A Deep Learning Recommender System for Anime

M.Sc. Research Project
Data Analytics

Vidyashree Mahaling Mutteppagol
Student ID: x19241283

School of Computing
National College of Ireland

Supervisor: Prof. Dr Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vidyashree Mahaling Mutteppagol
Student ID:	X19241283
Programme:	M.Sc. in Data Analytics
Year:	2021
Module:	M.Sc. Research Project
Supervisor:	Prof. Dr Christian Horn
Submission Due Date:	16/08/2021
Project Title:	A Deep Learning Recommender System for Anime
Word Count:	8132
Page Count:	21

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:	
Date:	16th August 2021

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):	

A Deep Learning Recommender System for Anime

Vidyashree Mahaling Mutteppagol
x19241283@student.ncirl.ie

Abstract

The extensive market available and the rising popularity of the anime industry necessitate a dedicated research study to help build a recommender system that can generate personalized and accurate anime recommendations for users. The prominent machine learning models such as the k-nearest neighbours, content-based, collaborative filtering, autoencoders and hybrid models, that are often used in recommender systems have a huge potential for development. These models experience data sparsity and cold-start issues and there is an opportunity for improvement in their predictive accuracies as well. As a result, a recommender system with a Deep Learning collaborative filtering-based model is proposed in this research which provides anime recommendations that are highly relevant to the users' likes and interests. This model will pre-process the data and transform it employing the techniques of embedding and batch normalization. The anime dataset used in the research is obtained from the open-source platform - Kaggle. The top anime recommendations are generated for the user in three cases - User-based CF model, Item-based CF model and our proposed Model-based CF using Deep Learning and the results are qualitatively measured and found to be very close to the interests of the users. In addition to generating the anime recommendations, the proposed model is evaluated using the metrics of Mean Squared Error (MSE) and Mean Absolute Error (MAE). The closer to zero values obtained for these metrics of MSE and MAE indicates that the model performs efficiently.

Keywords: Deep Learning, Collaborative Filtering, Anime, Data Sparsity, Recommender Systems, Cold-start

1 Introduction

1.1 Background

The volume of digital information at one's disposal has increased at an unparalleled rate since the inception of the internet. The task of separating useful data from heaps of useless data is becoming arduous with such vast amounts of information. Despite the fact that the search engines such as Google and Bing have been greatly perfected to make relevant recommendations, they fail to provide content that is highly personalized which is tailored to the unique likes and interests of individual users. This forms the primary reason for the increased popularity of the Recommender Systems (RSs) since the previous decade.

Currently, the RSs are being used in every domain, from recommending music, movies, books, e-commerce products to even making suggestions for the queries themselves when people seek answers from the internet regarding anything. In addition to making the lives of the end-users much simpler, this technology of RSs also helps all the affiliated businesses by increasing their revenue. When the market is this large, it only makes sense to strive to enhance the accuracy of the recommendations made.

Although a number of machine learning models have been employed to date to build the RSs, the one that has sparked a lot of research and attention in recent days is the use of Deep Learning models. Because of their competency in processing complicated data in multiple forms like audios, text, videos and pictures, a plethora of Deep Learning-based models for RSs has been developed in recent years. The application of Deep Learning in RSs also helps further improve the prediction accuracy and the trustworthiness in the recommendations made by retrieving the otherwise hidden significant features.

1.2 Motivation

One of the main objectives of building RSs is forecasting the future items that are most likely for the dataset in question. The output of this prediction can take the form of a numeric rating, or a binary score or a ranked list of recommended items. The numeric rating form of output is used by well-known tech companies like Netflix and Amazon. As a result, there is a lot of room for bettering the predictive accuracy of recommender systems. This also helps in the growth of the organizations because recommendations that are highly accurate lead to satisfied customers and their continued association with the products and services provided. There's also a huge scope for gaining profit by using efficient recommender systems with high predictive accuracies in the anime industry since it has a mass following in the area of entertainment. Studies on the anime industry estimated that the anime was worth USD 24.23 billion in the year 2020, and is anticipated to be worth USD 43.73 billion by the year 2027 [1]. The estimates of this study are an excellent indicator of the value that the anime industry holds and how building an RS that provides highly accurate recommendations can be a profitable venture. These estimates also suggest that, with the anime industry turning into such a strong market, the number of anime television series and movies produced each year would skyrocket. It will only make the task of discovering anime that appeals to an individual's tastes far more confusing and laborious. Therefore, designing and building a Deep Learning RS that addresses the issues of cold-start and provides highly accurate recommendations to the users is an intriguing and profitable research topic.

1.3 Target Users

This research aims in building a RS employing a Model-based Collaborative Filtering using Deep Learning technique that can generate anime recommendations to the users with great predictive accuracy. This will be of great benefit to the anime fans and the active members of the anime community since this RS can help them obtain high-quality suggestions based on their individual preferences and similar users in the community. This model could further be employed to boost the sales of the subsidiary businesses resulting from the anime industry, which mainly sell anime-related goods or streaming services.

[1] https://brandessenceresearch.com/technology-and-media/anime-market-size

1.4 Research Question

Is it possible to build a Model-based Collaborative Filtering recommender system for anime using Deep Learning?

1.5 Research Objectives

The following are the objectives of this research to help answer the research question presented in the above section 1.4:

1. *Design a Model-based Collaborative Filtering recommender system using Deep Learning that can provide highly relevant recommendations to the user based on their previous rating history and also similar users as them. In addition to that, predict the user's ratings to the animes recommended.*

2. *Does the change in activation function used in the Adam optimizer for the Deep Learning in the proposed recommender system influence the mean standard error and the recommended anime? How are the running times impacted?*

1.6 Justification and Scope

Though there have been numerous models built for recommending anime, there continue to exist problems of cold-start and data-sparsity. By employing the Deep Learning approach to collaborative filtering technique and effectively utilizing the data of similar users, the proposed model successfully the problems of data-sparsity and cold-start. In accordance with the computing power available, the number of hidden dense layers in the model can be increased to further enhance it's performance.

2 Related Work

2.1 Introduction

The success of any Recommender System (RS) relies on how relevant are the recommendations made to the user. And with the exponential rise in the magnitude of the data in recent days, this task is only getting arduous. On the bright side though, this has led to the development of highly advanced algorithmic models for the RSs. Zhang et al. (2019) classify the currently used RSs into three comprehensive categories – the content-based, the collaborative filtering and the hybrid RSs. Attempts to solve the problems in conventional RSs such as the data-sparsity, cold-start etc, resulted in the development of the Deep Learning models. Not only did these Deep Learning models allow the use of varied data types like audio, video and images along with the text, they were also easily scalable to huge volumes of data. The subsequent sections present and critically review the various Deep Learning models employed in the RSs today.

2.2 Model-Based Collaborative Filtering Recommender Systems

A novel model employing the Alternating Least Squares (ALS) method and using the CF technique as the base for recommending anime tv shows and movies, was proposed by Girsang et al. (2020). The ALS method helped match the history of the users. By normalizing the movie ratings in the dataset, the problem of cold-start was effectively addressed. And the SimRank tool helped tackle the data sparsity issue. The gradient descent use minimizes the cost function and the parameters with small values pertaining to the users are randomized. Post these steps, the model makes recommendations on the relevant anime. The dataset used for evaluating the proposed model was obtained from the site of Kaggle with 12294 rows of data for data on anime tv shows and movies and 73,516 records for users. In addition to generating recommendations based on the similarity index of the users, the study also evaluates the RMSE (which was found to be 2.537 for the proposed model) and predicts the anime ratings. If the additional features of the users and anime could be better utilized in addition to the user history, the proposed model generate highly relevant recommendations to the users. Therefore, this study provides adequate evidence that the CF models could be employed to generate optimum recommendations for our proposed research study with the inclusion of Deep Learning methodologies.

It is found that there is either implicit or explicit dependency on the feedback data when the RSs are modelled on the user preferences. The models based on the CF methods employ explicit feedback data and in cases where there is sparsity in the rating data, the performance presented by them is sub-par. And hence, Feng et al. (2021) claim that the aforementioned reasons render them highly disadvantageous and propose a novel CF ranking model that efficiently utilizes both the implicit and the explicit feedback data. When new users are added to the system, this model successfully addresses the cold-start and data-sparsity problems. The proposed CF ranking model is a result of an amalgamation of the pairwise-ranking approach of Bayesian Personalized Ranking (BPR) and the rating-oriented approach of Probabilistic Matrix Factorization (PMF) and is termed as the Rating Bayesian Personalized Ranking (RBPR). The datasets used for the evaluation of this model are - Ciao, MovieLens1M, FilmTrust and MovieLens100k and the metrics used are Mean Reciprocal Rank (MRR), Recall, Mean Average Precision (MAP) and Precision. The RBPR model successfully alleviates the cold-start problem in the RSs and outperforms the baseline models in all of the four datasets used for its testing. Despite these impressive results, the proposed model fails when recommendations are to be made in cases where the rating number of the user is not more than one.

2.3 Autoencoder–Based Recommender Systems

With an aim to take advantage of the multiple user preferences that are nonlinear, hidden yet significant and also attain recommendations that are highly accurate, Shambour (2021) proposed an autoencoders based Deep Learning model. This model makes use of the feedforward neural networks concept employed in the recommendation algorithm called the Autoencoder-based Multi-Criteria (AEMC). The datasets used for testing and evaluating the performance of the proposed model in the study were Yahoo! Movies and TripAdvisor. The Multi-Criteria User-based CF (MC-UCF), the Multi-Criteria User-based Artificial Neural Network (MC-ANN), and the Single-Criteria User-based CF (SC-UCF) models were used as the baselines against the proposed model. On evaluation, the proposed AEMC model was found to outperform the baseline models. The AEMC model presented enhancement in the MAE by 11.73%, 19.97% and 35.51%, and accuracy in the recommendations by 8.36%, 17.32% and 27.32% over the MC-ANN, the MC-UCF and the SC-UCF models respectively.

A common problem encountered by the widely-used models like the Matrix Factorization and the CF is the data-sparsity in their rating matrices. A recommended way to address this issue is to extract the important features from the side information available on both the users and items and integrate them into the model. Any noise present during these extractions has to be carefully excluded. Based on this idea, a novel model-based RS, named the Stacked Discriminative Denoising Autoencoder based Recommender System (SDDRS) was proposed by Wang et al. (2019). This model incorporated the rating information and the significant side information available in the data into a model based on Matrix Factorization that employed Deep Learning. The proposed SDDRS model was tested on datasets of varied sizes viz. MovieLens10M, MovieLens100k and MovieLens1M. A huge variety of baseline models were used against the SDDRS model for evaluating its performance such as the Sparse Linear Method (SLIM), the Sparse Linear Methods with Side Information (SSLIM), the Probabilistic Matrix Factorization (PMF), the Collaborative Denoising Auto-Encoder (CDAE) and the additional Stacked Denoising Auto-Encoder (aSDAE), and it outperformed them all. RMSE was used as an evaluation metric and the proposed model had the lowest values among all the models tested with values of 0.6501, 0.5021 and 0.5019 for the MovieLens10M, MovieLens100k and MovieLens1M datasets respectively. This model effectively addressed the issue of data sparsity. However, the model fails to address the cold-start issues in the RSs.

2.4 Fuzzy Logic–Based Recommender Systems

Although it might not be the commonly preferred model like the CF and the content-based models in building RSs but fuzzy logic can be employed to generate realistic recommendations. This is made possible by the fact that fuzzy logic helps examine the ambiguities and the uncertainties inherent in the calculation of similarities between the items and the users. Shojaei and Saneifar (2021) propose a novel model called the MFSR that determines the similarity based on a hierarchical order. MovieLens1M and MovieLens100K are the datasets and F1, Precision, MAE and Recall are the evaluation metrics employed in the evaluation of the model proposed. The New Heuristic Similarity Model (NHSM) and the Proximity-Impact-Popularity (PIP) were the state-of-the-art models that were used as the baselines against the MFSR model. The MFSR model outperformed these models with an F1 value of 0.654 and an MAE value of 0.423 with noteworthy enhancements in precision and recall values too. The proposed MFSR model could further be enhanced by incorporating the parameters from the social media networks and detecting the levels dynamically when multi-level similarities are evaluated.

2.5 Recurrent Neural Network–Based Recommender Systems

Nápoles, Grau and Salgueiro (2020) proposed a novel model for building the RSs based on the Long-Term Cognitive Networks (LTCNs). LTCNs are a kind of Recurrent Neural Networks (RNNs) that employ prior knowledge structures. The proposed study excluded human involvement of any kind, and therefore, employed the Pearson's correlation coefficients in place of the prior knowledge structures that are normally used in LTCNs. This model extended the general usually LTCN model by incorporating the Gaussian-kernel neurons and also used the user-item based rating matrix. The missing values for ratings in the dataset were compensated for by using the approximate values determined by the neurons and thus completed patterns were fed to the LCTN. Since this was an RNN based model, it comprised of a process that had a learning process with three steps. In cases where the prior knowledge structure was not available the weight matrix was calculated at first, then the Gaussian neurons were trained next and then in the last step, the non-linear neurons were fixed by employing a new variant of the non-synaptic backpropagation algorithm. The baselines models used against the proposed model were the matrix factorization (MF), the neural network autoencoder (AE), the k-nearest neighbour (kNN), the variational autoencoder (VAE) and the single value decomposition (SVD). The study used three sized variants (top 50, top 200 and top 500) of these three datasets - Netflix, MovieLens10M and anime for testing the proposed model. The proposed model outperformed the baseline models in the top 50 and top 500 categories for the anime dataset, the top 50 categories for the MovieLens10M dataset and had the fewest errors in the top 50 categories for the Netflix dataset. The cold-start problem which plagues most of the RSs was solved successfully in this model. This forms a major advantage of this proposed model in its consideration for application to our research work. An area that could use help or improvement in this model is utilizing the contextual information about the items when defining the weight matrix of the LTCN model.

2.6 Hybrid Recommender Systems

In an attempt to better the predictive accuracy, and solve the problems of the RSs' inability to learn non-linear latent features, an innovative Deep Learning hybrid model was proposed by R, Kumar and Bhasker (2020). The proposed model was based on a Deep Neural Network (DNN) that was built by the consolidation of the side information present on the items and users. The learning rate of the model is inversely proportional to its weight decay. Each of the model's three hidden layers comprises a linear function, a LeakyRLU and a dropout. The issues of overfitting data are managed by the application of techniques like dropout and regularization. The cyclical variations in values through multiple epochs help better the accuracy of the model. The datasets used in this study are FilmTrust, MovieLens100K, MovieLens1M, and Book-Crossing. The model is evaluated by employing the satisficing criterion of running time and the optimizing criterion of predictive accuracy. The predictive accuracy metrics used are the Mean Absolute Error (MAE), the Mean Squared Error (MSE), the R-squared (R^2) and the Root Mean Squared Error (RMSE), and the running time metrics are the SD and the Mean. The existing models are outperformed by the proposed hybrid model in the predictive accuracy criteria in the study. With improvements of 0.7%, 6.27%, 1.2%, and 2.5% for MAE, RMSE, MSE and R^2 for the dataset MovieLens100K, the proposed model outperforms the best baseline technique employed in the model. But there is scope for improvement in the running times of the proposed model. It consumes more time for training than almost all the baseline models except for the Nearest Neighbour model. Addressing the problems in streaming the data and the rankings instead of just solving the rating predictions problems would help better the proposed model.

Although the RSs built on the memory-based CF help increase the accuracy by employing the similarity features like the Pearson Correlation Coefficient (PCC), they can't be applied to large datasets that have varying data types. To address this issue, Mu et al. (2019) propose a model with a highly enhanced similarity measure called the Common Pearson Correlation Coefficient (COPC). To preserve the monotonicity property and help better the PCC measure, the proposed model makes use of a distance function. In addition to this, the model also has two additional measures – the Jaccard value and the Hellinger Distance (Hg) which is a global similarity measure. These measures help overcome the data sparsity problem and reduce the effects caused due to the lack of co-rated items. Anime, MovieLens, BookCrossing and Jester were the datasets used in the evaluation of the proposed model. The evaluation metrics such as F1-measure, precision and recall were used in the study and their values saw a measured rise as the number of recommendations increased. Concerning all the four datasets used, the proposed model outperformed the baseline models considered. An area of improvement for the proposed model in this study is the inclusion of other salient factors which could further help enhance the performance of the RS, such as the elements of diversity, popularity and viewing times. Data sparsity, which is a major problem that plagues the RSs, is very efficiently addressed in the proposed study.

A novel, matrix-based factorization model called the Bernoulli Matrix Factorization (BeMF) was proposed by Ortega et al. (2021) to generate both the prediction and reliability values. The major difference in this model lies in the fact that instead of employing the model-based CF technique, it uses the memory-based filtering model. It doesn't use the extended architectures that are commonly used to provide the reliability values. Most RSs use the regression models to design the classification systems in them but the proposed model uses the Bernoulli distribution-based model which is a huge improvement. The proposed model was evaluated on three datasets – MyAnimeList, FilmTrust and MovieLens. The baselines models used for comparison against the proposed BeMF model were the user-based KNN and the item-based KNN which employed the similarity metric - Jaccard and Mean Squared Difference (JMSD). The evaluation metric of the confusion matrix was used which showed that the model provided correct predictions for the majority of the cases. A plot on the quality of predictions also backed up the argument that the predictions of the BeMF model were highly accurate which had predicted ratings in the confidence interval of 30% to 50%. A major shortcoming of this study is that it doesn't address the cold-start user problems despite achieving high accuracy in predictions and reliability.

2.7 Dempster–Shafer Theory–Based Recommender Systems

Dempster–Shafer theory (DST) is a general framework that is used to solve uncertainty problems. It also helps in understanding other theories like the imprecise probability theory. By employing the DST, a novel method termed, the 2-probabilities focused union was proposed by Nguyen and Huynh (2017) for application in the RSs. The information on the user preferences around the products and the services was merged in this method using the DST. The top two focal elements with the highest probabilities, represented by the user preferences' mass functions were retained while discarding the rest as noise. To test the model, the Movielens and the Flixster datasets were used and to measure the accuracy of the recommendations made, the evaluation method of Dempster-Shafer Mean Absolute Error (DS-MAE) was used. For a 24-user combination, the model obtained very low values for the Standard Deviation (SD) and the Mean. For 4-users, the SD was 0.01 and for 1-user it was 0.106. Despite this, the model has its own advantages, in that, its computation time showed huge improvements over the 2-points basepoint model, and it could combine imperfect and uncertain information with the user preferences. Not only that but the user preferences obtained from diverse sources could also be combined and used in this model. The area that needs improvements for this model is the accuracy of the recommendations generated.

Providing soft ratings is a modern, novel technique that is employed in RSs nowadays.

This technique facilitates building a model that makes use of subjective, imperfect and qualitative data concerning the user preferences and also allows for consolidation of data on preferred services and products by the users. A Collaborative Filtering (CF) based model that provided soft ratings by consolidating all the available data from the social media handles of the users was proposed by Nguyen, Sriboonchitta and Huynh (2017). The data from the social media here was employed to eradicate the problems of data-sparsity and the cold-start, by extracting the community features. To test this model, the study used the Flixster dataset and provided highly accurate results. A model with a neighbourhood size of K was used as the baseline for comparison against the proposed model in the study and the plots of DS-Precision, DS-MAE, and DS-Recall were studied. It was found that the proposed model had better stability when compared to the baseline model. Although this is a novel idea where the social media data could be used for generating soft ratings, there is a cause for concern since there is a scope for overlapping of the communities in social media platforms. The study also fails to address the grey-sheep user problems.

3 Methodology

This research will follow the methodology of Knowledge Discovery in Databases (KDD) for the implementation of the RS model. The stages in the building of the proposed model and its evaluation such as data acquisition, cleaning, pre-processing and transformation, will be performed employing the KDD methodology guidelines.

3.1 Data Extraction

The dataset used in this research work is obtained from the open-source platform of Kaggle [2]. This dataset in turn was obtained by making HTTP requests to the public APIs [3] of the MyAnimeList website [4]. MyAnimeList is a comprehensive and possibly the largest database for all the related information on the anime tv shows and movies available today. The dataset has five files (anime.csv, anime_with_synopsis.csv, animelist.csv, rating_complete.csv, watching_status.csv) with various attributes. The attributes used and their description is presented in Table 1.

3.2 Data Pre-processing and Analysis

Different kinds of data cleaning techniques have been employed on the dataset because there are inconsistencies and errors present in it. This involved removal of null values, dropping unnecessary columns and merging of tables wherever necessary. The data from the column “Aired” was used to extract only the year for exploratory data analysis purposes. The Genre column that presents the genres that were tagged to each of the anime was also pre-processed. It was a single column with comma-separated values from which all the different genres were separated. There were also a large number of duplicate ratings for the anime by the same users. They were successfully dropped from the data frame. After extensive data cleaning and processing, this dataset consisted of 62,941 unique users and 17,559 unique anime shows and movies.

[2] https://www.kaggle.com/hernan4444/anime-recommendation-database-2020

[3] https://myanimelist.net/apiconfig/references/api/v2

[4] https://myanimelist.net/

Table 1: Dataset Attributes

Attribute name	**Type**	**Description**
anime_id	Integer	ID from MyAnimeList website that uniquely identifies each anime
Name	String	Full name of the anime
Genre	String	Comma separated values identifying the genre/genres that each anime belongs to
Type	String	Identifies type of the anime such as movie, TV, OVA, etc
user_id	Integer	ID generated by MyAnimeList for its users at random
rating	Integer	Rating assigned to an anime by a particular user
English name	String	Full name of the anime in English only
synopsis	String	Text in the form of string providing the synopsis of the anime
Aired	Date	The original broadcast date of the anime

A graph was plotted to understand the trends in the release of anime movies and TV shows between the year 1910 and early 2020. It can be seen from the plot in the Figure 1 that there was an exponential growth in the number of anime released every year. Please note that the drop in the count of anime near the year 2020 is because of the lack of records in the dataset for that year.

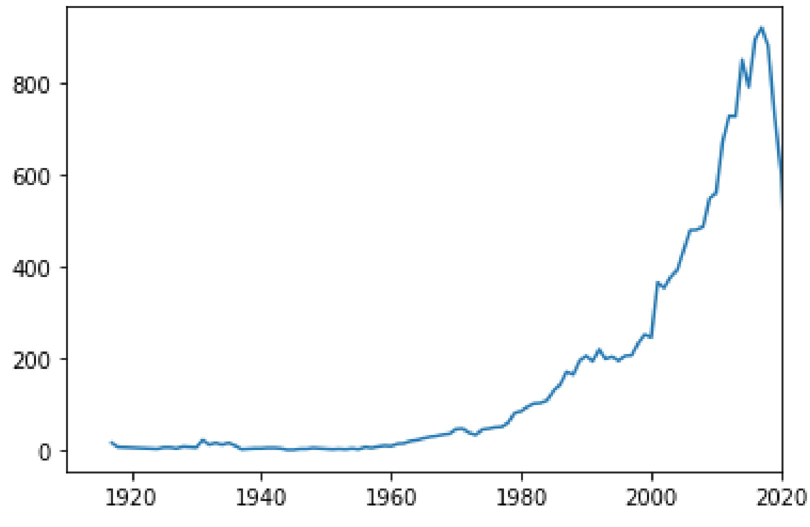

Figure 1: Plot of number of anime released from 1910 to early 2020

Once all the genres from the "Genre" column are separated during the data preprocessing techniques, to get an insight in to the top 100 genres of the anime. As seen from Figure 2, Comedy and Action are among the top common genres. If given enough data on the age group of the viewers, a lot of inferences can be drawn by correlating this data with that and business plans in the anime industry can be geared towards those profitable areas.

To better understand the distribution between the mean ratings and the count of the ratings, a joint plot between the two attributes is plotted as shown in Figure 3. It can

Figure 2: Word Cloud of the top 100 genres of anime

be inferred from this plot that for most anime, the mean ratings lie between 6.0 and 8.5 for the count of ratings ranging between 25,000 and 60,000. As the count of the rating increases beyond 100,000, we can see some outliers that have higher ratings. This can mean that the anime that are popular are also most often rated and that there might be an inherent bias present in their ratings because the opinions of the viewers are often influenced by the mass.

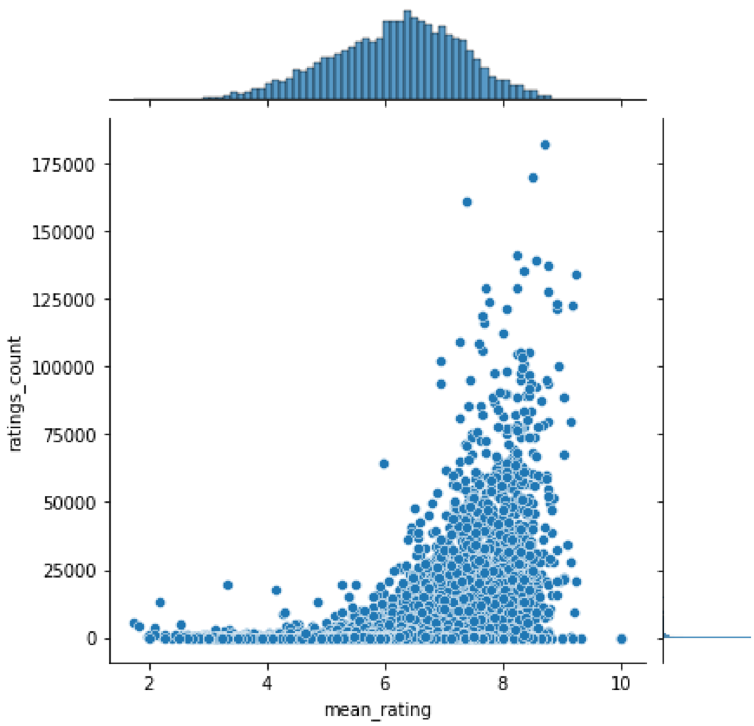

Figure 3: Plot of mean ratings Vs count of the ratings

3.3 Model Description

Based on the studies presented in section 2.2, a Model-based Collaborative Filtering using Deep Learning is proposed in this research. A traditional Collaborative Filtering RS makes recommendations or predictions for a user by amassing information or data on many users and items. This approach has an underlying assumption that, if users A and B have similar preferences for an item, then user A is more likely to have the same preferences as B for a totally different item than that of a randomly chosen user. Below are mentioned the types of the Collaborative Filtering technique.

3.3.1 Memory-based CF method

In this approach, the similarity between either the users or the items is computed by utilizing the available rating data from the users. If the similarity between users is computed, then it is termed as the User-based CF model and if the similarity between items is computed, then it is termed as the Item-based CF model. Here the similarity between the users is usually computed by either Pearson Correlation or the Cosine-based approach.

3.3.2 Model-based CF method

In the Model-based CF method, the ratings for the unrated items by the users are predicted by employing the different kinds of machine learning algorithms or data mining methods. Therefore, there is no need to calculate the similarity between the users or items in this approach. Instead, a user-item matrix is generated and this reduces the dimensionality in terms of latent factors. These rows of latent factor matrices are termed embedded vectors. Doing so effectively solves the problems caused by data-sparsity in high dimension vector matrices.

3.3.3 Proposed Model-based CF method using Deep-Learning

The application of Deep-Learning in the Model-based CF method has been a recent advancement in recommender systems. Here Matrix Factorization algorithms are used in the Deep Learning architecture. These algorithms decompose the user-item matrix into a product of two rectangular matrices that are of lower dimensionality. The first matrix consists of rows for each user and the second matrix consists of a column for every item. These rows or columns associated with particular users are termed latent factors. And at the end, the predicted ratings are computed. In mathematical terms, the predicted rating r that is computed for the user u for item i is given by this formula:

$$r_{ui} = \sum_{f=0}^{nfactors} H_{u,f} W_{f,i}$$

where, H belongs to the user-item rating matrix and is the value for user's latent factors and W belongs to the user-item rating matrix and is the value for item's latent factors.

In the proposed model of our research, we compute the predicted ratings based on the user-item matrix by employing the Keras Dense Layer of Deep Learning and generate recommendations. This is explained in detail in the sections of design and implementation.

Also, for the sake of comparisons, we generate user-based recommendations and item-based recommendations in this research, with our main objective being ranking-based recommendations with the proposed model.

3.4 Evaluation Metrics

In addition to qualitatively analysing the generated recommendations, the proposed model in this research will be evaluated to measure its performance. Two of the evaluation metrics that will be measured are given below:

3.4.1 Mean Squared Error (MSE)

The calculation that provides an estimate of the average of the squares arising when the predicted values differ from the actual values, is known as the Mean Squared Error. The values of the MSE are always positive only, and the closer those values are to zero, the better. For the proposed model in this research, the MSE is calculated as given below:

$$MSE = \sum_{i,j}^{m,n} T_{i,j}(r_{ij}^{actual} - r_{ij}^{predicted})^2$$

where, if the item j has been rated by the user i then T_ij is a matrix with a cell value equivalent to 1 or is 0, in case it is not rated. And r_ij is the rating for item j by user i in cases where the rating is present.

3.4.2 Mean Absolute Error (MAE)

The mean value of the absolute errors is given by the evaluation metric called the Mean Absolute Error. For the proposed model in this research, the difference between the actual rating of the anime (the true value) and the predicted rating for the anime (the predicted value) forms a prediction error. The better model always has a lower value for MAE. The mathematical equation for MAE is given below:

$$MAE = \sum_{i,j}^{m,n} T_{i,j}|r_{ij}^{actual} - r_{ij}^{predicted}|$$

4 Design Specification

Figure 4 represents the architecture for building a recommender system for anime by employing the proposed Model-based Collaborative Filtering using Deep Learning. This model is devised to provide the top recommendations for anime that are not yet watched by the user and also provide the predicted ratings for the same. The design consists of an input layer where the variables for the anime and the users are appropriately pre-processed and fed to the system. These variables are then embedded into vector matrices in the embedding layers and a dot product for the anime and user embedded matrices is computed. Embedding helps us find similarities between the users and the anime (which are discrete objects) and accomplishes the task of making those similarities apparent to the model. The dot product is further flattened so that it transforms into a long vector of input data that can be fed to the dense layer for Deep Learning.

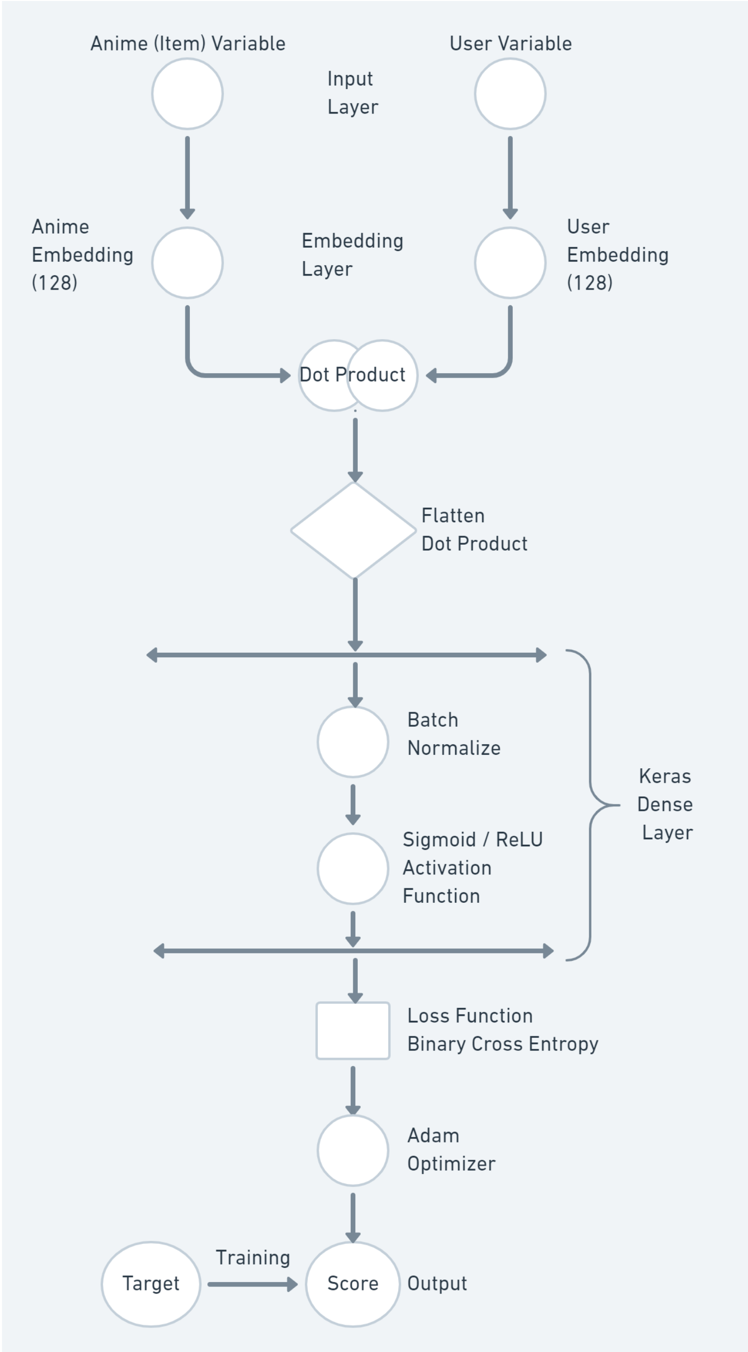

Figure 4: Proposed Model-based Collaborative Filtering Deep Learning model

The proposed model in this research utilizes one dense layer as represented in Figure 4. Depending on the requirements the hyper-parameters in this layer can be tuned. To overcome the problem of overfitting the data, batch normalization is employed here. Then to convert the linear output from the batch normalization step into non-linear output, we employ activation functions (either Sigmoid or ReLU – Rectified Linear Unit). This helps the model learn from the non-linear behaviour which is highly beneficial to the accuracy of the recommender. In addition to this, the problem of vanishing gradient is also solved by employing activation functions. When the gradient becomes zero or gets very close to zero value, the model or neural layer stops learning which is termed as vanishing gradient problem.

Sigmoid functions provide us with a weighted estimate which helps in the exact comparisons between two items. For example, an output from this function with a value of 0.734 between two items X and Y would indicate that X is 0.734 times more than Y. The sigmoid function's output is always in the range of (0,1) and is calculated by using the below equation:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The Rectified Linear Unit or ReLU function produces a zero output for all the inputs that are smaller than zero and a value of x for all values otherwise. This function requires very low resources for computation and learns faster and more evenly. It is given by the equation below:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

The output from the dense layer is then provided to the loss function that employs binary cross-entropy. This helps the model compute the cross-entropy loss that exists between the predicted values and the original or true values. The output is then fed to the Adam optimizer for training the data and the final output is obtained.

5 Implementation

5.1 Model Setup

For the implementation of the proposed model in this research, Google Collab was used. The Tensor Processing Units (TPUs) present in Google Collab is used as the runtime. Since Deep Learning models have huge computational demands for training the models, TPUs are used in this research which helps train the model much faster. So, the backend allocated in Google Collab using TPU as the runtime had the following system specifications – 12.69 GB RAM and 107.72 GB of allocated disk memory (Hard drive size). Python 3.7 is used as the programming language for the implementation of the model. For building the Deep Learning layer, TensorFlow and Keras libraries are used. For data pre-processing and the rest of the code, the following libraries are used in python – pandas, numpy, matplotlib, seaborn, and WordCloud.

5.2 Modelling of Deep Learning Collaborative Filtering Recommender System

As can be seen in the architecture presented in Figure 4, the data is subject to pre-processing by encoding and then embedded (128 bits embedding size) into two vector matrices – one for anime items and one for the users. A dot product is computed on these matrices to generate the anime-user matrix which is then flattened before being fed to the Deep Learning dense layer. The TensorFlow and the Keras libraries are used for the dense layer. A shallow dense layer employing batch normalization and Sigmoid or ReLU activation function is designed. The Adam optimizer using the binary cross-entropy as the loss function is used to train the model.

After the dense layer, we perform two experiments – first, we employ the Sigmoid activation function with 20 epochs to generate the recommendations and measure the evaluation metrics and second, we employ the ReLU activation function with 50 epochs and follow the same procedure. These activation functions help eradicate the problems of vanishing gradient and data sparsity and are also computationally cheaper. The model summary explaining the different variables factored in building this Deep Learning Model-based CF recommender system is given in Figure 5.

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
user (InputLayer)               [(None, 1)]          0
__________________________________________________________________________________________________
anime (InputLayer)              [(None, 1)]          0
__________________________________________________________________________________________________
user_embedding (Embedding)      (None, 1, 128)       8056448     user[0][0]
__________________________________________________________________________________________________
anime_embedding (Embedding)     (None, 1, 128)       2247552     anime[0][0]
__________________________________________________________________________________________________
dot_product (Dot)               (None, 1, 1)         0           user_embedding[0][0]
                                                                 anime_embedding[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 1)            0           dot_product[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 1)            2           flatten[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 1)            4           dense[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 1)            0           batch_normalization[0][0]
==================================================================================================
Total params: 10,304,006
Trainable params: 10,304,004
Non-trainable params: 2
__________________________________________________________________________________________________
```

Figure 5: Model summary with dense layer parameters

5.3 Training and Testing Details

After the initial data pre-processing, the dataset was split into training and testing dataset. The training data had 48,962,760 records and the testing data had 10,000 records. The validation loss at the end of each epoch was monitored when the data was being trained and the hyper parameters were appropriately tuned. The callback function was also employed in training the model where the regularization technique of early stopping monitored the validation loss with a patience of 3 epochs.

The parameters involved and learning details of the Deep Learning model built in this research are presented in the Table 2. Post the training and testing of the dataset, parameters of MAE and MSE were evaluated and graphs of Model-loss were plotted and customized functions were run to generate anime recommendations.

6 Evaluation

The proposed model in this research has been tested for the anime dataset by generating the anime recommendations and also by measuring the evaluation metrics - MSE and

Table 2: Parameters and Learning Details of the Proposed Model

Proposed Algorithm	**Parameter**	**Details**
	Batch Size	10000
	Adam Optimizer	Minimum Learning Rate = 0.00001, Maximum Learning Rate = 0.00005, Ramp up epochs = 5, Exponential decay = 0.8, Sustain Epoch = 0
	Regularization Technique	Early stopping while monitoring validation loss by employing a patience of 3 epochs
	Loss Function	Binary Cross-entropy
	Was the data shuffled?	Yes
Using Sigmoid Activation Function	Epochs	20
Using ReLU Activation Function	Epochs	50

MAE. The model loss was also plotted to analyse the difference between the training and testing loss. The details on the evaluation of the proposed model are as presented below:

6.1 Experiment 1 – Using Sigmoid Activation Function

In the first experiment using the Adam Optimizer, the Deep Learning layer in the proposed model is trained using the sigmoid activation function with 20 epochs. On evaluating the outcome of this model, the MSE was 0.07, the MAE was 0.1909 and the model loss was found to be 0.4842. The closer the MSE and MAE values are to zero, the better the model is. The Figure 6 shows the MSE, MAE and loss calculated in the code implementation. The Figure 7 gives the graph of the model loss where the model losses for training and test data was evaluated and plotted. This plot provides an insight into the performance trend of the model.

```
# Printing the MSE, MAE and the Loss of the model using the Sigmoid Activation function

score = model.evaluate(X_test_array, y_test)

print()
print(score)

313/313 [==============================] - 8s 22ms/step - loss: 0.4842 - mae: 0.1909 - mse: 0.0700

[0.48422756791114807, 0.1909279078245163, 0.06995591521263123]
```

Figure 6: MSE, MAE and Model loss using Sigmoid activation function

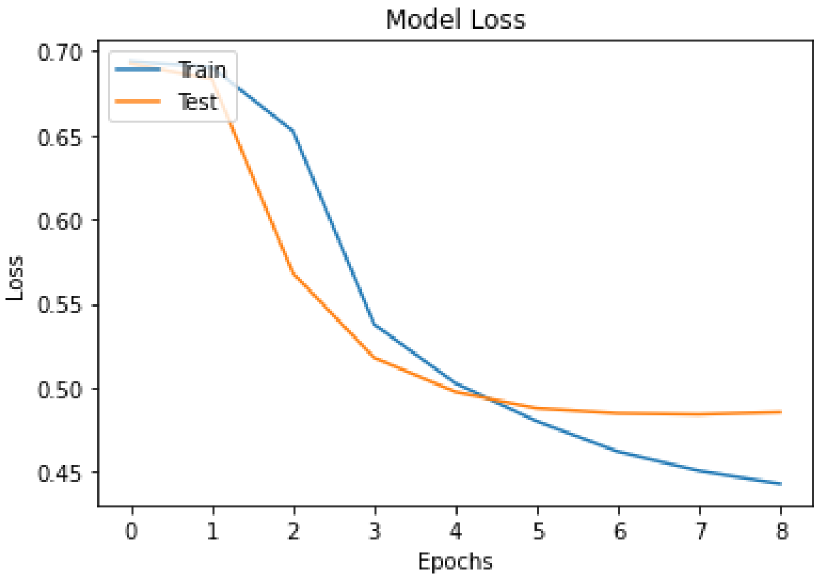

Figure 7: Model Loss using Sigmoid activation function

The ranking based recommendations for to the randomly picked user_id "100919" are the actual recommendations made by the proposed Model-based CF using Deep Learning model and the predicted ratings of the anime are also presented. This is depicted in Figure 8. Please note the column for synopsis is cropped due to heavy textual data.

```
Recommendations for user: 100919 based on rankings
==========================================================================
--------------------------------------------------------------------------
=> Top Anime Recommendations
--------------------------------------------------------------------------
```

	name	pred_rating	genre	
2	Usavich V	0.766885	Comedy	
1	The Familiar of Zero:Rondo of Princesses	0.497304	Action, Adventure, Harem, Comedy, Magic, Romance, Ecchi, Fantasy, School	Following his brave sacrifice in Blanc de La Vallière. With their venturing to a castle, appearance. The mystical and Moreover, she realizes that Sai between for
0	Kekkou Kamen	0.378737	Comedy, Ecchi	he academy Miami Takashi at identities and special detention voted "Least Likely to Survive mask, red boots and nothing els chopped pork out of the s

Figure 8: Example for ranking-based recommendations using Sigmoid activation function

6.2 Experiment 2 – Using ReLU Activation Function

In the second experiment using the Adam Optimizer, the Deep Learning layer in the proposed model is trained using the ReLU activation function with 50 epochs. On evaluating the outcome of this model, the MSE was 0.0926, the MAE was 0.2433 and the model loss was found to be 0.5530. The closer the MSE and MAE values are to zero, the better the model is. The Figure 9 shows the MSE, MAE and loss calculated in the code implementation. The Figure 10 gives the graph of the model loss where the model losses for training and test data was evaluated and plotted. This plot provides an insight into the performance trend of the model.

```
# Printing the MSE, MAE and the Loss of the model using the ReLU Activation function

score = model.evaluate(X_test_array, y_test)

print()
print(score)

313/313 [==============================] - 8s 22ms/step - loss: 0.5530 - mae: 0.2433 - mse: 0.0926

[0.5529688596725464, 0.24330636858940125, 0.09263932704925537]
```

Figure 9: MSE, MAE and Model loss using ReLU activation function

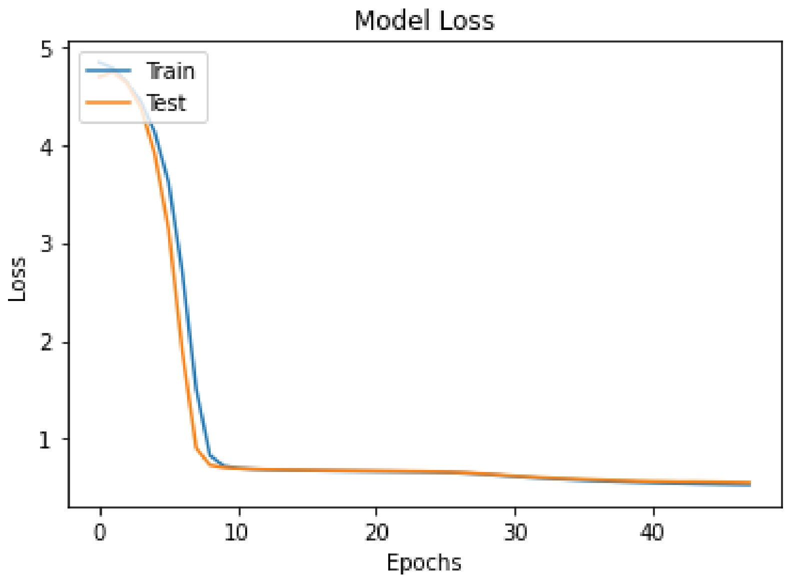

Figure 10: Model Loss using ReLU activation function

The ranking based recommendations for the randomly picked user_id "100919" are the actual recommendations made by the proposed Model-based CF using Deep Learning model and the predicted ratings of the anime are also presented. This is depicted in Figure 11.

```
Recommendations for user: 100919 based on rankings
==========================================================================
--------------------------------------------------------------------------
=> Top Anime Recommendations
--------------------------------------------------------------------------
```

	name	pred_rating	genre	
0	I''s Pure	0.565012	Comedy, Drama, Ecchi, Romance, School	Seto Ichitaka has a crush on Yoshizuki Iori club at school. But only a miracle coul been too shy to even talk to her. This mirac Welcoming Party. It seems that things start
1	Hokuto no Ken: Raoh Gaiden Gekitou-hen	0.546266	Action, Adventure, Martial Arts, Romance	The third chapter of the pentalogy (3 M
2	Igano Kabamaru	0.383127	Action, Adventure, Comedy, Romance, School, Shounen	fter the death of Saizou, Kabamaru's Ookubo, claims that she received a letter with Lady Ookubo to the big city Tokyo
4	Chuuzumou	0.372810	Comedy	Once upon a time, there lived an old man to somewhere. He followed them and s mice living in their house always
3	Galactic Whirlwind Braiger	0.358309	Action, Sci-Fi, Adventure, Space, Mecha	In the year 2111, the solar system has been are helpless. In order to battle the evi Blaster Kid, Stephen Bowie, and Angel with their robot Braiger... for a price. Mea 30 more planets for human colon

Figure 11: Example for ranking-based recommendations using ReLU activation function

6.3 Discussion on Experiments 1 and 2

The proposed recommender system employing Model-based Collaborative Filtering using Deep Learning is evaluated on the anime dataset using two separate activation functions and a different number of epochs. This helps us better understand if they impact the quality of anime recommendations generated and if there is a change in the performance of the model indicative in its evaluation metrics – MSE and MAE. The first experiment where the Sigmoid activation function is used with 20 epochs results in MSE of 0.07 and MAE of 0.1909. The second experiment where the ReLU activation function with 50 epochs is used results in an inferior MSE value of 0.0926 and MAE value of 0.2433. At first, this might not look significant but, the training and computational time, taken by ReLU are significantly higher than that of the Sigmoid activation function model. Also, the recommendations made by the Sigmoid-based model are qualitatively better than the ones made by ReLU. This is indicative in their predicted ratings for the recommended anime and can be verified by looking at the Figure 8 and the Figure 11.

7 Conclusion and Future Work

One of the main objectives while implementing the Model-based Collaborative Filtering using Deep Learning for recommending anime was to address the problem of data sparsity and cold-start. By successfully employing a Deep Learning layer in combination with the Collaborative Filtering approach, we addressed the aforementioned issues and generated top recommendations to the user with predicted ratings.

In an attempt to find the best parameters of performance for the proposed model, hyper-parameter optimization was performed. It was found that the Sigmoid activation function with 20 epochs provided the best results. An MSE as low as 0.07 and an MAE of 0.1909 was achieved. The closer to zero values of these metrics indicates that they performed best with these hyper-parameters.

Since the dataset used for generating recommendations in this research was huge and taken from the real-time website "MyAnimeList", which is an encyclopaedia of all things anime, this model can very well be used to be implemented in a real-time application for recommending anime. As a part of the future scope, the proposed model can further be improved by using the demographic information of the users and making recommendations based on the age groups of the users. It'd be interesting to see how would the recommendations differ for every user based on these factors.

Acknowledgement

I extend my sincere thanks to Prof. Dr Christian Horn for his constant guidance (even until the last minute of submission) in helping me achieve the objective of my research thesis. The meetings he held with me every week encouraging me to be inquisitive about obscure areas in recommender systems helped me make this work so much better and the process smoother. I couldn't have asked for a better supervisor.

References

Feng, Junmei et al. (2021). “RBPR: A hybrid model for the new user cold start problem in recommender systems”. In: *Knowledge-Based Systems* 214, p. 106732. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106732.

Ortega, Fernando et al. (2021). “Providing reliability in recommender systems through Bernoulli Matrix Factorization”. In: *Information Sciences* 553, pp. 110–128. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.12.001.

Shambour, Qusai (2021). “A deep learning based algorithm for multi-criteria recommender systems”. In: *Knowledge-Based Systems* 211, p. 106545. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106545.

Shojaei, Mansoore and Hassan Saneifar (2021). “MFSR: A novel multi-level fuzzy similarity measure for recommender systems”. In: *Expert Systems with Applications* 177, p. 114969. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114969.

Girsang, A S et al. (June 2020). “Collaborative Recommendation System in Users of Anime Films”. In: *Journal of Physics: Conference Series* 1566, p. 012057. DOI: 10.1088/1742-6596/1566/1/012057.

Nápoles, Gonzalo, Isel Grau and Yamisleydi Salgueiro (2020). “Recommender system using Long-term Cognitive Networks”. In: *Knowledge-Based Systems* 206, p. 106372. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106372.

R, Kiran, Pradeep Kumar and Bharat Bhasker (2020). “DNNRec: A novel deep learning based hybrid recommender system”. In: *Expert Systems with Applications* 144, p. 113054. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.113054.

Mu, Yi et al. (2019). “An Efficient Similarity Measure for Collaborative Filtering”. In: *Procedia Computer Science* 147. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, pp. 416–421. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.01.258.

Wang, Kai et al. (2019). “SDDRS: Stacked Discriminative Denoising Auto-Encoder based Recommender System”. In: *Cognitive Systems Research* 55, pp. 164–174. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2019.01.011.

Zhang, Shuai et al. (Feb. 2019). “Deep Learning Based Recommender System: A Survey and New Perspectives”. In: *ACM Comput. Surv.* 52.1. ISSN: 0360-0300. DOI: 10.1145/3285029.

Nguyen, Van-Doan and Van-Nam Huynh (2017). “Two-probabilities focused combination in recommender systems”. In: *International Journal of Approximate Reasoning* 80, pp. 225–238. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2016.09.005.

Nguyen, Van-Doan, Songsak Sriboonchitta and Van-Nam Huynh (2017). “Using community preference for overcoming sparsity and cold-start problems in collaborative filtering system offering soft ratings”. In: *Electronic Commerce Research and Applications* 26, pp. 101–108. ISSN: 1567-4223. DOI: https://doi.org/10.1016/j.elerap.2017.10.002.