Final Project in Machine Learning (Recommendation System)

Kollimenos Lampros (2920) - Refenes Alexandros (3134) - Karvelis Ioannis (3100)

January 2024

1 Abstract

In this report, we delve into the domain of collaborative filtering for personalized movie recommendations, employing the renowned MovieLens 100k dataset as our primary source of interaction data. Our motivation stems from the increasing importance of recommendation systems in enhancing user experience across various platforms. By exploring a variety of techniques, we aim to uncover patterns in user preferences and provide insightful movie suggestions.

The methodology employed involves the utilization of well-established collaborative filtering algorithms, including memory-based approaches and matrix factorization techniques. Leveraging the MovieLens 100k dataset, which comprises 100,000 ratings from approximately 943 users on 1682 movies, we seek to understand and implement effective personalized movie recommendations.

Our report discusses the significance of collaborative filtering in the context of movie recommendations, the challenges associated with traditional methods, and the advancements in matrix factorization, hybrid approaches and deep learning. The results obtained from our analysis are presented, shedding light on the effectiveness of the implemented algorithms in capturing user preferences and delivering accurate movie suggestions.

Through this exploration, we contribute to the understanding of recommendation systems in the context of movie recommendations and offer insights into the performance of various algorithms on the MovieLens 100k dataset. Our findings aim to inform future endeavors in building robust and personalized recommendation systems.

2 Introduction

The exponential growth of digital content platforms has led to an abundance of movies, posing a challenge for users to discover titles aligning with their tastes. To address this, sophisticated recommendation systems have become integral to user experience, particularly in the context of movie suggestions. This project focuses on the widely recognized MovieLens 100k dataset, a valuable resource for exploring collaborative filtering algorithms.

2.1 Problem Statement and Significance

The surge in streaming services and digital content availability has created an information overload, making it difficult for users to navigate through the vast array of movies. Personalized movie recommendations offer a solution to this problem, enhancing user engagement and satisfaction. Our project seeks to contribute to this solution by leveraging collaborative filtering to provide accurate and relevant movie suggestions.

The significance of personalized movie recommendations extends beyond entertainment platforms. Such systems contribute to user engagement, satisfaction, and retention, playing a pivotal role in the success of streaming services.

2.2 Motivation

Our motivation for undertaking this project lies in the desire to explore a variety of recommendation techniques, including collaborative filtering algorithms, matrix factorization, hybrid models, and neural networks using Keras. By leveraging the MovieLens 100k dataset, we aim to gain insights into the challenges and opportunities associated with recommendation systems, contributing to the broader understanding of various recommendation techniques.

2.3 Scope and Objectives

Our scope encompasses the implementation and evaluation of collaborative filtering algorithms on the MovieLens 100k dataset, focusing on providing personalized movie recommendations. The primary objectives include exploring memory-based collaborative filtering algorithms, investigating matrix factorization techniques, evaluating hybrid recommendation approaches, and analyzing the results to understand the strengths and limitations of the implemented algorithms.

By achieving these objectives, we aim to contribute to the existing body of knowledge on collaborative filtering for movie recommendations and provide practical insights that can inform the development of recommendation systems in real-world scenarios.

In the subsequent sections, we delve into the existing literature on collaborative filtering, describe the MovieLens 100k dataset and our chosen features, detail the methodologies employed, present experimental results, and conclude with insights for future work.

2.4 Input-Output Framework

In the context of movie recommendation systems explored in this report, the input to our algorithms is derived from user-movie interactions within the MovieLens 100k dataset. Specifically, this input consists of historical user ratings for various movies, forming a comprehensive user-item interaction matrix. Each entry in the matrix represents a user's rating for a particular movie. The output of our algorithms is personalized movie recommendations for users based on their historical preferences. The goal is to predict, with high accuracy, how users might rate unseen movies, thereby facilitating a more engaging and tailored content discovery experience. This process involves leveraging collaborative filtering techniques, deep learning models, and hybrid approaches to generate recommendations that align closely with individual user preferences within the given dataset.

3 Literature Review

3.1 Collaborative Filtering in Movie Recommendations

Collaborative filtering (CF) has been a cornerstone in the realm of recommendation systems, particularly in the context of movie recommendations. Numerous studies have explored collaborative filtering as an effective method for predicting user preferences based on historical interactions. Sarwar introduced memory-based collaborative filtering, where user-item similarities are computed to make personalized predictions. While this approach demonstrated success, it faces challenges with scalability and sparsity. For more information on item-based collaborative filtering recommendation algorithms, you can visit this link.

3.2 Matrix Factorization Techniques

Matrix factorization techniques have gained prominence in collaborative filtering to address scalability issues. The work of Koren introduced Singular Value Decomposition (SVD) for collaborative filtering, decomposing the user-item interaction matrix into latent factors. This approach proved effective in capturing intricate patterns in user preferences. However, it may struggle with handling cold-start problems for new users or items. For more information on recommender systems, you can check out the document Recommender Systems [Netflix].

3.3 Hybrid Recommendation Systems

To enhance recommendation accuracy and address the limitations of individual methods, hybrid recommendation systems have been explored. Burke proposed combining collaborative filtering with content-based approaches to leverage the strengths of both. This fusion aims to mitigate cold-start issues while providing accurate and diverse recommendations. For a survey and experiments on hybrid recommender systems, you can visit this link.

3.4 Evaluation Metrics in Recommender Systems

Measuring the effectiveness of recommendation systems involves various evaluation metrics. Metrics such as Root Mean Squared Error (RMSE), Precision, and Recall are commonly employed.

Herlocker discussed the importance of evaluating recommendation algorithms using relevant metrics, highlighting the trade-offs between accuracy and diversity.For information on evaluating recommender systems, you can check out the document Evaluating Recommender Systems.

3.5 State-of-the-Art in Collaborative Filtering

The current state-of-the-art in collaborative filtering often involves advanced matrix factorization techniques, deep learning models, and the incorporation of side information. Xiangnan He proposed Neural Collaborative Filtering (NCF), integrating neural networks to enhance recommendation accuracy. These advancements aim to address the challenges posed by traditional collaborative filtering methods. For more information on the topic, you can read the paper on arXiv: A Comprehensive Survey on Graph Neural Networks.

3.6 Comparison with Existing Work

In comparing our work with existing literature, we acknowledge the strengths and weaknesses of memory-based and model-based collaborative filtering approaches. We build upon the insights from matrix factorization techniques, incorporating them into our models for personalized movie recommendations on the MovieLens 100k dataset.

This literature review provides a foundation for our understanding of collaborative filtering methods, guiding our approach to building an effective recommendation system for movie preferences.

4 Dataset and Features

4.1 MovieLens 100k Dataset Overview

Our study utilizes the MovieLens 100k dataset, a well-established benchmark in the field of collaborative filtering and recommendation systems. This dataset comprises interactions between users and movies, containing 100,000 ratings ranging from 0.5 to 5. Specifically, it includes data on approximately 943 users and 1682 movies, forming the basis for our exploration into personalized movie recommendations.

4.2 Preprocessing and Data Characteristics

To ensure the robustness of our models, we conducted preliminary preprocessing steps on the MovieLens 100k dataset. This involved handling missing values, if any, and ensuring a balanced distribution of ratings. We also split the dataset into training, validation, and test sets to facilitate model evaluation.

Furthermore, normalization techniques were applied to scale the ratings appropriately, enhancing the convergence and interpretability of our collaborative filtering algorithms. Our goal was to provide a clear understanding of user preferences and movie affinities.

4.3 Feature Extraction and Representation

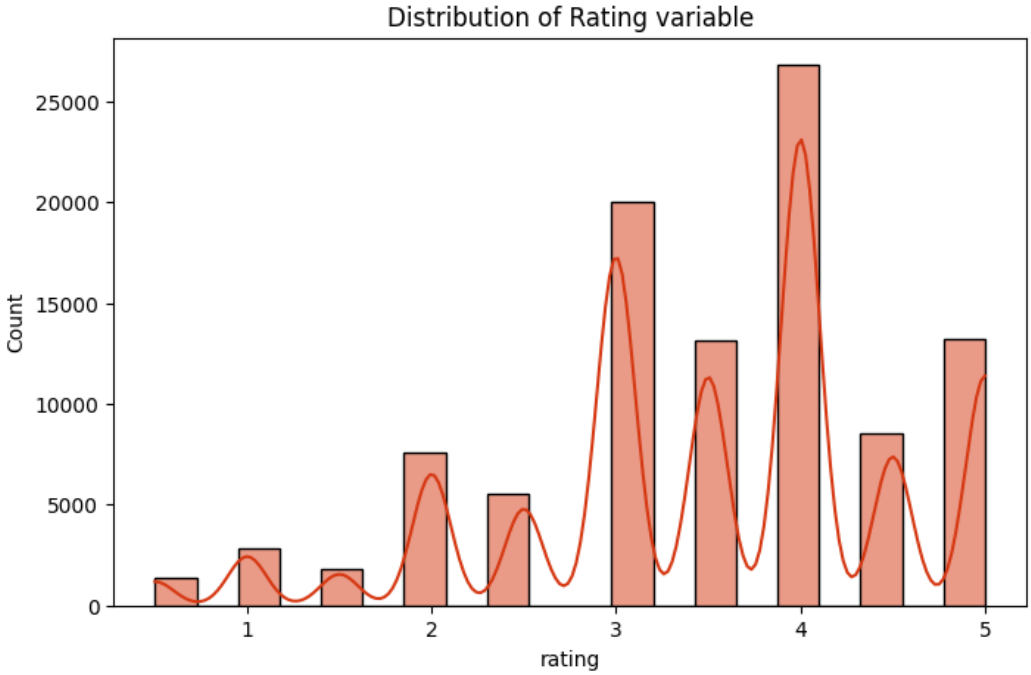
In the context of collaborative filtering, the primary features in our dataset are user-item interactions, representing the user's rating for a particular movie. These interactions are pivotal for constructing user-item matrices, forming the foundation for recommendation algorithms.

While the MovieLens 100k dataset itself doesn't involve explicit features beyond user-item interactions, the collaborative filtering algorithms employed inherently leverage latent factors to represent users and movies in a feature space. These latent factors capture underlying characteristics that contribute to personalized recommendations, enhancing the performance of our models.

4.4 Example Instances

In our exploration of the MovieLens 100k dataset, we began by analyzing the distribution of user ratings, a crucial aspect of understanding user preferences. The histogram below illustrates the distribution of ratings on a scale of 0.5 to 5. The plot, created using Seaborn, provides insights into the prevalence of different rating values across the dataset. The kernel density estimation (KDE) overlay enhances our understanding of the continuous distribution. Notably, we observe a concentration of

ratings around values that are higher than 3, shedding light on the general sentiment of users toward the movies in the dataset, which seems to be mostly positive. This analysis sets the stage for further investigation into personalized movie recommendations by establishing a baseline understanding of how users rate the movies in our dataset.



5 Methods

5.1 Collaborative filtering - User-based

In the ever-expanding landscape of online content, personalized recommendation systems play a pivotal role in enhancing user experience. Among these, user-based collaborative filtering stands out as a prominent methodology that leverages the collective wisdom of like-minded users to provide tailored suggestions. The fundamental idea behind this technique is to identify individuals with similar preferences based on their historical interactions with items and recommend unseen items to a target user, drawing upon the tastes and choices of those who share comparable interests. This collaborative approach taps into the power of user consensus, allowing for the discovery of relevant content in a manner that transcends traditional genre or content-based categorizations. As we delve into the functionality of user-based collaborative filtering, we uncover a mechanism that not only reflects the intrinsic social nature of user preferences but also addresses the inherent challenge of information overload, guiding users toward a more personalized and engaging content discovery journey.

5.2 Collaborative filtering - Item-based

The fusion of item-based collaborative filtering with k-nearest neighbors (KNN) enhances recommendation systems, providing nuanced and personalized content suggestions. Item-based collaborative filtering excels by identifying similarities between items based on user interactions, offering insights into user preferences. KNN refines this approach by focusing on the most relevant items with similar user engagement patterns.

In the Surprise library, item-based collaborative filtering seamlessly integrates with models like KNNWithMeans. Configured with options such as cosine similarity, this model utilizes patterns in user-item interactions to deliver personalized recommendations. Importantly, it considers mean ratings, contributing to a nuanced understanding of user preferences. This dynamic approach leverages the collective wisdom of similar items for a refined user experience in digital content consumption.

This process is represented by this equation:

$$\text{cosine\_similarity}(U_i, U_j) = \frac{U_i \cdot U_j}{\|U_i\| \cdot \|U_j\|}$$

5.3 Matrix factorization

Introducing Singular Value Decomposition (SVD) functionality within the Surprise library unveils a powerful dimensionality reduction technique that significantly advances collaborative filtering in recommendation systems. SVD, a matrix factorization method, excels in capturing latent features underlying user-item interactions, thereby enabling the model to uncover intricate patterns and relationships within large datasets.

As applied within the Surprise library, the SVD algorithm facilitates the seamless extraction of latent factors, providing a foundation for precise and personalized predictions. This introduction sets the stage for a detailed exploration of how SVD, embedded in the user-friendly environment of Surprise, contributes to the efficacy and sophistication of collaborative filtering models for recommendation systems.

SVD is given by the following mathematical equation:

$$R \approx U\Sigma V^T$$

5.4 Hybrid Model

The hybrid recommendation model for MovieLens ML Latest Small combines collaborative filtering (CF) with singular value decomposition (SVD) and incorporates a 20% content-based filtering component. Collaborative filtering leverages user-item interactions to identify patterns and similarities among users or items, enhancing the accuracy of recommendations. SVD, on the other hand, decomposes the user-item interaction matrix to capture latent features, providing a nuanced understanding of user preferences. The inclusion of content-based filtering, accounting for 20% of the model, ensures a more comprehensive recommendation approach by considering the inherent characteristics of movies, such as genres. This hybrid model strikes a balance between personalized user preferences derived from collaborative filtering and the richness of content-based features, resulting in a robust and effective recommendation system for MovieLens ML Latest Small dataset.

5.5 Neural Network CF

The neural network model is created using TensorFlow and Keras, consisting of embedding layers for users and movies, followed by flattening, concatenation, and dense layers. The model is compiled with mean squared error loss and trained on the training data, with ModelCheckpoint used to save the weights of the best-performing model based on validation loss. After training, the best model weights are loaded, and the total training time is printed. This collaborative filtering model aims to predict movie ratings based on user and movie interactions, providing recommendations by capturing user preferences through embeddings. The training process involves optimizing the model parameters to minimize the mean squared error between predicted and actual ratings, enhancing its ability to make accurate predictions on unseen data.

This process is described by this equation:

$$\hat{y}_{u,i} = g(f(U_u, V_i))$$

5.6 Neural Network CF with Matrix Factorization

A collaborative filtering neural network for movie recommendation using embeddings. The model takes two inputs: user IDs and movie IDs. It then creates embeddings for both users and movies using separate embedding layers. These embeddings are vectors in a latent space with a specified number of factors (nfactors). The embeddings are flattened and fed into a Dot layer, which computes the dot product of the user and movie embeddings, representing the predicted user-movie interaction or rating. The network is trained to minimize the mean squared error between the predicted ratings and the actual ratings in the training data. The Adam optimizer is used for optimization with a learning rate of 0.001. The training data is split into training and validation sets using the train-test-split function from scikit-learn. The model is compiled, trained using the fit method, and evaluated over 10 epochs. The summary of the model shows the architecture, including input and output shapes, the number of parameters, and the layers involved. The embedding layers contribute significantly to the number of parameters in the model. The embeddings capture latent

features of users and movies, allowing the model to generalize and make predictions for unseen user-movie pairs based on learned patterns from the training data. The collaborative filtering approach enables personalized movie recommendations by leveraging the preferences of similar users.

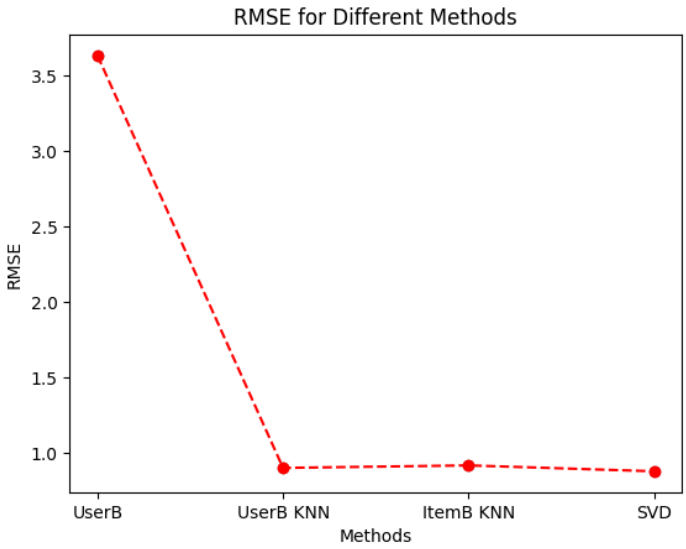
5.7 Long Sort-Term Memory Neural Network

The model architecture involves embedding layers for both users and movies, concatenating these embeddings, and passing the result through dense layers with dropout for regularization. Additionally, an LSTM layer captures temporal dependencies in the data. The model is compiled with mean squared error loss and trained on the training data, with validation performed on the test set. The training process is monitored using ModelCheckpoint to save the weights of the best-performing model. After training, the saved weights of the best model are loaded, and the total training time is printed. This collaborative filtering model aims to predict movie ratings based on user and movie interactions, offering recommendations by capturing both user preferences and temporal patterns in their ratings.

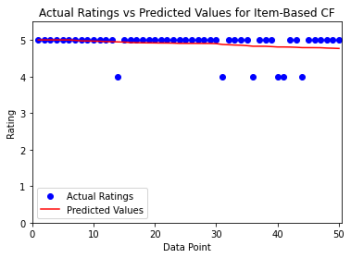
6 Experiments

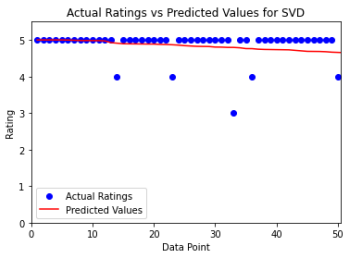
6.1 Machine Learning Methods

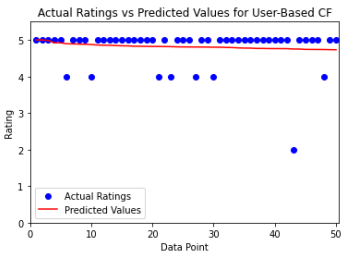
In a world where personalised recommendations rule supreme, our story unfolds against the backdrop of collaborative filtering techniques such as user-based and item-based recommendation systems, with the intricate algorithms of k-nearest neighbours (KNN), singular value decomposition (SVD), and the pursuit of the elusive root mean square error (RMSE) serving as the benchmark for optimising the delicate balance between accuracy and scalability.



Now we proceed with further testing by comparing the actual to the predicted ratings of 50 movies user1(for example) has already rated.





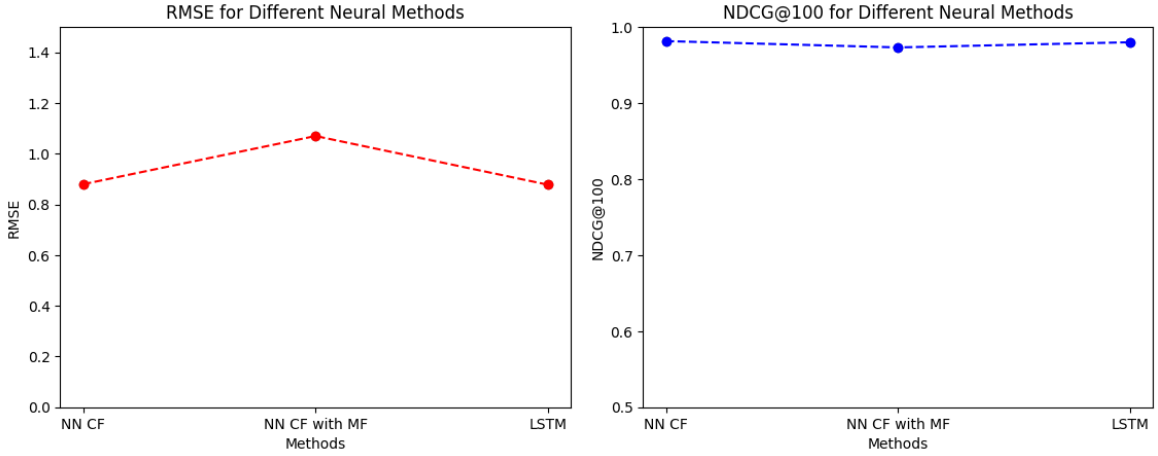


Observations

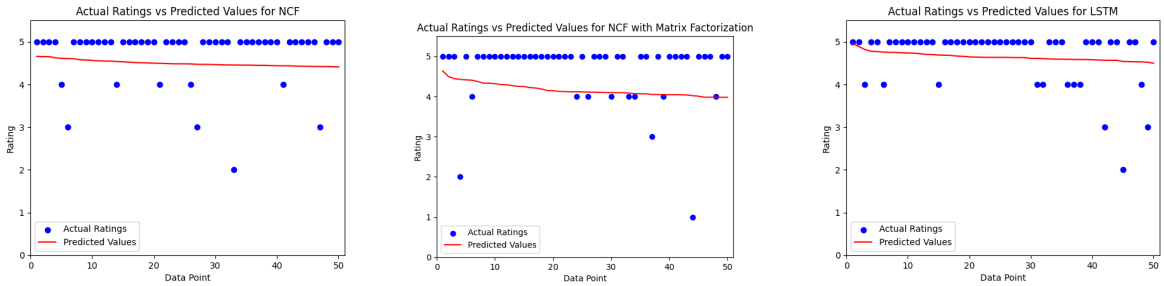
- As we can see from the experiments above 3 of our models seem to have very comparable results in the predictions.
- The first method we used for the User-Based CF with matrix similarity was the least effective.
- Also we saw a noticeable time difference between item and user based CF (both with KNN). Item based seemed to be much slower in the execution. One thought for this for sure the high sparsity of the item user matrix

6.2 Deep Learning Methods

The effectiveness of the proposed methods, namely Long Short-Term Memory (LSTM) Neural Network, Neural Network Collaborative Filtering (CF) with Matrix Factorization, and Neural Network CF, was thoroughly evaluated. These methods were meticulously implemented and configured based on the methodologies outlined in the previous section. To assess the performance of each model, two widely recognized evaluation metrics, Root Mean Square Error (RMSE) and Normalized Discounted Cumulative Gain at 100 (NDCG@100), were employed. RMSE served as a quantitative measure of the models' predictive accuracy, quantifying the average magnitude of the errors between predicted and actual values. Meanwhile, NDCG@100 provided insights into the ranking quality of the recommended items. The combination of these evaluation metrics offers a comprehensive analysis of the models' efficacy in capturing complex patterns and generating accurate recommendations.



Now we proceed with further testing by comparing the actual to the predicted ratings of 50 movies user1(for example) has already rated.



Optimizers

Optimizers play a crucial role in training neural networks, and choosing the right optimizer can significantly impact the performance of your models. While Adam is a popular optimizer due to its adaptive learning rates and momentum, we tried different optimizers that can help in finding the one that works best for our recommendation system architecture.

After compiling our models with SGD, Adam, Nadam and RMSprop we came to the conclusion that in our recommendation system for our dataset only SGD seems to be slightly underperforming in comparison to the others, although its training time is only 60 seconds when the other three optimizers needed 80 seconds.

Observations

- Firstly we can see that the method with the Matrix Factorization is the worst by far.
- Long Short-Term Memory seems to have the best performance even though it is known to be suitable for sentiment analysis and our dataset does not have written reviews.
- Different optimizers did not seem to have a noticeable effect on the results.
- NDCG metrics are very high across the board indicating that all three models are performing well with the matrix factorization model being only slightly worse.
- There seem to be an overfit of our models. In the train set we have much lower RMSE than in the test set

6.3 Comparison of Machine and Deep Learning

The experimentations and evaluations of both traditional machine learning (ML) methods and deep learning (DL) techniques provide valuable insights. The machine learning methods, encompassing user-based and item-based collaborative filtering with algorithms like k-nearest neighbors (KNN) and singular value decomposition (SVD), serve as foundational pillars for recommendation systems. The meticulous comparison of actual and predicted ratings reveals nuanced differences in performance, with some methods demonstrating higher effectiveness than others. On the other hand, the integration of neural network models, including collaborative filtering with matrix factorization, neural network collaborative filtering, and long short-term memory (LSTM) neural network, introduces a deep learning dimension to recommendation systems. Surprisingly, the LSTM neural network emerges as a robust performer and would have theoretically proven even more performant if we had trained it on a dataset with written reviews instead of just ratings. Additionally, the exploration of different optimizers highlights the adaptability of DL methods. Ultimately, the comprehensive evaluation, considering metrics like Root Mean Square Error (RMSE) and Normalized Discounted Cumulative Gain at 100 (NDCG@100), provides a holistic understanding of the efficacy and trade-offs between traditional machine learning and deep learning approaches in the context of recommendation systems. Ultimately, machine learning methods seemed to function slightly better than our deep learning models eluding to the fact that our dataset does not have adequate features or items for the more sophisticated deep learning models to show their advantages over regular machine learning models. It is also noteworthy that deep learning models are more computationally expensive and therefore take substantially more time to train, in our case that time was upwards of 80 seconds.

7 Conclusion

In conclusion, our exploration of movie recommendation systems on the MovieLens 100k dataset revealed notable insights. The collaborative filtering with KNN, the SVD and the LSTM neural network emerged as the top-performing models. The success of these algorithms was influenced by the dataset's characteristics, including sparsity and the absence of additional features. Notably, the LSTM neural network showcased adaptability, excelling in capturing temporal dependencies within user ratings despite the dataset's limitations. Some directions for future research and improvement are:

- The project could use a larger and more comprehensive dataset with more ratings, users, movies, and features. This could improve the accuracy and robustness of the recommendation systems, as well as allow for more fine-grained analysis and comparison of different methods.
- The project could incorporate other features such as written reviews into the input or output of the recommendation systems. This could enhance the quality and relevance of the recommendations, as well as provide more information and context to the users.

8 Contributions

All three members of the team contributed to research of useful material. Then we developed our first model (User-based with matrix similarity). After that we splitted development of our three other machine learning models to each member of the team. Refenes Alexandros worked mainly on the SVD model, Kollimenos Lampros on Item-based CF with KNN and Ioannis Karvelhs on the User-Based CF with KNN. Having done that we shifted to research of deep learning models. After some consideration and planning each member developed one deep learning model. Kollimenos Lampros worked on our Neural Collaborative Filtering model, Ioannis Karvelhs worked on our collaborative filtering approach with matrix factorization and Refenes Alexandros mainly developed our Long Short-Term Memory model. Finally all three members served as editors on this report.

References

https://www.nickmccullum.com/recommendation-engines-collaborative-filtering-python/
https://surprise.readthedocs.io/en/stable/knn_inspired.html
https://grouplens.org/datasets/movielens/
https://ieeexplore.ieee.org/abstract/document/7977363