A comparative analysis on the different Recommendation Engines on the movies Dataset

MSc Research Project
MSc AI for Business

Madhusudhan Purushothama
Student ID: x23137193

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Madhusudhan Purushothama

Student ID: x23137193

Programme: MSc AI for Business **Year:** 2024

Module: MSc Research Project

Supervisor: Rejwanul Haque

Submission Due Date: 12/08/2024

Project Title: A comparative analysis on the different Recommendation Engines on the movies Dataset

Word Count: **10336** **Page Count : 22**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Madhusudhan Purushothama

Date: 12/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (-if applicable):	

Table of contents

A comparative analysis on the different Recommendation Engines on the movies Dataset

Madhusudhan Purushothama

X23137193

Abstract

There has been an increase in volume of information due to the increase in growth of internet and easily accessible mobile devices and businesses that depend on the internet. This has led to the need to create a system than has the capacity to filter out important information for users. To solve for this issue a recommendation system can be used to provide the consumers the necessary information about the items or services based only on the requirement of the individual. A lot of research has been made on recommender systems and have developed various filtering algorithms that can increase the efficiency and effectiveness for the users and the system. In this work, I have conducted a comprehensive analysis of recommendation engines, evaluating them using the identical MovieLens dataset. This study involves comparing several distance approaches and norms for implementing recommender filtering systems. In this abstract, we aim to analyze various recommendation engines, including collaborative filtering, content-based filtering, machine learning models such as and k-NN, and deep learning models like LSTM and Siamese Networks and Sentence Transformers. The aim is to proceed with building a multimodal recommendation system to recommend movies.

Keywords: Recommendation Engines, Similarity Check, Cosine Similarity, Collaborative Filtering, Content-Based Filtering, Hybrid Filtering, Classification Techniques, K Nearest Neighbours, L2 Norm, L1 Norm, L-p Norm

1 Introduction

The streaming services such as Netflix and Amazon Prime Video are valuable companies in the video content industry as they have most number of users who use them and the scale of business they have. They also generate a lot of revenue. *Giving the right recommendation to the company's user base is the main objective of all for profit organizations who are in the pursuit of increasing their income.* To achieve this companies such as Netflix and Amazon Prime invest heavily on research and developing a recommendation system and how to give their users the recommendation of items or services that they most likely will buy.

The business objective of this work is to give an overview of different types of recommendation engines and understand how they are built and scaled and how can they act as a foundation for generating more revenue. We have utilized a Group-lens 100K Movie-lens dataset, and the technique is built to provide the best fit for the data set under consideration. A detail comparison of various methodologies has been conducted which are based on hyperparameters, metrics and other factors. These have been supported by going through the performance of the model in various use case scenarios which assist in predictive analytics of a movie based on the interest of the user. Since the whole idea revolves around the fact about a complete end-to-end analysis of different recommendation filtering techniques used in the famous OTT platforms.

The research questions can be formulated in the following manner,

RQ1: What are the differences between the different Recommendation Platforms like Content-Based, item Based, Collaborative, Hybrid Filtering, Machine Learning based Filtering and Deep Learning based Filtering?
Solution: We will analyse in deep down towards the way these recommendation engines are working. Based on this we will also look which solutions are easy to use and can give maximum impact in a favourable recommendation.

RQ2: How the Recommendation Engines are used for the development of famous OTT platforms like Netflix, Amazon Prime etc.?
Solution: We should understand the concepts so that the research can be fruitful while making a solution out of it.

RQ: Can machine learning models, deep learning models be applied along with recommending filtering to achieve the hybrid movie recommendation?
Solution: We will try to make a hybrid pipeline which will analyse the users interest and based on this gives a collected best recommendation output.

Research Outcome

The study concluded with the development of hybrid recommendation system which combines collaborative filtering, content-based filtering, machine learning models such as Naive Bayes and k-Nearest Neighbours, and advanced deep learning models like Long Short-Term Memory (LSTM) networks and Siamese Networks and Large Language Models that it becomes a greedy searched multi modal system. The constraints of individual methodologies were addressed, this multi modal approach on the GroupLens 100K MovieLens dataset shows a significant improvement in the accuracy of the recommendation. The study shows the uses of integrating conventional and modern ways to develop a recommendation engine that is both customized and adaptable. It gives an in-depth analysis of these models. The suggested hybrid pipeline model shows the possibility of practical application in subscription-based video on demand platforms like Netflix and Amazon Prime Video which shows the importance of customizing recommendations depending on the preference of the user. This study acts as the base for future improvements which also guarantees the flexibility of the system including the changes to the behaviour of the user and technological breakthroughs.

Research Outline

In this section we will discuss the research papers that has been studied and implemented as of now and look into the mathematics of the recommender engines. The works of the recommender engines is based on the principles of linear algebra and matrix computations, these two subjects will be of utmost importance in the implementation of this idea. In the Implementation chapter, we will be discussing the way the whole of the idea is implemented and discuss on how the different models or algorithms, or the ideas are implemented. This chapter is followed up with the results and analysis and we will be discussing the different results of the part of the code. Finally, we will conclude and discuss the future implementation that needs to be taken into the account for future work.

2 Related Work

The internet has become affordable and as a result the number of users who use the internet is growing every day. This leads to an increase in the amount of data that is being sent in the internet. This has resulted in the increase in the amount of information and knowledge that is being shared. (Sánchez Moreno et al., 2020).
The huge increase in data has started a new era of handling information. This data is used to develop systems that are more efficient, inventive and effective. When it comes to stitching the data the recommendation engine helps in forecasting the ratings that a user would give to the items or services of interest.

2.1 Research on the different recommendation Engines

The movie recommendation engine is discussed in this section as an example of adaption of the big data technique. A movie recommendation engine according to the general definition is a filtering-based system that assesses the quality of the search result and gives a recommendation based on the relevancy of the search items based on the search history of the user. (Amato et al., 2017; Alam & Awan, 2018; Rehma et al., 2018).

User ratings that could be given to an item of interest are predicted by recommendation engines. These recommender systems can be found in various websites and are used for retrieving information and extracting information from heterogenous data sources among various other applications. (Amato et al., 2017; Alam & Awan, 2018), among other applications. This chapter talks about recommender engines which is used as a tool for marketing for internet firms. This recommender system is considered as a intelligent and knowledgeable sales person who knows the preferences of the users and can make informed decisions about which item or service recommendation will be beneficial to the user and is likely to convert. It was initially used in e commerce and is now being used in the media industry Akerkar (2019).

Recommender engines have become a critical part of multiple fields, and they have been used in many sites like Netflix, Amazon and YouTube among many other sites. In a few sites these recommender engines give recommendations based on the search history of the user and the platform they are using. These are used by sites like LinkedIn, Amazon, Facebook and Netflix. The sites are designed in such a way that it benefits the user by giving them a more enjoyable experience. (Amato et al., 2018). The recommender system can be considered as a system that does data analysis and stitches the necessary data. These systems are getting more popularity and are being used in various sectors like movies, news, sports. At the moment sites like Spotify, Netflix and other multimedia companies have a variety of interesting features that will help in enhancing the experience of the user.

This holds true for a lot of applications that are being used in the mobile phone. The ability of these recommender systems is very important for these applications. The user’s taste, experience, behaviour and interest is considered by the recommender system. This can help in improving the customer experience and also lower the cancellation rates (Amato et al., 2018). The recommender system uses a strategy called as stepwise where a user rates different items and the system is then able to figure out what the user might rate other items that the user has not rated (Khadse et al., 2018). Content based and collaborative filtering are the methods that are used most frequently to make a recommender system. Collaborative filtering can be used in the user-based apps to give rapid and effective personalized recommendations. In order to develop this collaborative filtering system that the recommender system makes use of the training data we make use of machine learning method (Chen et al., 2021).

This model can be used to give more accurate recommendations. If the recommendation system is trained on huge amounts of data, it will be able to perform more efficiently and effectively. As there has been a rise in the amount of data that is being generated it has become more challenging to extract data from different sources (Khalid & Zebaree, 2021). The big data in movie recommendations provides a huge amount of user data like the behaviour of the viewers, ratings, site activity on the basis of ratings and the data of the users (Shen et al., 2020; Bazai et al., 2021). A computing system called Apache spark is a clustering based computing system that has high performance, open source and is openly accessible (Bazai et al., 2021), it is a framework that is freely available and is used in big data analytics by using continual algorithms. It supports tools that were written using MLlib and APIs (Myung & Yu, 2020). Spark is able to give a greater range of components when we compare it to MapReduce function that is being used in Hadoop. It also provides better performance and offers other applications in various supervised learning based models (Morfino & Rampone, 2020). For programming its functionalities use a resilient distributed dataset (RDD).

2.2 Advances in the Movies Recommendation

The use of parallel programmes can have a very strong point. Spark has various libraries and tools like Python, Scala and R that help in the management of data and scaling the system. (Bazai & JangJaccard, 2020). This recommender system was developed using alternating least square (ALS) model and matrix factorization and was implemented using Spark machine learning and is popularly now as MLlib (Meng et al., 2016) for the application programming interfaces (APIs) in machine learning and are available through the spark framework. A machine learning toolkit has various a variety of methods which includes linear regression, frequent pattern matching, clustering, Spark ML, linear algebra, Classification, and recommendation in a single package. This paper talks about how collaborative filtering was used to address the cold start problem. This method uses Spark MLlib where the gaps in the users item-based matrix is filled. It uses latent features to separate the rating dataset into matrices which are item based and user based. They used the elements from the new user matrix instead of creating a new model each time which reduces the time and effort. The researchers achieved the desired points by multiplying the matrices in order to reduce the time. The complexity of the system might have decreased due to categorizing the recommendation system into various collective services and making classification process leads to the training continuous models which results in a strong collaborative system.

This model-based approach has a major benefit as it could recommend huge number of items to various users and this was achieved due to assigning explicit ratings to the data that was collected when the data was less. This method gives better results if we use sparce matrices instead of memory-based algorithms. The website Classmates.com that began in the year 1995 became popular as a result of word of mouth. This has motivated other communities to develop their own services that address niche audience.

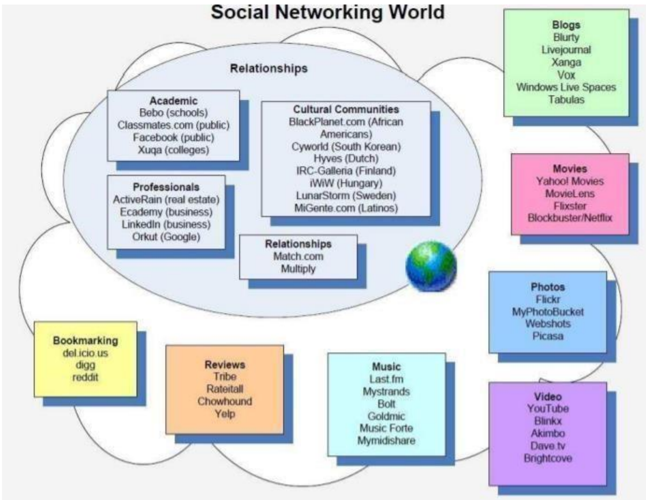


Figure 1: Social Networking world for the Movies Analysis and Recommendation (Meng et al., 2016)

The websites such as Blockbuster gives personalized recommendation based on the ratings of the user and does not include any social networking features (Akerkar, R., 2019). Yahoo! Movies uses personal ratings to recommend movies that might be shown in theatre or television. It utilizes its user base to recommend a list of movies based on the users who are like you and the users reviews and their rating. Flixter recommends movies to users based on what the friends of the user have rated by creating a web-based community around movies.

2.3 Advanced models in the movie recommendation

A RNN variant that analyses sequential data in the recurrent structures is the Gated Recurrent Unit (GRU). The extracted features and the proposal from the rectangular region is combined in the Faster RCNN, which is an improved version of Fast RCNN. The trailer of the movie is segmented into frames manually and is pre-processed for image analysis by using a fuzzy elliptical filter. A text analysis is performed by tokenising the movie reviews. This text in then fed to classify the movies as offensive or non-offensive through the GRU. The Faster RCNN is now fed with pre-processed visuals to classify the movies as violent and non-violent by using the features extracted from the trailers. The best movies is selected by the fuzzy decision making system by analyzing the outputs form Mamdani fuzzy interface system using gauss membership function and the four categorized outputs.

The performance of these deep neural networks were analyzed using the recall, specificity, precession, accuracy and F1 score. The accuracy range of FRCNN is 98.42% for movie trailers and accuracy range of GRU for reviews is 97.73%. The rise in streaming services like Netflix and Disney+ contribute s to the rise in the industry. The users face challenges in choosing the films that they might like as they have to pick form a variety of different films. The use of a recommender system can relieve this issue. Twitter was used to gather reviews of the movies. The item recommendation was given to the user by developing a recommender system built using weighted hybrid filtering and GRU algorithms. The weighted hybrid filtering is used by combining collaborative and content-based filtering methods. The dataset was acquired by crawling tweets that had the feedback of a certain accounts about a film. This resulted in a dataset that has 854 films, 45 users and 34,086 tweets in total. Test sizes of 40%, 30%, and 20% are used. SGD, Adadelta, Nadam, Adam, and Adagrad optimization techniques were used. The results show that the Nadam optimization approach gives the best result. The recall of 88.63%, F1score is 86.30% and accuracy is 88.63%, and precision is 85.74%.

Movie Name	Release Year	Director Name	Genres	Cast	Writer	IMDb Rating
3 Idiots	2009	Rajkumar Hirani	Comedy, Drama	Aamir Khan, Madhavan, Mona Singh	Abhijat Joshi	8.4

Movie's Summarised Story-Line	Trailer Video	Trailer Audio
Two friends are searching for their long lost companion. They revisit their college days and recall the memories of their friend who inspired them to think differently, even as the rest of the world called them "idiots".		

Figure 2: A showcase of the multimodal system for the movie recommendation using different data sources (Valentino & Setiawan, 2024)

To be able to recommend accurate movies to the users we have to first build a sequence of their behaviour. The movie sequence embedding, and user information are used as input features after dimensionality reduction which are fed into the transformer architecture and multilayer perceptron (MLP). The accuracy of the system is improved by using a transformer layer with positional encoding for sequencing the activity of the user and multi head attention algorithms. To make the recommendation more accurate to each user the system uses K Means clustering to group these movies according to the embeddings of the genre. The anticipated rating is then combined with this information. The 100 K and 1 M MovieLens datasets showed improvement on testing the model. The 100 K dataset had RMSE, MAE, precision, recall, and F1 scores of 1.0756, 0.8741, 0.5516, 0.3260, and 0.4098, while the 1 M dataset had 0.9927, 0.8007, 0.5838, 0.4723, and 0.5222. This method solves the scalability and cold start problems and also performs better than the baseline models in recommending the Top-N items.

The feedback from the user is in three categories I) dislike, ii) like, and iii) neutral/not viewed. The dataset used here is from the Audio-Video data from the Flickscore dataset. The classification model that uses two factors were studied. The GenreLike-score GL-score will help the users find movies according the genre that is preferred by the user and (ii) Different Audio/Video embeddings. The GLscore was better in predicting the preference and this was given by testing various combination of these parameters on different modalities of the dataset. The study used various embedding strategies and the keyframe extraction approaches were studied.

One of the examples of the recommendation system is a movie recommendation system which plays an important role on going through the entire collection of the movies data ad recommending the movies based on the preference of the user. The previous study in the field of recommendation system have focused on the output assuming that the input for the model will remain the same for every user. The embedding in the input vector changes for various users. The perspective of the film changes based on the Genre and meta information like plot and who directed the movie and the cast of the movie. The TestLike score and GenreLike score were used to formulate this. This cross attention-based model was successful in competing with the state-of-the-art model. The datasets used to assess these models were MovieLens-100K (ML-100K) and MFVCD-7K. The audio video data was used for considering the multimodality and the textual data was used for computation of the score.

The Transformer architecture for feature fusion for various modalities, the ViT model for poster modality extraction and the BERT model for textual modality extraction were used to predict the choice of the user. We can give better recommendation by making use of both pretrained fundamental models with smaller datasets that is used in the downstream application. The dataset used is the MovieLens

100K and 1M datasets. This cross-modal algorithms is able to predict the ratings of the user better than the baseline model.
We can also make use of a new candidate ranking technique to keep the users out of the recommended item maze. This model worked on MovieLens (ML) 1M and had a success rate of 71% for short-term predictions and a hit rate of 58% overall. It also manages two essential parameters—item popularity which has a 34.28% success rate) and diversity 59.22% in an implicit manner.

The machine learning algorithms have contributed to the process of personalized content distribution which has increased the accuracy of the recommendation system. The analysis and comparison has been made using three datasets MovieLens Dataset, The Movies Dataset, and TMDB Movie Dataset. SVD had a better performance on the MovieLens 20M Dataset, AUTOENCODER had a better performance on The Movies Dataset, and the hybrid recommender performed best on the TMDB 5000 Movie Dataset. The best recommendation algorithm can be choose by considering particular dataset properties. We can improve the accuracy of these systems by combining contact-based filtering with content filtering and neural collaborative filtering.

2.4 Research Summary for the advanced models

Sl.no	Name	Author	Dataset	Model	Result
1	Movie recommendation system via fuzzy decision making based dual deep neural networks	Aramuthakannan, S., Ramya Devi, M., Lokesh, S. and Manimegala	MovieLens	GRU, FRCNN	The proposed GRU yields accuracy range of 97.73% for reviews and FRCNN yields the accuracy range of 98.42% for movie trailer.
2	Movie Recommender System on Twitter Using Weighted Hybrid Filtering and GRU	Valentino, N. and Setiawan, E.B.,	The dataset used in this study was obtained by crawling tweets relevant to	weighted hybrid filtering and GRU methods.	The performance evaluation yielded 85.74% precision, 88.63% recall, 88.63% accuracy, and 86.30% F1score.
			the feedback of specific accounts regarding a film		
3	Enhancing Sequence Movie Recommendatio n System Using Deep Learning and KMeans	Siet, S., Peng, S., Ilkhomjon, S., Kang, M. and Park, D.S.,	MovieLens datasets (100 Kand 1 M)	RSs model,K means Clustering	It reduces the cold-start and scalability issues but also surpasses baseline techniques in Top-N item recommendations, highlighting its efficacy in the contemporary environment of abundant data.

4	Genre Effect Toward Developing a Multi-Modal Movie Recommendatio n System in Indian Setting	Mondal, P., Kapoor, P., Singh, S., Saha, S., Singh, J.P. and Singh, A.K.,	Indian languagebased multimodal Hindi movie dataset,Flicks core dataset	GenreLike-score GLscore,Different Audio/Video embeddings	GL-Score of 89 was achieved using the Hindi movie dataset
5	Impulsion of Movie's Content-Based Factors in Multimodal Movie Recommendatio n System	Mondal, P., Kapoor, P., Singh, S., Saha, S., Onoe, N. and Singh, B.,	MovieLens-100K(ML100K) and MFVCD-7K	TextLike_score (TL_score) and Genr eLike_score (GL_score)	It is proved that the Cross-Attention-based multi-modal movie recommendation system with the proposed Meta_score successfully covers all the analytical queries supporting the purpose of the experiment.
6	Movie Recommendatio n with Poster Attention via Multi-modal Transformer Feature Fusion	Xia, L., Yang, Y., Chen, Z., Yang, Z. and Zhu, S.,	MovieLens 100K and 1M datasets	BERT model	The prediction accuracy of user ratings is improved in comparison to the baseline algorithm, by this means demonstrating the potential of this cross-modal algorithm to be applied for movie or video recommendation.
7	Collaborative recommendation model based on multi-modal multi-view attention network: Movie and literatur e cases	Hu, Z., Cai, S.M., Wang, J. and Zhou, T.,	MovieLens1M and Book-Crossing datasets	**C**ollaborative **R**ecommendation **M**odel based on **M**ultimodal multiview **A**ttention **N**etw ork (CRMMAN)	Compared with the state-of-the-art knowledge-graph-based and multimodal recommendation methods, the AUC, NDCG@5 and NDCG@10 are improved by 2.08%, 2.20% and 2.26% on average of two datasets.
8	Remembering past and predicting future: a hybrid recurrent neural network based recommender system	Bansal, S. and Baliyan, N.,	MovieLens (ML) 1M	Hybrid Recurrent Neural Network	It implicitly handles two important parameters, i.e., diversity and item popularity with a success rate of 59.22% and 34.28%, respectively.
9	Next-Generation Movie Recommenders: Leveraging Hybrid Deep Learning for Enhanced Personalization	Alsekait, D.M., Shdefat, A.Y., Mostafa, N., Hamdy, A.M.M., Fathi, H. and AbdElminaa m, D.S.,	MovieLens Dataset, The Movies Dataset, and TMDB Movie Dataset	Hybrid Deep Learning (RNN)	Integrating Contact Based Filtering, Content Filtering, and Neural Collaborative Filtering significantly improves recommendation system accuracy..

The analysis of literature focuses on the developments in fields of recommendation systems, particularly in the movie recommendations using methods such as collaborative filtering, matrix factorization, and deep learning models, viz., GRU and transformer. In any case, all these systems have certain drawbacks, which can be stated as follows. The first is the cold-start problem where the recommendation for new users or items is problematic due to a minimal history. Another drawback is scalability, because the management of huge and various data can be time-consuming. Moreover, whereas the use of hybrid models and the approaches based on multiple modalities enhances the performance of the recognition system, the employment of such methods is associated with the use of optimization algorithms and huge databases, which increases the computational load. The issue of personalization still persists since the optimization objectives depend on both the user variability and the prediction of user preferences, which will result in reduced relevance of the recommendations. Finally, in the case of social networks and user feedback incorporation, which seems advantageous, there exists an additional step which has to be taken into account while designing a model.

In this research, greedy selection pipeline will be proposed for recommendation engine with the use of multiple models. Feedback from the users of the system will be integrated into the system to generate better recommendations while the system will self-optimize in anticipation of better performance in the near future.

3 Research Methodology

To develop a multimodal recommendation system using Siamese Networks and LSTM we make use of the ML-100K . The data is loaded and then prepared for further processing. The ML-100K dataset has movie details, user ratings and information about the user. To maintain the chronological order, we sort the dataset by User ID and timestamp. We will then a sequence of (item_id, rating) pairs by grouping the User ID making sure that rating history of each user is preserved. After training the we will assess the model on the validation set. The predictive accuracy of the model is given by the mean absolute error on the validation dataset, the lower values show better performance.

Dataset – Movie Lens Dataset

The data has been previously pre-processed by the teaching assistant of the course CSE 8403-00, Mr. Kyi Thar. The pre-processed data may be found at the following link: https://github.com/kyithar/class. We move to the exploratory data analysis (EDA) and then go to the development of out pipeline model that encompasses Collaborative filtering, Machine learning based models Long Short-Term Memory (LSTM) model and Siamese networks. In this work we will give a popularity score to movies and the focus is to improve the accuracy of the model to the maximum level. Required packages for installation: The library used is the Python libraries including the pandas, NumPy, matplotlib, and seaborn. The version of Python used in the current work is 3. 6. Machine learning libraries are such as TensorFlow, keras and pytorch. Characteristics of a dataset the dataset is ml-latest and it contains information of free-text tagging activities and 5-star ratings by a movie recommending website, movielens. org. The ratings are received in the amount of 26,024,289 and the tags are applied in the amount of 753,170 for 45,843 movies. The data set was collected from the 270896 users and the collect from the January 09 1995 to August 4 2017. The dataset was formed on the date which is considered to be 04.08.2017. You can check the content of README. txt file for instructions on, among other things, the organization and encoding of the initial data set.

Content-Based Filtering

This algorithm helps in recommending items and services that the user has previously enjoyed maximizing the satisfaction of the user. Consider the following example If a user likes "Batman," the algorithm will recommend more movies in the same genre. How does the algorithm know which genres to choose from and which movies to recommend?

Netflix saves the information about its users in a vector format. The profile vector provides the information on the behaviour of the user, such as their tastes in movies and the ratings given to those films. The item vector gives information about the movies. The information about the actor, director and the genre of the movie is included in the item vector. Below are the equations for Cosine similarity, Euclidian's distance and Pearson correlation.

$$sim(A, B) = Cos(\theta) = < A, B > = \frac{A \, . \, B}{||A||||B||}$$

$$Euclidean\ Distance\ and\ Norm - 2 = \sum_{i=1}^{n} ||x_i - y_i||_2^2 \ \forall\ i\ \in \{1, 2\ 3, \ldots, n\} R^n$$

$$sim(u, v) = \frac{\sum(r_{ui} - r_u`)(r_{vi} - r_v`)}{\sqrt{\sum(r_{ui} - r_u`)^2}\sqrt{\sum(r_{vi} - r_v`)^2}}$$

Collaborative Based Filtering

This method can be better explained with an example. If a user A likes sandwiches, fries and coke and a user B like wraps, sandwiches and coke then the user B is likely to enjoy fries. They user A and user B have similar interests and are likely to enjoy similar items. The collaborative filtering algorithm makes use of the user behaviour to recommend things.

The collaborative filtering method is one of the most used algorithms by the businesses as it not dependent on any other information to function well. This algorithm calculates the similarity between the users. This similarity is then used to identify users who are similar to each other and the recommendation systems suggests the items based on the items that are liked and used by these users who are similar to each other. The same methods are used to recommend movies to the users who are similar to each other. We can predict an item for user u by calculating the weighted total of the user ratings given by other users to the item that has to be recommended. The following is the forecast $P_{u,i}$:

$$P_{u,i} = \frac{\sum_v (r_{v,i} * S_{v,i})}{\sum_v S_{u,v}}$$

The forecast of the item is given by symbol $P_{u,i}$. The $R_{v,i}$ is the rating given by a user v to a movie i. $S_{u,v}$ shows how two users are similar. This table can explain the user-movie rating matrix.

User	Movie1	Movie2	Movie3	Movie4	Movie5	Rating
A	4	1	-	4	-	3
B	-	4	-	2	3	3
C	-	1	-	4	4	3

Table 1: User – Movie Reference Matrix. This is the Matrix that plays the pivot role in the proper recommendations

This table shows a matrix of user-generated movie ratings. Let us observe the similarity between users (A, C) and (B, C) in the table to have a better understanding of it. Movies x2 and x4 are among the most frequently rated by A and C, while movies x2, x4, and x5 are among the most frequently rated by B and C.

$$r_{AC} = [(1-3)*(1-3) + (4-3)*(4-3)]/[((1-3)^2 + (4-3)^2)^{1/2} * ((1-3)^2 + (4-3)^2)^{1/2}] = 1$$

$$r_{BC} = [(4-3)*(1-3) + (2-3)*(4-3) + (3-3)*(4-3)]/[((4-3)^2 + (2-3)^2 + (3-3)^2)^{1/2} * ((1-3)^2+(4-3)^2 + (4-3)^2)^{1/2}] = -0.866$$

The correlation between A and C is two times more than the correlation between B and C. Users A and C are more similar and it is likely that the movies user A like will also be liked by User C. It requires a huge amount of processing time to calculate how each user is similar to each other and to calculate their similarity. We can deal with this by making predictions only for a limited number of users instead of all of them.

3.2.3 Item to Item Filtering

In this approach, we determine the degree to which two objects are similar to one another. We will look for similarities between each movie pair in our scenario, and then we will propose comparable movies to people who have previously expressed an interest in them based on their findings. This method operates like user-user collaborative filtering, except for one minor difference: instead of taking the weighted sum of ratings from "user-neighbours" we take the weighted sum of ratings from "itemneighbours" instead. We will now look for similarities between the two objects.

$$P_{u,i} = \frac{\sum_N (S_{i,N} * R_{u,N})}{\sum_N (|S_{i,N}|)}$$

Now that I have a better understanding of the similarities and differences between each movie and its ratings, predictions can be formed and based on those predictions, comparable movies may be recommended.

$$sim(i, j) = \cos(i, j) = \frac{i \, . \, j}{||i||_2 * ||j||_2}$$

Machine Learning Models

Instead of increasing the difficulties of personalization when attempting to scale the models outcomes for the exponentially increasing amount of internet users, Machine Learning models consist of an array of different algorithms that are constantly changing. The recommendation of products and services utilize data from the customers' historical buying behaviours, preferences and actions to estimate what they are most likely going to buy next.

K NN (k-Nearest Neighbour)

k-NN (k-Nearest Neighbour) (Li, G. and Zhang, J. , 2018, October) is a machine learning method which is unsupervised in nature. It is among the simplest of the machine learning algorithms out there. This means that, when training the given model, no assumptions are made concerning the data distribution and that the model architecture is restricted to what can be learned from the data. The aspects that depict it a type of Lazy or instance-based learning are firstly it does not require any training data samples for model building, and secondly the entire training data set is utilized in testing. In order to process k-NN, we will do these steps:

- Step 1: For every sample in the training dataset it will calculate and saves k-nearest neighbours for the subsequent comparing process. In this step it finds and stores the k nearest neighbours for all the samples in the training set.
- Step 2: In this phase, it acquires from the dataset the k closest neighbours for an unlabelled sample of the previous step. Then among those it votes for its k-nearest neighbours, it then predicts the class for any of the neighbours (majority wins in this case).

Recurrent Neural Networks and LSTM

To advance a multimodal recommendation system from the ML-100K dataset with the combined use of Siamese Networks and LSTM, the following elaborate data preprocessing steps are followed. The ratings of 100,000 tuples that involves 943 users and 1,682 movies of the ML-100K are useful for the project. We will start by reading and understanding the structure of the data which has various features like rate, time stamp, user id and movie id. Besides, we employ extra user (e. g., age, gender, occupation) and movie (e. g., genre, title) information to extract more features. Preprocessing includes the scaling of the ratings to the same range, the division of the data into the training, validation, and testing sets (which is usually in the range of 80/10/10), and the conversion of texts like the movie titles and genres to numerical arrays by means of W2V or GloVe.

Where, to find out how similar a user is to a movie, we combine these embeddings and choosing metrics like cosine similarity. The latest layer gives the likelihood that a user will like the movie, with the input layer that applies the sigmoid or softmax function based on the problem. During the training of the model, it is necessary to select an appropriate loss function, for example, mean squared error for regression problems (for rating predictions) or binary or categorical cross-entropy for classification problems (for predicting users' preference). A frame that enables efficient training is created by optimizers such as Adam or RMSprop. In the training process, there is a validation set used for tuning the hyperparameters and the early stopping technique to prevent model over-training.

3.5 Siamese Network

Separate encoders are defined for each type of embedding: `encoder_user` for the user embeddings, `encoder_item` for the item embeddings And `encoder_video`, `encoder_audio`, `encoder_text`, `encoder_meta` for its five modules. To improve feature representation attention mechanisms are incorporated. Inside each modality, self-attention allows standing out the most relevant features regarding the specific modality; instead, attention between the different modalities enables the model to appreciate how each modality depends on the other. This entails the formulation of query, key, and value tensors to calculate the attention weights and gather information. It is processed together with user embeddings in the "Siamese network". It has several dense layers using LeakyReLU activation to learn compact representations for the subsequent comparison. The last procedure involves a feedforward network (FFN) with input being concatenated values obtained from Siamese network to give rating prediction. Finally, this FFN uses linear layers together with ReLU activations to scale the embeddings to the rating range of {0, 1, 2, 3, 4, 5}.

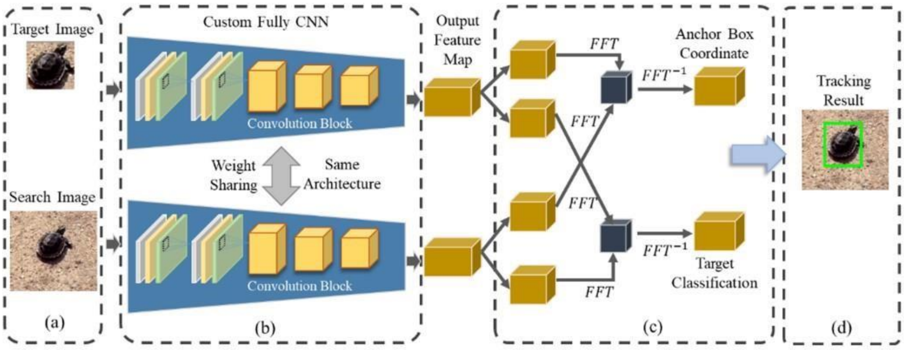


Figure 3: An example of the Siamese network how it works in order to give a prediction result (Source: Lim, S.-C.; Huh, J.-H.; Kim, J.-C 2023)

The model is optimized using a combination of “loss functions”: Weighted MSE Loss to adjust each class’s weight according to the number of instances from each class, and, Cosine Embedding Loss to improve the distance of items and users that share close distances. In the “training and evaluation” phase, the Adam optimizer is utilized in optimizing the model concerning the two loss functions.

3.6 Sentence Transformers

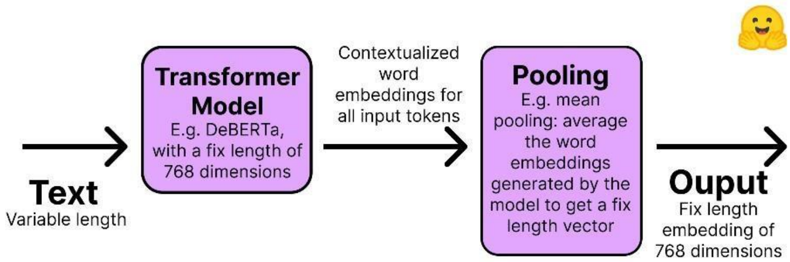


Figure 4: Functioning of a Sentence Transformer (Source: Hugging Face)

This transformer converts texts and picture pixels of different lengths into a embedding of a constant size which can give us the meaning of the input. The first layer has a pre-trained Transformer model that is used to process the input text. The model makes use of a base called as "distilroberta-base" . There is an embedding for every word token in the text which is produced by the transformers as an output for all input tokens. The second layer has a pooling layer which gathers all the embeddings into a single embedding of a specified length. Mean pooling can be used to calculate the average of the model's embeddings.

4 Implementation

In this section, we will discuss the different techniques that we have taken into the account for the implementation of recommender engines. The files contain information on MovieLens, a movie recommendation service, including its ratings and free-text tagging activities. A total of 27278 movies are represented by 20000263 ratings and 465564 tag application applications. Between January 09, 1995, and March 31, 2015, 138493 unique users contributed to the collection of this information. This dataset was created on October 17, 2016, according to the most recent available data. Users were chosen for inclusion at random from a pool of applicants. All of the individuals that were chosen had rated at least 20 films. There is no demographic information provided. A unique identifier is assigned to each user, and no additional information is shared.

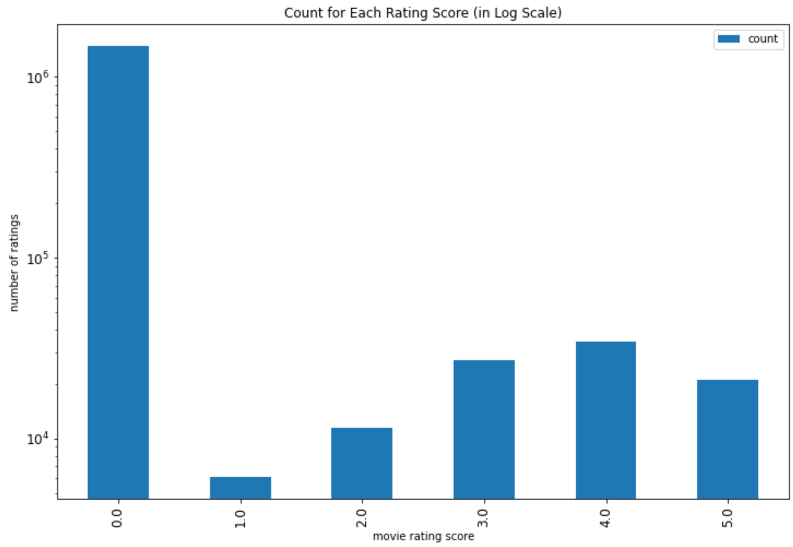


Figure 5: Count of Each Rating Score

movie.csv is a CSV file that provides information on movies with movieId, title, genres. link.csv is a CSV file that includes IDs that may be used to link to other sources with movieId, imdbId, tmbdId. The file genome scores.csv provides movie-tag relevance data, with movieId, tagId, relevance. The file genome tags.csv contains the following tag descriptions as tagId, tag. In this scenario, we have used the Google Collab platform to develop this whole analysis and used the above-described data for building the model.

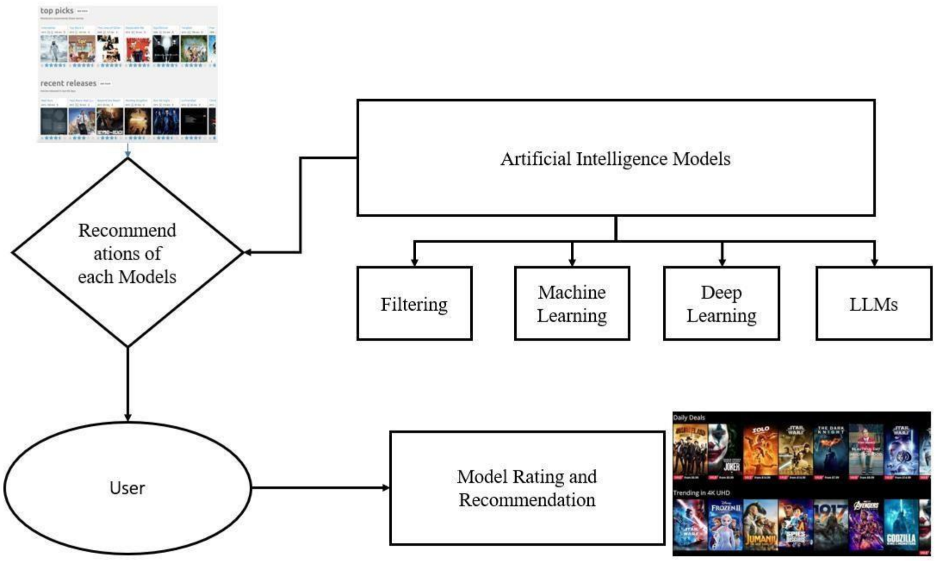


Figure 6: Proposed solution for training the movie lens data with different models

---

Algorithm Flow

Step 1: Dataset Acquisition: The dataset that will be taken for the analysis of the model consist of two types,

Standard Dataset which can be either scrapped or researched. In this case to train the model we have used the MovieLens dataset.

Step 2: Machine Learning Modelling:

a. In this step the first part is the training and testing the dataset samples for the model to fit
b. Training the model to learn the insights
c. Fix the model

Step 3: Recommendation in the real time: Once the model has been trained, we will check how the model performs in the real time. Two concepts will be used,

a. Distance Matrix
b. How the options are close to one's interest

Step 4: Deployment: The trained model is connected to some Platform for the real time Use

Step 5: Feedback: The performance of the model for the second criterion of Step 3, can be used as factor of likeliness and dislikeliness to re-tune the model and continue with the step 4.

Dataset Acquisition

This project makes use of a dataset which is a small package of data and a dataset which is obtained from live data. The MovieLens dataset is used to build the recommendation system. It is one of the most used and is publicly available dataset and has information about the user ratings, genre and other information of the user. Real and rich datasets are required as the benchmark for this approach because it covers almost all categories, and this dataset fulfils the requirement well, in addition to being widely used in academia and industries'.

Training and Testing the Dataset

The acquired datasets will have the training sets and the testing sets depending on the goals of the model. Conventional division of data is the 80/20 division whereby 80% of the data is adopted for training with the remaining 20% being used for an evaluation. Other ways of validating the model may include kfold validation whereby the data is split into k subset and each subset is used once to test the prediction model while the other folds are used for training the model.

The models to be trained on the training data shall be collaborative filtering, LSTM based modelling and a matrix that involves enhanced filtering using sentence transformers. This way, each of the models will be devoted to reflecting certain parameters of users' preferences: the degree of their previous interactions (the collaborative filter), temporal relations between actions (LSTM), and semantic proximity between texts (sentence transformer).

Machine Learning Model Training

Each model will have to go through a cycle of training in which the model learns to make predictions about the users' preferences given the inputs. Thus, whereas collaborative filtering models will learn by detecting patterns in the user-item interaction, LSTM models will learn from the sequence in which a user has watched a particular movie. The user reviews and the description of the movies will be transformed into a similar high-dimensional space that will capture the semantic relations. More on training, hyperparameters will be adjusted to allow the model to perform as required.

Model Fixing
After training and evaluation on the test set, the best performing model (or model combination) will be 'finalized' for operational use. The model will be trained on the training set and its performance will be tested on the test set so that it optimally works on unseen data. In case of satisfactory performance, the model will be firmed which implies that the model is ready to be deployed to a live environment.

Real-Time Recommendation
As soon as the model is trained and fixed, the following phase would be to check the model in real-time processing. This involves evaluating the movies recommended by the system based on the preference of the user.

Distance Matrix: The elements in the recommendation system of the work will be based on the distance matrix that compares differences between various movies in real-time. This matrix will be computed by determining the (for instance) distance between two vectors that depict movies. Such features could be genres, ratings, users' preferences and even Embeddings that can be obtained using a sentence transformer. Distance matrix will in effect assist the system to find out movies the user is likely to like based on the previous choices made.
User Interest Proximity: The system will then rate the level of relevance of the recommended options with the user's current interests. This is in regard to the user's recency, including movies he or she has been watching recently, ratings given to certain films and genres of movies he or she has been inclined towards. The system will sort the involved recommendations, based on a distance matrix as the repository will be made closer to the user's interests. The use of the given approach contributes to the fact that not only are the recommendations introduced relevant for the user, but also corresponds to their mood and preferences at the time of their interaction with the application.

Deployment
Having trained the model and testing the real-time efficiency, the subsequent phase is to take the model into live systems, available to the users. The specified model will run online, providing recommendations non-stop as people engage with the app. It is necessary to give real-time data feeds and respond the request by the user and to provide recommendations with minimum latency. Real-time operations can be achieved by hosting the system on cloud services such as AWS or Google Cloud; infrastructure can be implemented on infrastructure using Kubernetes to support the containers along with load balancing. Furthermore, real time databases like Redis or Elasticsearch can be used to store data and pull data as the need arises.

5 Result and Analysis

5.1 Using Filtering Models - Content-Based Recommender

The "**movie description-based recommender**" depends on summaries and slogans, to make a set of movies which are similar. For instance, when querying for '*The Godfather*', the top suggestions included 'The Godfather: 'Part II', which is suitable for lovers of black and white crime dramas. In the same way, the powered suggestion of 'The Dark Knight' include 'The Dark Knight Rises' which has the themes belonging to the Batman series.

```
get_recommendations('The Godfather').head(10)
```

	title
973	The Godfather: Part II
8387	The Family
3509	Made
4196	Johnny Dangerously
29	Shanghai Triad

```
get_recommendations('The Dark Knight').head(10)
```

	title
7931	The Dark Knight Rises
132	Batman Forever
1113	Batman Returns
8227	Batman: The Dark Knight Returns, Part 2
7565	Batman: Under the Red Hood

Figure 7: Content Based Filtering results for two user inputs

The "**metadata-based recommender**", gives recommendations based on cast, crew, genres and keywords and came up with good recommendations. The suggested movies related to 'The Dark Knight' were other Batman movies, while 'Mean Girls' suggested other high school and teen movies.

```
get_recommendations('The Dark Knight').head(10)
```

	title
8031	The Dark Knight Rises
6218	Batman Begins
6623	The Prestige
2085	Following
7648	Inception

```
get_recommendations('Mean Girls').head(10)
```

	title
3319	Head Over Heels
4763	Freaky Friday
1329	The House of Yes
6277	Just Like Heaven
7905	Mr. Popper's Penguins

Figure 8: Impact of addition of more features

Just to overcome the gaps of basic recommendations, the enhancement filter 'popularity and ratings' was added. This adjustment was expected to improve recommendations by increase rating and popularity of the recommended films.

```
improved_recommendations('The Dark Knight')
```

	title	vote_count	vote_average	year	wr
7648	Inception	14075	8	2010	7.917588
8613	Interstellar	11187	8	2014	7.897107
6623	The Prestige	4510	8	2006	7.758148
3381	Memento	4168	8	2000	7.740175

Figure 9: Result of similarity and weights matrix

It is important to outline that these systems often exaggerate the concept of attributes of metadata. So using cast, crew, and keywords could be helpful, but at the same time, it could lead to a situation where recommendations which are given are not quite related to context. The actual recommendations could have two movies labelled with the same keywords but can belong to different genres, which is not useful for users who want to watch a particular type of movie.

5.2 Using Filtering Methods – Collaborative Filtering

Collaborative Filtering improves recommendations from the aspect of users' behaviours. The value of the predicted rating for user 196 for movie 926 was almost 4.71 . This system predicts simply the rating of a movie based on the behavioural history of the user and not the features which can give personalized recommendation. The SVD model was utilized to recommend items most relevant to each client's needs. For instance, it recommended the best 10 movies for user 196 in terms of the predicted ratings on unseen movies. The reason for this approach is to be able to recommend items that are probably the same as other users who have the same tastes.

```
Top 10 movie recommendations for user 196:
Movie ID: 926, Estimated Rating: 4.71
Movie ID: 318, Estimated Rating: 4.70
Movie ID: 3035, Estimated Rating: 4.65
Movie ID: 7361, Estimated Rating: 4.65
Movie ID: 1252, Estimated Rating: 4.64
```

Figure 10: Movie Id's generation with the User Ratings

The problem with collaborative filtering is that it can only work if you have similar preferences as another user. It does not have a feature where it can recommend things that are not necessarily related to the theme. It also depends on user interactions; therefore, it may not give optimal results with new users or items with few ratings (cold startup issue). The collaborative filtering can have problems with the recognition different tastes, it deals with universally liked items.

5.3 Using Filtering Methods – Hybrid Recommender

We presented more personalized movie recommendations using features of content-based and collaborative filtering in the hybrid recommender. It is an interface where the information about the user–the user ID, and relevant information about the movie required are input, and the system returns a list of similar movies ranked according to the user-specific predicted ratings. The mentioned approach combines the similarities between movies based on content (by calculating the cosine similarity) and their ratings forecasted by the user's collaborative filtering model. For example, while using the hybrid recommender to search for the user 1 and the movie Avatar as well as for the user 500 and the movie Avatar the system suggested different films for each of the user. The results shows that the hybrid model is better at recommending based on the preference of the user and it improves the chance of a better match.

```
hybrid(1, 'Avatar')
```

	title	vote_count	vote_average	year	id	est
1011	The Terminator	4208.0	7.4	1984	218	3.459929
974	Aliens	3282.0	7.7	1986	679	3.352664
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.312752
2014	Fantastic Planet	140.0	7.6	1973	16306	3.118456
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.055605

```
hybrid(500, 'Avatar')
```

	title	vote_count	vote_average	year	id	est
2014	Fantastic Planet	140.0	7.6	1973	16306	3.222594
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.207002
974	Aliens	3282.0	7.7	1986	679	3.203834
1011	The Terminator	4208.0	7.4	1984	218	3.179103

Figure 11: Movie Titles with Hybrid Filtering using Content based and collaborative

As depicted in the following figures, it was evident that the hybrid recommender proposed diverse suggestions for the same movie depending on the user – a clear indication of its ability to combine the content-based similarities and the ratings given by the specific viewer. This helps in having a more comprehensive solution which helps in overcoming the limitations of each method.

5.4 Using Machine Learning Based Models

In this context, we adopted and deployed a K-nearest neighbour (k-NN) recommendation system with regard to users' ratings and movies' metadata. The system employs the collaborative filtering model to find movies that are similar to a given movie by calculating a cosine similarity of movies forming the rating matrix.

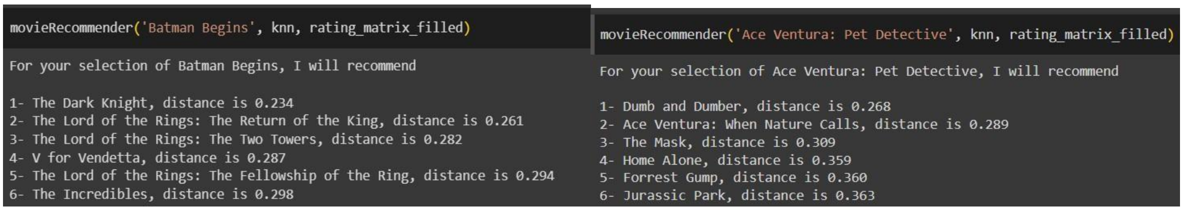


Figure 12: K-NN based recommendation

If we query the recommender with Ace Ventura: The query Pet Detective or Batman Begins returns the title and link to other films similar in terms of rating based on the cosine similarity. The output has a list of recommended movies along with the distance measure, We can calculate how close the recommendations are to the input movie in terms of rating space.

5.5 Using LSTM based Models

After defining the model architecture, we compile the Adam optimizer and using the mean squared error (MSE) as the loss function. The mean absolute error (MAE) is used to measure performance.

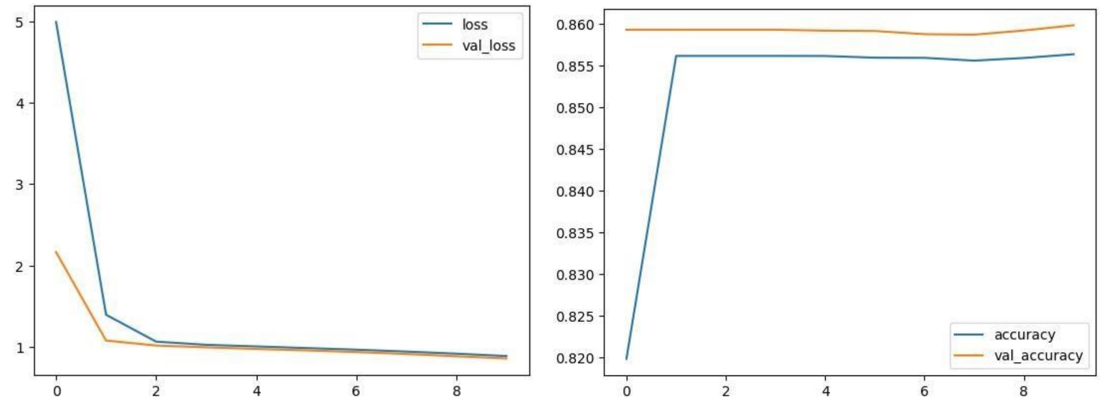


Figure 13: Loss and Accuracy graphs for LSTM

The model is trained for 10 epochs on the training dataset with the batch size equals 32. In this process, the validation set is used to check and prevent overfitting of the model as it commences to learn patterns beyond the scope of sets A and B. As part of the learning algorithm in the training phase the data is continually fed through the model to adjust the parameters in order to minimize the objective function. Upon the model training is done, the performance of the model is tested on the validation set, and the most preferred measure using the degree of accuracy is the MAE. MAE is inversely related to accuracy, therefore the lower the values of MAE, better the model. The steps of data preprocessing and modeling by LSTM together with high-quality evaluation allow killing the full max potential of the ML-100K dataset in creating a stable and efficient recommendation system. The loss curve shows a descending trend on both training loss as well as the validation loss across the epochs, which signifies the model's better fitting and convergence. The values of the metrics show that the training accuracy is plateauing at approximately 0. 855 and validation accuracy at 0. 850.

User 1: ['Toy Story (1995)', 'L.A. Confidential (1997)', 'Star Trek: First Contact (1996)', 'Men in Black (1997)', 'Broken Arrow (1996)', 'Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963)', 'Kolya (1996)', 'Hunt for Red October, The (1990)', 'Aladdin (1992)', 'Silence of the Lambs, The (1991)']

5.6 Siamese Networks

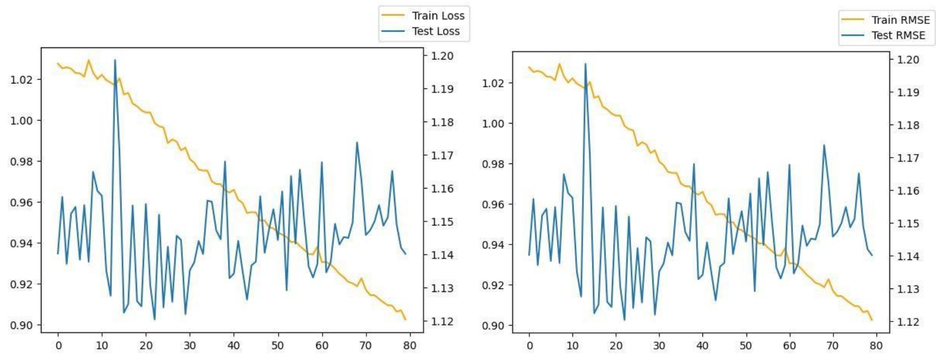


Figure 14: Training performance graphs for siemese networks

The RMSE curve shows the fluctuation train and test RMSE values across epochs. Despite the oscillations, both RMSE values generally decrease, indicating the model's gradual improvement with some instability, possibly due to high learning rates or data variability. The precision curve shows fluctuating train and test precision values across epochs. Despite the oscillations, both precision values generally improve, indicating the model's gradual enhancement with some instability, possibly due to high learning rates or data variability.

User 1: ['2001: A Space Odyssey (1968)', 'Close Shave, A (1995)', 'Some Like It Hot (1959)', 'Monty Python and the Holy Grail (1974)', 'Duck Soup (1933)', 'Wizard of Oz, The (1939)', "It's a Wonderful Life (1946)", 'Silence of the Lambs, The (1991)', 'Princess Bride, The (1987)', 'Butch Cassidy and the Sundance Kid (1969)', 'Apollo 13 (1995)', 'Raging Bull (1980)', 'Fargo (1996)',

5.7 Sentence Transformers

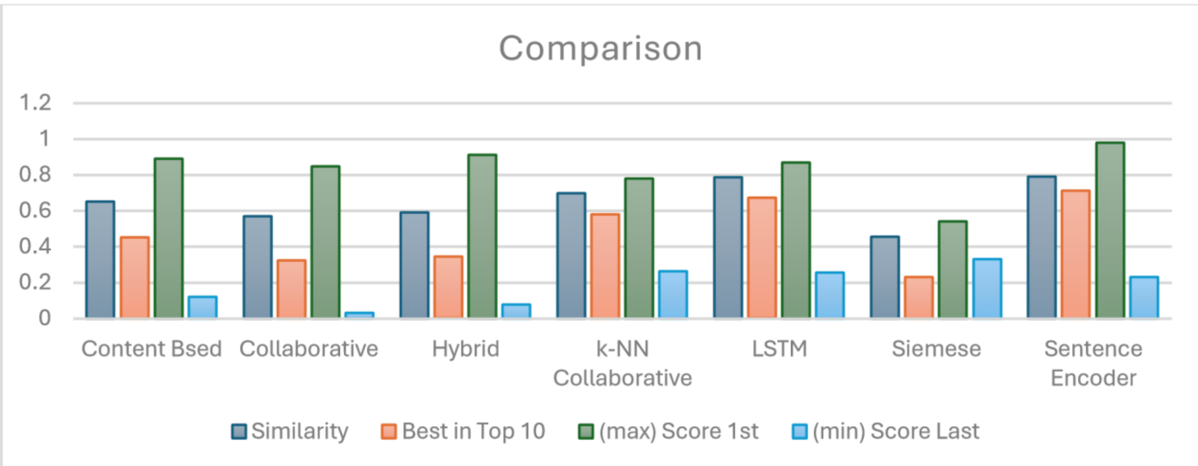
The Sentence transformer makes use of the ability to use artistic and thematic components of a movie to improves the process of recommending a movie. For instance, this model elevates the prominence of "V for Vendetta" positions it with 0. 287 distance and 'The Incredibles' with 0. 298 distance. Many of such movies like "Batman Begins" have elements of heroism, dark tales and characters' transformation, which categorize them as being closer in theme to what the user likes. In the cases of summary and top suggestions of the term transformer model, it is striking that "The Lord of the Rings" is not included. This exclusion even goes further to depict how this approach can erase choice that is completely different thematically hence providing a more tailored suggestion.

```
1- The Dark Knight, distance is 0.234
2- The Lord of the Rings: The Return of the King, distance is 0.261
3- The Lord of the Rings: The Two Towers, distance is 0.282
4- V for Vendetta, distance is 0.287
5- The Lord of the Rings: The Fellowship of the Ring, distance is 0.294
```

Figure 15: Recommendations in Sentence Transformers

Further, the deviations in the distances among the recommended movies give more information. For instance, the distance ‘’0’’ can be observed in the movie ‘Spider-Man 2’. Innovativeness is assigned the 320 in both models which suggest that its significance is well understood across the two models. However, the reduction in distance observed in the case of the sentence transformer model of wrecks like “V for Vendetta”, “The Incredibles” and the like are indicative of a higher association with the probable user choice. The reason for this is that such films are more semantically related to ‘Batman Begins’ than some of the options that take the higher rank according to the collaborative filtering method.
In general, when applying the model supported by the transformer for sentences, the capacity to improve recommendations compared to the one based only on rating matrix turns out to be higher since in addition to the user ratings, the model takes into account the similarities in the narrative and the materials contained in different films. This, in turn, results in the production of a more fine-grained and contextually relevant set of recommendations that is going to closely match the user’s expected preferences given the choice of “Batman Begins.”

6 Evaluation



These models are compared based on various factors like Similarity, Best in top 10, Max for Score 1st and Min for Score Last. To elaborate the similarity score is the value that gives how the similarity of the recommended movie is to the taste of the user. Best in top 10 gives talks about how many movies did the recommendation system give that was with respect to the users interest. Max score 1st talks about the maximum score the top recommendation movie had which was suggested by the recommendation system. Min score last talks about the minimum score the last movie had that was suggested from the recommendation system. We can observe that tge sentence encoder has performed very well.

The sentence encoder has one of the highest similarity of about 80 percent followed by LSTM and KNN. Sentence encoder performs the best in Max Sore first having a value of close to 100 percent followed by hybrid recommender system that has a value of about 95 percent. Sentence encoder has the highest score in best in top 10 at around 65 percent and LSTM has around 63 percent.

Adding a rating score model into the three previous models comprising of collaborative filtering, LSTM-based modelling, and a filtering matrix with the improvement of a sentence transformer is an improvement to the pipeline of recommendation. The rating score model will act as a feedback tool that allows users to rate the recommendations resulting from each of the three methods. This approach also solves the problem of each man to his taste in movies, which will result in a variety of reactions for the same suggestions.

7 Conclusion and Future Work

However, to analyze user input results one ought to use Large Language Models (LLMs) for the following recommendations. Thus, the language learning machines (LLMs) have capabilities of computing and understanding the complex natural language data and can easily recognize the latent relationships and structures that exist between the users' preferences and movie types. In so doing, this integration is likely to bring about more complex models that will adequately explain the moderating features of the user preferences in regards to the suggestions provided.

Last of all, when the rating score model is incorporated, and improved information regarding the users is available, one can consider the possibilities of incorporating continuous learning in the recommendation system. This can be done by updating in real time so that the model is the most recent data of the users. It also identifies the changes in the users interests and make sure the recommendation is dynamic. Taking into account all the conditions, the recommendation of the rating score model, combined with the calculating and learning capabilities of the LLMs, constitutes a mise-en-place as a strategy to enhance the efficiency of the reconsideration and the importance of the recommendation system. Through these enhancements, it will be possible to maintain the high reactivity our system shows to users' choices, creating satisfaction.

References

Akerkar (2019). Recommendation Engines. In Artificial Intelligence for Business (pp. 41-52). Springer, Cham.

Alam & Awan (2018). Domain analysis of information extraction techniques. Int. J. Multidiscip. Sci. Eng., 9, 1–9.

Alsekait et al. (2024). Next-Generation Movie Recommenders: Leveraging Hybrid Deep Learning for Enhanced Personalization. Appl. Math, 18(5), 957-981.

Amato et al. (2017). Kira: A system for knowledge-based access to multimedia art collections. In Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 January–1 February 2017; pp. 338–343.

Amato et al. (2018a). Multimedia story creation on social networks. Future Gener. Comput. Syst., 86, 412–420.

Amato et al. (2018b). Multimedia summarization using social media content. Multimed. Tools Appl., 77, 7803–17827.

Aramuthakannan et al. (2023). Movie recommendation system via fuzzy decision making based dual deep neural networks. Journal of Intelligent & Fuzzy Systems, 44(3), 5481-5494.

Azevedo & Santos (2008). KDD, SEMMA and CRISP-DM: a parallel overview. IADS-DM.

Bansal & Baliyan (2023). Remembering past and predicting future: a hybrid recurrent neural network based recommender system. Journal of Ambient Intelligence and Humanized Computing, 14(12), 16025-16036.

Batagelj (1990). Norms and distances over finite groups. Inst., Department, Univ.

Bazai & Jang-Jaccard (2020). In-Memory Data Anonymization Using Scalable and High Performance RDD Design. Electronics, 9, 1732.

Bazai et al. (2021). Scalable, High-Performance, and Generalized Subtree Data Anonymization Approach for Apache Spark. Electronics, 10, 589.

Chen et al. (2021). Improving the Prediction Quality in Memory-Based Collaborative Filtering Using Categorical Features. Electronics, 10, 214.

Greub (2012). Linear algebra (Vol. 23). Springer Science & Business Media.

Harper & Konstan (2015). The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19.

Hu et al. (2023). Collaborative recommendation model based on multi-modal multi-view attention network: Movie and literature cases. Applied Soft Computing, 144, 110518.

Khadse et al. (2018). Recommendation Engine for Predicting Best Rated Movies. Int. J. Adv. Sci. Techn., 110, 65–76.

Khalid & Zebaree (2021). Big data analysis for data visualization: A review. Int. J. Sci. Bus., 5, 64–75.

Li & Zhang (2018). Music personalized recommendation system based on improved KNN algorithm. In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 777-781). IEEE.

Lim et al. (2023). Siamese Trackers Based on Deep Features for Visual Tracking. Electronics, 12, 4140. https://doi.org/10.3390/electronics12194140

Meng et al. (2016). Mllib: Machine learning in apache spark. J. Mach. Learn. Res., 17, 1235–1241.

Mondal et al. (2023a). Genre Effect Towards Developing a Multi-Modal Movie Recommendation System in Indian Setting. IEEE Transactions on Consumer Electronics.

Mondal et al. (2023b). Impulsion of Movie's Content-Based Factors in Multi-modal Movie Recommendation System. In International Conference on Neural Information Processing (pp. 230-242). Singapore: Springer Nature Singapore.

Morfino & Rampone (2020). Towards Near-Real-Time Intrusion Detection for IoT Devices using Supervised Learning and Apache Spark. Electronics, 9, 444.

Myung & Yu (2020). Performance Prediction for Convolutional Neural Network on Spark Cluster. Electronics, 9, 1340.

Rehma et al. (2018). Comparison and Evaluation of Information Retrieval Models. VFAST Trans. Softw., 6, 7–14.

Sánchez-Moreno et al. (2020). Exploiting the User Social Context to Address Neighborhood Bias in Collaborative Filtering Music Recommender Systems. Information, 11, 439.

Shen et al. (2020). Collaborative filtering-based recommendation system for big data. Int. J. Comput. Sci. Eng., 21, 219.

Siet et al. (2024). Enhancing sequence movie recommendation system using deep learning and kmeans. Applied Sciences, 14(6), 2505.

Valentino & Setiawan (2024). Movie Recommender System on Twitter Using Weighted Hybrid Filtering and GRU. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 159-172.

Xia et al. (2024). Movie Recommendation with Poster Attention via Multi-modal Transformer Feature Fusion. arXiv preprint arXiv:2407.09157.