CS550 Massive Data Mining
Movie Recommendation system

Sanket Dattakumar Dalvi
Rutgers, The State University of New Jersey
sanket.dalvi@rutgers.edu

Swapnil Shrikrishna Verlekar
Rutgers, The State University of New Jersey
swapnil.verlekar@rutgers.edu

Saurabh Sunil Kamble
Rutgers, The State University of New Jersey
saurabh.kamble@rutgers.edu

ABSTRACT

One of the most widely used types of entertainment is movies, but it may be challenging to select the right ones because there are so many of them released every year. In recent years, recommendation systems have grown in popularity as a means of giving people individualized recommendations based on their tastes. One of the most used methods in recommendation systems is collaborative filtering. It does have some drawbacks, though, such the cold start issue and sparsity problems. In order to get over these drawbacks, we present in this work a system for movie recommendations that combines collaborative filtering with content-based filtering methods. The suggested approach seeks to increase the accuracy and performance of a standard filtering technique. We assess the effectiveness of our suggested approach using a number of criteria, including RMSE, MAE, precision, accuracy, recall, and F1-score.

KEYWORDS

Movie Recommendation System, Singular Value Decomposition, K-Nearest Neighbors, Pearson Correlation Coefficient, Content based filtering, Collaborative filtering, Movielens dataset

1 INTRODUCTION

Movie recommendation systems are applications that suggest movies to users based on their preferences, behavior, and context. To improve user experience and engagement, online platforms like Netflix, YouTube, and Amazon frequently employ them. Systems for movie recommendations can also assist consumers in finding fresh entertainment that they would otherwise miss.
Building movie recommendation systems may be done in a variety of ways, including content-based, collaborative filtering, and hybrid techniques. The commonality of film characteristics, such as genres, actors, directors, etc., is the foundation of content-based techniques. Methods of collaborative filtering make use of the ratings or comments of other users who share similar preferences. To get around their drawbacks, hybrid systems mix content-based and collaborative filtering strategies.

In this research, we outline our system's design, implementation, and assessment while contrasting it with current practices. We also talk about the difficulties and potential futures of movie recommendation algorithms.
Dataset:
The primary dataset used in this study is the GroupLens movie review dataset. It contains 27,753,444 ratings from 283,228 users for 58,098 movies. The ratings range from 0 to 5 and are part of the MovieLens movie recommendation service's 5-star rating and free-text tagging system. The dataset includes additional information such as the timestamp of the review, the movie's genre, keywords.
Data Preprocessing:
In order to build an effective Movie Recommendation System using the MovieLens dataset, it is essential to properly preprocess the data. One important step in this process is to split the dataset into training and testing sets based on UserID. This can be achieved using the train_test_split method from the scikit-learn library, with 80% of each user's reviews being allocated to the training set and the remaining 20% to the testing set. The stratify attribute of the function can be used to specify the feature on which the split is based. Additional pre-processing steps such as data cleaning and normalization is done based on model requirement.

2 RELATED WORK

[1] Describes an experimental methodology to compare recommendation algorithms for Collaborative Filtering and Content-Based Filtering. Three algorithms were tested: a baseline for Collaborative Filtration and two algorithms for Content-based Filtering. The experiments demonstrate the behavior of these systems in different data sets, their main characteristics, and the complementary aspect of the two main approaches. The goal is to go beyond the "precision of the predictions" in comparing the different approaches.

[2] Describes a technique that integrates content information into item-based collaborative filtering using clustering. The group rating information obtained from the clustering result introduces content information into collaborative recommendation and solves the cold start problem. Experiments on MovieLens data show that this approach improves prediction quality of item-based collaborative filtering. This is especially true for the cold start problem.

[3] Explains how to apply the IKNN algorithm, which has compression and a global impact, for recommendation systems. The two fundamental functions of recommendation systems—score prediction and Top-N recommendation—are the subject of this research. The experimental findings demonstrate that the recommendation system's score predicted mean square difference (RMSE) has decreased

Conference'17, Washington, DC, USA
© 2016 ACM. 978-x-xxxx-xxxx-x/YY/MM...$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

greatly and improved recommended precision when employing the IKNN algorithm.

[4] The singular value decomposition (SVD), a traditional theory in matrix computation and analysis as well as a potent tool in machine learning and contemporary data analysis, is discussed in this work. The course first explores the fundamental idea behind SVD before illuminating how crucial SVD is to matrices. The work examines variational concepts of singular values and eigenvalues using majorization theory. The study explores unitarily invariant norms, which are subsequently used to establish broad conclusions for matrix low rank approximation, based on SVD and a theory of symmetric gauge functions. The subdifferentials of unitarily invariant norms are also studied in this study. Numerous machine learning issues, including matrix completion and matrix data classification, could benefit from these findings.

3 PROBLEM FORMALIZATION

(1) Distribution of Movies by Genre: The graph below illustrates the distribution of ratings by genre. It is clear from the data that Drama and Comedy are the most popular genres among users, while Documentary and Film-Noir are the least preferred. In particular, the Drama genre has received more than 40,000 ratings, indicating that it is the most frequently watched category.

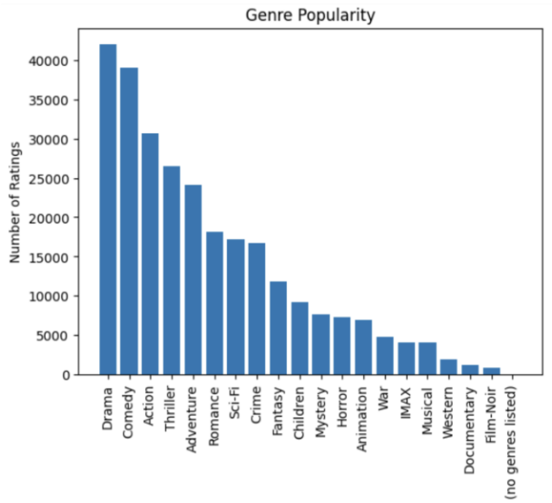


Figure 1: Distribution of Movies by Genre

(2) Average Number of rating compared to ratings given: The graph below illustrates the distribution of average ratings given by users. The data shows that the majority of average ratings fall within the range of 3 to 4, with a peak at 3.5. This suggests that users tend to give movies an average rating of 3.5

In light of the aforementioned analysis, the objective of this project is to develop movie recommendation system models utilizing various algorithms and concepts. To address the movie recommendation problem, this project proposes the implementation of multiple recommendation models based on both collaborative filtering and content-based filtering. The problem statement for this project involves addressing data inconsistencies and enhancing the quality of the dataset to facilitate the development of a high-quality recommendation system.

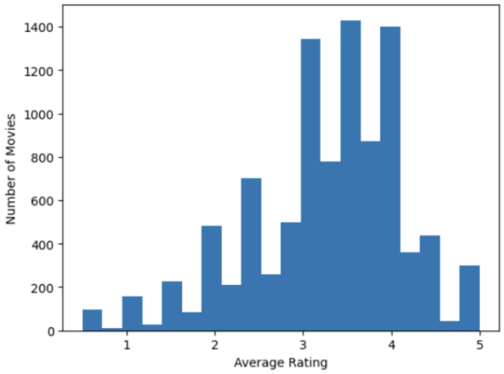


Figure 2: Plot of average rating given by users

4 THE PROPOSED MODEL

1. Content Based Filtering With KNN:
To predict rating, in our project we use content based filtering along with knn algorithm. The movies and ratings dataset are merged to get how much a user rated for a specific movie, along with movie name and genre. The data is normalized by transforming features by scaling each feature. A movie vector is formed by transforming the movie titles and genres into a tfidf matrix. A tfidf matrix is a matrix where each row represents a document and each column represents a unique word in the corpus. The value in each cell represents the TF-IDF score for that word in that document.. A mapping between movie IDs and row indices in the tfidf matrix is created.

KNN algorithm is used to find the k nearest nighbors for the given data point and predict how much rating will the user give to the movie that they haven't seen.

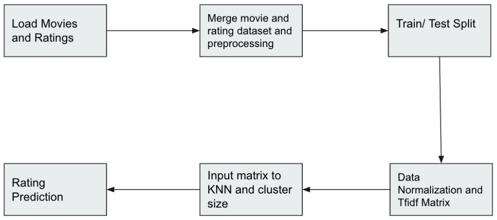

Load Movies and Ratings
Merge movie and rating dataset and preprocessing
Train/ Test Split
Data Normalization and Tfidf Matrix
Input matrix to KNN and cluster size
Rating Prediction

Figure 3: Rating prediction flowchart

2. Collaborative filtering using Pearson Correlation based on User Ratings:

In our project, we used collaborative filtering based on user ratings and Pearson correlation. Collaborative filtering based recommendation system begins by preprocessing the Movies and Ratings datasets. It then takes into account the user's input data, such as the movies they have already watched and the ratings they have given to those movies. Based on this information, the dataset is preprocessed to create groups of users who have watched the same movies. The similarity between all users in the dataset and the input user is calculated using Pearson correlation. Movie ratings are then calculated based on a weighted average of all users, with the Pearson correlation coefficient "r" serving as the weight. The top 10 movies are recommended to the input user based on these calculated movie ratings.

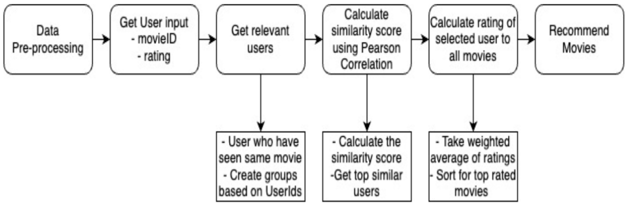

Data Pre-processing
Get User input - movieID - rating
Get relevant users
Calculate similarity score using Pearson Correlation
Calculate rating of selected user to all movies
Recommend Movies
- User who have seen same movie - Create groups based on UserIds
- Calculate the similarity score -Get top similar users
- Take weighted average of ratings - Sort for top rated movies

Figure 4: Collaborative filtering with Pearson Correlation Coefficient flowchart

Pearson correlation is a statistical measure that calculates the relationship between two continuous variables. It is considered the best method for measuring the association between variables because it is based on covariance. Pearson correlation provides information about both the magnitude and direction of the relationship between variables.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

Figure 5: Pearson Correlation Coefficient

The Pearson correlation coefficient can range from +1 to -1. A value of +1 indicates a perfect positive relationship, while a value of -1 indicates a perfect negative relationship. A value of 0 indicates that there is no relationship between the variables.

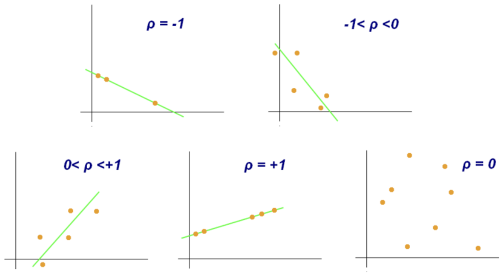


Figure 6: Examples of scatter diagrams with different values of correlation coefficient

One of the reasons why Pearson correlation is used in recommendation systems is because it is not affected by scaling. This means that multiplying all elements by a nonzero constant or adding any constant to all elements does not change the Pearson correlation. For example, if we have two vectors X and Y, pearson(X, Y) will be equal to pearson(X, 2 * Y + 3). This property is important for recommendation systems because two users may rate items on different scales but still have similar preferences.

3. Singular Value Decomposition based Model:

The Singular Value Decomposition (SVD), a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K¡N). In the context of the recommender system, the SVD is used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

Our aim is to utilize the SVD method, which employs latent factors, in our system. The SVD approach considers a reduced dimensional representation of movies and groups together individuals who have similar preferences. By applying this model, we can minimize the root mean square error (RMSE) for unobserved data. Furthermore, we implemented the GridSearchCV technique to determine the optimal hyperparameters that produce the lowest RMSE.
$A = UVS^T$

5 EXPERIMENTS

To assess the performance of our models, we will employ various evaluation parameters. The metrics of RMSE and MAE are utilized to evaluate the accuracy of rating prediction, while metrics such as Precision, Recall, and F-Score are employed to assess the accuracy of item recommendations for each user. Formulas for computing RMSE and MAE are given below.

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

Figure 7: Formula for computing RMSE

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

Figure 8: Formula for computing MAE

For the Content-Based Filtering model utilizing KNN, the optimal RMSE and MAE values were obtained with a k size of 100, yielding values of 0.6152 and 0.5312, respectively. It is apparent that the accuracy of the model improves with appropriate selection of the k-value. A plot of time in seconds versus the k-value used for the model reveals that the time required by the model increases with increasing k-value, indicating a tradeoff between time and accuracy. Although the model was constructed using a smaller dataset due to the high computational power required for larger datasets, it can be inferred that the accuracy of the KNN-based model for rating prediction would improve with the use of a larger dataset.

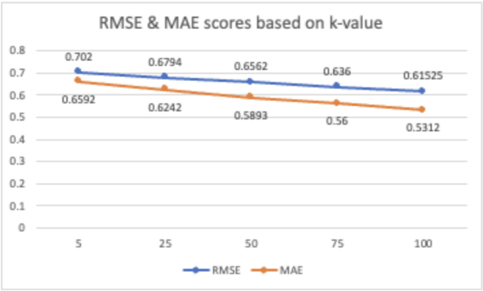


Figure 9: RMSE & MAE scores based on k-value

The model utilizing the Pearson Correlation Coefficient demonstrated strong performance, providing 10 high-quality movie recommendations based on new user rating data. As a Collaborative filtering-based model, the selection of similar users from the dataset that are most closely aligned with the input user is crucial for enhancing the accuracy of movie predictions.

The aforementioned formatted output presents the top 10 recommended movies based on the input user ratings. These recommendations are derived from the weighted average of ratings provided by selected similar users, with the Pearson Correlation Coefficient serving as the weight.

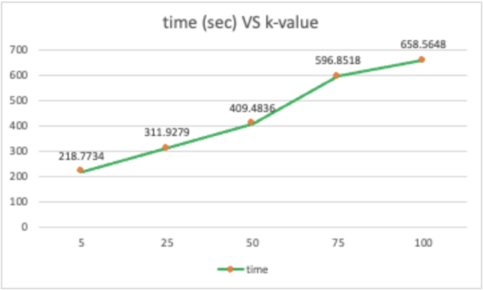
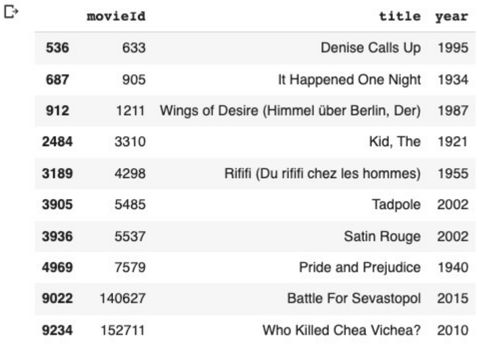


Figure 10: Time VS K-value plot for KNN model



	movieId	title	year
536	633	Denise Calls Up	1995
687	905	It Happened One Night	1934
912	1211	Wings of Desire (Himmel über Berlin, Der)	1987
2484	3310	Kid, The	1921
3189	4298	Rififi (Du rififi chez les hommes)	1955
3905	5485	Tadpole	2002
3936	5537	Satin Rouge	2002
4969	7579	Pride and Prejudice	1940
9022	140627	Battle For Sevastopol	2015
9234	152711	Who Killed Chea Vichea?	2010

Figure 11: Top recommended movies

Another collaborative model was implemented utilizing Singular Value Decomposition for movie recommendation, yielding high accuracy values. The table below presents the accuracy values for the developed model.

Table 1: Accuracy measures for SVD.

RMSE	MAE	Precision	Recall	F-Measure	NDCG
0.880741	0.676173	0.805710	0.400003	0.534599	0.961140

6 MODEL CORRECTNESS (OPTIONAL TASKS)

a. Transparency and Explainability of Recommender Systems:

To explain the transparency and explainability of the recommender system we can have a look at the TFIDF matrix produced when we vectorize the dataset into an object for rating prediction. The value in the matrix and later calculated cosine similarity between the datapoints helps us explain the reason for the rating predicted by the KNN. Similarly for Movie recommendation the computed Pearson Correlation can be inspected to figure out the movie recommended to the user.

b. Fairness and unbiasedness of Recommender Systems:
To achieve fairness and unbiasedness we perform data analysis to understand the distribution of data. This gives us an insight on how the skewed the dataset helps us train and perform validation in such a way that the model does not exhibit any biases. We can also split the train and test data in a stratified way in order to have equal representation of all classes. This can achieve us fairness and unbiasedness in a Recommender System

c. Privacy Protection for Recommender Systems:
For Privacy protection we anonymize the data so that any new recommendation made to a user cannot be traced back to where the recommendation came from. To achieve this abstraction we can encode the data, drop private data and use ID mapping so that privacy protection is achieved in the recommendation system and no personal data of user is visible on the client end.

7 CONCLUSIONS AND FUTURE WORK

The models implemented in this project demonstrate satisfactory prediction accuracy and provide high-quality movie recommendations. Both Content-based filtering and Collaborative filtering exhibit distinct advantages and disadvantages, as evidenced by the models implemented for each approach. Collaborative filtering benefits from considering the ratings of other users and adapting to the user's evolving interests without requiring the analysis or extraction of information from the recommended item. However, it also presents several drawbacks, including a slow approximation function, a limited number of users for approximation, and privacy concerns that are addressed by anonymizing user ratings.

In contrast, Content-based filtering learns the user's preferences and offers a highly personalized experience. Nevertheless, it does not consider the opinions of others regarding the item, which may result in recommendations of low-quality items. Furthermore, data extraction is not always straightforward, and identifying the characteristics of the item that the user likes or dislikes is not always evident. The models implemented were based on a small dataset; by transitioning to a larger dataset and utilizing greater computational power, the accuracy and quality of movie recommendations could be improved.

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to Professor Yongfeng Zhang and Teaching Assistant Mr. Sam (Guo) Lin for their invaluable guidance and support throughout the course of this project. Their expertise and insights have been instrumental in helping us complete this project successfully. We are truly grateful for their unwavering commitment to our growth and development.

REFERENCES

[1] Rafael Glauber and Angelo Loula. 2019. Collaborative filtering vs. content-based filtering: differences and similarities. *arXiv preprint arXiv:1912.08932* (2019).

[2] Qing Li and Byeong Man Kim. 2003. An approach for combining content-based and collaborative filters. *In Proceedings of the sixth international workshop on Information retrieval with Asian languages* (2003), pp. 17–24.

[3] Sailuo Wan Hua Xia Li, Bin and Fengshou Qian. The research for recommendation system based on improved KNN algorithm. *In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)* (????), pp. 796–798.

[4] Zhihua Zhang. 2015. The singular value decomposition, applications and beyond. *arXiv preprint arXiv:1510.08532* (2015).