Analysis of Movie Recommendation System Using Big Data and Machine Learning Algorithms

Abhiram Iyengar[1], Preet Jain[1], Harsh Singhania[1] and Vaibhav Godbole[2]

[1]Fr. Conceicao Rodrigues College of Engineering (Department of Information Technology), Mumbai (Maharashtra), India. s

abhiramiyengar29@gmail.com

preetjain99@gmail.com

hsharsh.singhania@gmail.com

godbole@fragnel.edu.in

Abstract. With the surge of movies available on streaming platforms from across the globe and increase in number of users, a recommendation system plays a vital role in offering ideal recommendations to users based on their tastes. From a vast list of movies and their reviews depending on different users, we are dealing with large amounts of data from a variety of sources and in a variety of formats, such as in the case of videos, where a large amount of data must be computed and then recommended to a person, it entails a number of factors and the computation of data using the information of other users. In this paper, big data tools like Apache's Hadoop and Hive are deployed and various machine learning algorithms are implemented and analysed for efficient movie recommendations.

Keywords: Apache Hadoop, Apache Hive, Machine Learning, Content-based filtering, Collaborative filtering, Hadoop Distributed File System, Recommender system, etc.

1 Introduction

A recommendation system is a type of data filtering system that challenges you to prioritize a user and make recommendations based on the user's priorities. Movies, podcasts, novels, and videos, as well as colleagues and stories on social media, items on e-commerce platforms, and individuals on business and dating sites, all have different content. These systems will often retrieve and filter data on a user's interests, which they can then use to advance their recommendations into the next cycle. For an occurrence, LinkedIn can recommend you profiles of users who are in the same job category. Numerous multiple times, these frameworks can be ad-libbed based on interests of countless individuals. For instance, if Amazon sees that countless clients who purchase a gaming console likewise purchase a gaming controller. They can recognize a controller to another client who has recently added a gaming console to his cart. Because of the advances in recommender frameworks, clients continuously anticipate great outcomes. In the event that a music streaming application can't anticipate and play tune that the client likes, at that point the client will simply quit utilizing it. This has prompted a lot of technical enterprises on improving their recommendation structures. The taste of a single customer can differ depending on a large number of aspects, content of the movie, genre, or his/her favourite actor, director etc. For example, a person likes to watch horror and suspense movies. and he finds romantic movies very boring and he doesn't like them. Likewise, another user likes to watch romantic movies and action movies, but doesn't like comedy movies. Sometimes you can see that a person you like doesn't like a movie genre, let's say comedy in this case, but gets a recommendation because it has been seen by many of the other users due to its high rating. This is the situation where both approaches to the recommendation system (content-based filter and collaborative filter) are used.

2 Related Work

In paper [1], introducing the concept of content genome. This paper proposed an algorithm, which is suitable for mass recommend object-collaborative filtering algorithm based on the contents of the genome. The results indicate that the new algorithm can effectively improve the precision and accuracy of the similarity calculation, and has a better stability. Presented in [10], is a comprehensive analysis of a new recommendation framework that is both optimal and fast. The derived methodology is highly optimised, and real-time analytics is also possible, since it only uses movie ranking as the primary component of the feature set. The recommender model is complex, and it can create new recommendations based on subscriber changes. The paper defines a movie recommendation system focused on items that uses movie ratings as a function. Using a modified cosine similarity matrix, it finds movies to suggest. The experiment's Root Mean Square Error (RMSE) is 1.01. In the item-based collaborative filtering algorithm, the accuracy is 79.72 percent. In paper [12], a distributed filtering-based movie recommendation framework has been implemented. This framework uses Apache Mahout to provide movie recommendations that take into account the ratings provided to movies. When market requirements become more complex, there is a greater reliance on extracting useful knowledge from massive amounts of raw data to push business solutions. The same can be said about automated recommendation services, which are becoming commonplace in consumer sectors like books, music, clothes, movies, news stories, locations, and utilities. Users' input is collected by the applications in order to enhance potential recommendations. The paper describes how two collaborative filtering algorithms were used in conjunction with Apache Mahout to build a movie recommender system.

Table 1. The following table gives a summary of research work conducted from various technical papers on similar topics.

Research Paper Title	Algorithm Used	Objective	Advantages	Limitations
[1] A collaborative filtering recommendation algorithm based on Contents' Genome	Content similarity matrix	Proposes the concept to solve low accuracy leaded by content-based recommendation algorithm in mass recommendation object.	1. The mean absolute error (MAE) value calculated using this algorithm is lower when compared to traditional collaborative filtering algorithms. 2. Another advantage is that we will not be biased by item popularity.	As for all Content-Based (or similar) approaches, one shortcoming is that we will not be able to uncover links between items that are not given by their intrinsic properties.
[2] A Content-based Movie Recommender System Based on Temporal User Preferences	Bayesian non-parametric framework: Interests extraction, inferring of preferences and prediction	To provide a user-centred framework that incorporates the content attributes of rated movies (for each user) into a Dirichlet Process Mixture Model to infer user preferences	The proposed method used a temporal preference model of the user to recommend favours movies to the users. For user modelling, the proposed method after	Improvement of the diversity and serendipity measure of Recommendation could be made in this proposed methodology of the recommendation system.

		and provide a proper recommendation list.	extracting user interests from user profile, the priority of each interest is inferred as the user preference.	
[3] An effective collaborative movie recommender system with cuckoo search	K-means & cuckoo search optimization. K-means-cuckoo based collaborative filtering framework	To suggest a recommender system through data clustering and computational intelligence.	The experiment outcomes on the Movielens dataset discussed indicated that the approach that was discussed provides high performance regarding accuracy and is capable of providing reliable and personalized movie recommendation systems with the specific number of clusters.	If the initial partition does not turn out to work well then, efficiency may decrease.
[4] An improved content based collaborative filtering algorithm for movie	Applied a set matching comparator instead of Naive Bayes for collaborative filtering.	To give a novel method to find the similar content between two items	On comparing with Pure CF, it was found that for high sparsity levels this approach works better than Pure CF and the MAE values were better	The results of this approach may not be the same when tested with a much larger dataset.

recommendations			than the one's generated from the Pure CF. At high sparsity, the data available is less and hence Pure CF doesn't perform better.	
[5] Big Data Analysis: Recommendation System with Hadoop Framework	Nearest N Users Algorithm, Mahout supported algorithms,Similarity Measures	To propose a recommendation system for the large amount of data available on the web in the form of ratings, reviews, opinions, complaints, remarks, feedback, and comments about any item (product, event, individual and services) using Hadoop Framework.	Resultant graph is showing that whenever file size is increasing the execution time is not increasing in the same ratio and we know that data size that are in the form of ratings, ranks, review, feedback are increasing drastically.	Here the Recommendation system by applying the weightage of summarized reviews and opinions on the rating of item can be the future enhancement of this work.
[6] Design and Implementation of Movie Recommender System Based on Graph Database	User-based Collaborative Filtering, Similarity measures : Euclidean Distance	To use graph database which is good at dealing with complex relations along with using the traditional method for	The data model obtained by using graph database to organize data is simpler and more expressive than using traditional relational	Graph databases are not as useful for operational use cases because they are not efficient at processing high volumes of transactions and they are

		collaborative filtering.	database or other NoSQL database. Finally, the degree of recommendation of films is distinguished by color and size, allowing users to have a better experience.	not good at handling queries that span the entire database.
[7] Fast Collaborative Filtering with a k-Nearest Neighbour Graph	k-Nearest Neighbor Graph Construction Fast Recommendation Algorithm	To present a fast Collaborative Filtering algorithm using a k-Nearest Neighbor graph.	This algorithm predict the preferences of the k-nearest neighbor items and decreases the execution time by calculating a k-nearest neighbor item graph in less time based on greedy filtering.	As the number of users grow, the algorithms suffer scalability issues. If you have 10 million customers and 1,00,000 movies, you would have to create a sparse matrix with one trillion elements.
[8] Exploiting Visual Contents in Posters and Still Frames for Movie Recommendation	Unified Visual Contents Matrix Factorization(UVMF) The Learning Algorithm of UVMF	To use visual contents to improve the performance of movie recommendation and propose a novel movie recommendation model named unified visual contents matrix factorization	UVMF outperforms several outstanding methods on real-world data. Taking visual feature extraction and recommendation into a unified optimization process	It takes time to compile. Even if it is flexible, it needs time with the original draft.

		(UVMF) that integrates visual feature extraction and recommendation into a unified framework.	improves the accuracy of recommendation.	
[9] Item-Based Collaborative Filtering in Movie Recommendation in Real Time	Computation of Similarity Matrix Prediction of Unknown Ratings Recommendation of Top k Items	To provide prediction of the different items in which a user would be interested in based on their preferences.	The derived system is quite optimized and real-time analytics is also possible.	The model accuracy (79.72%) is somewhat low as compared to contemporary recommendation models.
[10] Movie Recommendation System Based on Movie Swarm	Movie Swarm Mining Interesting and Popular Movie Mining (IPM)	To recommend movies to a new user as well as the others.	This algorithm solves the new item recommendation problem and provides an idea about the current trends of the popular movies and user interests.	The method has a shortcoming of finding the group of a user depending on the movie genre, if he/she enjoys a diverse set of movies.
[11] Movie Recommendation System Using Collaborative Filtering	User-based Filtering Item-based Filtering Model Building	To describe the implementation of a movie recommender system via two collaborative filtering algorithms	The movie recommender system built in this paper facilitates the understanding of how a	The recommender system could be developed using hybrid filtering approach instead of CF as hybrid systems

		using Apache Mahout.	recommender system works.	provide better accuracy.
[12] Movie Recommendation Using Map Reduce	Sequential Algorithm Parallel Algorithm	To present the algorithm used in the implementation of an application, which supports collaborative filtering algorithm and recommends movies using the Parallel Java 2 Library.	Using the ability of parallel computing, the running time of the recommender system is reduced.	The recommender system can be improved by considering more factors than just ratings. These factors could be gender of the user and age of the user.
[13] User-based Collaborative-Filtering Recommendation Algorithms on Hadoop	Map Reduce for CF Data Partitioning phase Map phase Reduce phase	To implement user-based CF algorithm on a cloud computing platform, namely Hadoop, to solve the scalability problem of CF.	The design algorithms enable Collaborative Filtering algorithm in Hadoop platform to take the good performance.	The Collaborative Filtering algorithms on Hadoop platform cannot reduce the recommendation response time for a single user.
[14] Movie Recommender System Based on Percentage of View	Collaborative filtering, Content-based filtering along with Residual method	To prove the effectiveness of the approach, it is shown that this feature can be a good indicator of users' like and dislike. The best approach is	To improve the performance of movie recommender systems on implicit feedback.	To test the system using more data and improve the accuracy of the system. To model users better to improve the

		determined and used for recommendation.		accuracy of the system.
[15] MovieSwarm: Information Mining technique for Movie Recommendation System	Movie Swarm Mining	Generating movie swarm which is very useful for movie producers and can solve new items problem. Also finds out which genres of movie should be recommended among followers, that solves new users' recommendation problem	Generates a new system that mines the movie swarm with two pruning rule and used vertical data format for frequent items mining.	Users clustering techniques using movie rating and expecting to provide trust recommendation system without using trust matrix.
[16] ORBIT: HYBRID MOVIE RECOMMENDATION ENGINE	Hybrid Recommendation Algorithm	Comparative case study of conventional recommendation algorithms to ORBIT's Hybrid movie recommendation algorithm has also been studied.	It lets the system cons ider collaborative data without relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item.	Content description and Over-specialization

[17] Personalized Movie Recommender System Using Rank Boosting Approach on Hadoop	MapReduce	This service recommender system uses keywords to indicate user preferences and a variation of collaborative filtering algorithm called user based collaborative filtering is used to provide recommendation to customers.	Scalability, Flexibility and Fast.	To incorporate the dictionary feature, which will help to identify words with same meanings.
[18] Rating Prediction on Movie Recommendation System: Collaborative Filtering Algorithm (CFA) vs. Dissymetrical Percentage Collaborative Filtering Algorithm (DSPCFA)	DSPCFA, CFA and RMSE evaluation method	This study also uses two similarity measurement methods, namely the pearson correlation similarity method and the cosine similarity method as a comparison to determine the characteristics of each measurement method.	No domain knowledge necessary, Serendipity	Cannot handle fresh items and Hard to include side features for query/item
[19] Research and Implementation of Hybrid Recommendati	Hybrid Recommendation Algorithm	Use the Word2Vec model to train the tag information of	It lets the system cons ider collaborative data without	Content description and

on Algorithm Based on Collaborative Filtering and Word2Vec		the mobile data to get the similarity between the tags, and recommend applications to the user according to the similarity.	relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item.	Over-specialization.
[20] Simulation of Genre based Movie Recommendation system using Hadoop MapReduce Technique	Hybrid Recommendation Algorithm	In this paper the objective is to explain how the results from two algorithms which are run on Hadoop are combined to get more accurate movie recommendations.	It lets the system consider collaborative data without relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item.	Content description and Over-specialization.

3 Proposed Work

3.1 Problem Statement

This model implements content-based filtering and collaborative filtering on a given dataset using Hadoop and Hive framework. It will be recommending movies based on these filtering techniques using various machine learning algorithms.

3.2 Dataset

The recommendation system uses statistical data as input. The dataset is called TMDB 5000 movies dataset. The Movie Database (TMDb) API was used to build this dataset. It also has information on a variety of other films, actors and actresses, crew members, and television programmes. Following pre-processing, the dataset is educated using Machine Learning techniques. The numerous parameters which the movie dataset consists are as follows: Title, Overview, Budget, Cast, Date of release, Crew members, Genre, Vote count, etc. Along with the above-mentioned parameters we have standard parameters such as movie id, movie name and other parameters which come under the metadata of the movies like popularity, original language, runtime, IMDB id, homepage, revenue etc. These are not quite rarely used and referred in coming up with a recommendation of a movie for a user.

3.3 Algorithm

3.3.1 Collaborative filtering. To decide the top-N recommended items for a given consumer, collaborative filtering systems typically use a three-step mechanism. They begin by locating the k database users who are the most comparable to the currently active user. In the second stage, they add up the objects these users bought and give a weight to each one depending on its value in the list. They choose and recommend the N items with the highest weight that have not yet been bought by the successful customer from this union in the third and final level. Singular Value Decomposition (SVD) is one of the most commonly used unsupervised learning algorithms, and it's at the heart of multiple recommendation systems at major companies including Google, Netflix, Facebook, and YouTube.

Table 2. The below table depicts the analysis of various machine learning algorithms used to perform collaborative filtering.

	Random Forest	**Logistic Regressio n**	**Linear Discrimin ant analysis**	**AdaBoost Classifier**	**Gaussian Eliminatio n**	**K Neighbor s**
Accur acy	28.749%	28.491%	27.399%	29.131%	27.854%	41.855%
CPU time	0:00:03.1 84568	0:00:08.9 44910	0:00:00.0 69813	0:00:03.1 09400	0:00:00.0 19977	0:00:00.1 20746

Table 3. The below table depicts the analysis of using SVD for collaborative filtering

Sr.no	**Parameters**	**Value**

1	MAE	[0.68432502, 0.69153599, 0.69482132, 0.69874779, 0.68923494]
2	RMSE	[0.88462886, 0.89607402, 0.9042461, 0.90577317, 0.8972781]
3	Fit time	(3.6630747318267822, 3.68796706199646, 3.7052431106567383, 3.697650909423828, 3.686905860900879)
4	Test Time	(0.10649442672729492, 0.1664433479309082, 0.10650634765625, 0.10571694374084473, 0.10575079917907715)

3.3.2 Content-based filtering. We look for item profiles and their corresponding user ratings when creating a user profile. The number of the item profiles makes up the user profile, with the sum reflecting the feedback given by the customer or user. After the user profile has been developed, we use cosine resemblance between the user generated profile and the object profile to approximate the user profile's similarity to all the items in the database.

3.3.3 Description based recommendation.

3.3.3 Description based recommendation. In this model, while implementing content-based filtering technique over the movies dataset the description parameter is used as a feature to compute the likeliness of that movie for a particular user. The text punctuation and many common words in English language which are present in the description or overview the movies (stop words) are eradicated and then using TF-IDF feature the importance of the remaining words are learnt. The stop words basically are the words which make the learning of the description/overview of a movie more complex and hence it needs to be removed before the processing of the data. This will ensure more accurate understanding of the movie description and hence result in more accurate recommendations.

3.3.4 Genre, Keywords and Crew details-based recommendation. In this approach, the parameters such as genres, crew details, keywords and etc are extracted and processed. Then based on the information on the director, cast member or genre from a high-rated or recently watched movie by a user similar movie involving similar features of the parameters are suggested**.**

The steps which are taken in order to essentially create a keyword list are lowercasing words, removing stop words, removing punctuation and lemmatizing words.

3.4 Implementation

The sklearn model performs pre-processing on the data provided from the dataset. This model is trained over the dataset which is further split into the testing and training dataset. The dataset is stored on Hive database and retrieved using Hadoop. Once the model is trained a recommendation algorithm (collaborative or content-based filtering) can be implemented over it to obtain the necessary recommendations.

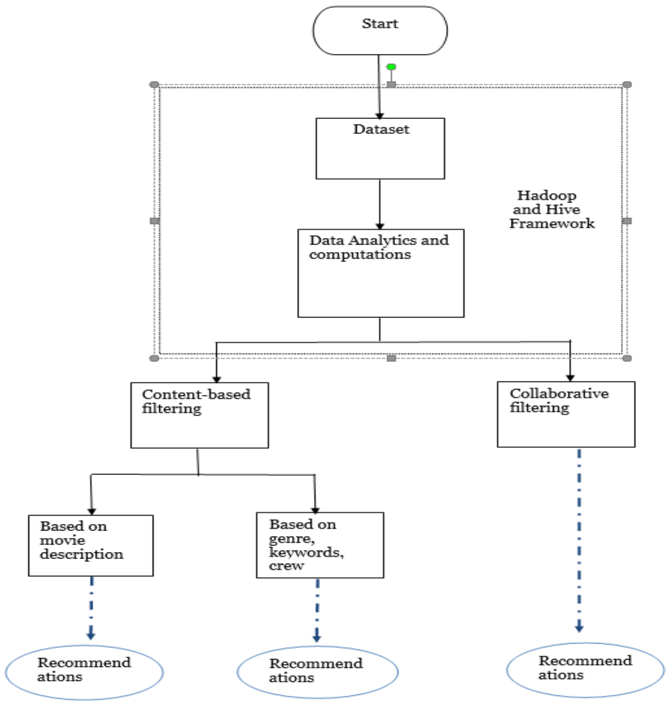

Fig.1. Architecture of the recommendation system

Surprise scikit makes it easy to implement new algorithm ideas as well. The aim is to find movies to recommend to a consumer based on his previous movie choices. We find users who have scored the same movie for each movie, and then we find movies rated by each of those users - the number of times a related movie is listed for a particular movie indicates how similar it is to the original movie, and therefore has a higher likeability for the user.

3 Results Analysis

The best results were obtained when singular value decomposition algorithm was used. Many other machine learning algorithms were used to get recommendations but their accuracy level was not at par with the results achieved from singular value decomposition method.

4.1 Hadoop and Hive cluster

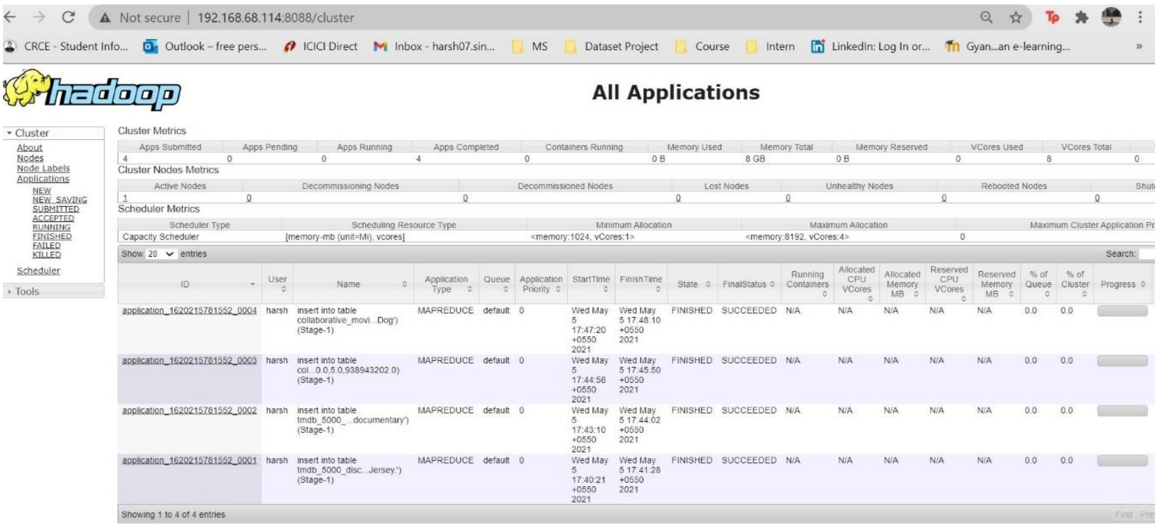

Fig .2. Screenshot of Hadoop cluster metrics

The Hadoop cluster metrics displays four applications successfully completed along with the application type i.e., MapReduce, application id, the start time and the finish time of the applications.

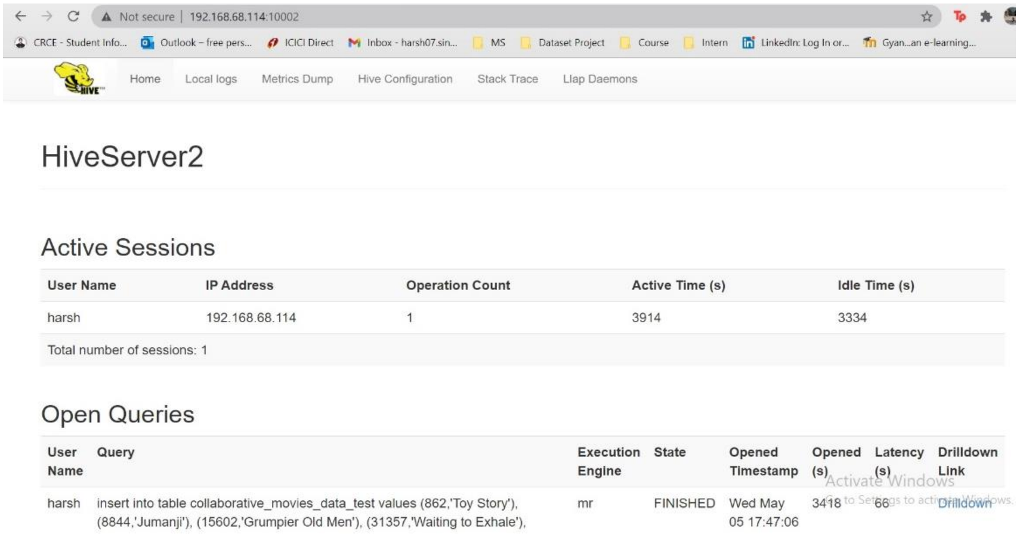

Fig.3. Screenshot of Hive server

Hive framework is used to insert the queries on top of Hadoop clusters. The above screenshot depicts various parameters of the server such as timestamp, latency and status.

4.2 Collaborative Filtering

```
best_movies_for_user(1)
```

```
The Million Dollar Hotel
The Red Elvis
Sleepless in Seattle
Eight Miles High
The Good Thief
Nell
Flags of Our Fathers
Los Olvidados
The Thomas Crown Affair
Sicko
```

```
best_movies_for_user(3)
```

```
Sleepless in Seattle
Galaxy Quest
The Sixth Sense
Broken Blossoms
While You Were Sleeping
Cheerleaders' Wild Weekend
Dead Man
Totally Blonde
Crank
Nell
```

Fig.4. Recommendations from user based collaborative filtering method

The above results for recommendations of top 10 movies for a particular user were obtained using SVD. The recommended movies have not been watched by the particular user, but by other individuals with similar liking in that genre of movies.

```
svd = SVD()
evaluate(svd, data, measures=['RMSE', 'MAE'])
```

```
{'test_rmse': array([1.09624665, 1.05031068, 1.04592488, 1.0456505 , 1.06268   ]),
 'test_mae': array([0.87501212, 0.84258385, 0.84624695, 0.84181778, 0.85818033]),
 'fit_time': (0.511338472366333,
  0.5747692584991455,
  0.5378849506378174,
  0.4835829734802246,
  0.5236251354217529),
 'test_time': (0.016027450561523438,
  0.01406407356262207,
  0.013962984085083008,
  0.00797891616821289,
  0.014028310775756836)}
```

Fig.5. RMSE and MAE values using SVD

The root mean square error and mean absolute error calculated for collaborative filtering using SVD is shown above.

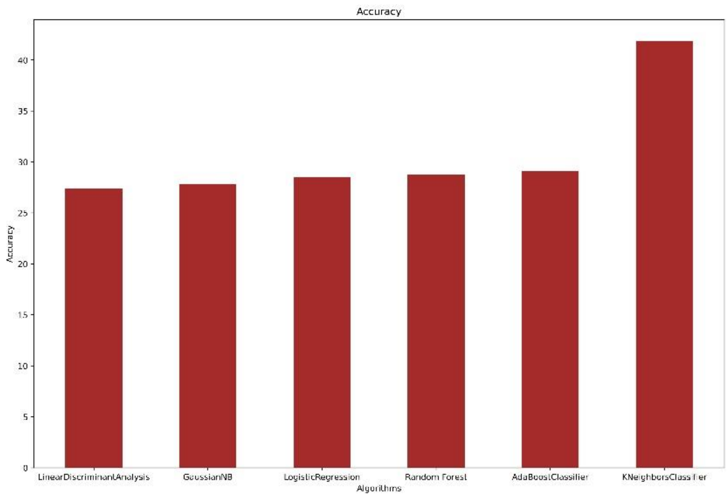

Fig.6. Graphical representation of accuracy of various ML algorithms other than

The other machine learning algorithms used over the movie's dataset were Random Forest, Logistic Regression, Linear Discriminant Analysis, Gaussian elimination, K-Neighbours and Decision Tree algorithms. From the above analysis, these algorithms have a low accuracy around the same range and this is the reason why SVD is preferred in contrast to these algorithms.

4.3 Content-based Filtering

```
get_recommendations('Batman & Robin' , cosine_sim)
```

```
210

210                              Batman & Robin
299                              Batman Forever
1359                                     Batman
428                              Batman Returns
212                      The Day After Tomorrow
514                       Ice Age: The Meltdown
4771                         The Exploding Girl
3857    Batman: The Dark Knight Returns, Part 2
3                         The Dark Knight Rises
65                              The Dark Knight
9            Batman v Superman: Dawn of Justice
Name: title, dtype: object
```

Fig.7. CBF: Recommendations based on movie description

With content-based filtering, the similar movies to a movie which is liked by the user is recommended. To find the similarity among the movies, cosine similarity is used. For example, when the movie "Batman & Robin" is liked by user, then the movies based on the similar description is recommended as shown above.

```
get_recommendations('Iron Man', cosine_sim2)
```

```
68

68                         Iron Man
79                       Iron Man 2
7           Avengers: Age of Ultron
16                     The Avengers
26       Captain America: Civil War
31                       Iron Man 3
39                     TRON: Legacy
4401             The Helix... Loaded
83                        The Lovers
193                      After Earth
4117               Six-String Samurai
Name: title, dtype: object
```

Fig.8. CBF: Recommendations based on movie genre, keywords & crew details

If the movie "Iron Man" is liked or highly rated by a user, then in this case, the movies recommended would be based on the similar genre, cast, crew details. As it can be from the above picture the movies Iron Man 2, Iron Man 3, The Avengers etc have identical cast and crew and belong to the same genre.

4 Conclusion and Future Work

Using Hadoop and the Hive Framework, we suggested a recommendation method for the vast amount of movie dataset available on the web in the form of scores. The aim of this method is to provide the consumer with a forecast of which movies he or she might enjoy based on his or her tastes. Our proposed novel recommendation framework solution is really good. Since, our model uses movie ratings and description & genre, keywords and crew details in our system as the principal component for collaborative and content-based filtering respectively, the system that is derived is quite optimized. Our model is quite dynamic and generates recommendations based on the ratings provided by the various number of users.

This recommender system can be improved by considering more factors. For example, we could use the age of a user to recommend them movies that are watched by most of the users of that particular age. Also, a very large dataset of movies can be used in the same way and better results can be obtained in this way. Multiple computers could be used for processing the large dataset of movies and by using big data tools like Apache’s Hadoop and Hive. In the future, new movies that are released can also be considered instead of a particular dataset of movies in a real time model where an application can be created and this model could be implemented.

ACKNOWLEDGEMENT. Firstly, we express our profound sense of reverence to our mentor Prof. Vaibhav Godbole, for his continuous guidance and help throughout the entire course. We are also grateful to Dr. Jagruti Save (Head of Department) and Dr. Srija Unnikrishnan (Principal of Fr. Conceicao Rodrigues College of Engineering) for their endless support. We also thank all staff members of the Department of Information Technology, for providing assistance and cooperation whenever required.

REFERENCES

[1] A collaborative filtering recommendation algorithm based on contents' genome, IET International Conference of Information Science and Control Engineering 2012, Ying-Wei Chen, Xin Xia, Yong-Ge Shi School of Mathematics and Computer Science, JiangXi Science Technology Normal University, NanChang, 330013 China

[2] 2017 3rd Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS) A Content-based based on Temporal Movie User Recommender Preferences System Bagher Rahimpour Cami, Hamid Hassanpour, Hoda Mashayekhi Faculty of Computer Engineering & IT Shahrood University of Technology Shahrood, Iran

[3] W.-K. Chou Department of Computer Science and Information Management, Providence University, Taichung, Taiwan,Springer Science+Business Media Singapore 2016 J.C. Hung et al. (eds.), Frontier Computing, Lecture Notes in Electrical Engineering

[4] Egyptian Informatics Journal July 2017, Pages 105-112 An effective collaborative movie recommender system with cuckoo search, Rahul Katarya,Om Prakash Verma,Department of Computer Science & Engineering, Delhi Technological University, Delhi, India

[5] Proceedings of 2017 Tenth International Conference on Contemporary Computing (IC3), 10-12 August 2017, Noida, India, An Improved Content Based Collaborative Filtering Algorithm For Movie Recommendations, Ashish Pal, Prateek Parhi and Manuj Aggarwal Department of Computer Science ARSD College, University of Delhi

[6] 2015 IEEE International Conference on Computational Intelligence & Communication Technology Big Data Analysis: Recommendation System with Hadoop Framework Jai Prakash Verma Assistant Professor, Institute of Technology, Nirma University, Ahmedabad

[7] 2017 14th Web Information Systems and Applications Conference Design and Implementation of Movie Recommender System Based on Graph Database, Ningning Yi School of Computer Science Communication University of China Beijing, China

[8] This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 20110017480). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2012R1A1A1043769).

[9] XIAOJIE CHEN, PENGPENG ZHAO, JIAJIE XU, ZHIXU LI, LEI ZHAO,YANCHI LIU, VICTOR S.SHENG, AND ZHIMING CUI

[10] 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)

[11] 2012 Second International Conference on Cloud and Green Computing

[12] A. V. Dev and A. Mohan, "Recommendation system for big data applications based on set similarity of user preferences," 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, 2016, pp. 1-6. doi: 1O.1109/ICNGIS.2016.7854058

[13] Zhi-Dan Zhao and Ming-Sheng Shang, User-based Collaborative Filtering recommendation algorithms on hadoop, Knowledge Discovery and Data mining 2010, Third International Conference, 9th Jan -10th Jan 2010, 478-481.

[14] This work was partially supported by the National Natural Science Foundation of China under Grant Nos. (60973069, 90924011), and by the China Postdoctoral Science Foundation under Grant No. 20080431273.

[15] 5th Conference on Knowledge-Based Engineering and Innovation, Iran University of Science and Technology, Tehran, Iran, Ramin Ebrahim Nakhli, Hadi Moradi, Mohammad Amin Sadeghi, School of Electrical and Computer Engineering, University of Tehran

[16] 2012 7th International Conference on Electrical and Computer Engineering
20-22 December, 2012, Dhaka, Bangladesh, Sajal Halder, Kyung Hee University
Md. Samiullah, University of Dhaka, Bangladesh, A. M. Jehad Sarkar, Hankuk University of Foreign Studies, South Korea

[17] 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN 2013), Dharmendra Pathak, Sandeep Matharia, Chameli Devi School of Engineering, Indore, India

[18] International Journal for Innovative Research in Science & Technology| Volume 2 | Issue 02 | July 2015, Ms. Annies V Jose, Ms. Jini KM, University of Calicut NCERC, PAMPADY, Thrissur.

[19] 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), Johan Eko Purnomo, Sukmawati Nur Endah, Universitas Diponegoro, Semarang, Indonesia

[20] 2015 8th International Symposium on Computational Intelligence and Design, Yao Xiao, Quan Shi, Nantong University, Nantong, China