Bengali Movie Recommendation System using K Nearest Neighbor and Cosine Similarity

1st Anan Ehsan Siam
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
aanan.ehsan.siam@g.bracu.ac.bd

2nd Sanzida Akter
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
sanzida.akter@g.bracu.ac.bd

3rd K.M. Mahfuzul Monir
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
k.m.mahfuzul.monir@g.bracu.ac.bd

4th Md Humaion Kabir Mehedi
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

5th Md Sabbir Hossain
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

6th MD. Mustakin Alam
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
md.mustakin.alam@g.bracu.ac.bd

7th Annajiat Alim Rasel
dept. of Computer Science and Engineering(CSE)
School of Data and Sciences(SDS)
Brac University
Dhaka,Bangladesh
annajiat@gmail.com

***Abstract*—Each of us needs entertainment to recharge our spirits and energy in this fast-paced world. The confidence we acquire from entertainment allows us to work harder and enthusiastically. Movie can be one of the finest source of this entertainment. But finding a good movie can be hectic sometimes. That's why recommendation system can be a big solution for this problem where we can find our preferred movies among all of these different kinds of movies with the use of movie recommendation systems, which saves us the stress of having to spend a lot of time finding our preferred movies. As a result, it is essential that the system for suggesting movies to us is very trustworthy and gives us recommendations for the films that are either most similar to or identical to our tastes. This movie recommendation system is established using KNN and cosine similarity . Cosine Similarity increases the likelihood that the two comparable documents will be oriented closer together, even if they are separated by a large Euclidean distance because of the size of the documents.At the same time, since it produces extremely accurate predictions, the KNN algorithm can compete with the most accurate models.KNN algorithm identifies groups of people with similar movie rating preferences and makes predictions based on the average rating of the top k neighbors.**

Index Terms*—Movie Recommendation System , K-Nearest Neighbor Algorithm, Cosine Similarity , User Ratings, Root Mean Square Error(RMSE), Mean Absolute Error(MAE)

I. Introduction

A movie recommender system uses machine learning to predict or filter users' film interests based on their prior decisions and actions. In recent years this system has become a necessity as the social media and the internet streaming media is spreading widely and there are a lot of movies to watch having tons of segments and genres.Eventually, it becomes difficult for a consumer to pick up their preferred movies.There have been a number of works that are relevant to this over the years, but rarely are there recommendations for any Bengali movie review system in particular. This movie recommendation system proposes appropriate Bengali films based on a user's interests and previously rated films.It aims to increase the quality and scalability of the Bengali movie recommendation system, where cosine similarity is a metric for comparing two numerical sequences and K Nearest Neighbor(KNN) which is a non-parametric form of supervised learning, was utilized to finish the challenge. So, this recommendation system can be a better choice for the consumers to choose their movie even wisely and make their entertainment even more easier and better.

II. Research Motivation and Objective

As there are many works which has been done related to Recommendation System using various algorithms and techniques, we are motivated to build our own personalized Recommendation System which will be based on Bengali movies as most of the works are being done using English movies and others. The primary goal of this research is to provide a personalized recommended system using K Nearest Neighbor algorithm, Cosine Similarity and Bengali Movie Dataset collected from Chorki and Hoichoi to provide users Bengali movie recommendations based on user's ratings.

III. Related Work

The total architecture of this movie recommendation system is mainly based on three major parts, those are: data acquisition and repository, Recommendation System (RS) and user interface [1]. At the same time, all the registration information including user demographics and movie reviewer ratings was stored in a particular data structure. In this system, a recommendation system subunit was also established for some specifications like collaborative filtering, and calculating Euclidean scores for use in movie selection Through the graphical user interface, movie preferences, ratings, and suggestions (which are gleaned from the end-user). The dataset which was used to collect data to build this recommendation system was Movie Lens, which was accepted by The University of Minnesota. There were three primary files in this Movie Lens dataset, which are: 1. Rating file 2. Movie list file and 3. User information file. To communicate with the recommendation engine, A user interface is also developed. Three different metrics are utilized to analyze the movie recommendation engine: Precision, recall, and F-measure. The aforementioned three metrics have always been the go-to technique to gauge how well information retrieval systems perform in their output. In the proposed approach, standard user demographics such as gender, age, and occupation are used. User ratings are used to map out other users with similar tastes, and recommendations are made by excluding common entities. So this can be a tremendous attempt for movie recommendation business online.

Another recommendation system is mainly a hybrid process where an optimized clustering algorithm is introduced to detect the user profiles that are represented by denser profile vectors after Principal Component Analysis transformation [2]. There are two phases to the entire mechanism: both an online and offline phase. When in offline mode, The clustering model is structured in small sample size, and dimensional space, and prepares to target active users into many groupings. On the other hand, in online, a TOP-N film A user's active user recommendation list is displayed as a result of predicted movie ratings. Traditional CF searches the whole space to locate the k-nearest neighbors for a target user. However, considering the super high dimensionality of user profile vectors, it is hard to calculate a similarity to find similarities based on ratings that sometimes lead to poor recommendations because of sparse. To avoid this type of problem there is an alternative which is the offline clustering module. This offline clustering module involves two phases: 1) concentrating feature information into a relatively low and dense space using the PCA technique; 2) To build an effective GA-KM clustering algorithm based on the transformed user space. In this system, pre-processed data is used where they employed a linear feature extraction technique to transfer the original high space into a relatively low space which carries denser feature information. Since the high dimensionality of a user-rating matrix is mostly empty at the beginning, which makes the similarity computation very difficult, the approach is started with a PCA-based dimension reduction process. Memory-based CF systems have two key drawbacks, which are the cold start and data scarcity.

Moreover, a recommendation system has been proposed, which is mainly a solution for improving the scalability and quality of the system [3]. Here, a Hybrid approach is used in which it is unified Content-Based Filtering and Collaborative Filtering so that the approaches can profit from each other. For implementing the content-based filtering, they did several things like:

- Terms Allocation,
- Terms Representation,
- Learning Algorithm Selection and
- Provide Recommendations.

K-Means Algorithm was also adjusted over here where the numbers of clusters and items attribute features were given in the input and lastly cosine similarity measure. The dataset that was used in this system has some features like, has the ratings being assigned on a scale of 1 to 5 where 1 is very terrible and 5 being excellent. This is totally a user-based recommendation system where at least 20 movies have been rated by each user, and the simple demographic data of the users e.g. age, gender, occupation, and zip are also provided.

Furthermore, this movie recommendation system is established under the content-based movie recommendation approach. This system records the user's past behavior and recommends things according to that [4]. Like this, the system also suggests the movies according to the genres as well. For example, if a user likes a movie that is from the action genre and he/she rates it higher, the system will record this action and after that, it will show the user the movies from the same genres. The dataset which is used here is mainly divided into two sections where the first section contains the list of movies along with the genre which is being listed and in the second section the dataset contains a list of rating from 1 to 5 where 1 means poor and 5 stands for excellent. To make it easy, the ratings are converted into binary values, where from 1 to 3 the value is 0 and for 4 and 5 the value is 1. Here, a method named Offloading is used for transferring resource-intensive applications from portable devices to remote servers by considering different parameters. Offloading mechanisms involve three tasks and They are - partitioning, profiling, and offloading decision.

This entire system is mainly based on the review given by the user and converts the review into binary and afterward provides the recommendation according to that particular user's taste

IV. Working with Dataset

A. Dataset Collection

There are many online sources for movie datasets. The goal of this method is to work with Bengali movies. However, we have collected Bengali movie ratings of different users from the IMDb database. We mainly focused on Bengali movies from Chorki and Hoichoi. Chorki and Hoichoi both are subscription-based media services based on Bengali movies.

B. Dataset Description

For this system, we have made our own dataset based on Bengali movies from Chorki and Hoichoi platforms. There are in total 381 movies from different genres. 164 movies are from Chorki website and the rest are from Hoichoi. Each movie has an individual movie ID for user ratings. The genres include drama, fiction, comedy, romantic, thriller, series, etc. with Director and Cast name. This information is gathered in a particular file named **movie.csv**. Another CSV file named **ratings.csv** contains user ratings and the time duration of the movies. For each movie, each individual rating has been collected. There are total 1,00,000 ratings for 381 Bengali movies from each user. Around 550 people gave ratings for those movies.

C. Data Preprocessing

For data preprocessing, we have merged the whole dataset with movie ID, movie title and user ratings. We analyzed all the data and deleted unnecessary columns. Then, the NaN values from the dataset have been removed. After combining the file, a pivot table has been made. The dataset has been transformed into another format after deleting the unnecessary columns. After the transformation of the dataset, 373 out of 381 movies are found as unique. Movies that received the highest number of ratings from users have been sorted. Finally, by classifying different ratings, a graph has been drawn for the highest to lowest range of user ratings (0-5) :

D. Data Exploration and Fetching

The next step was Data exploration and fetching. For this part, a normal recommendation of a searched movie has been found using Cosine Matrix. In the next step, movies that are rated by a particular user have been gathered for the normal recommendation. The final step was to predict a user's rating for a particular movie, which is randomly chosen. In this part, Single Value Decomposition (SVD) algorithm is applied to evaluate the regression model. By evaluating the Root Mean Square Error(RMSE), and Mean Absolute Error(MAE) of algorithm SVD on 5 splits, a prediction of user rating for a particular movie has been predicted which is similar to the actual rating in the dataset.

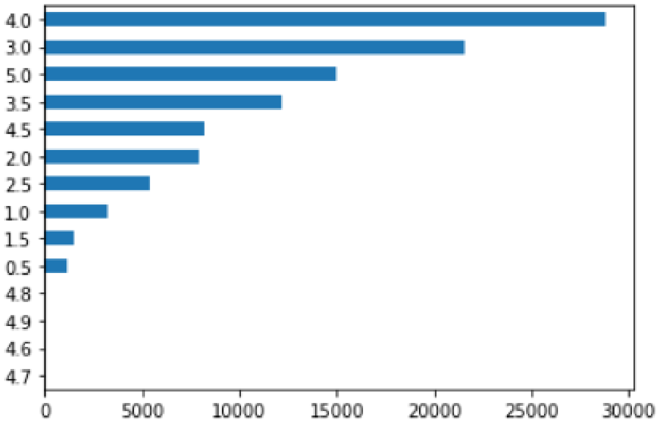
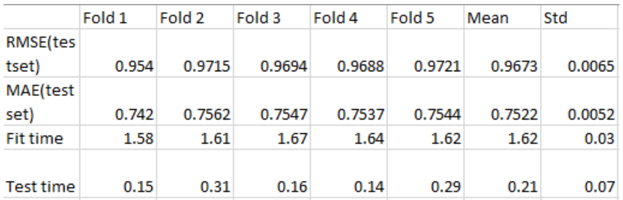


Fig. 1. Classifying Different Ratings

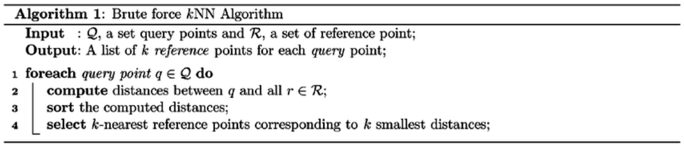


	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE(testset)	0.954	0.9715	0.9694	0.9688	0.9721	0.9673	0.0065
MAE(test set)	0.742	0.7562	0.7547	0.7537	0.7544	0.7522	0.0052
Fit time	1.58	1.61	1.67	1.64	1.62	1.62	0.03
Test time	0.15	0.31	0.16	0.14	0.29	0.21	0.07

Fig. 2. Evaluating RMSE, MAE of algorithm SVD on 5 splits

V. Proposed Methodology

In this system, two separate dataset has been used which contained information like user ID, movie ID, title, genre, rating timestamp then later merged together. Reconstructed the merged dataset by deleting unnecessary columns, removing the NaN values from the dataset, combining the files and making a pivot table, using data transformations, cleaning data and fetching detailed information. Later, we have done collaborative filtering using the KNN brute force algorithm.



```
Algorithm 1: Brute force $k$NN Algorithm
Input  : $\mathcal{Q}$, a set query points and $\mathcal{R}$, a set of reference point;
Output: A list of $k$ reference points for each query point;
foreach query point $q \in \mathcal{Q}$ do
    compute distances between $q$ and all $r \in \mathcal{R}$;
    sort the computed distances;
    select $k$-nearest reference points corresponding to $k$ smallest distances;
```

Fig. 3. Brute Force Algorithm

Collaborative filtering (CF) is basically the predictive method that uses algorithms in order to filter data to make recommendations for people with similar tastes or preferences, and there are three common types from which we have used neighbor-based filtering to predict the nearest neighbors of each movie based on the user ratings. Then, we chose a random movie using the np.random.choice function and generated recommendations using KNN for the selected movie.

In addition, we have also used Cosine Similarity, which is a metric used to measure how similar the records are irrespective of their size. It measures the cosine angle between two vectors projected in a multidimensional space. The smaller the angle, the higher the cosine similarity. If A and B are two vectors and theta is the angle between them, then the cosine similarity is:

$$\cos\theta = \frac{A.B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig. 4. Formula for Cosine Similarity

The cosine similarity value is ranged between 0 and 1 which is the same as the cosine angle range. If the angle value between A and B is 90 degrees then the cosine value is 0 which means they have no similarity where, whereas if the value is 15 degrees then it indicates that they have quite a similarity between them. The recommendation system uses measurement to learn the similarity between users and items to recommend them accurately.

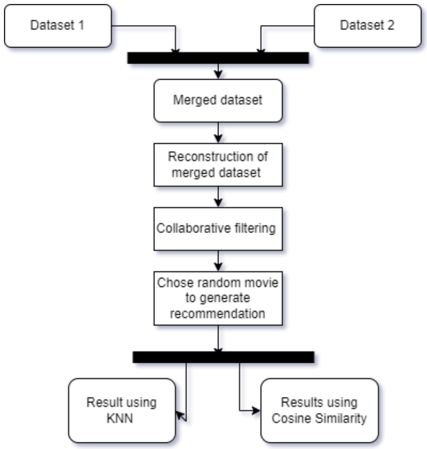

Dataset 1
Dataset 2
Merged dataset
Reconstruction of merged dataset
Collaborative filtering
Chose random movie to generate recommendation
Result using KNN
Results using Cosine Similarity

Fig. 5. Flowchart for the Bengali Movie Recommendation System

VI. Experimental Result

The suggested movie recommendation system is based on Cosine Similarity and K-Nearest Neighbor algorithm.In this system , a random movie is selected and based on that movie an user can see the list of other recommended movies.

A. Using Cosine Similarity

In the first step, normal recommendation using Cosine Matrix has been found. A random movie has been selected in this part and based on that movie there are top nine movies are shown in the recommendation list.

After that for best movie recommendation , cosine similarity has been used where again a random movie has been selected and similarity of that movie based on movie rated by an user has been filtered.Regardless of the size of the movies, cosine similarity is a measurement being used determine how alike they are. It determines the cosine of the angle formed by two vectors that are projected onto a multidimensional space.

B. Using K-Nearest Neighbor Algorithm

A movie is randomly selected for generating K-Nearest Neighbor algorithm.KNN depends on component feature similarity rather than making any assumptions about the fundamental distribution of the data. Whenever KNN draws conclusions about a movie, it first determines the ”distance” between the targeted film and every film in its database, sorts those distances, and afterwards provides the top K nearest neighbor movies as the most equivalent movie suggestions.

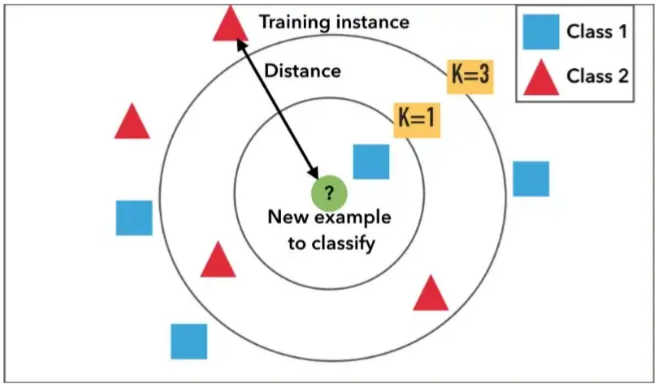


Fig. 6. Preview of how KNN suggests movie recommendation based on a selected movie

VII. Future Work

As for future work, we will try to improve the system by focusing on the user interface. We will try to implement and improve the AI so that the system can automatically detects the user’s preferable movie as well as the trending movies of the user’s preferable genres are automatically generates. Hopefully this will reduce the hassle more for a particular user to choose the right movie for them.

VIII. Conclusion

The entire recommendation system is being established to get comparatively better and refined result in terms of searching for a bengali movie. For this reason, K- Nearest Neighbors algorithm is used for this recommendation system. At the same time, to organize all the similarities together, Cosine similarity was used and this can be a tremendously amazing solution for the movie enthusiasts to get the perfect entertainment by finding the best movie for them. The suggested system is so user friendly that this will also reduce the stress of spending a mammoth amount of time for making the right decision according to someone's choice.

References

[1] V. Subramaniyaswamy, R. Logesh, M. Chandrashekhar, A. Challa, and V. Vijayakumar, "A personalised movie recommendation system based on collaborative filtering," *International Journal of High Performance Computing and Networking*, vol. 10, no. 1-2, pp. 54–63, 2017.

[2] Z. Wang, X. Yu, N. Feng, and Z. Wang, "An improved collaborative movie recommendation system using computational intelligence," *Journal of Visual Languages Computing*, vol. 25, no. 6, pp. 667–675, 2014, distributed Multimedia Systems DMS2014 Part I. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1045926X14000901

[3] S. Agrawal and P. Jain, "An improved approach for movie recommendation system," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, pp. 336–342.

[4] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," in *Smart Intelligent Computing and Applications*, S. C. Satapathy, V. Bhateja, and S. Das, Eds. Singapore: Springer Singapore, 2019, pp. 391–397.