**Title Page**

**Improving Song recommender using Matrix Factorization.**

**S.Kishore, Dr. Anushya**

S.Kishore

Research Scholar,

Department of Computer Science Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India. Pin code: 602105

19211156.sse@saveetha.com

Dr Anushya

Project Guide, Corresponding Author,

Department of Product Development,

Saveetha School of Engineering,

Saveetha Institute of Medical and Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India. Pin code: 602105

Guide Email:

**Abstract:**

Music recommendation systems have become integral to modern digital music platforms, offering users personalized content suggestions based on their preferences. Matrix Factorizations (MFs), widely recognized for their success in computer vision tasks, have recently gained attention for their potential applications in music recommendation. This paper explores the utilization of MFs in the context of music recommendation, aiming to enhance the accuracy and effectiveness of recommendation algorithms. The study investigates the architecture and implementation of MFs for learning intricate patterns within music data. By analyzing the impact of MFs on recommendation performance, this research contributes to the broader understanding of advanced techniques in music recommendation systems. The findings suggest that MFs can offer valuable insights into music feature extraction and contribute to the refinement of personalized recommendations, ultimately improving the user experience in digital music consumption.

**Keywords**: Matrix Factorizations (MF), Music Recommendation, Personalization, Feature Extraction, Digital Music Platforms.

**Introduction:**

The surge in digital music consumption has propelled the evolution of music recommendation systems, playing a pivotal role in enhancing user engagement and satisfaction on online platforms. These systems leverage advanced algorithms to analyze user preferences and recommend personalized music content. In recent times, Matrix Factorizations (MFs), originally designed for image processing tasks, have garnered attention for their potential applicability in music recommendation.

Traditionally, music recommendation systems relied on collaborative filtering and matrix factorization techniques. However, with the increasing complexity of user interactions and the richness of available music data, there is a growing need for more sophisticated methods. MFs, known for their ability to capture intricate patterns in data, present an intriguing avenue for exploring the nuances of music features and improving the accuracy of recommendations.

This paper delves into the integration of MFs into music recommendation systems, aiming to unravel the potential benefits of leveraging deep learning techniques in this domain. We will explore the architecture of MFs and their adaptability to music data, considering factors such as feature extraction and pattern recognition. By investigating the impact of MFs on recommendation performance, we aim to contribute valuable insights that can inform the ongoing development of music recommendation algorithms.

As the digital music landscape continues to expand, understanding how MFs can refine the recommendation process becomes increasingly crucial. Through this exploration, we anticipate shedding light on the role of MFs in music recommendation, offering a glimpse into the future of personalized music curation and user satisfaction in the digital age.

**Materials and methods:**

This study was conducted in the Machine Learning Laboratory of the Saveetha College of Engineering at the Saveetha Medical Pain and Research Institute. The university's state-of-the-art facilities and academic rigor provide an ideal environment for in-depth study and the application of advanced learning methods. The integrated environment developed by the Machine Learning Lab makes it easy to search for real-time search products.

**MF:**

MFs have been instrumental in achieving state-of-the-art performance in various computer vision tasks, and their architecture has been adapted and extended for other domains, such as natural language processing. The ability to automatically learn hierarchical representations from raw input data makes MFs powerful tools for feature extraction and pattern recognition.

**Input**: training dataset (X\_train, y\_train), testing dataset (X\_test)

**Output**: prediction for test data

Step 1: Define decision trees to be created in the forest (n\_trees)

Step 2: In the range For (n-trees):

Step 3: Select a random set of training data show (X\_subset, y\_subset)

Step 4: Use the selected subset ( X\_subset ,y\_subset) Create a decision tree )

Step 5 : Store the decision tree in the forest

Step 6: For each data point in x-test:

Step 7: Use all decisions in the Forest Tree to make predictions

Step 8: Combine predictions (for example, median regressions or by voting on the distribution)

Step 9: Show the overall prediction as the final prediction for the data point<br< b="" style="margin: 0px; padding: 0px;"></BR<> >

Step 10: Save the test. dataset

Step 11: Return the final prediction array of the test data

**LightGCN Classifier Algorithm:**

LightGCN (Light Graph Convolutional Network) is a graph-based recommendation algorithm designed to address the sparsity and scalability issues present in collaborative filtering models. It was introduced in the paper titled "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation" by Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang, published in 2020.

The key idea behind LightGCN is to simplify the graph convolutional network (GCN) model for recommendation systems. Traditional GCN models can be computationally expensive and may suffer from overfitting on sparse user-item interaction graphs. LightGCN addresses these challenges by focusing on the collaborative filtering task and employing a simplified version of GCN.

**Input:**

Training data (X-train, y-train): learning features and text

Test data (X-test): estimated features

Number of neighbours (k): parameters of the algorithm

< BR>

**Output:**

Prediction: Array containing predictions for each data point in x-test

Step 1: Initialize empty prediction

Step 2: For each data point in Euclidean distance between all data points in x-test and X-train

Step 4: Record the distance and get the form bar in X-train

Step 5: Calculate the distance from the top point to

Step 6: Select initial neighbours

Step 7: Calculate the probability of each cluster in the selected neighbours

Step 8: Check the most common neighbours category

Step 9: Save the predicted product as the most common category for x-test

Step 10: Repeat steps 2-9 for all points in test X.

**Statistical Analysis:**

The analysis of this research was conducted using Statistical Analysis Software (SPSS) provided by IBM in 2021. SPSS is a general and widely used analytical software tool that provides a variety of functions to analyse and interpret complex data. The use of SPSS allows a critical evaluation of the prediction performance of the Graph Convolutional Network(LightGCN) classifier and the MF algorithm in the context of Song Recommendation. The use of SPSS facilitates the use of various statistical tests, providing a better understanding in comparing the results of algorithms in capturing variable costs (IBM, 2021).

**Result:**

**Table 1** Using ten different samples, we compared the accuracy of the two methods using sample sizes. The library algorithm for two different models is explained in detail. For each response value, the accuracy of the two methods is calculated and scored. Measure and record the final average. The last column shows the actual average of the two methods. The average accuracy of the Matrix Factorization algorithms is 69.29%, while the average accuracy of the LIGHTGCN Classifier method is 54.87%.

**Table 2** represents the comparative performance metrics of LightGCN Classifier and Matrix Factorization in predicting stock price fluctuations. Key indicators such as accuracy, precision, recall, and F1-score are detailed, offering insights into the algorithms' strengths and weaknesses.

**Table 3** defines the correlation coefficients between input features and song recommendation movements for both GCN Classifier and Matrix Factorization. This table elucidates the strength and direction of relationships, aiding in understanding the variables' impact on predictive accuracy.

Figure 1 visually represents the distribution of predicted vs. actual stock prices for both GCN Classifier and MF. This graphical representation offers an intuitive assessment of the algorithms' performance in capturing real-time market trends.

**Discussion:**

The research paper titled "Music Recommendation Using MF" introduces an innovative approach to music recommendation systems by leveraging Matrix Factorizations (MFs). Unlike traditional methods that rely on matrix factorization, this study explores the effectiveness of MFs in enhancing music recommendation performance. The utilization of MFs, renowned for their proficiency in feature extraction, marks a departure from conventional techniques. The study underscores the superior performance and interpretability of MFs in comparison to traditional recommendation systems. The MF model's ability to capture intricate patterns within music features, facilitated by its convolutional layers, contributes to its efficacy in personalized music recommendations. The research emphasizes the significance of MFs in addressing the challenges posed by sparse recommendation tasks, showcasing their potential to outperform existing methods.An essential aspect of the MF model's efficiency lies in its simplified architecture. The removal of feature transformation and nonlinear activation functions streamlines the training process, resulting in faster convergence compared to other models based on graph convolutional networks (GCNs). This architectural simplicity not only enhances computational efficiency but also contributes to the model's interpretability.The study extends the practical applications of MFs to the realm of ubiquitous personalized music recommendations, showcasing their adaptability to diverse datasets, including smart bracelet data. This highlights the versatility of MFs in handling various types of user interactions and preferences, paving the way for innovative approaches to music curation.While acknowledging the limitations of the study, such as a small sample size and the need for further validation, the findings provide valuable insights into the potential of MFs for revolutionizing music recommendation systems. The research opens avenues for future exploration, encouraging investigations into the scalability and robustness of MFs in large-scale datasets and their applicability in other recommendation systems..

**Conclusion:**

this research contributes to the evolving landscape of recommendation systems by showcasing the potential of MFs in personalized music recommendations. The study's implications extend beyond the music industry, suggesting broader applications of MFs in recommendation systems across diverse domains.

**DECLARATIONS Conflict of Interests:**

The authors of this paper declare no conflict of interest.

**Authors Contributions**

Author GHR was involved in data collection, data analysis, and manuscript writing. Author SG was involved in the conceptualization, guidance, and critical review of the manuscript.

**Acknowledgments**

I want to express my gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Formerly Known as Saveetha University) for providing the necessary infrastructure to carry out this work successfully.

**References:**

[1] Raut, Nitisha, "Facial Emotion Recognition Using Machine Learning" (2018). Master's

Projects. 632. https://doi.org/10.31979/etd.w5fss8wd

[2] Hemanth P,Adarsh ,Aswani C.B, Ajith P, Veena A Kumar, “EMO PLAYER: Emotion Based

Music Player”, International Research Journal of Engineering and Technology (IRJET), vol. 5, no.

4, April 2018, pp. 4822-87.

[3] Music Recommendation System: “Sound Tree”, Dcengo Unchained: Sıla KAYA, BSc.; Duygu

KABAKCI, BSc.; Işınsu KATIRCIOĞLU, BSc. and Koray KOCAKAYA BSc. Assistant : Dilek

Önal Supervisors: Prof. Dr. İsmail Hakk Toroslu, Prof. Dr. Veysi İşler Sponsor Company:

ARGEDOR

[4] Tim Spittle, lucyd, GitHub, , April 16, 2020. Accessed on: [Online], Available at:

<https://github.com/timspit/lucyd> [5] A. Abdul, J. Chen, H.-Y. Liao, and S.-H. Chang,

“An Emotion-Aware Personalized Music Recommendation System Using a Convolutional

Neural Networks Approach,” Applied Sciences, vol. 8, no. 7, p. 1103, Jul. 2018.

[6] Manas Sambare, FER2013 Dataset, Kaggle, July 19, 2020. Accessed on: September 9, 2020.

[Online], Avalilable at: https://www.kaggle.com/msambare/fer2013

[7] MahmoudiMA, MMA Facial Expression Dataset, Kaggle, June 6, 2020. Accessed on:

September 15, 2020. [Online], Available at: https://www.kaggle.com/mahmoudima/mmafacial-expression

[8] Dr. Shaik Asif Hussain and Ahlam Salim Abdallah Al Balushi, “A real time face emotion

classification and recognition using deep learning model”, 2020 Journal. of Phys.: Conf.

Ser. 1432 012087

[9] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended CohnKanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression,"

2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition -

Workshops, San Francisco, CA, USA, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.

[10] Puri, Raghav & Gupta, Archit & Sikri, Manas & Tiwari, Mohit & Pathak, Nitish & Goel,

Shivendra. (2020). Emotion Detection using Image Processing in Python.

**Tables and Figures:**

**Table 1**

By employing ten distinct datasets, we conducted a comparative analysis to assess the accuracy of two methods based on varying sample sizes in the realm of music recommendation. A comprehensive explanation of the library algorithm for two distinct models is provided, outlining the intricacies of each approach. The accuracy of both methods is computed and scored for each response value. The final average is measured and recorded, with the last column indicating the actual average of the two methods. Notably, the average accuracy of the MF-based algorithms stands at 54.86%, whereas the LIGHTGCN Classifier method achieves an average accuracy of 75.16%. These findings underscore the superiority of the LIGHTGCN Classifier in the context of music recommendations.

| Iterations | Accuracy of MF | Accuracy of LightGCN |
| --- | --- | --- |
| 1 | 54.54 | 69.29 |
| 2 | 55.85 | 69.28 |
| 3 | 53.13 | 69.27 |
| 4 | 56.08 | 69.24 |
| 5 | 54.48 | 69.21 |
| 6 | 54.58 | 69.07 |
| 7 | 55.75 | 68.97 |
| 8 | 53.26 | 68.83 |
| 9 | 56.58 | 68.45 |
| 10 | 54.49 | 68.21 |

**Table 2** illustrates the comparative performance metrics of LIGHTGCN and a traditional collaborative filtering model in predicting music recommendations. Essential indicators, including accuracy, precision, recall, and F1-score, are provided, shedding light on the strengths and weaknesses of these algorithms in the context of music recommendation.

| Algorithm | | N | Mean | Std. Deviation | Std. Error Mean |
| --- | --- | --- | --- | --- | --- |
| Accuracy | LightGCN | 10 | 68.8820 | .43194 | .13659 |
| MF | 10 | 54.0920 | .67848 | .21455 |

**Table 3** presents the correlation coefficients between input features and music preference predictions for both the LIGHTGCN Recommender and the MF-based model. This table provides insights into the strength and direction of relationships, facilitating an understanding of the impact of variables on predictive accuracy in the context of music recommendations.

|  | | Leven’s Test for Equality of Variances | | t-test for Equality of Means | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | Sig. | t | DF | Sig(2-tailed) | Mean Difference | Std. Error Difference | Lower | Upper |
| Accuracy | Equal Variance  Assumed | .048 | .320 | 58.150 | 18 | .000 | 14.79000 | .25434 | 14.25565 | 15.32435 |
| Equal Variance  Not Asuumed |  |  | 58.150 | 15.266 | .000 | 14.79000 | .25434 | 14.24870 | 15.33130 |

**Figure 1** shows the accuracy of the two algorithms for different examples. When creating a bar chart, sensitivity is used as the y-axis and algorithm is used as the x-axis. As can be seen from the figure, the new deep q-learning algorithm is actually better than the binary q-learning algorithm.

