Recommendation System for Movies

Yashi Sharma (58), Nachiket Nasa (35), Khushi Jain (25), Purnima Kumar (41)
Department of Computer Science
University of Delhi

Under the supervision of
Dr. Ankit Rajpal
Assistant Professor
Department of Computer Science
University of Delhi

1. Abstract

Our project discusses the importance of explanations in recommendation systems. It introduces the Explainable Matrix Factorization (EMF) technique for collaborative filtering (CF) recommender systems, which aims to provide accurate and explainable recommendations without the need for additional data sources. The paper (section 3.5) also introduces new explanation quality metrics, Mean Explainability Precision (MEP) and Mean Explainability Recall (MER), to evaluate the quality of explanations in EMF.

2. Introduction

Recommender systems have become integral components of various online platforms, shaping user experiences and driving engagement. These systems employ algorithms to predict user preferences and recommend items or content tailored to individual interests. From e-commerce websites to streaming services and social media platforms, recommendation algorithms play a crucial role in guiding user choices, increasing engagement, and driving revenue. However, designing effective recommender systems requires a deep understanding of user behavior, algorithmic techniques, and evaluation methodologies.

Recommender systems can be categorized into several types based on the underlying algorithms and data sources. Content-based filtering recommends items similar to those previously liked by the user, while collaborative filtering suggests items based on the preferences of similar users. Hybrid recommender systems combine multiple approaches to overcome the limitations of individual methods, while knowledge-based recommender systems leverage domain knowledge to provide personalized recommendations.

Our project addresses the problem of information overload experienced by internet users, caused by the overwhelming amount of data available to them. This problem leads to difficulties in finding relevant information and can impact user experience. By introducing a Movie Recommender System, the research aims to personalize the search for movie recommendations according to user preferences and reduce information overload. This approach not only assists users in finding relevant content but also enhances the delivery of online content in various digital applications.

In our project we have studied 5 different techniques mainly used in recommendations systems namely :

1. Text Mining
2. K-Nearest Neighbor
3. Neural Network
4. Singular Value Decomposition
5. Explainable Matrix Factorization

Out of these 5 techniques, we have implemented Explainable Matrix Factorization for collaborative filtering recommender systems.

3. Related Work

3.1 Recommendation of Scientific Publications—A Real-Time Text Analysis and Publication Recommendation System by Midhad Blazevic, et. al

3.1.1 Background

When conducting research, scholars encounter the challenge of navigating extensive volumes of existing information to identify research gaps, extract insights, and further develop existing work. However, data is present in abundance within specific fields. It's difficult for scholars to process such a large amount of data considering factors like time constraints and the relevance of existing publications. To solve this problem, Midhad Blazevic et al. introduce a recommendation system that leverages real-time text analysis and similarity algorithms. This system assists researchers in making informed decisions regarding the relevance of existing publications to their work by recommending publications that facilitate successful research progress.

The main contributions of the paper are as follows:

1. The authors introduce a novel approach to recommend publications based on real-time text analysis and content-filtering.
2. The authors examine combinations of topic modeling and similarity algorithms to determine which ones provide adequate recommendations.
3. The authors examine the impact of text length on topic modeling.

3.1.2 Materials & Methods

Scientific publications, extracted from the Springer Nature API, and real-time text from a LaTeX editor serve as inputs. After discarding documents under 300 words, 92,651 publications are stored. Utilizing an NLP pipeline, preprocessing includes removing special characters, tokenization, stop word removal (plus LaTeX commands), and lemmatization. A document term matrix is generated, filtering out terms occurring too frequently or infrequently. Topic modeling identifies relevant documents based on user input. A similarity metric compares the relevant documents to the input, with the top 10 recommended to the user. Seven metrics (cosine similarity, Euclidean distance, etc.) and three topic modeling models (LSA, LDA, NMF) are available for user selection.

3.1.3 Results

Human judgment is used to rate the publications recommended by different combinations of topic models used with similarity metrics. 5 publications are selected and different combinations of text are experimented with for the user input. The system recommends 150 papers for each possible combination of topic model and similarity metric.
The major results are mentioned below:

1. Cosine similarity with LDA is the best approach with 60+ highly similar recommendations.
2. Canberra and Manhattan yield the worst results with all three models with 140+ useless recommendations each.
3. It's also observed that Cosine and Jensen combined with LDA, or LSA gives mostly useful results.
4. Improvement in results is seen when text length is increased.

3.1.4 Conclusion

The paper concludes by highlighting a research gap within recommendation systems in science and academia. Despite the potential benefits of recommendation systems, these supportive functions remain underutilized. And the novel approach presented in the paper is first of its kind.

3.2 Movie Recommender System Using K-Nearest Neighbors Variants by Sonu Airen, et. al

3.2.1 Background

The paper addresses the problem of information overload experienced by internet users, caused by the overwhelming amount of data available to them. This problem leads to difficulties in finding relevant information and can impact user experience. By introducing a Movie Recommender System, the research aims to personalize the search for movie recommendations according to user preferences and reduce information overload. This approach not only assists users in finding relevant content but also enhances the delivery of online content in various digital applications.

The paper proposes different variations of the K-nearest neighbors with various similarity measures, which enable personalized movie recommendations by finding similar users to an active user and utilizing their preferences. The system calculates similarities between users and items using the rating matrix and then combines ratings from neighboring users to generate personalized recommendations for the active user. By employing different KNN algorithms and similarity measures, the system aims to enhance the accuracy of movie recommendations and improve the user experience.

3.2.2 Materials & Methods

1. **Dataset**: The study utilized the ML-100K dataset from MovieLens for evaluating the performance of different KNN algorithms with various similarity measures. The dataset likely includes user-item ratings for movies, which served as the basis for building the recommendation system.
2. **Algorithm Implementation:** The section details the implementation of different variations of KNN algorithms, namely KNN-Basic, KNN-WithMeans, KNN-WithZScore, and KNN-Baseline. Each algorithm was integrated with different similarity measures such as cosine, msd, pearson, and pearson baseline to assess their impact on the recommendation accuracy.

3. **Evaluation Metrics:** Error measures like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and accuracy measures like Precision@k, and Recall@k were used to assess the effectiveness of the algorithms in generating movie recommendations.
4. **Experimental Setup:** The experiments were conducted using a five-fold cross-validation approach to ensure robust evaluation of the algorithms. The performance of each algorithm with different similarity measures was compared across varying numbers of nearest neighbors (K) to analyze their impact on recommendation quality.

3.2.3 Results

The results of the study on the Movie Recommender System using K-Nearest Neighbors variants are as follows:

1. **Optimized Value of K:** The study found that the error metrics stabilize after the neighborhood size of 40 neighbors. Therefore, the optimized value of K (number of nearest neighbors) for the dataset was determined to be 40.
2. **Error Metrics:** The RMSE values for the proposed KNN algorithms with similarity measures were all found to be less than 1.233, which was the RMSE value for an existing technique. This indicates that the proposed techniques outperformed the existing one in terms of RMSE .
3. **Limitations and Future Work:** The study acknowledged limitations such as performance being tested only on the ML-100K dataset and the potential memory and runtime implications for larger datasets. To improve the system, use better distance measures like Mahalanobis distance and incorporate other algorithms for enhanced performance.

3.2.4 Conclusion

The authors conclude that the system can significantly reduce information overload and improve the user experience by providing personalized movie recommendations. The paper not only addresses the problem of information overload but also offers a practical solution for enhancing the delivery of online content in digital platforms.

3.3 Fully Content Based movie recommendation system with feature extraction using neural network by Hung-Wei Chen, et. al

3.3.1 Background

Most movie recommender systems are based on collaborative filtering to recommend movies. The CF can predict user preferences by analyzing a user's browsing history and other user's preferences. However, the CF methods suffer from the cold start problem : it fails when no usage data of items is available. Another option we have is content-based filtering which does not have such drawbacks. The content-based filtering use some additional information about movies so that the system treats new movies just like old ones. In this paper authors proposed a method which recommended movies based on only content information (e.g. Directors, Actors, Genres, etc.) as their training data.

3.3.2 Materials & Methods

For the purpose of training textual data and extracting a set of vector features corresponding to every single word in the input data, Word2Vec model is used.

1. **Dataset & Preprocessing** : MovieLens-20M dataset for experiments. It contains about 27K movies, 138K users and 20 M ratings. Content data for 23k movies was retrieved from Open Movie Database (OMDb) API and stored into JSON files. Parts of movie metadata are treated as words (Movie Sentence Dataset) for input to the Word2Vec model. Feature extraction is based on the Continuous Bag-of-Words (CBOW) model which uses the Weighted Movie Sentence dataset as the training dataset.
2. **Similarity Measurement** : Semantic similarity between two words is measured by cosine similarity. Authors have proposed a method to measure similarity between two movies using movie sentences. There are two adjustable details regarding cosine similarity, first is how to choose the cosine similarity. The paper mentions two approaches, one is Same Metadata Only (SMO) and the other one is Fully-connected (FC).
   The total average similarity adds up all the cosine similarities and treats the averaged sum as the similarity score for two movie sentences.
3. **Recommendation List Generation** : The similarity between movies is measured and a recommendation list is generated for every movie in the dataset. This list is for movies and not the users, which means the list doesn't change itself no matter who uses the list.

3.3.3 Experiments

In order to measure the performance of the model Precision@K is used which shows the performance of the model in top K recommendation. High precision with lower K is considered to be a better system.
Two ways to calculate similarity : Fully-Connected (FC) and Same Metadata Only (SMO)
Two ways to calculate the average similarity : Total Average (TA) and the Metadata-Based (MB)
(D, A, G : Director, Actor and Genre)
The authors found DAGY_S_M to be the best combination with the best performance. This combination consists of Director, actor, genre, year, SMO as the parameter to calculate similarity and MB as the parameter to calculate the average similarity.

3.3.4 Conclusion

The authors conclude the paper by stating that the results of the experiments support the intuition behind the proposed method. Future works include trying to use more metadata as input features and use different ratings to weight the dataset instead of IMDb ratings. Also, combining the proposed process with collaborative filtering is a promising way to yield better results.

3.4 Recommendation Algorithm Based on Matrix SVD with Exponential Correction by Zhang et. al.

3.4.1 Background

The paper discusses improvements to the Singular Value Decomposition (SVD) algorithm used in collaborative filtering recommendation systems. Traditional SVD algorithms have limitations in accuracy due to the specificity of user-item matrices and iterative algorithm constraints. The paper aims to address these issues by implementing improved SVD algorithms, considering the time factor and exponential correlation and analyzing the performance of various algorithms.

3.4.2 Materials & Methods

1. **Dataset**: - This study utilized Movielens100K dataset, which includes more than 100,000 ratings from 1000 users on 1700 movies. 20% of each sub-data set is divided into the training set, and the remaining 80% belongs to the test set.

2. **Algorithm Implementation:** - 5 algorithms were considered in this study:
    a. <u>(Basic) SVD</u> - It is based on matrix decomposition to predict unknown elements in a user-item rating matrix. It uses user matrix **P** and a project matrix **Q** for training, aiming to fit the scoring matrix **R** as closely as possible. For unknown elements, the estimation is done using the inner product of **P** and **Q**.
    b. <u>RSVD</u> - It introduces bias terms for users and items to reflect individual differences and preferences. Regularization terms are included to prevent overfitting during the training process.
    c. <u>SVD++</u> - It uses implicit feedback information, such as user interactions with items, to predict unknown movie ratings. Estimation formula includes a factor vector associated with each item.
    d. <u>timeSVD++</u> - It modifies SVD++ model by including time-based biases, considering temporal dynamics of user preferences and item popularity.
    e. <u>timeSVD++EXP</u> - It applies an exponential decay function to model how the influence of past interactions diminishes over time. This reflects the idea that more recent interactions are more indicative of current preferences.

3. **Evaluation Metrics:** - Root Mean Squared Error (RMSE) is used as the metric for evaluating the goodness of a model.

3.4.3 Results

BasicSVD is used as a standard to compare the performance of each algorithm. RSVD and SVD++ algorithms bring accuracy improvements of 0.82% and 1.52% respectively. TimeSVD++ brings accuracy improvements of 3.37%. After the introduction of exponential modification, accuracy is further improved by 5.06% compared to the base algorithm.

As the number of hidden features increases, the prediction accuracy of each SVD algorithm improves, indicating more captured interactive information between users and items. However, there is a limit to the performance improvement from increasing hidden features.

3.4.4 Conclusion

The study's findings indicate that modifying the iterative approach of the Basic SVD algorithm enhances its efficacy, and the incorporation of temporal elements and exponential adjustments further amplifies this enhancement.

3.5 Explainable Matrix Factorization for Collaborative Filtering by Behnoush Abdollahi, et. al

3.5.1 Background

The paper addresses the importance of explanations in recommendation systems, particularly in Collaborative Filtering (CF) and Matrix Factorization (MF) models. Traditional explanation methods often require additional data sources, which may not be available for MF-based systems. To bridge this gap, the paper introduces Explainable Matrix Factorization (EMF), a technique that provides accurate and explainable recommendations solely based on rating data. EMF utilizes an explainability bipartite graph and introduces new metrics (Mean Explainability Precision (MEP) and Mean Explainability Recall (MER)) to evaluate the quality of explanations. This approach enhances user trust and recommendation validity without the need for extra data sources.

3.5.2 Materials & Methods

The method leverages a novel approach to incorporate explainability into MF models without requiring additional data sources. The study compares EMF with five baseline methods, including Non-Negative Matrix Factorization (NMF), Probabilistic Matrix Factorization (PMF), classical user-based and item-based top-n techniques, and non-personalized top-n most popular items.

The proposed Explainable Matrix Factorization (EMF) technique involves several key components:

- It leverages an explainability bipartite graph, where edges connect users to items based on the rating distribution within the active user's neighborhood. This graph captures the explainability of recommended items.
- The technique formulates a new objective function that incorporates explainability and accuracy, balancing the two aspects during optimization. By using stochastic gradient descent, the EMF approach updates the representations of users and items in the latent space to generate accurate and explainable recommendations.
- The proposed technique is evaluated using metrics such as Root Mean Squared Error (RMSE) and Normalized Discounted Cumulative Gain (nDCG@10) for rating prediction accuracy and

Mean Explainability Precision (MEP) and Mean Explainability Recall (MER) for measuring the quality of explanations.

3.5.3 Results

The results indicate that EMF outperforms the baseline methods in terms of explainability metrics (MEP and MER), showcasing significantly better performance in generating explainable recommendations. The study demonstrates that EMF can effectively balance recommendation accuracy and explainability, offering a promising approach for enhancing user trust and recommendation validity in CF systems without the need for additional data sources.

3.5.4 Conclusion

The research addresses the critical need for transparent and trustworthy recommendations in collaborative filtering (CF) recommender systems. The proposed Explainable Matrix Factorization (EMF) technique introduces a novel approach to generate accurate top-n recommendations that are explainable without relying on additional data sources. The findings of this research have significant implications for enhancing user trust and understanding of recommendations, ultimately improving the user experience in recommendation systems.

4. Dataset

For the purpose of this experimentation we have used two datasets, MovieLens-100k and MovieLens-1M. These datasets are a widely used benchmark in the field of movie recommendation research.

MovieLens - 100k:

a. Size : It consists of 100,000 ratings from 943 users on 1,682 movies.
b. Contents:
    - Ratings : User rate movies on a scale of 1 to 5.
    - Users : The dataset includes information about 943 users.
    - Movies : It contains details for 1,682 unique movies, including genre and titles.
      The dataset also contains information about users such as age, gender, occupation.

MovieLens-1M:

a. Size : The dataset comprises 1,000,209 ratings from 6,040 users on 3,952 movies.
b. Contents:
    - Ratings : User rate movies on a scale of 1 to 5.
    - Users : The dataset includes information about 6,040users.
    - Movies : It contains details for 3,952 unique movies, including genre and titles.

The dataset also contains information about users such as age, gender, occupation.

5. Experimental Details

In this study, we propose a novel approach termed Explainable Matrix Factorization (EMF) for personalized recommendation systems. The methodology encompasses several key stages including data preprocessing, model construction, training, and evaluation.

Core libraries such as numpy, pandas, matplotlib.pyplot, and seaborn provided essential functionality for numerical computations, data manipulation, visualization, and enhanced data visualization, respectively. Additionally, we leveraged custom libraries including 'core.UserToUser' for user-to-user similarity computation and 'core.preprocessing' for data preprocessing tasks such as normalization, encoding, and train-test splitting.

The specific versions used for the libraries are:

1. matplotlib==3.2.2
2. numpy==1.18.1
3. pandas==1.0.5
4. python==3.6.10
5. scikit-learn==0.23.1

The code was executed on a system with an Intel(R) Core(TM) i5-1035G1 CPU.

5.1. Data Preprocessing

A. Data Encoding : User and item identifiers within the dataset were encoded using a standard procedure to facilitate numerical processing.

B. Normalization : To ensure consistency and mitigate biases, we normalized the ratings by subtracting the mean rating of each user from their respective ratings. This process helps in improving the generalization and convergence properties of our model.

5.2. Model Construction

A. User-to-User Similarity : To quantify the similarity between users based on their ratings.

B. Explainable Score Computation : User-to-user similarity was used to assign appropriate weights to each interaction, reflecting its relevance to the user's preference and interests.

C. Explainable Matrix Factorization (EMF) : The model uses the calculated explainable score to enhance traditional matrix factorization. EMF incorporates an additional regularization term, coupled with explainability weights, to learn latent factors.

5.3. Model Training

A. Initialization : Latent factor matrices (P and Q) were initialized randomly to start the training process. Hyper parameters such as learning rate, regularization coefficients, and the latent factors were set based on empirical observations and experiments.

B. Updation : A gradient-based optimization approach to update latent factor matrices was used iteratively.

5.4. Model Evaluation

Model is evaluated for both the datasets for the Mean Absolute Error (MAE) to evaluate the model's predictive accuracy on the test set.

Implemented functionality within the ExplainableMatrixFactorization class to generate personalized recommendations for users based on their previous ratings and preferences.

6. Results & Discussion

After passing the test/validation data to our model for 10 epochs, we observe a loss of

- 0.797 on the validation data for Movielens100k dataset, without normalization (see fig. 6.1)
- 0.783 on the validation data for Movielens100k dataset, after normalizing the data (see fig. 6.2)
- 0.738 on the validation data for Movielens1M dataset, without normalization (see fig. 6.3)
- 0.758 on the validation data for Movielens1M dataset, after normalizing the data (see fig. 6.4)

It is observed that a greater size of the dataset used in training and testing the model led to a comparatively lower loss value. Normalizing the data was beneficial when dealing with a smaller dataset. However, normalization worsened our loss value when dealing with a considerably larger dataset, Movielens1M.

The plot of train loss in Movielens100K Normalized dataset (fig. 6.2) shows a sudden change from epoch 3 to 4, when compared to the change from epoch 2 to 3. This may be considered as a temporary fluctuation since the loss starts decreasing and the graph seems to converge eventually, following the trend of all the other plots.

Movielens1M dataset resulted in worse performance after normalizing the data (0.738 without normalization vs. 0.758 with normalization). This may indicate loss of information after normalization e.g. important variations in the data, data not following normal distribution, or that the hyperparameters of the model require retuning. This observation may require further studies to investigate the cause of worse performance and find their respective solutions (e.g. retuning hyperparameters if found that current hyperparameters are causing the issue).

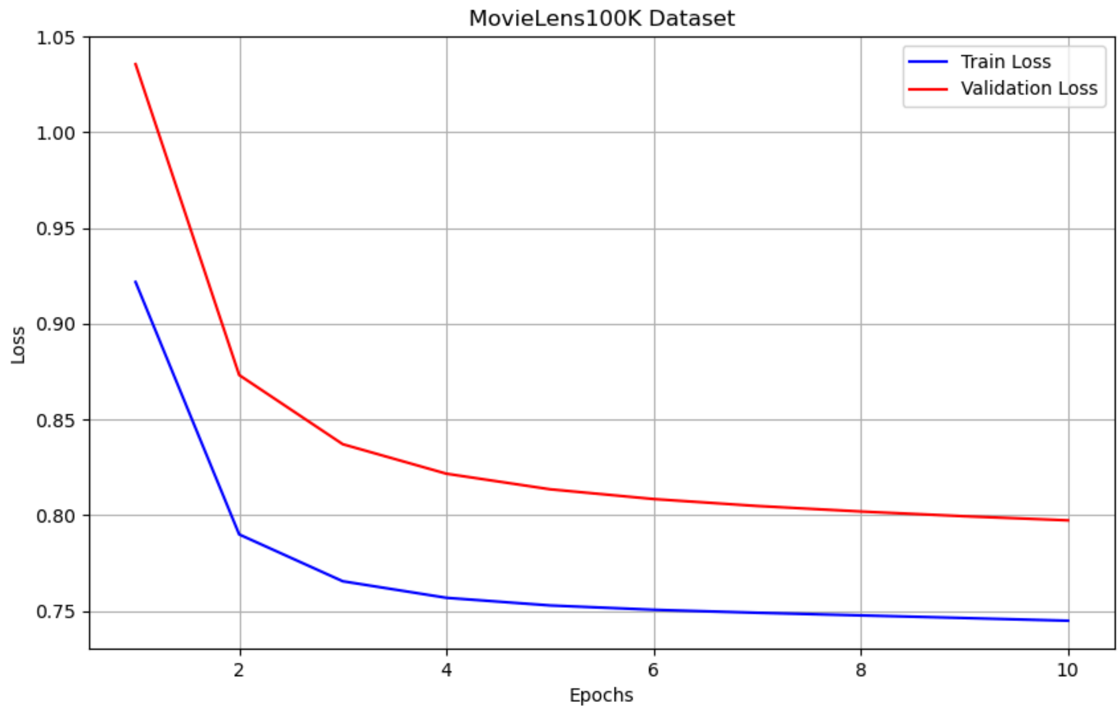

Fig. 6.1: Graph showing training and validation loss on Movielens100K Dataset without using normalization

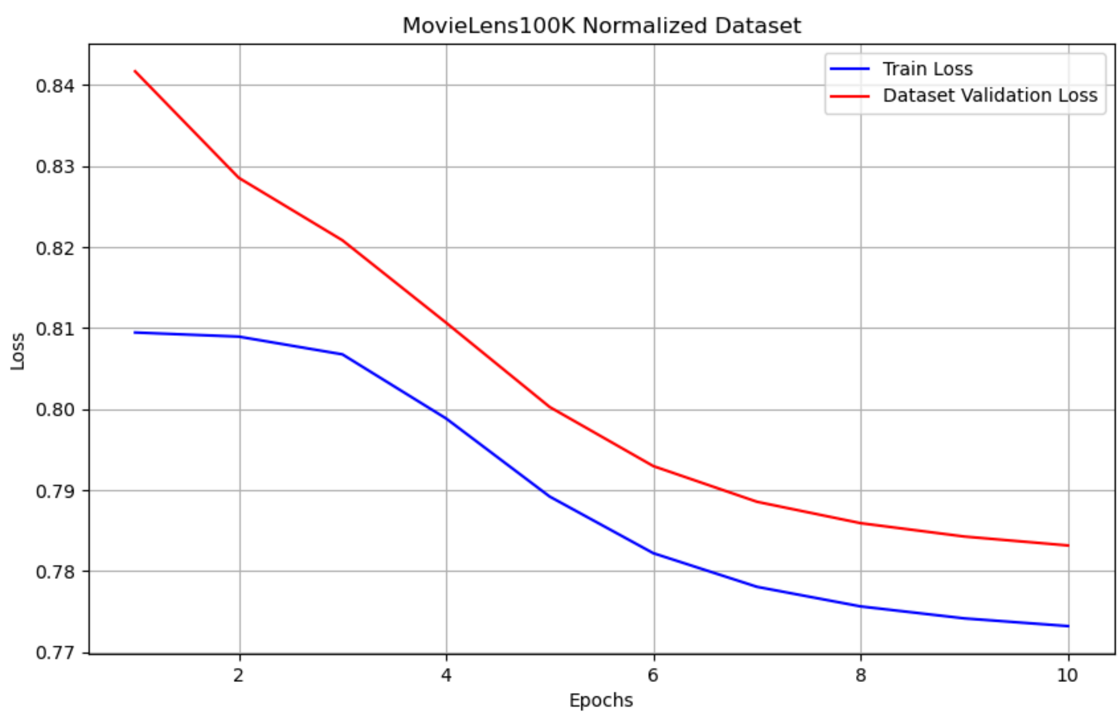

Fig. 6.2: Graph showing loss in training and validation loss on Movielens100K Dataset with normalization

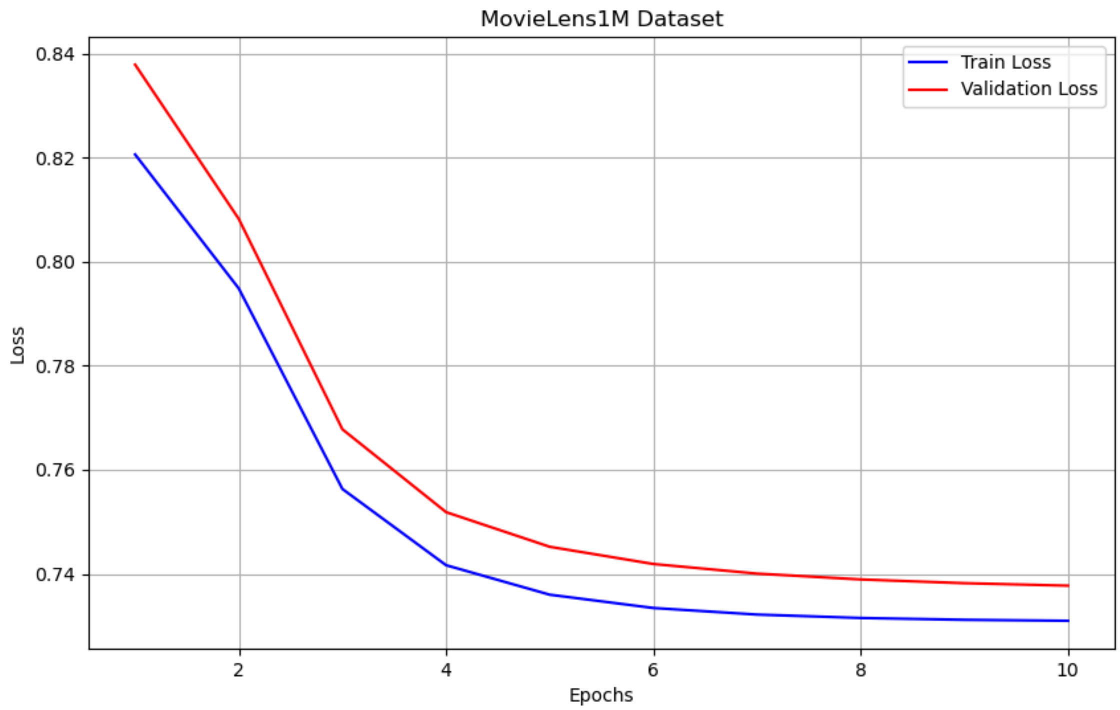

Fig. 6.3: Graph showing loss in training and validation loss on Movielens1M Dataset without normalization

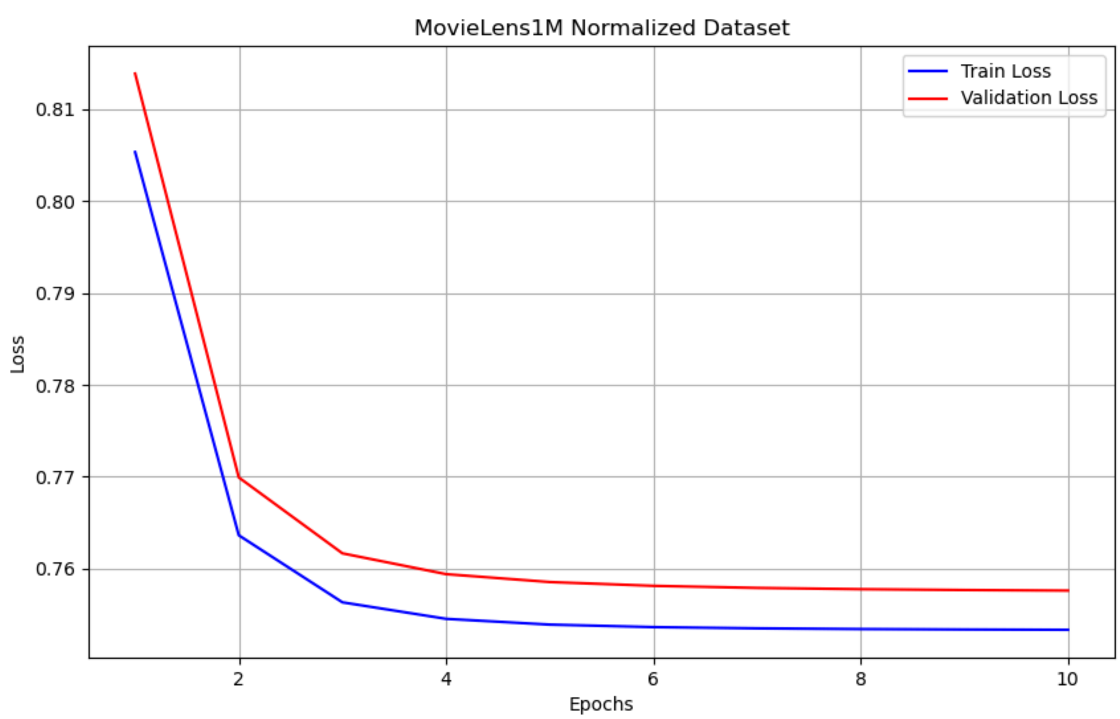

Fig. 6.4: Graph showing loss in training and validation loss on Movielens1M Dataset with normalization

7. Conclusion

The Explainable Matrix Factorisation (EMF) model shows a consistent decrease in both training and validation loss over epochs, indicating good learning and generalization. The loss obtained on Movielens1M dataset is slightly higher than that of Movielens100K dataset, suggesting the dataset size and complexity affect model accuracy. Normalization lead to marginal improvement in performance when considering Movielens100K dataset, but resulted in worse performance in Movielens1M dataset. EMF model has potential for further optimization through hyperparameter tuning and data preprocessing, of which the visuals provided above are clear indications.

References

1. Blazevic, M.; Sina, L.B.; Secco, C.A.; Nazemi, K. Recommendation of Scientific Publications—A Real-Time Text Analysis and Publication Recommendation System. Electronics 2023, 12, 1699. https://doi.org/10.3390/electronics12071699
2. Airen, S., Agrawal, J. Movie Recommender System Using K-Nearest Neighbors Variants. *Natl. Acad. Sci. Lett.* 45, 75–82 (2022). https://doi.org/10.1007/s40009-021-01051-0
3. H. -W. Chen, Y. -L. Wu, M. -K. Hor and C. -Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017, pp. 504-509, https://doi.org10.1109/ICMLC.2017.8108968
4. Bingqian Zhang, Xinyu Zhou, Jiaming Li, and Liang Li. 2020. Recommendation Algorithm Based on Matrix SVD with Exponential Correction. In Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education (CIPAE 2020). Association for Computing Machinery, New York, NY, USA, 71–75. https://doi.org/10.1145/3419635.3419659
5. Abdollahi, Behnoush & Nasraoui, Olfa. (2016). Explainable Matrix Factorization for Collaborative Filtering. https://doi.org/10.1145/2872518.2889405