Comparing Collaborative Filtering based Recommender and Hybrid Recommender System

Swathi Gurijala
Artificial Intelligence and Machine Learning.
Lambton College,
Toronto, Canada
C0790294@mylambton.ca

Varadharajan Kalyanaraman
Artificial Intelligence and Machine Learning.
Lambton College,
Toronto, Canada
C0793756@mylambton.ca

Vignesh Kumar Murugananthan
Artificial Intelligence and Machine Learning.
Lambton College,
Toronto, Canada
C0793760@mylambton.ca

Abstract— ***Here, we are building a collaborative filter matrix factorization-based hybrid recommendation system for recommending movies to users based on emotions generated from Twitter tweets and other vectors generated by users in previous activities. It will be later cleaned up, stemmed, lemmatized, and developed emotional values to calculate the dynamic data collected by Twitter using the Twitter developer API and disposal technology. These values are combined with the recorded film data to form the main data frame. Traditional approaches, such as collaborative and content-based filtering, have limitations because they require prior user activity to carry out recommendations. To reduce this dependency, a hybrid is used that combines both collaborative and content-based filtering techniques with the emotions generated above.***

Keywords: sentiment analysis, restaurant analysis, natural language processing, classification

I. Introduction

Understanding and creating a customer's digital profile plays a vital role in increasing sales and attracting more customers in the digital age. However, doing this manually for millions of customers is a tedious and error-prone task. They are introducing machine learning and data science that simplifies this process and produces better results. Recommending the best products to customers and suggesting the best movies to customers at Amazon is one of the most significant achievements of their domain. Machine learning has already curated the model for this purpose. Customers' previous behaviors, such as purchases made in the past and movies seen, are essential data inputs for the model. A vector of recent events is constructed to depict a user's basic profile. Matrix factorization, which uses a vector of latent characteristics to represent a person or an object, is one of the most widely used strategies for protecting people and items in the shared latent space. As a result, the inner product of a product's and user's latent vectors is used to compute their interaction.

The matrix factorization has risen in popularity thanks to a competition organized by Netflix, and most of the research is focused on the same approach. Despite matrix factorization's usefulness in collaborative filtering, it is discovered that the choice of interaction function and inner product might improve its performance. While it may appear to be a minor adjustment, it has the benefit of allowing for the development of a better, specialized interaction function for modeling latent feature interactions between users and products.

II. Literature Review

In 2020, Murali Krishna Rao[1] proposed using collaborative filtering to recommend the system to provide related products for e-commerce sites with very little latent time. RMSE must be optimized to make this happen, so he proposed a solution that makes the system select appropriate learning rates and regularized parameters for the given data. To achieve this, he has gone through two approaches, one is collaborative filtering, and another one is content-based filtering.

In 2020, R.Barath, P.Chitra [2] came up with an approach to apply Matrix factorization in collaborative filtering to overcome data sparsity issues and inaccurate rating prediction. They based their recommendation model on using SVD based strategies to overcome the problem in sparsity and finally incorporated that into the framework; they tested this with movie lens data set, and the results were better than the existing system.

In 2020, Funakoshi and his team worked on a hybrid recommended model that combines the benefit of both collaborative and content best filtering here, and each user profile is a percentage as a matrix that depends upon the value concerning the other users according to the keyword. Their simulation provided relevant results that the user recommended appropriate documents with higher precision than non-hybrid filtering models.

In 2017, XIN GUAN[4] and his team worked on collaborative filtering algorithms such as Matrix factorization techniques with enhanced SVD to increase the prediction accuracy and the density Matrix has been unless to explore its potential this analysis has been evaluated on both the famous Netflix and movie lens data sets they, in turn, increase the effectiveness in terms of accuracy and efficiency when compared to traditional methods and active learning methods.

In 2013, Navgaran and his team identified that collaborative filtering recommendation systems with Matrix factorization do not have good execution times during large data sets, so they introduced a new collaborative filtering recommendation system based on MF using a genetic algorithm. The algorithm's effectiveness for predicting unrated items is factorizing randomly in a single chromosome to form a genetic algorithm population simulating these results on different data sets.

In 2018, Sanandaj[6] and his peer proposed the hybrid recommendation system based on collaborative filtering and content is filtering with additional information such as demographic information of users. They included object-based ontological schematic filtering to rent the underlying schematic relationship among items person and demographic information to make an appropriate regression tree; this increases the accuracy in real-time.

In 2021, Sarker[7] and his peer published a work that uses deep healing to recommend implicit feedback. This architecture of combining Matrix package session method's your network gives many improvements over other popular methods. The model effectively uses rating information with user experience and demographic information, as genre helps recommend cold-start situations.

In 2020 Suguna[8] and her team proposed a system to identify and understand internet users' behavior, which was challenging. They used UIR Matrix and Naive Beye's classifier to increase the perfectness of the recommendation in this type of project; more reference is given to the user location to identify the similarity between them.

In 2014, Farooque and his team partitioned before clustering the data; this played a significant role in introducing the computations and increasing the model's accuracy. This partitioning further reduces the load, and that way increases the accuracy. They have also normalized the cosine letter scoring to dampen the effect and eliminate any significance weighing.

In 2020, Singh[10] used hybridized factorization machines and SVD models to increase the accuracy with deep neural networks; supplementing the near neighbors in KNN also increased the model's accuracy. The accuracy improved slightly at the recall cost; this hybrid model got a better position but a lower recall value.

III. Methodology

3.1: We experimented with two major recommendation algorithms in this part of the project. The main goal is to build a project using a collaborative filter-based recommendation system with the use of matrix factorization and also on hybrid recommender using the neural embodying layer from the PyTorch package and compare the results between them using various methodologies.

3.2: For this project, we are using data sets from a movie lens project that has been conducted long back and place as a vital road for any collaborative filtering recommendation system. Many academic scholars and research scientists have used this data set to understand and analyze accommodation and to try out new techniques that they have developed on this. For this, we need data from Twitter to analyze the information on the movies, and we can create sentiment scores based upon the commonly used sentiment analyzer and use them for further processing.

3.3: As per the project guidelines, we have extracted tweets from teachers using tweepy and an available package that connects with Twitter API to collect various sweets and additional information from Twitter for the entire data set.

3.4: Matrix factorization is a way to generate latent images when multiplying different kinds of entities. It has been used for collaborative filtering to identify the relationship between the users and items.

In recommendation systems, it can predict what the users can get based on their reviews and ratings on the available items.

Algorithm : Matrix Factorization based Recommendation

input : Set of users U, items Y,
Matrix of users-items rankings R,
Number of latent features l,
Number of items to be recommended $N \in \mathbb{N}$,
output: Top-N recommendations $R(U) \in \mathcal{I}^N$

for $i \leftarrow 1$ **to** numIterations **do**
  foreach $user \in U$ and $item \in Y$ $with$ $rating$ $R[u,i]$ **do**
    $predicted\_rating = U[u,] * M[i,]^t$
    $err = R[u,i] - predicted\_rating$
    $U[u,] = adjust\_by\_gradient\_descent(U[u,])$
    $M[i,] = adjust\_by\_gradient\_descent(M[i,])$
return N items with highest predicted ranking

Figure 1: Matrix Factorization Algorithm

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

Pi and Qj are the vectors for items and users, and K denotes the dimension of the latent space.

3.5: On the whole, the concept of embodying is finding a relationship between discrete categorical variables to a vector of continuous numbers. Neural embedding will reduce the dimensionality of the categorical variables and then represent them in transformed space.

Neural embedding has three important steps:

1. Figuring out the nearest neighbors in the space so that recommendations can be created in users' interest.
2. Become an input to the model that has been created for the supervised task.
3. For visualizing the concepts and relations between categories.

Neural network embedding can overcome the two limitations of the standard method in one hot encoding.

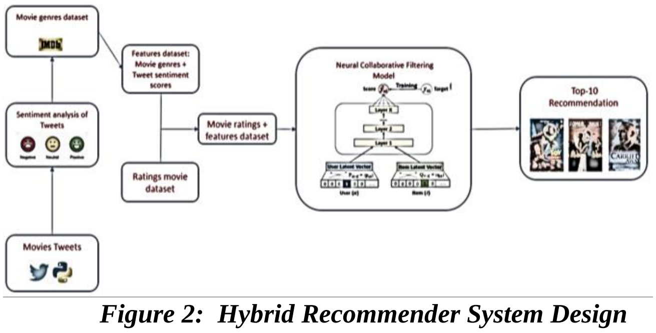

Figure 2: Hybrid Recommender System Design

3.6: Sentiment Analysis s is the process of identifying the sentiment value from any given text or context. Dada multiple methods to do something analysis on any given document here in our case, we are using tweets as comments from users for the movies, so we will go ahead with VADER sentiment analysis tool to find the scores for the tweets. It makes identifying the sentiment very easy because it can understand slang, conjunctions, punctuations, and many more. It works expeditionary well with social media text since, in our case, we are going again with tweets VADER would be the right choice.

3.7: Linear activation function was used to multiply the final Matrix from the embedding layers, and then the output is given to the sigmoid function to account for non-linearity.

IV. Experiments

The primary purpose of this project is to create a collaborative filtering, matrix factorization, and hybrid recommender system that uses tweets from Twitter to recommend movies to users.

4.1 Experiment Design:

First, we changed the movie data extracted from the file into a data frame and used the Twitter API and scrap to get tweets related to movie hashtags. Then, using matrix factorization, a coordinated filter-based recommendation system was built.

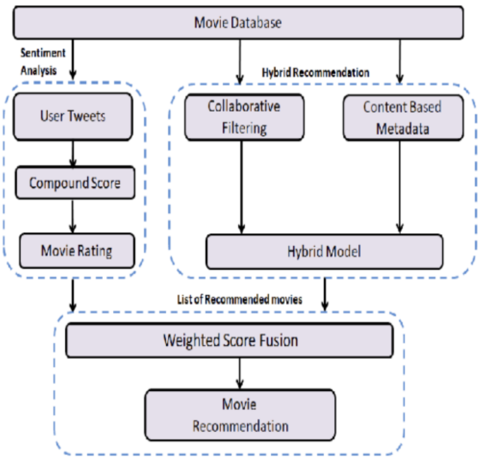

Figure 3: Overall Architecture

4.2 Dataset Preparation:

We have two sets of data available from the dataset movie lens. We have used the first file to have the movie data, as shown below.

Input Items: This is movie data which contains characteristics like movie name, release date, and IMDB URL to check movie rating, as well as one-hot encoded features like unknown, action, adventure save, animation, kids, comedy, crime, documentary, drama, fantasy, film noir, horror, music, mystery, romance, sci-fi, thriller, war, West.

Extracting sentiment from tweets is one of the essential steps, and it is done using the Text blob package; here is the result.

Movie data metrics:

Movies	1682
Unique users	1642
Ratings	72954

Sentiment distribution chart for the collected tweets.

Using the Twitter API and developer console credentials, we extracted tweets from Twitter using the movie name in the above data file as a hashtag. These tweets will later be loaded into the CSV data frame from there.

A limit of five tweets per movie was considered during extraction. Based on the collected tweets, "The Big Squeeze" topped the movie with the highest positive review. Followed by "Til There was you," with a 93% average sentiment score.

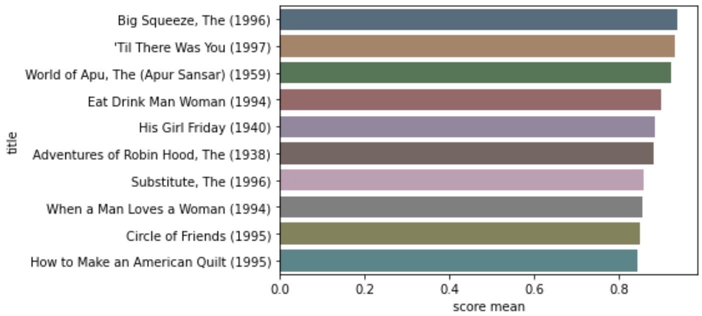

Figure 4: Bottom 10 Movies

The movie with the lowest sentiment score turned out to be "Killer: A Journal of Murder (1995)" with -0.84, followed by "Faster Pussycat! Kill! Kill!" with -0.83

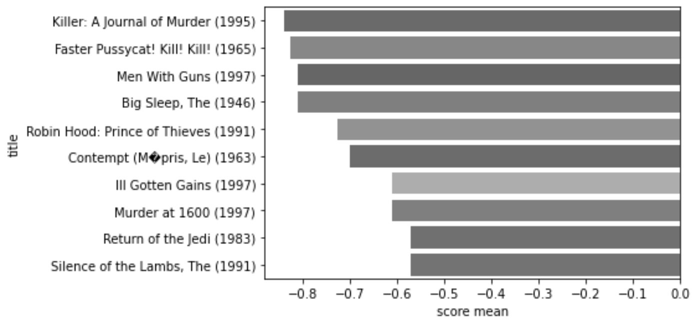

Figure 5: Top 10 Movies

Using NLTK and Vader Sentiment Analysis, Twitter data was cleaned up with less than three letters of words, extra spaces, duplicates, punctuation, stems, lemmatization, and positive, negative, and neutral score calculations. The next step was to map those tweets or comments to the movie name taken from the above file.

4.3 Evaluation Metrics:

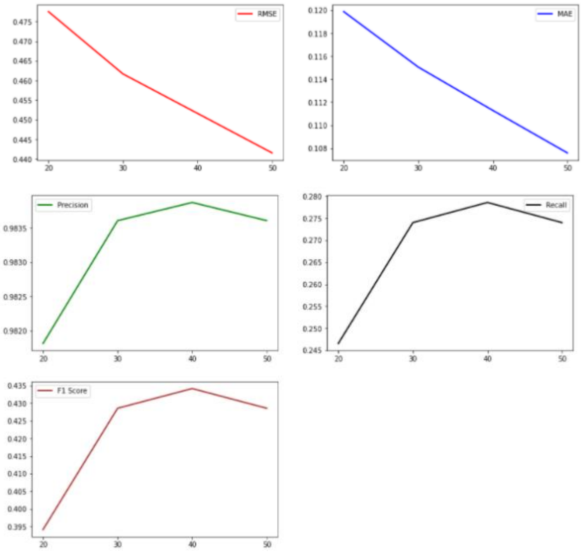

Figure 6: Evaluation Scores

Evaluations metrics are quantitative metrics resulting from the datasets and model; here are a few.
Two metrics have been used to compare the two models: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

RMSE measures the error of a model in predicting quantitative data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The prediction error is as follows

$$Prediction\ Error = Actual\ Value - Predicted\ Value$$

Absolute Error:

$$Absolute\ Error \rightarrow |Prediciont\ Error|$$

Finally, we calculate the mean for all recorded absolute errors (Average sum of all fundamental errors).

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

A recall is the number of correct results divided by the number of results that should have been returned.

$$Recall = \frac{True\ Positives}{True\ Positives +\ False\ Negatives}$$

F-score (8) combines precision and recall into a single measure that captures both properties.

$$F\_score = \frac{2*Precision*Recall}{Precision+Recall}$$

4.4 Results and Analyses:

We got the following interpretations upon RMSE, precision, recall, and F1 scores. Our model scored an RMSE value of 0.4315 in training and 1.2271 in testing.

The curve below shows that collaborative filtering works, the cost function decreases at each iteration of gradient descent, reaching a plateau of about 200, and the change is constant after that.

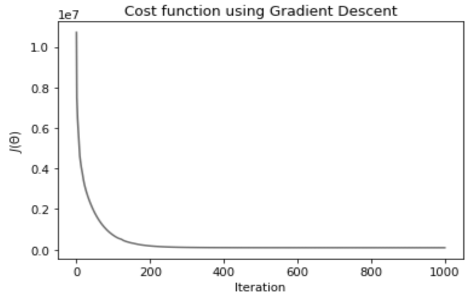

Figure 7: Cost Function

4.4.1 Hyperparameter Selection

To build a hybrid recommender system, we used a neural network layer that was fully connected with various linear convolution layers. We have defined all channels as one and kernel size as 1. We used the Pytorch SGD Optimizer with a learning rate of 1e6. For better results, use the Relu activation function with the SGD optimizer.

4.4.2 Results of Recommendation Models:

I received the following interpretations of RMSE, precision, recall, and F1 values. Our model achieved an RMSE value of 0.4315 in training and 1.2271 in the test phase. The evaluation recommendation system uses MSE metrics, but calculating the MSE for training and testing is not very useful, so a learning curve to understand better the performance quality of models trained in different train sets. It would help if we did an analysis. We set the size and evaluated the MSE scores for both the training and test sets for a particular train and test set size.

```
MAE: 0.7404742048501969
MSE: 0.874679306441374
RMSE: 0.9352429130681169
R-Squared: 0.3059656687608454
Precision 0.12087912087912088
Recall: 0.7534246575342466
F1-score: 0.20833333333333331
```

Figure 8: Matrix Factorization Metrics

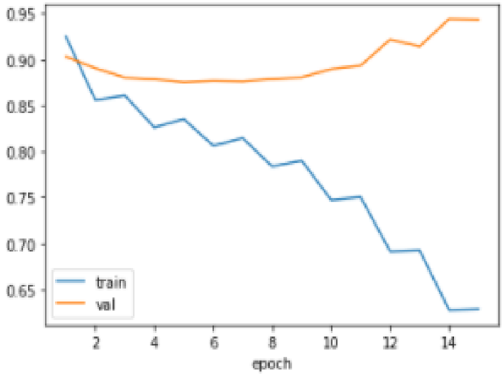

Figure 9: Model Comparison

```
RMSE:  0.4775128762588308
MAE:  0.1198866174513692
Precision:  0.9818181818181818
Recall:  0.2465753424657534
F1 Score  0.3941605839416058
```

Figure 10: Embedding layer Metrics

V. Conclusion

Recommender systems play an essential role in e-commerce, movie publishing sites, etc. They are used to filter and categorize information and customers for greater profits. This project used Twitter emotional data and user past activity to recommend movies. Sentiment analysis is used to provide data about a user's reaction to a particular movie worldwide. Therefore, this project was created using hybrid methods to get results that users can run without prior activity. As we understand, hybrid models work correctly and produce accurate results. This helps expand areas that are not limited to movies and the like.

Two different recommendation systems were implemented to build a Recommender System. Matrix Factorization and Hybrid Recommender, driven by the newly Neural Collaborative Filtering, were the primary methods used and analyzed during the development phase.

In conclusion, Hybrid Recommender exhibited better performance than the other method. Thus, it can be considered a potential method to improve the traditional recommender systems.

VI. References

[1] Murali Krishna Rao "A Collaborative Filtering Recommender System with Randomized Learning Rate and Regularized Parameter." ASTROSAT Payload Operation Centre Computer Centre, IUCAA Pune, India. 2020 1-4

[2] R.Barath, P.Chitra "Applying Matrix Factorization In Collaborative Filtering Recommender Systems." 2020, 6th International Conference on Advanced computing and communication systems(ICACCS). 2020. 1-2.

[3] Kaname F'unakoshi, Takeshi Ohguro. "A content-based collaborative recommender system with detailed use of evaluations." Fourth International Conference on knowledge-Bpsed Intelligent Engineering Systems &Allied Technology, 30 "Aug-1 "Sept 2000, Brighton, UK. 2020. 3-4.

[4] XIN GUAN 1, CHANG-TSUN LI1,2, AND YU GUAN3. "Matrix Factorization With Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems." 2017 IEEE. 2017. 3-4.

[5] Dariush Zandi Navgaran, Parham Moradi, and Fardin Akhlaghian. "Evolutionary Based Matrix Factorization Method For Collaborative Filtering Systems." 2013 IEEE. 2013. 4-5.

[6] Didar Divani Sanandaj, Sasan H. Alizadeh. "A hybrid recommender system using Multi-Layer Perceptron Neural Network." 2018 Artificial Intelligence and Robotics (IRAN OPEN). 2018. 5-6

[7] Rafidul Islam Sarker, Abdul Matin. "A Hybrid Collaborative Recommendation System Based On Matrix Factorization And Deep Neural Network." 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), 27-28 February, Dhaka. 2021. 3-4.

[8] Suguna, Sathishkumar, Deepa, "User Location and Collaborative based Recommender System using Naive Bayes Classifier and UIR Matrix." Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA-2020). 2020. 1-2

[9] Umar Farooque, Bilal Khan, Abidullah Bin Junaid. "Collaborative Filtering Based Simple RestaurantRecommender." 2014 IEEE. 1-4

[10] Jagendra Singh. "Collaborative Filtering based Hybrid Music Recommendation System". Proceedings of the Third International Conference on Intelligent Sustainable Systems [ICISS 2020]. 2020. 2-3