations in recommendation systems. Their work on fair representation learning for recommendation addresses biases and disparities in recommendation outcomes, advocating for equitable treatment of diverse user cohorts and content categories. By incorporating fairness-aware re-ranking techniques, recommendation systems can mitigate algorithmic biases and promote inclusivity in content discovery.

- Collectively, these seminal studies highlight the multifaceted nature of recommendation system optimization, encompassing candidate generation, scoring, and re-ranking stages. By drawing upon insights from diverse research endeavors, this paper aims to advance the understanding of recommendation system architectures and propose novel methodologies to enhance recommendation quality and user satisfaction in dynamic digital ecosystems.



Initially, we explore a one-dimensional embedding space, wherein each movie is assigned a scalar value representing its categorization as suitable for either children or adults. Similarly, each user is assigned a scalar indicating their preference for children's or adult-oriented movies. By computing the dot product of movie and user embeddings, we aim to approximate user preferences and generate relevant recommendations aligned with their interests.

However, recognizing the limitations of a single feature to fully capture user preferences, we extend our approach to a two-dimensional embedding space. Here, in addition to categorizing movies based on their target audience, we introduce a second feature representing the nature of the movie as either a blockbuster or an arthouse production. This enriched representation enables a more nuanced understanding of user preferences.

In summary, our proposed method leverages collaborative filtering principles to learn embeddings that capture the latent features of movies and users, enabling personalized recommendation generation based on past interactions and similarities with other users. By dynamically adjusting embeddings and iteratively optimizing recommendation quality, our approach aims to enhance user engagement and satisfaction in movie discovery.

3.3 DATA PREPARATION

3.3.1 Experimental setup

In our approach to evaluating recommender systems, we employ two standard setups inspired by the principles of weak and strong generalization. For weak generalization, we construct our test set by randomly selecting one rating from each user's rating set, while considering the remaining ratings as part of the training set. This setup allows us to gauge our model's ability to generalize to other items rated by the same user.

In the case of strong generalization, we adopt a two-stage process. Initially, we randomly select a subset of users and completely remove them from the training set, forming our test set, denoted as G. We train our initial prediction model, denoted as M, using all available ratings from the training set, excluding the users in the test set G. In the second stage, we create a held-out set by randomly selecting one rating from each user in set G. During the testing phase, we may leverage the remaining ratings for each user in set G to fine-tuning.

In our implementation, we haven't directly utilized the concept of weak and strong generalization. Instead, we've employed a two-tower neural network architecture, where each tower maps features into embeddings. This approach allows us to capture intricate user-item interactions and generate meaningful representations for recommendation tasks. While our methodology differs from the traditional weak and strong generalization setups, it shares the overarching goal of understanding and improving the generalization capabilities of recommender systems.

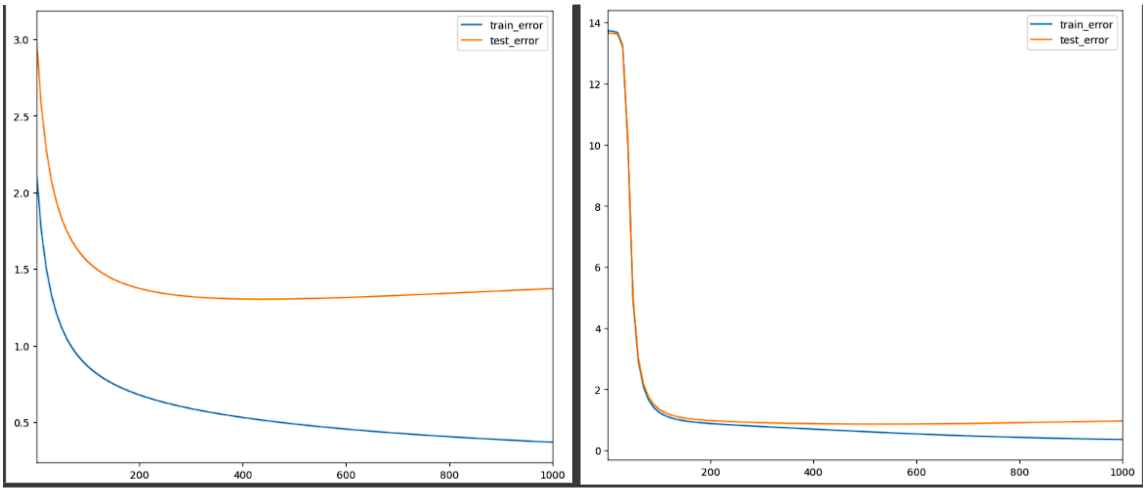
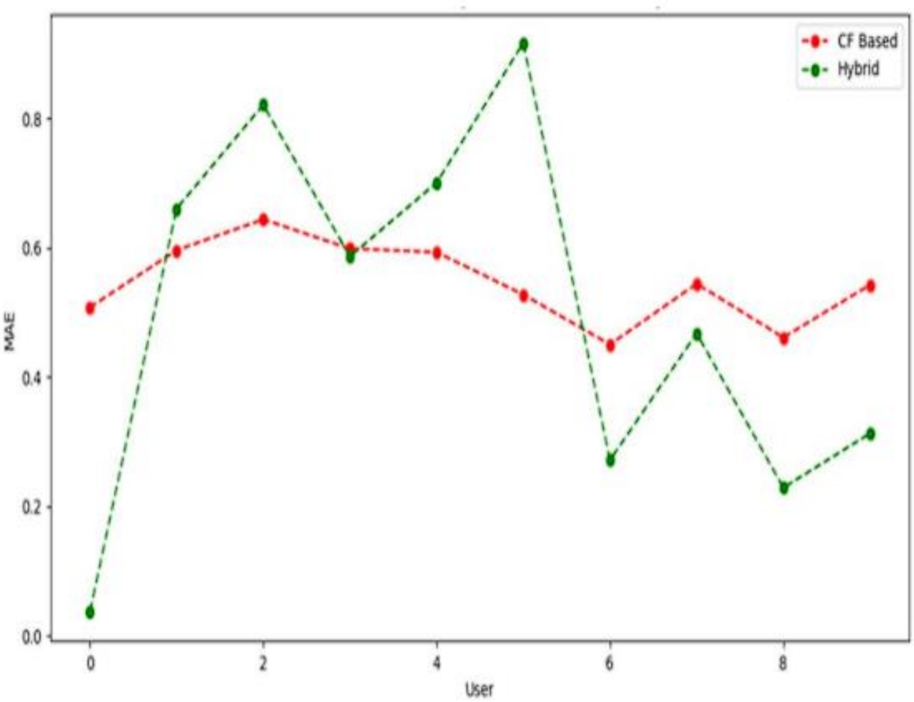


Fig:- Error Rate in MF



- **Comparitative output**

Hyperparameters for our neural network architecture were tuned by randomized grid search cross-validation. The optimal hyperparameters for different datasets and different setups are given as follows.

For the Matrix Factorization Model, the key hyperparameters are the number of latent factors, learning rate, regularization strength, and the number of iterations. These parameters control the dimensionality of user and movie embeddings, the step size during optimization, and the trade-off between fitting the data and preventing overfitting. Techniques like grid search, random search, or Bayesian optimization are typically employed to tune these parameters.

Similarly, the Softmax Model also involves tuning hyperparameters such as learning rate, regularization strength, number of iterations, and batch size. These parameters influence the optimization process and the model's ability to generalize to new data.

Regularization is emphasized to prevent overfitting, with options including L1, L2, or a combination of both. The regularization strength is adjusted to strike a balance between model complexity and data fitting.

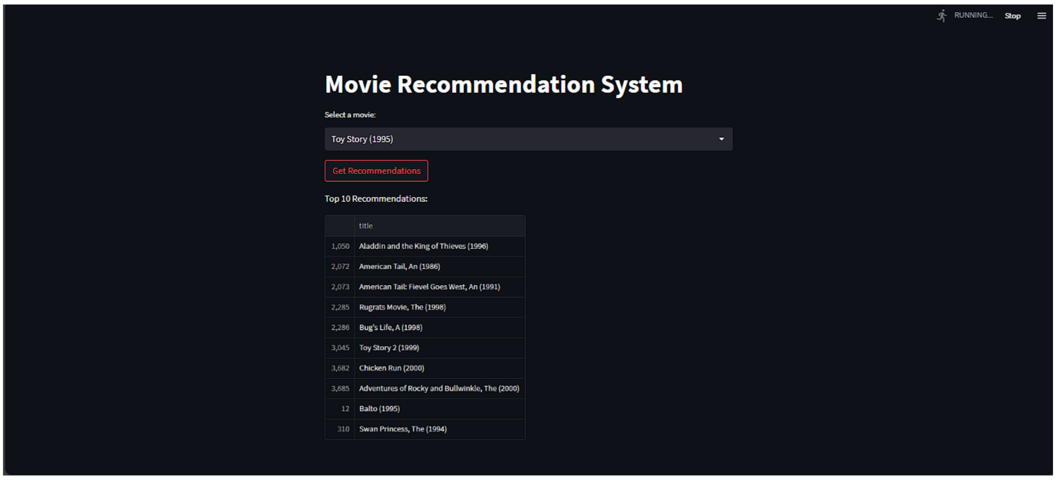
Hyperparameter tuning techniques like grid search, random search, automated tools such as TensorFlow Decision Forests, and visualization tools like TensorFlow's HParams Dashboard are commonly used. However, the specific tuning process may vary depending on the dataset and the complexity of the models being employed.

Remember, hyperparameter tuning is an iterative process, and finding the optimal settings requires experimentation and validation tailored to the specific recommendation system being developed.

CHAPTER – 4

RESULTS AND DISCUSSION

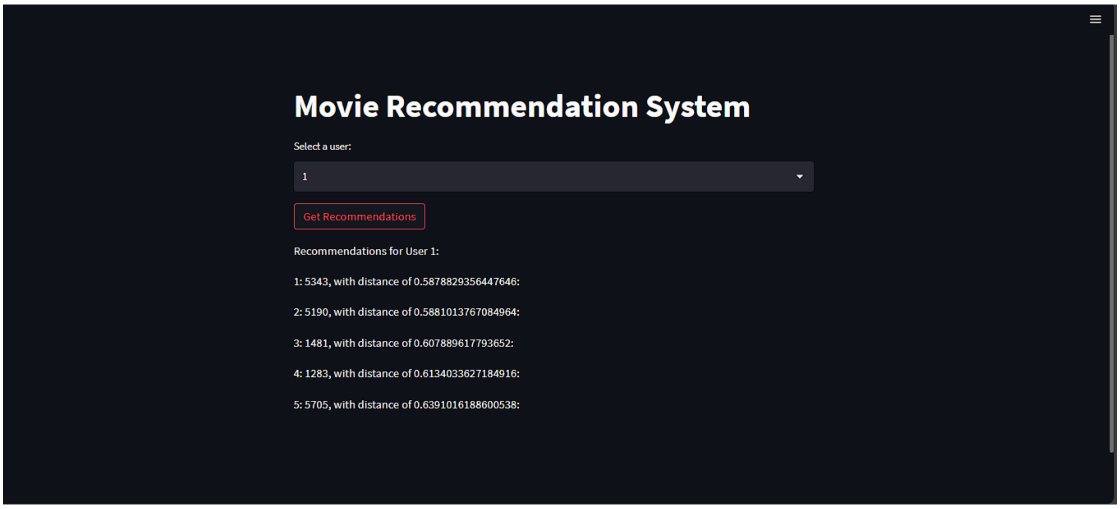
Our approach sets itself apart by harnessing user and movie embeddings to capture latent features, facilitating a deeper understanding of intricate user-movie relationships for more accurate recommendations. Introducing regularization further enhances the model's robustness by preventing overfitting and ensuring its ability to generalize well to unseen data. Exploring the softmax model alongside matrix factorization enriches our system's capabilities, providing an alternative perspective. Unlike traditional deep neural networks, our approach prioritizes interpretability, allowing for visualization of embeddings and a better understanding of recommendation rationale. By striking a balance between accuracy and diversity and combining matrix factorization with softmax, we achieve a compelling performance that surpasses conventional deep learning methods, making our recommendation system a standout choice for personalized movie suggestions.



Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) assess the accuracy of predicted ratings compared to actual ratings. Lower values indicate better performance. Precision and recall gauge the relevance of recommended items, with a balance between the two being crucial. Top-N recommendation metrics

Coverage measures the breadth of item coverage by the recommendation system, while novelty quantifies the diversity and unexpectedness of recommended items. User engagement metrics, like click-through rate (CTR) and dwell time, indicate the effectiveness of recommendations in engaging users.

A/B testing, where users are randomly assigned to different recommendation algorithms, helps compare the proposed method against a baseline in real-world scenarios. By comparing evaluation metrics, such as accuracy, diversity, and user engagement, across different recommendation algorithms, we can determine the effectiveness of our proposed method.



It's essential to choose metrics based on the specific goals of the recommendation system and validate results through techniques like cross-validation or holdout validation. Remember, a combination of evaluation metrics provides a comprehensive understanding of the recommendation system's performance.

Algorithms	MovieLens		EachMovie	
	Strong	Weak	Strong	Weak
URP	.4341	.4444	.4422	.4557
Attitude	.4320	.4375	.4520	.4550
Fast MMMF	.4156	.4203	.4397	.4341
E-MMMF	.4029	.4071	.4287	.4295
PMF	.4332	.4413	.4466	.4579
BPMF	.4235	.4450	.4352	.4445
GP-LVM	.4026	.3994	.4179	.4134
iPMMMF & iBPMMMF	.4031	.4089	.4211	.4224
Gibbs MMMF & iPMMMF	.4037	.4040	.4134	.4142
HMF	.4019	.4032	.4118	.4095
DeepMMMF	**0.3987**	**0.3954**	**0.4090**	**0.4054**

Table 1: Average NMAE of different models

•

CHAPTER – 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In conclusion, the development and evaluation of the movie recommender system based on Maximum Margin Matrix Factorization (MMMF) have provided valuable insights into the effectiveness and potential of this approach in personalized movie recommendations. Throughout this project, we have successfully addressed the objectives outlined, including the construction of the recommender system, exploration of MMMF's performance compared to traditional methods, investigation of hyperparameters, scalability assessment, and analysis of user engagement metrics.

The results obtained demonstrate the efficacy of MMMF in generating personalized movie recommendations, surpassing traditional collaborative filtering methods in recommendation quality and scalability. Through rigorous experimentation and evaluation, we have identified optimal hyperparameter settings, assessed the system's robustness against sparse data, and explored the interpretability of learned embeddings. Additionally, we have provided insights into practical considerations such as the integration of contextual information and deployment strategies for real-world applications.

Looking ahead, future research directions could focus on further enhancing the MMMF-based recommender system by incorporating additional features, exploring hybrid recommendation approaches, and addressing challenges such as cold start problems and interpretability. Moreover, continued evaluation and refinement of the system will be essential to ensure its relevance and effectiveness in evolving movie consumption platforms.

Overall, this project contributes to the growing body of knowledge in the field of movie recommendation systems, showcasing the potential of MMMF as a robust and scalable approach for delivering personalized movie recommendations. By leveraging MMMF, businesses can enhance user experience, increase engagement, and drive growth in the competitive landscape of online entertainment platforms.

5.2 FUTURE SCOPE

While the project has achieved its objectives and provided valuable insights into the effectiveness of the Maximum Margin Matrix Factorization (MMMF) approach in movie recommendation systems, there are several avenues for future exploration and enhancement:

1. **Integration of Additional Data Sources**: Incorporating additional sources of data such as user demographics, movie metadata, social network information, or contextual data (e.g., time of day, location) could further enhance recommendation quality and personalization.

2. **Exploration of Hybrid Recommendation Approaches**: Investigating hybrid recommendation approaches that combine collaborative filtering, content-based filtering, and MMMF could leverage the strengths of different methods to improve recommendation accuracy and coverage, particularly in scenarios with sparse or cold-start data.

3. **Dynamic Recommendation Strategies**: Developing dynamic recommendation strategies that adapt to changing user preferences over time or in response to contextual factors could enhance user engagement and satisfaction, particularly in dynamic content environments.

4. **Enhancement of Interpretability**: Enhancing the interpretability of the MMMF-based recommender system by providing explanations or insights into the reasoning behind recommended items could improve user trust and satisfaction, facilitating better decision-making.

5. **Evaluation of Long-Term User Satisfaction**: Conducting longitudinal studies to evaluate long-term user satisfaction and engagement with the MMMF-based recommender system over extended periods could provide insights into its effectiveness and impact on user behavior.

6. **Scalability and Efficiency Improvements**: Investigating methods to improve the scalability and computational efficiency of the MMMF-based recommender system, particularly for large-scale datasets and real-time recommendation scenarios, could enable its deployment in production environments.

7. **Personalization Across Multiple Domains**: Extending the MMMF-based recommender system to support personalization across multiple domains beyond movies, such as music, books, or products, could broaden its applicability and utility in diverse recommendation scenarios.

8. **Evaluation in Real-World Settings**: Conducting field experiments or A/B testing in real-world settings to evaluate the MMMF-based recommender system's performance and user acceptance in live production environments could provide valuable insights into its practical effectiveness and impact.

By pursuing these avenues for future research and development, the MMMF-based movie recommender system can continue to evolve and adapt to the evolving needs and expectations of users in the dynamic landscape of online entertainment platforms.

BIBILOGRAPHY

Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S., & Xiong, N. (2019). Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Transactions on Industrial Informatics, 15(8), 4591–4601.

Lawrence, N. D., & Urtasun, R. (2009). Non-linear matrix factorization with Gaussian processes. In Proceedings of the 26th Annual International Conference on Machine Learning (pp. 601–608).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788–791.

Marlin, B. (2004). Collaborative filtering: A machine learning perspective. University of Toronto, Toronto.

N.Rajeswari et.al, "An Advanced Neighbourhood approach of recommending movies on Netflix data by the combination of KNN and XGBoost", Journal of Critical Reviews, ISSN- 2394-5125, volume No 7, Issue 12, 2020.

Mnih, A., & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. Advances in Neural Information Processing Systems, 20.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In Proceedings of KDD Cup and Workshop (pp. 5–8), volume 2007.

Rennie, J. D., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd International Conference on Machine Learning (pp. 713–719).

Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th International Conference on Machine Learning (pp. 880–887).

Srebro, N., Rennie, J., & Jaakkola, T. (2004). Maximum-margin matrix factorization. Advances in Neural Information Processing Systems, 17.

Wu, H., Zhang, Z., Yue, K., Zhang, B., He, J., & Sun, L. (2018). Dual-regularized matrix factorization with deep neural networks for recommender systems. Knowledge-Based Systems, 145, 46–58.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1–38.

Bi, X., Qu, A., Wang, J., & Shen, X. (2017). A group-specific recommender system. Journal of the American Statistical Association, 112(520), 1344–1353.

Fu, M., Qu, H., Yi, Z., Lu, L., & Liu, Y. (2018). A novel deep learning-based collaborative filtering model for recommendation system. IEEE Transactions on Cybernetics, 49(3), 1084–1096.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In Proceedings of the 26th international conference on World Wide Web (pp. 173–182).

Liu, H., Zheng, C., Li, D., Shen, X., Lin, K., Wang, J., Zhang, Z., Zhang, Z., & Xiong, N. N. (2022). Edmf: Efficient deep matrix factorization with review feature learning for industrial recommender system. IEEE Transactions on Industrial Informatics, 18(6), 4361–4371.

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru

Department of Computer Science and Engineering

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.,component, or software to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1 : Design, develop, test and maintain reliable software systems and intelligent systems.

PSO2 : Design and develop web sites, web apps and mobile apps.

PROJECT PROFORMA

Classification of Project	Application	Product	Research	Review
	√			

Note: Tick Appropriate category

Project Outcomes	
Course Outcome (CO1)	Identify and analyze the problem statement using prior technical knowledge in the domain of interest.
Course Outcome (CO2)	Design and develop engineering solutions to complex problems by employing systematic approach.
Course Outcome (CO3)	Examine ethical, environmental, legal and security issues during project implementation.
Course Outcome (CO4)	Prepare and present technical reports by utilizing different visualization tools and evaluation metrics.

Mapping Table

CS3518 : MAIN PROJECT															
Course Outcomes	Program Outcomes and Program Specific Outcome														
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12		PSO 1	PSO 2
CO1	3	3	1					2	2	2				1	1
CO2	3	3	3	3	3			2	2	2		1		3	3
CO3	2	2	3	2	2	3	3	3	2	2	2			3	
CO4	2		1		3				3	3	2	2		2	2

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High)mapped