EasyChair Preprint

№ 13177

Recommendation System in Machine Learning

Ayush Hedaoo

May 6, 2024

Priyadarshini J.L. College of Engineering, Nagpur
An Autonomous Institute affiliated to R.T.M Nagpur University
Name of the Student: Ayush M. Hedaoo
Email Id: ayushhedaoo1203@gmail.com

Recommendation System in Machine Learning

ABSTRACT

The goal of a recommendation system is to predict user interests and infer their mental processes. Based on the user's demands and while taking into account their interests, this system can give them the information they need. A more thorough analysis of the data is required to provide better recommendations. Numerous recommendation systems have been developed using diverse methodologies. As OTT platforms, shopping, travel, and other websites proliferate and strive to quickly improve their user suggestions, the research into such systems has gained popularity up to this point. In this paper, we have implemented movies recommendation system using machine learning techniques. We have studied and compared different recommendation models and using the best model we have implemented the movies recommendation system for recommending movies to the user. Machine learning is used in the movies recommendation system because it gives an entity the potential to learn artificially without explicit programming

Keywords : Recommendation System, Machine Learning, Movies, Recommendation models, Content filtering, Collaborative filtering

1. INTRODUCTION

Systems for making recommendations are widely utilized today in everything from entertainment to retail applications. Additionally, the data needed for these applications is growing every day as a result of the internet's widespread accessibility. Therefore, there is room for improvement and a need to offer better suggestions that can effectively handle enormous amounts of data. Our recommendation engine for movies is primarily built using machine learning, cosine similarity metrics, and content- based filtering approaches. Based on the user's past behaviour or explicit feedback, content-based filtering techniques employs movie features to suggest additional films that are comparable to the user's favorites. Two videos can be viewed as two vectors in m dimensional user space in cosine similarity. The cosine of the angle between the vectors is used to calculate how similar they are to one another. In our system, machine learning is employed to create recommendation models and to retrieve information. An entity can learn artificially through machine learning without explicit programming.

We know that in the content filtering[9] we recommend.
movies to the user based on the movie details like title, actors etc. and also based on the user's past history. Recommendation system which are purely based on content filtering have certain drawbacks like there isn't enough variety or novelty, Scalability is difficult etc. And in the collaborative filtering[1], It compiles the user ratings for services like items, movies, etc., finds patterns among users based on their ratings, and generates fresh recommendations for the user based on inter-user comparisons. Recommendation systems which are purely based on the collaborative filtering have certain drawbacks like cold start problem, hard to include side features for services like item, movies etc. The side elements for movie recommendations may include a user's country or age. Including available side features raises the model's calibre. Using a machine learning technique in the movies recommendation will surely help to improve the efficiency of the recommendation system. We have used Tmdb dataset for the movies recommendation. In this paper, we have first studied the dataset properly and then done the exploratory data analysis on the dataset to recognize the patterns and understand it properly. Then we have done preprocessing of the dataset. Creating different machine learning based recommendation models using content and collaborative filtering. Then we have done training and testing of these models. Then using the best recommendation model we have created the Rest API and then created the recommendation system GUI for the movies. Then in the GUI we need to enter movies details like title etc. and then we will recommending similar movies to the user.

2. LITERATURE REVIEW

There are various recommendation approaches like content, collaborative filtering, demographic etc. We can use those recommendation approaches along with various machine learning techniques for improving recommendation for the movies. There are different methods and techniques in the machine learning which can be used in the recommendation system for improving recommendations for the user. Different recommendation approaches has advantages and disadvantages that could impact the precision and effectiveness of a system.

In this paper they have created a collaborative filtering-based recommender system for new trends in any research field has been developed in this study. The three main building blocks for the recommender system that is here proposed are datasets, prediction ratings based on users, and cosine similarity. The quantity of accurate ratings submitted by users will decide how accurately they are rated. Cosine similarity is then used to order the findings.

In this paper they have created a movie recommendation using cosine similarity and KNN. This study outlines a method that provides users with generalized suggestions based on the popularity and/or genre of a film. The implementation of the Content-Based Recommender System involves several deep learning techniques. This study also provides a glimpse into the difficulties that content-based recommendation systems encounter, along with our efforts to address them.

In this paper they have created a recommendation system using KNN and cosine similarity, the authors of this research developed a recommendation system. They have worked on machine learning based technology that helps to comprehend requirements and provides recommendations for the user's chosen product. In this research, different machine learning algorithms are compared for the suggestion of different product purchase patterns by users and provides more accurate search result.

In this paper the authors have created a recommendation system using cosine similarity. The algorithm not only offers recommendations but also details about the movie you searched for. The rating of the film, its premiere date, cast, and genres are among the supplementary information. The system also offers more details about the cast. The system also conducts sentiment analysis on the movie reviews, categorizing them into two categories, "Good" and "Bad," to aid the user in saving time when reading reviews.

In this paper the CCAM (co-clustering with augmented matrices) has been used by the authors of this paper to develop a variety of techniques, including heuristic scoring, conventional classification, machine learning, and the incorporation of content-based hybrid recommendation systems in conjunction with collaborative filtering models, to build a recommendation system.

the authors of this study have created the collaborative filtering-based ALS Algorithm. ALS works to address the scalability problem of large datasets. This work created a movie recommendation system for predicting user ratings using the ALS algorithm. This system cannot display slightly better results since the Restricted Boltzmann Machine (RBM) has not been improved.

They have put out a graph clustering methodology that uses contextual correlation to identify groups in a graph that exhibit multiview vertex properties. The methodologies used before this model, however, were focused on the features of a single view and ignored the contextual link between features. To fulfil the task of clustering in the multiview featured graph, their solution combines graph clustering and multiview learning. The model's unsupervised learning foundation means that it does not allow vertex embedding for attributed graphs, which is a feature of supervised learning models.

They suggested integrating the recommendation system with user reputation. Using information about the users' interests or user types, online recommendation systems make recommendations to users. However, recommendation systems occasionally push particular goods or services without confirming their reputation. Suggest removing the skewed user ratings by examining the user's historical

rating history and user credibility. To identify biased consumers, one might employalgorithms like the cumulative sum method algorithm. recommended using collaborative filtering. Their approach gives genuine users greater reputation value while giving fraudulent users less. Therefore, their system is unable to distinguish between distinct users if this value is the same for both types of users.

3. PROPOSED SYSTEM

3.1 Problem Statement

"To implement the movies recommendation system usingmachine learning techniques"

3.2 Problem Elaboration

Today recommendation system is used in various domains and the major challenge is to provide better recommendation of services to the user by using huge amount data present within the application. In this project, we are doing the comparative study of machine learning techniques which are implemented using collaborative filtering and content filtering. After comparison, whichever is the best technique amongst them will be used in building the movies recommendation system. Also, the purpose of using machine learning in recommendation system is to create the model for prediction of movies etc. in the recommendation system instead of doing it programmatically explicitly each time.

3.3 Proposed Methodology

Following our research and literature review, we found that systems that were only based on content and collaborativefiltering had a number of disadvantages. Therefore, we combined these filtering approaches with a number of machine learning algorithms, including ALS (Alternating Least Square), SVD (Single Valued Decomposition), KNN (K-Nearest Neighbor), Co-clustering, and cosine similarity, to improve this recommendation system. After comparing these strategies, we will decide which is best and can be employed moving forward to construct the ultimate movierecommendation system. In order to obtain information onthe movies, we used the TMDB dataset. During implementation, we intend to use the following categories ofmachine learning techniques (or algorithms):

1) ALS (Alternating Least Square)

Collaborative filtering uses the alternating least squares (ALS) algorithm, which is a very well-liked technique. A matrix factorization approach called ALS recommendation makes use of Alternating Least Squares with Weighted-Lamda-Regularization (ALS-WR). It executes the ALS algorithm in parallel and factors the user to item matrix Ainto the user to feature matrix U and the item to feature matrix M. In order to reduce the least squares difference between anticipated and actual ratings, the ALS algorithm seeks to identify the latent components that best explain the observed user to item evaluations.

2) KNN (K- Nearest Neighbors)

One categorization technique that makes the assumption that comparable entities reside nearby is KNN. This algorithm places the new case in the category that matches the available categories the most by assuming similarity between the new case/data and existing cases. In order to classify a new data point based on similarity, it stores all of the existing data.

3) SVD (Singular Value Decomposition)

In the fields of data science and machine learning, Singular Value Decomposition (SVD), a traditional linear algebraic approach, is becoming more and more well-liked. This popularity results from its use in creating recommender systems. Many online user-centric apps, such video players, music players, e-commerce applications, etc., suggest additional content for users to interact with. It can be difficult to find and suggest numerous acceptable products that users will like and choose. SVD is one of thevarious strategies that are employed for this goal.

4) Co-clustering

Co-clustering focusses on grouping by similar rows and columns while focusing on both the row and column dimensions. The key distinction from the standard K means algorithm is that the row cluster centroid and the column cluster centroid are calculated from the co-cluster centroid.

Cosine Similarity Cosine similarity is a metric used in a variety of machine learning techniques, including the KNN for calculating the distance between neighbors, recommendation systems for

suggesting comparable movies, and textual data for determining the similarity of text in a document. Applications like data mining and information retrieval use machine learning and cosine similarity.

4. IMPLEMENTATION

4.1 Data Collection

The TMDB dataset provides information about every movie. The Movie Database is a collectively created film and television database (TMDB). Since 2008, the community's amazing individuals have uploaded every piece of data. The vast data set and significant focus on foreign markets offered by TMDB are mostly unmatched. The size of all these files in the dataset is around 900MB.This collection of data includes a number of movie-related files, including: -

movies.csv - This file contains data on 45500 films that are included in the entire movielens dataset. Adult, Budget, Genres, Homepage, Id, Imdbid, Original Language, Original Title, Overview, Popularity, Poster Path, Production Companies, Production Countries, Release Date, Revenue, Runtime, Spoken Languages, Status, Tagline, Vote Average, and Vote Count are the columns in this file.

credits.csv- This file contains the columns like cast, crew and id. This file has around 45500 rows. It contains details of cast and crew for all the movies. It is available in the form of a stringified JSON object.

keywords.csv- It includes the MovieLens movie plot keywords for the films. It is accessible as a stringified JSON object. This file has around 46,000 rows. It contains columns like id, keywords etc.

links.csv- This file contains all of the Full MovieLens dataset's movie TMDB and IMDB IDs. This file consists of around 45,800 entries. It contains the columns like: - Movieid, Imdbid, Tmdbid etc.

Links_small.csv- It contains the TMDB and IMDB IDs for a small subset of the Full Dataset's 9,000 movies. It contains the same columns as present in the links.csv file. (6) Rating_small.csv- It contains the portion of 100,000 reviews left by 700 individuals for 9,000 films. This file contains around 1,00,000 entries. It contains the columns like: Userid, movieid, rating, timestamp etc.

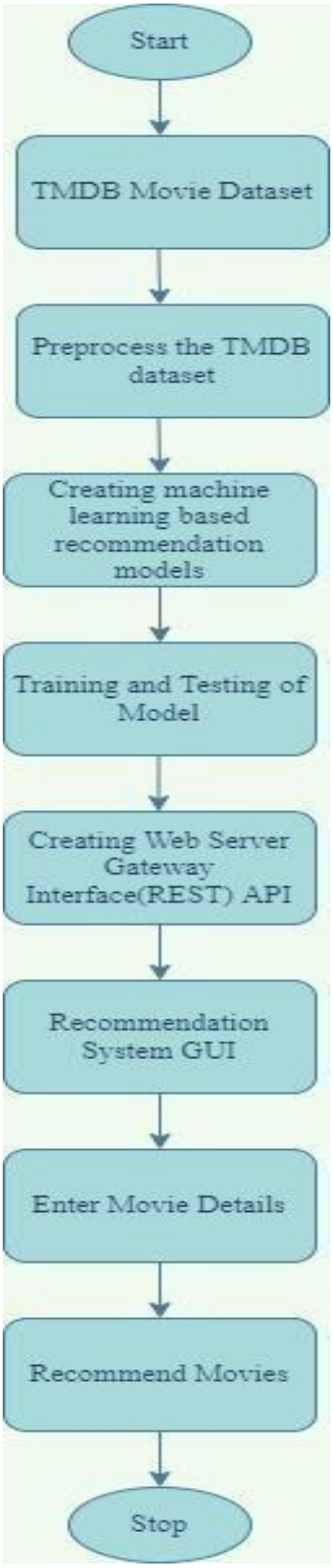
4.2 Preprocessing and creating machine learning models

The additional data that is present in the existing dataset has to be cleaned up and preprocessed. We will deal with duplicate, invalid, and null values in the dataset. This is necessary to transform the raw data into a more comprehensible, practical, and effective manner.

After the dataset was preprocessed, we developed a number of machine learning-based recommendation models, including ALS (Alternating Least Square), SVD (Single Value Decomposition), Co-clustering, and KNN (K-Nearest Neighbor), which were implemented using the collaborative filtering approach, and a cosine similarity-based recommendation model that was implemented using the content filtering approach. The best model is then used in the ultimate movies recommendation system when a comparative analysis of the models is completed.

4.3 Training and Testing

We must train and test the model when it has been generated. The dataset has been divided in half, 80:20. 20% of the dataset is used to test the model, while the remaining 80% is utilized to train the model. For evaluating the models, we have used metrics like RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). Root Mean Square Error is a statistic that reveals how far, on average, a model's projected values and observed values differ from one another. Mean Absolute error in the context of machine learning refers to the size of the discrepancy between the forecast of an observation and its actual value. Below is the workflow diagram for the project: -



5. RESULTS

This section contains a discussion of the outcomes from our experimentation and implementation of various machine learning-based recommendation algorithms. We have used two metrics namely RMSE (Root Mean Square Error) and MAE (Mean Square Error) for evaluating various recommendation models.

Table -1: Comparison of recommendation models

Recommendation Models	**RMSE**	**MSE**
Collaborative + SVD (Singular Value Decomposition)	0.8675	0.6729
Collaborative + K-Nearest Neighbors	0.9552	0.7257
Collaborative + Co-Clustering	0.9544	0.7249
Collaborative + ALS (Alternating Least Square)	0.8219	0.7632
Content + Cosine Similarity	0.7481	0.6316

In the above table, the model which have low value of RMSE and MSE is considered as best model as it is having less error. So, the model using content and cosine similarity is best as compared to the other models. For creating movies recommendation GUI, we have used python Flask as IDE and created the API for the best machine learning based recommendation model. Below is the screenshot for recommending movies to the user using cosine similarity.

Author	Existing Proposed Algorithm and Methods	Metrics	Outcome Results
Sobecki (2014)	Ant colony optimization (ACO), particle swarm optimization (PSO), bat algorithm (BA), Bee colony optimization (BCO), and Intelligent weed optimization (IWO).	Mean absolute error.	System provides MAE value for five SI algorithm as 0.41, 0.56, 0.68, 0.77 and 0.53 respectively.
Wang et al. (2014)	K-means clustering, genetic algorithm (GA) and principal component analysis (PCA)	Mean absolute error, precision, recall, and t-value.	The system provides mean absolute error, t-value as 0.78 and 13.85 respectively.
Hatami et al. (2014)	Cuckoo optimization algorithm (COA)	Mean absolute error, Coverage, precision, and recall.	COA method is more stable than the GA method.
Wasid et al. (2015)	Fuzzy particle swarm optimization-collaborative filtering (FPSO-CF)	Mean absolute error, Coverage.	The system provides mean absolute error, coverage as 0.80 and 0.96 respectively.
Om Prakash et al. (2016)	Particle swarm optimization (PSO) and Fuzzy c-means (FCM)	Mean absolute error, Standard deviation.	The system provides mean absolute error, Standard deviation (SD) as 0.75 and 5.067, respectively.
Prakash et al. (2017)	K-means clustering algorithm and cuckoo search optimization algorithm.	Mean absolute error, Standard deviation, root mean square error and t-value.	The system provides mean absolute error, accuracy as 0.68 and 63.22, respectively.
Katarya et al. (2018)	Artificial bee colony (ABC) optimization technique and k-means clustering technique.	Mean absolute error, precision, recall, and accuracy.	The system provides mean absolute error, precision, recall, and accuracy as 0.42, 0.64, and 53.22, respectively.
Katarya et al. (2018)	Gray wolf optimizer and Fuzzy c-means clustering (FCM).	Mean absolute error, Standard deviation, precision and recall.	The system provides mean absolute error, Standard deviation, precision and recall as 0.68, 0.54 0.55 and 0.49 respectively.
Yadav et al. (2018)	Bat algorithm	Mean absolute error, precision, Recall, F-score.	By using the Bat algorithm, we get 6.9% improved result compared to existing ABC regarding MAE and F-score.

6. CONCLUSION

In this paper, we have implemented various recommendation models using content and collaborative filtering based on different machine learning techniques to improve the user recommendation in the movies recommendation system. After studying, comparing and experimenting various recommendation model we have realized that model based on content filtering and cosine similarity was better as compared to the other models. Using the best model, we have created API for it using pythonflask and using it in movies recommendation GUI. Finally, we are recommending similar movies to the user. In order to boost user satisfaction, our suggested solution would enable the system to make a recommendation to the user that is more accurate. In future we can implement recommendation system which can work on real time information of users. Also, we can try to implement cross domain recommendation system in future.

7. REFERENCES

[1] M Viswa Murali, Vishnu T G, Nancy Victor," A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research",2019, (ICACCS),DOI:10.1109/ICACCS.2019.8728409

[2] Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, Gaurav Srivastav," Movie Recommendation System using Cosine Similarity and KNN",2020,((IJEAT), DOI: 10.35940/ijeat.E9666.069520

[3] Shivganga Gavhane,Jayesh Patil,Harshal Kadwe,Projwal Thackrey,Sushovan Manna, "Recommendation System using KNN and Cosine Similarity",2020,

[4] Shubham Pawar, Pritesh Patne, Priya Ratanghayra, Simran Dadhich, Shree Jaswal, "Movies Recommendation System using Cosine Similarity", (IJISRT), Volume 7, Issue 4, April – 2022, 342-346, April 2022.

[5] Y. C Chen, "User behavior analysis and commodity recommendation for pointearning apps," In 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE,2016.

[6] Y.H Zhou, D. Wilkinson, R. Schreiber, "Large scale parallel collaborative filtering for the Netflix prize," In Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management (pp. 337–348). Shanghai: Springer,2008

[7] Tiantian He , Yang Liu, Tobey H. Ko , Keith C. C. Chan , and Yew-Soon Ong "Contextual Correlation Preserving Multiview Featured Graph Clustering",(2019),(IEEE transactions)

[8] Zhiheng Wu,Jinglin Li,Qibo Sun,Ao Zhou,"Service recommendation with context-aware user reputation evaluation",(2017),(IEEE conf)

[9] Khamael Raqim Raheem; Israa Hadi Ali, "Content-based Recommender System Improvement using Hybrid Technique", (2020) (IEEE Xplore)

[10] Shailesh Kalkar, Prof. Pramila Chawan, "A Survey on Recommendation System based on Knowledge Graphand