DESIGNING HYBRID COLLABORATIVE FILTERING

MOVIE RECOMMENDATION SYSTEM - PYTHON

BY

ADHARSH R

Roll No: 2019201002

MBA (2019-2021)

A PROJECT REPORT

For Summer Internship

Submitted to the

FACULTY OF MANAGEMENT STUDIES

MASTER OF BUSINESS ADMINISTRATION

COLLEGE OF ENGINEERING,

GUINDY

ANNA UNIVERSITY

CHENNAI 600025

OCTOBER, 2020

Abstract

Traditional collaborative filtering algorithm does not take into account the user's interest factors, at the same time, it has the problem of sparse data, poor scalability and so on, which directly affects the quality of the recommendation. Through this project, a hybrid model which clusters users and then recommends movies based on collaborative filtering is designed. Firstly, according to the user's existing score, a method is used to cluster the users. Then, a collaborative filtering algorithm (Alternative Least Square) is used to recommend movies which would effectively improves data sparsity.

TABLE OF CONTENTS

1. INTRODUCTION AND BACKGROUND

Usage of recommender systems have increased across businesses to gain the continues consumer relationship. Almost every day, customers use online services to find new items or service to purchase based on several variables of interest. The interest of users can be predictable always and may change abruptly over time; and hence, the concept of recommender system. Recommender systems uses algorithms that are mostly dependent on machine learning to suggest items for users. Recommendations are usually based on User information or item information.

Flip Kart, Amazon, Prime Video, Netflix, YouTube, Facebook and many others are increasingly using recommender systems to increase the good experience to users. Movies and music sites usually gain a significant advantage using recommendation systems in enhancing user interaction and engagement. This is easily done by monitoring user ratings, number of clicks, the relevance of the recommendation and the accuracy.

In systems with huge amount of data, recommendation systems perform with great accuracy and provide a great profitability for businesses. Recommender systems fall into three categories: collaborative, content-based, and hybrid filtering.

The collaborative filtering makes recommendations based on the similarity among the users or the items. The content-based approach depends on item metadata to establish the relationship between items for recommendation. The third, hybrid filtering approach, is a diverse combination of collaborative and content-based approaches.

While a hybrid approach to collaborative filtering focuses on combining the varied approaches to collaborative filtering, an adaptive approach to collaborative filtering focuses on changing the dynamics of user preference. Due to the unpredictable nature of user preferences on a system and the abrupt

nature with which the preferences change, there is the need for these systems to adjust properly using some algorithms and the need for diverse combinations of the collaborative and content-based approaches (hybrid), thus introducing the concept of adaptive hybrid collaborative recommender systems.

Recommender systems are plagued with computational and modelling problems with the advent of making successful recommendations. The Netflix (DVD Rental Company) prize challenge set the perfect tone for the advancement of research in recommendation systems, having the collaborative filtering (CF) approach gaining much attention and use in many domains.

CF approaches include model-based (using data mining techniques in making a recommendation), memory-based (using explicit data such as ratings to make recommendations), and hybrid approaches (combining the model and memory-based approaches).

In this project, a Hybrid Collaborative Filtering model susceptible to user preference is created. Specifically, K-Means clustering and the ALS (Alternate Least Square) model are combined together.

1.1.OBJECTIVES

This project is to achieve the following objectives:

1. Design, develop and deploy a hybrid recommender system.

2. Evaluate the computational efficiency in terms of speed.

3. Evaluate performance metrics for the hybrid recommender system

2. LITERATURE REVIEW

Many studies implement the memory and model-based approaches for making a recommendation of items. These systems differ algorithmically from data search algorithms in their ability to produce results without an explicit request from the user. Prevalent among these are the content-based, collaborative, and hybrid approaches. They constitute 35% of sales for Amazon,2/3 of Netflix movies viewed, and 38% of Google's News click-through. The hybrid approach to recommender systems, as of 2016, suffers from minimal research efforts, although some studies indicate its high accuracy level compared to the other methods. Some ML techniques, like K-Means, despite its popularity, have not been researched enough.

Collaborative filtering recommendation systems that implement single algorithms have many drawbacks. K. Haruna, M. A. Ismail, D. Damiasih, J. Sutopo, and T. Herawan assert that the user-based approach, for example, based on the behaviour of other users; thus, a new behaviour does not lead to an update of items recommended. The item-based approach on the other hand, focuses on item metadata which presents many challenges, especially when there is a large number of items. These problems ushered in the mixed recommendation algorithms, thus, the hybrid collaborative filtering techniques. Y. Song, W. Ji, and S. Liu, use user and item combination as a hybrid approach. This makes recommendations by weighted summing of the recommended results of two algorithms. Others also merged the user-based CF with the content-based.

While a hybrid approach to collaborative filtering is a common practice among many recommender systems and following the success of the collaborative filtering techniques, their adaptivity hasn’t gained much attention as these systems in themselves have a chunk of challenges. This project is aimed to overcome some challenges by creating a hybrid model.

Netflix serves as a subscription model for personalizing recommendation on movies and shows a user may be interested in. Fundamentally, estimations of the chances of watching a particular title in the catalogue based on ratings and viewing history, similarity among users, title, genre, etc. information are taken into account. The time of the day, device type, and length of viewing are also employed in the recommendation process.

On adding a new profile or registering as a new user, suggestions are made and as a user selects these optional suggestions, similar recommendations are made; otherwise, popular movies are recommended.

Amazon Machine Learning, Strand, Peeruis, Azure ML and IBM Watson are some proprietary recommender models that are employed in the industry, while, The Universal Recommender, Raccoon Recommendation Engine, HapiGER and easyRec are open source models for the recommendation.

3. METHODOLOGY

The data is collected from the database and it is pre-processed to create a hybrid model. Figure 0.1 shows the overall design of the hybrid model.

The users are clustered through K means clustering as described in Figure 0.2.

And the movie recommendation is made as described in Figure 0.3

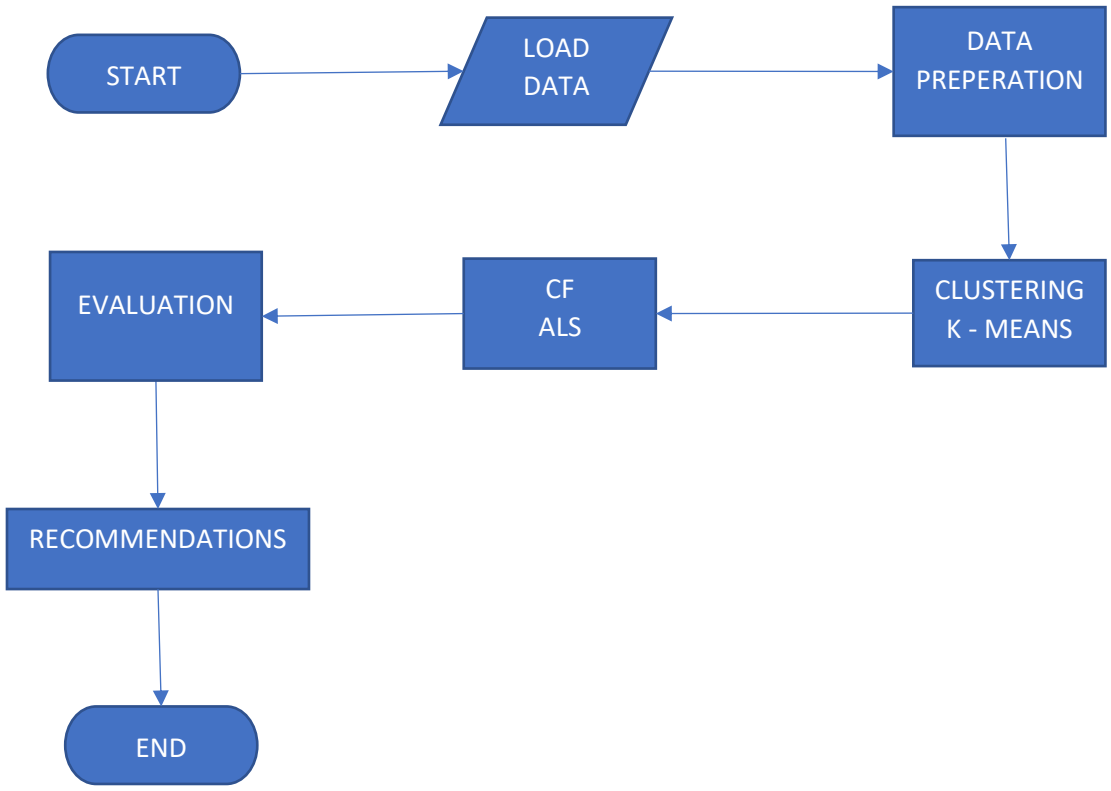


Figure 3.1 Hybrid model system design

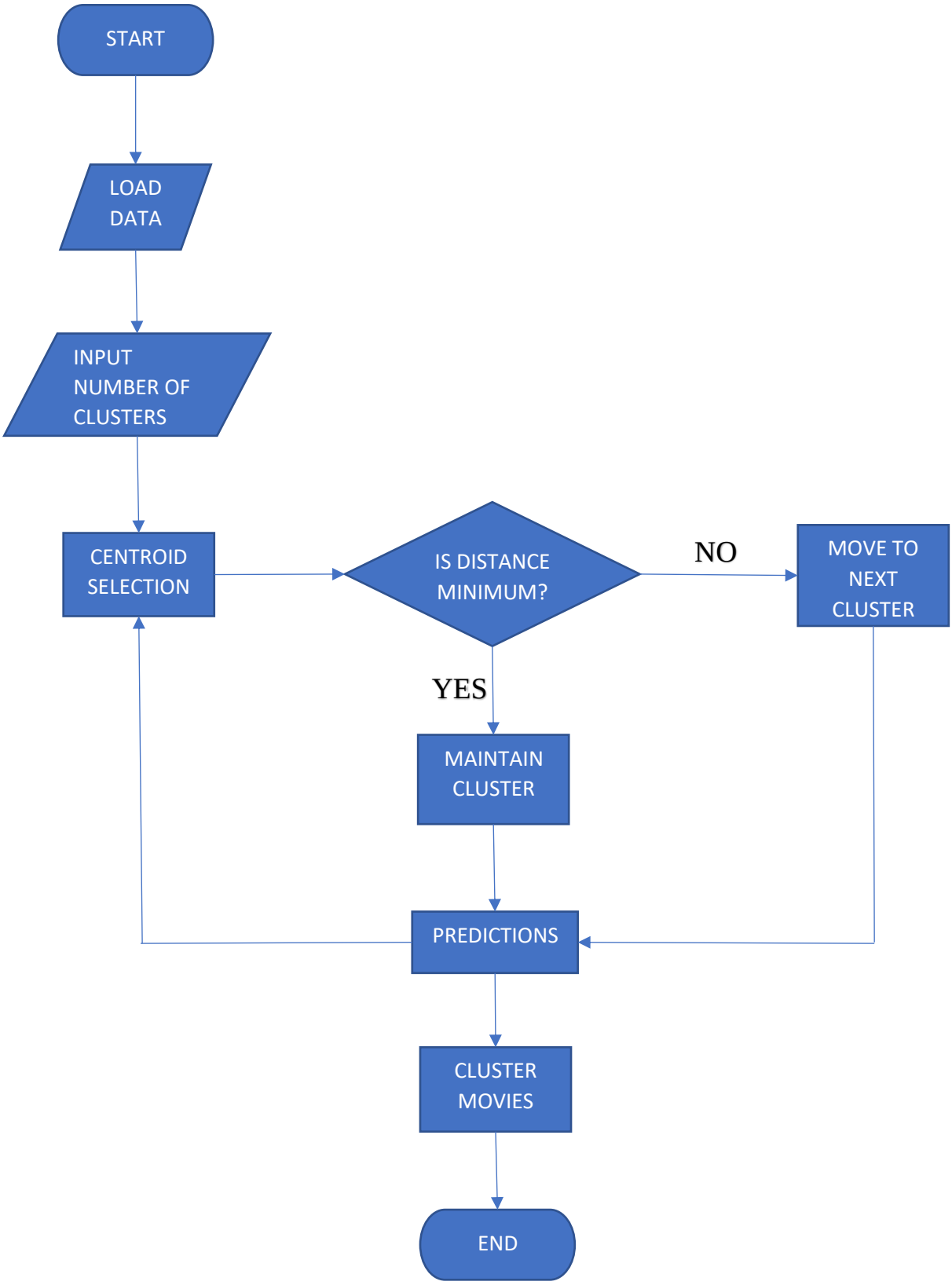


Figure 3.2 K Means Clustering Algorithm Implementation

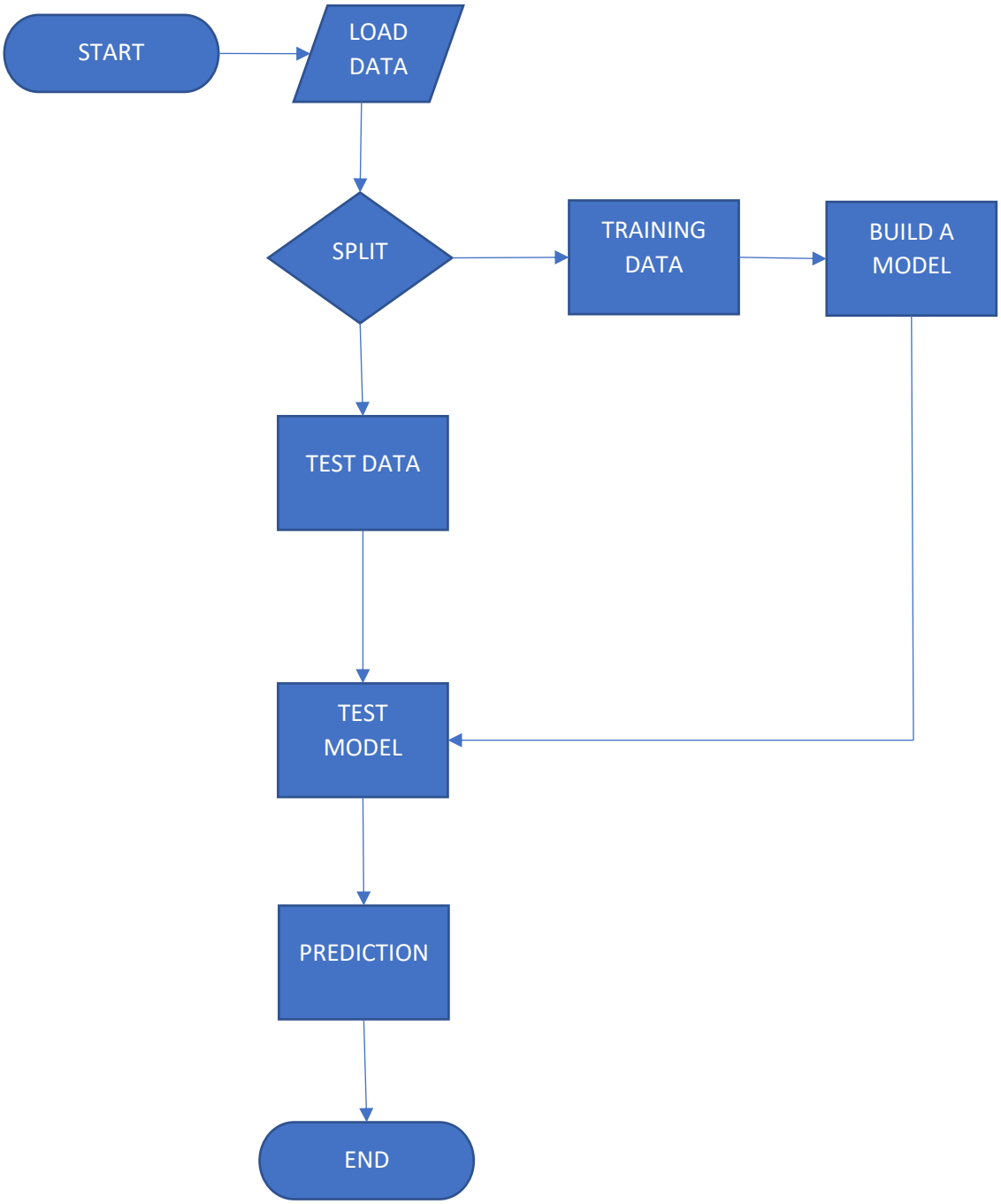


Figure 3.3 Alternating Least Square (ALS) Model

4. RESULT AND DISCUSSION

4.1.SYSTEM IMPLEMENTATION:

4.1.1. ABOUT THE DATASET

The dataset used is Movie Lens Dataset obtained from Kaggle which contains the data of movies released on or before July, 2017.

This dataset has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website. But for the simplicity, a subset of the data which has 670 users and their ratings is chosen.

4.1.2. DATA PRE-PROCESSING

The Rows with empty Genre, movie Id, user Id are removed and data is checked for missing values.

Different tables (ratings and movie metadata) are merged together.

Data is normalised by dividing the rating with overall mean rating of the user and for each genre the average normalised score is calculated.

With the average normalised score for each genre for each user, cross tabulation is created as shown in Figure 4.1.

genres	Action	Adventure	Animation	Comedy	Crime	Documentary	Drama	Family	Fantasy	Foreign	History	Horror	Music	Mystery	Rom
userId															
1	0.000000	0.000000	0.000000	1.142857	0.000000	0.000000	0.857143	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.00
2	0.947712	0.947712	1.421569	0.956328	1.137255	1.421569	0.952451	0.000000	0.852941	0.0	0.0	1.042484	0.000000	0.000000	1.13
3	1.059322	1.200565	0.000000	1.165254	0.800377	0.000000	0.975860	0.000000	0.000000	0.0	0.0	0.776836	0.000000	0.000000	1.12
4	1.045585	1.092402	0.000000	0.957626	1.024127	1.170431	0.955325	0.000000	1.111910	0.0	0.0	1.053388	1.014374	0.000000	0.93
5	0.979525	0.958231	0.000000	0.922741	1.043407	1.022113	1.070025	0.000000	1.047666	0.0	0.0	0.958231	0.000000	0.000000	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
667	1.128713	0.917079	0.000000	1.034653	0.705446	1.128713	0.929528	0.000000	0.846535	0.0	0.0	0.705446	0.000000	1.410891	0.84
668	0.952381	0.000000	0.000000	0.000000	0.000000	0.952381	1.011905	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.00
669	1.309524	0.000000	0.000000	1.047619	0.916667	0.000000	1.047619	1.047619	1.047619	0.0	0.0	0.000000	0.000000	0.000000	0.52
670	1.328125	0.000000	0.000000	1.062500	0.000000	0.000000	0.990057	0.000000	0.796875	0.0	0.0	0.000000	0.000000	0.000000	0.00
671	1.055410	0.986877	0.863517	0.937533	1.233596	0.986877	1.027997	0.000000	0.986877	0.0	0.0	0.863517	0.000000	0.986877	0.86

.

Figure 4.1

Normalisation Score = Rating by the user for particular movie /

Overall Average rating by the same user

Mean Normalisation Score (Genre Wise) = Sum of Normalisation Score for all movies rated by the user in same genre /

Number of movies rated by the user in the genre

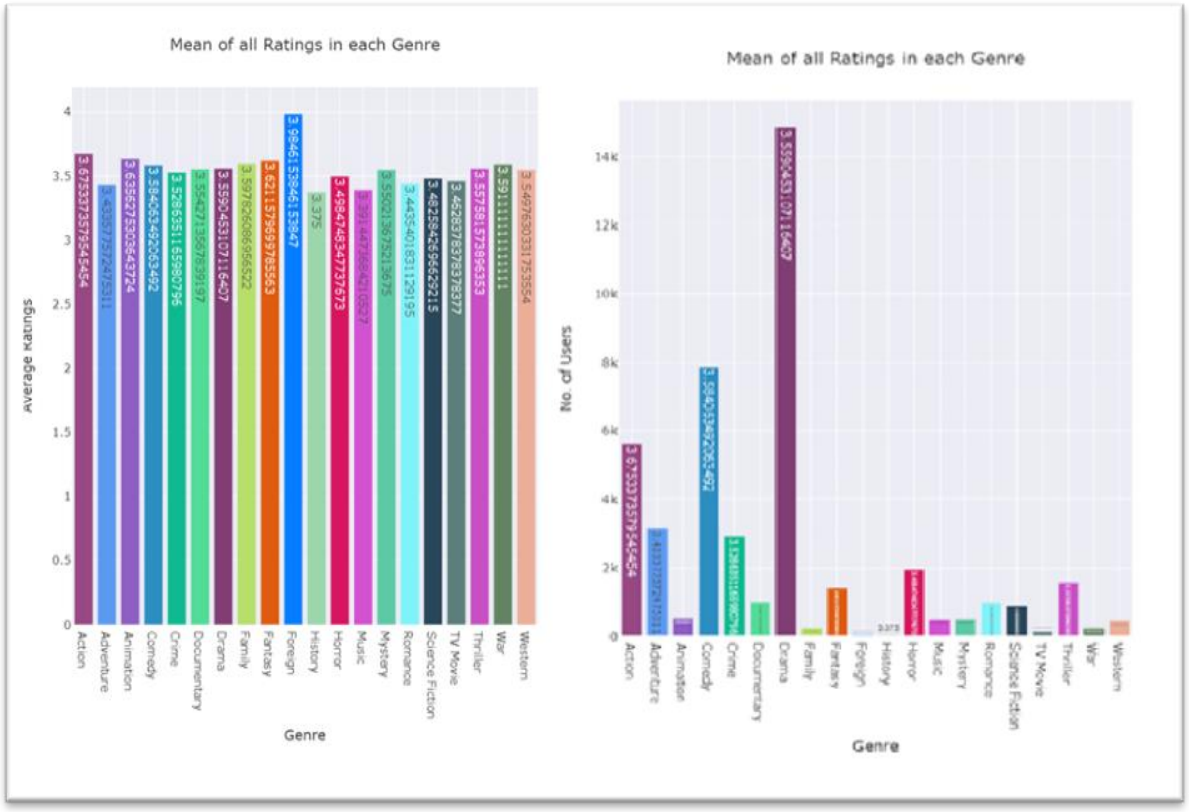


Figure 4.2

4.1.3. CLUTERING THE USER (K MEANS)

Data clustering is known to be NP-hard for finding groups in heterogeneous data. K-Means is notable for its efficiency and usage with large datasets, although its operations require foreknowledge of the cluster. It is the process of grouping similar data items together based on some measure of similarity (or dissimilarity) between items. In this case, K values are randomly selected between 5 and 100. There are various variants of the K-Means clustering algorithm.

The focus will be to reduce the squared error function (Within Cluster Sum of Squared Error-WCSS), given by:

$$\sum_{j=1}^{k=1}\sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2$$

Where $\| x_i - c_j \|^2$ *is* the Euclidean distance between *x i* and *c j*. *x i*, equals the number of data

points in the *i* th cluster, c shows the number of cluster centres. Users closest to their cluster centres are the one's representative of that cluster (most relevant users).

Within Cluster Sum of Squares with respect to number of clusters is plotted to determine the number of clusters in the users. The plot is shown in Figure 4.3.

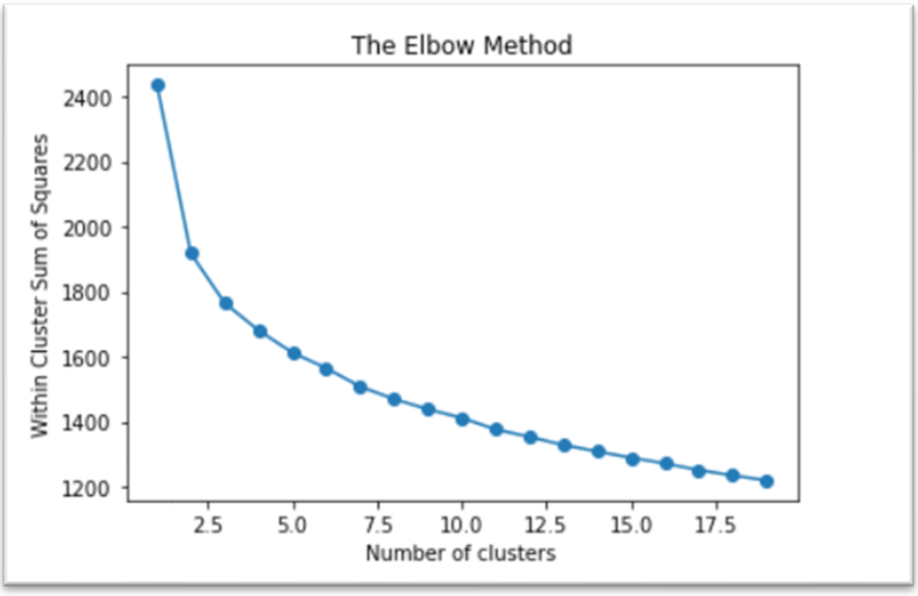


Figure 4.3

From the graph (Figure 4.3), number of clusters is taken as 4 as 3 to 4 has a big jump in WCSS value compared to 4 to 5. So, users are clustered using their normalised score with respect to the genres into 4 clusters as shown in Figure 4.4.

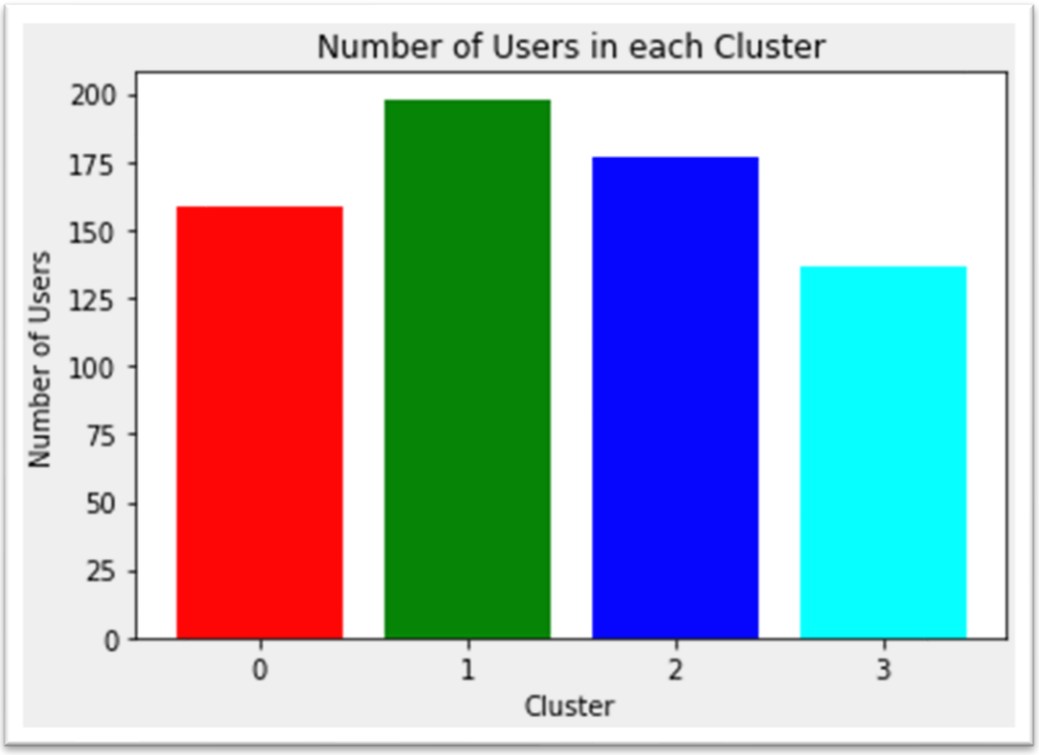


Figure 4.4

4.1.4. ALTERNATIVE LEAST SQUARE ALGORITHM

Recommender system (RS) is becoming growingly popular. In this work, Apache Spark (through pyspark library) is used to demonstrate an efficient parallel implementation of a collaborative filtering method using ALS.

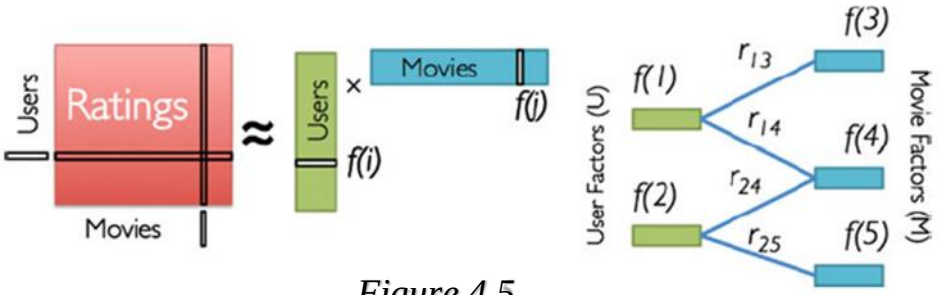


Figure 4.5

ALS is used for dimensionality reduction purpose which helps in overcoming the limitations of collaborative filtering such as data sparsity and scalability. The challenges of data sparsity are appearing in numerous situations, specifically, another problem, when a new an item or user has just added to the system, it is difficult to find similar ones since there is no sufficient information, this problem is called cold start problem. When selecting the ALS algorithm as a part of building the proposed movie recommender system, there are basic parameters through them can determine the best rating of users for given movies.

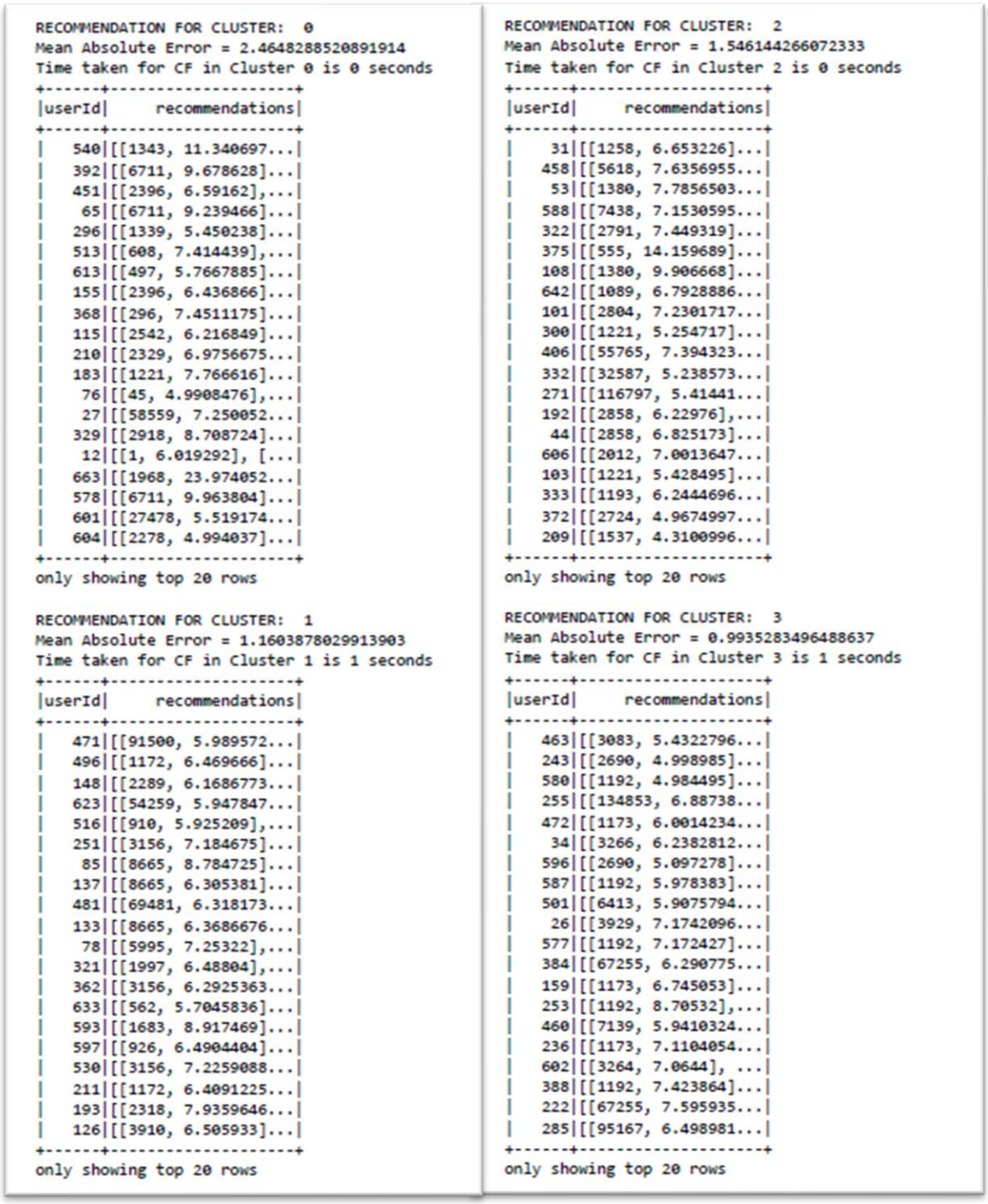
```
Mean Absolute Error = 0.8807316987986559
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|   471|[[106696, 5.45254...|
|   463|[[2043, 5.6891346...|
|   496|[[4965, 8.114831]...|
|   148|[[2043, 5.327806]...|
|   540|[[482, 9.491585],...|
|   392|[[4728, 13.234947...|
|   243|[[96588, 5.674291...|
|   623|[[5114, 6.9909487...|
|    31|[[5791, 5.6127615...|
|   516|[[1353, 6.015499]...|
|   580|[[2043, 5.321213]...|
|   251|[[3260, 7.314558]...|
|   451|[[4232, 7.6334977...|
|    85|[[4467, 8.7528925...|
|   137|[[6413, 7.905879]...|
|    65|[[1162, 10.191819...|
|   458|[[7327, 7.437454]...|
|   481|[[1232, 5.899743]...|
|    53|[[2132, 11.092971...|
|   255|[[3260, 6.179349]...|
+------+--------------------+
only showing top 20 rows

The time taken for CF is: 2 seconds
```

Figure 4.6

4.1.5. HYBRID MODEL

This model first clusters the users using K-means Clustering and then use ALS algorithm for each cluster to recommend movies. As number of clusters is taken as four, four sets of recommendations are made for each cluster.



```
RECOMMENDATION FOR CLUSTER:  0
Mean Absolute Error = 2.4648288520891914
Time taken for CF in Cluster 0 is 0 seconds
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|   540|[[1343, 11.340697...|
|   392|[[6711, 9.678628]...|
|   451|[[2396, 6.59162],...|
|    65|[[6711, 9.239466]...|
|   296|[[1339, 5.450238]...|
|   513|[[608, 7.414439],...|
|   613|[[497, 5.7667885]...|
|   155|[[2396, 6.436866]...|
|   368|[[296, 7.4511175]...|
|   115|[[2542, 6.216849]...|
|   210|[[2329, 6.9756675...|
|   183|[[1221, 7.766616]...|
|    76|[[45, 4.9908476],...|
|    27|[[58559, 7.250052...|
|   329|[[2918, 8.708724]...|
|    12|[[1, 6.019292], [...|
|   663|[[1968, 23.974052...|
|   578|[[6711, 9.963804]...|
|   601|[[27478, 5.519174...|
|   604|[[2278, 4.994037]...|
+------+--------------------+
only showing top 20 rows

RECOMMENDATION FOR CLUSTER:  1
Mean Absolute Error = 1.1603878029913903
Time taken for CF in Cluster 1 is 1 seconds
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|   471|[[91500, 5.989572...|
|   496|[[1172, 6.469666]...|
|   148|[[2289, 6.1686773...|
|   623|[[54259, 5.947847...|
|   516|[[910, 5.925209],...|
|   251|[[3156, 7.184675]...|
|    85|[[8665, 8.784725]...|
|   137|[[8665, 6.305381]...|
|   481|[[69481, 6.318173...|
|   133|[[8665, 6.3686676...|
|    78|[[5995, 7.25322],...|
|   321|[[1997, 6.48804],...|
|   362|[[3156, 6.2925363...|
|   633|[[562, 5.7045836]...|
|   593|[[1683, 8.917469]...|
|   597|[[926, 6.4904404]...|
|   530|[[3156, 7.2259088...|
|   211|[[1172, 6.4091225...|
|   193|[[2318, 7.9359646...|
|   126|[[3910, 6.505933]...|
+------+--------------------+
only showing top 20 rows

RECOMMENDATION FOR CLUSTER:  2
Mean Absolute Error = 1.546144266072333
Time taken for CF in Cluster 2 is 0 seconds
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|    31|[[1258, 6.653226]...|
|   458|[[5618, 7.6356955...|
|    53|[[1380, 7.7856503...|
|   588|[[7438, 7.1530595...|
|   322|[[2791, 7.449319]...|
|   375|[[555, 14.159689]...|
|   108|[[1380, 9.906668]...|
|   642|[[1089, 6.7928886...|
|   101|[[2804, 7.2301717...|
|   300|[[1221, 5.254717]...|
|   406|[[55765, 7.394323...|
|   332|[[32587, 5.238573...|
|   271|[[116797, 5.41441...|
|   192|[[2858, 6.22976],...|
|    44|[[2858, 6.825173]...|
|   606|[[2012, 7.0013647...|
|   103|[[1221, 5.428495]...|
|   333|[[1193, 6.2444696...|
|   372|[[2724, 4.9674997...|
|   209|[[1537, 4.3100996...|
+------+--------------------+
only showing top 20 rows

RECOMMENDATION FOR CLUSTER:  3
Mean Absolute Error = 0.9935283496488637
Time taken for CF in Cluster 3 is 1 seconds
+------+--------------------+
|userId|     recommendations|
+------+--------------------+
|   463|[[3083, 5.4322796...|
|   243|[[2690, 4.998985]...|
|   580|[[1192, 4.984495]...|
|   255|[[134853, 6.88738...|
|   472|[[1173, 6.0014234...|
|    34|[[3266, 6.2382812...|
|   596|[[2690, 5.097278]...|
|   587|[[1192, 5.978383]...|
|   501|[[6413, 5.9075794...|
|    26|[[3929, 7.1742096...|
|   577|[[1192, 7.172427]...|
|   384|[[67255, 6.290775...|
|   159|[[1173, 6.745053]...|
|   253|[[1192, 8.70532],...|
|   460|[[7139, 5.9410324...|
|   236|[[1173, 7.1104054...|
|   602|[[3264, 7.0644], ...|
|   388|[[1192, 7.423864]...|
|   222|[[67255, 7.595935...|
|   285|[[95167, 6.498981...|
+------+--------------------+
only showing top 20 rows
```

Figure 4.7

4.2. COMPUTATIONAL TIME

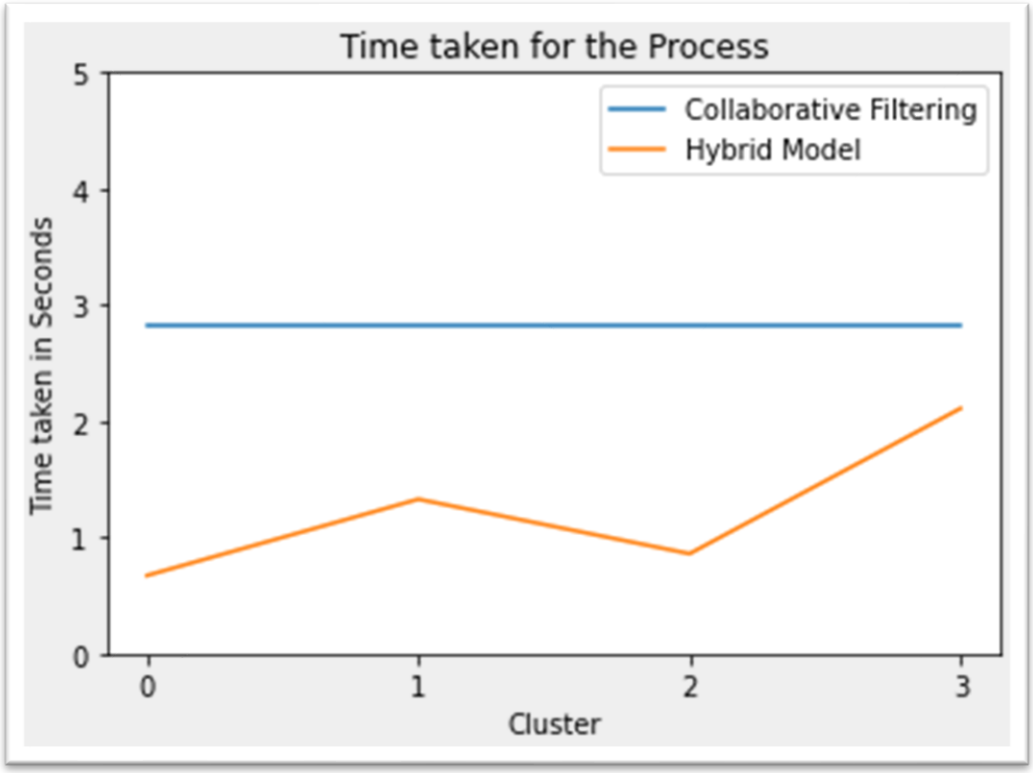


Figure 4.8

The computational time taken for hybrid model to execute and the normal collaborative filtering model using ALS algorithm is compared in Figure-4.8.

As the hybrid model clusters the users first, the movie recommendation can be done for the cluster in which the targeted user is present.

This makes the time taken for the process to fall sharply, which in real-time and large data save more time and resources.

It will have a huge impact as the size of data grows.

4.3.MODEL PERFORMANCE

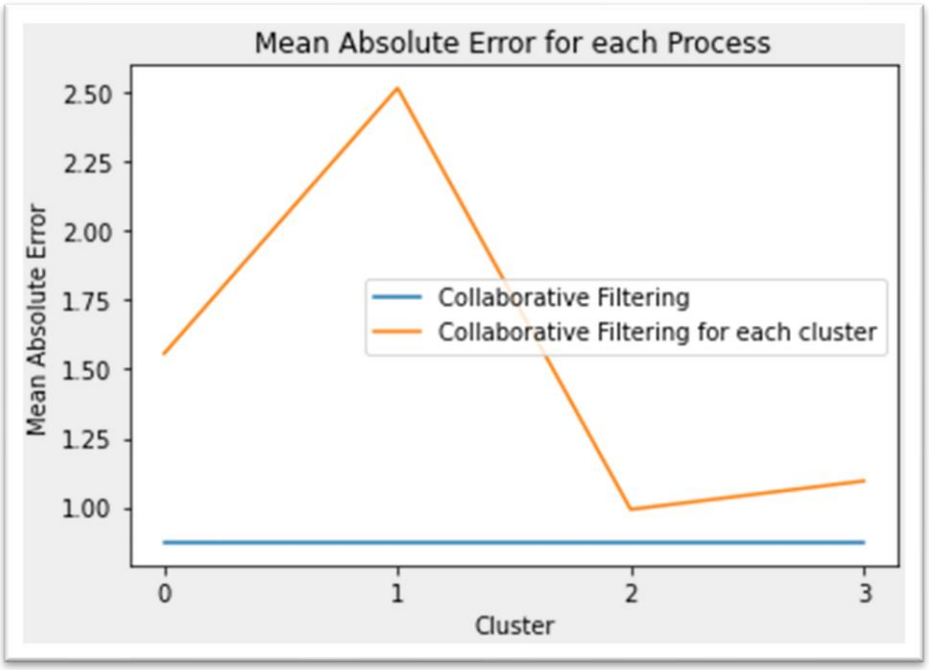


Figure 4.9

Mean Absolute Error is adopted in the ALS to measure the accuracy of the predictions. Small MAE (w) values are preferred.

$$w = \frac{\sum (m_1 - m_2)}{n}$$

From Figure 4.9, MAE value of the Hybrid Model has significantly increased in cluster 1 and slightly increased in cluster 2 and cluster 3. This may be due to the small dataset chosen for this experiment. With larger data, this model is expected to perform better than the conventional model.

4.4.LIMITATIONS

The dataset chosen for this project is small due to the limited resources which affected the performance of the Hybrid Model greatly.

Then for simplicity, only the primary genre of a movie is taken into consideration.

5. CONCLUSION AND RECOMMENDATIONS

Movie recommender system plays a significant role in identifying a set of movies for users based on user interest. Although many move recommendation systems are available for users, these systems have the limitation of not recommending the movie efficiently to the existing users.

This project a hybrid movie recommender system based on clustering the users using K-means Algorithm and collaborative filtering using ALS algorithm.

From the results, the computational time of the proposed model is almost half of the conventional model and also it is expected to improve drastically with larger dataset.

However, the accuracy of the proposed model drops. The size of the dataset can be a reason for this and the clustering method can be analysed and tuned further. The selection of parameters for the ALS algorithm also can be analysed because it affects the performance of building the movie recommendation system.

Other features of the movies such as popularity, crew, language, original language, tags, etc., can be considered for clustering.

6. REFRENCES

- "Research on Recommendation System based on Interest Clustering" Yunfei Yu and Yinghua Zhou. AIP Conf. Proc. 1820, 080021-1–080021-7; doi: 10.1063/1.4977377 - Published by AIP Publishing.

- "Improved Collaborative Filtering Recommendation Algorithm of Similarity Measure" - Baofu Zhanga) and Baoping Yuanb). AIP Conf. Proc. 1839, 020167-1–020167-6; doi: 10.1063/1.4982532 Published by AIP Publishing.

- "A Literature Review on Recommender Systems Algorithms, Techniques and Evaluations" - Kasra Madadipouya & Sivananthan Chelliah Volume 8, Issue 2, July 2017, ISSN 2067-3957 (online), ISSN 2068-0473

- Data Source - https://www.kaggle.com/rounakbanik/the-movies-dataset

- https://www.kaggle.com/agewerc/als-model-in-pyspark

- https://www.kaggle.com/alfarias/movie-recommendation-system-with-als-in-pyspark/

- "Adaptive Hybrid Collaborative Filtering Recommendation System (AHCF)" - Robert Agboyi, 2019.

- "A collaborative approach for research paper recommender system" - K. Haruna, M. A. Ismail, D. Damiasih, J. Sutopo, and T. Herawan,