Information Retrieval Project

CF based Recommendation System

Group 6

Submitted to: **Dr. Vinti Agarwal**

Group Members:

2017A7PS0084P : Aniruddha Jayant Karajgi
2017B4A70633P : Rohit K Bharadwaj
2017A7PS0215P : Jhaveri Ayush Rajesh
2017A7PS0036P : Rahul Jha
2017B3A70565P : Pranay Khariwal

November 2020

Abstract

Recommender Systems are used almost everywhere in today's world, from e-commerce websites, streaming services to various social media websites. These systems have become an integral part of our day to day life that we may not even realize that we are using one. In this project we are implementing collaborative filtering based recommender systems. There are two main approaches of collaborative filtering:

1. **User-based**
2. **Item-based**

We explore several user-based collaborative filtering techniques on **MovieLens** Dataset, and try to predict the top 5 similar movies for a given target user. The approaches used in this project have been novel and are able to give lower MAE values as compared to the standard implementations.

Table of Contents

Figures

Tables

1. Basic Implementation

User based collaborative filtering works based on the assumption that the users who have liked similar movies in the past will tend to like similar movies in the future. Hence, the first step in recommending movies to target users is to find similar neighbours of the given target user. The similarity metric used in our study is Pearson Correlation. After calculating the similarity score of target user with every other user, we consider top 10 most similar users to the target users in our further calculations. The top 10 most similar users are used in Resnick Prediction formula and the rating of the unseen movie for the target user is predicted. We evaluate the model performance by performing 5-fold validation split.

MAE Values					
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.70208	0.67017	0.66331	0.67546	0.68368	0.6823

Table 1: MAE Values.

1.1 Questions

1. **What is the significance of multiplying the value of rv,m by the similarity of user u to user v, in resnick prediction formula?**
   The similarity of user u to user v is used give the appropriate weight to the normalised rating of user v to predict the rating of user u. The higher the similarity between user u and user v, the greater the weight we give to the rating of user v. This includes negative weights which consider the negative impact due to ratings of dissimilar users.

2. **What could the additional possible information from data to be exploited to improve existing system developed in part A?**
   The dataset had user tags and genres for most of the movies, which was not utilized in part A approach. In one of our improvements, we have utilized the user tags data and incorporated them with movie ratings to compute a weighted score of ratings and tags. We then use this weighted score instead of just the user ratings.

2. Improvement No.1

2.1 Motivation

The existing model considered only ratings of the users to calculate user similarity, we feel that this might not be sufficient and the resulting model is not complex enough to capture the latent relations of user similarity. Thus, we incorporated data from user tags as well. In this case, it is expected that the model complexity increases and thus generalization of the model would increase.

2.2 Assumptions

The main assumption behind our approach is that tags can be used as a metric of similarity between users. For example, users that tagged a movie with "a good plot" enjoyed the movie, and may enjoy other movies with the same tag watched by those users. Likewise, a user may not enjoy a movie tagged as "bad plot" by a user with similar tags in those movies they have both watched.

Another assumption is that the rating a user gives to a movie can be complimented by the tags that user uses to describe that movie. A low rating of 1.5 would reflect tags like "bad plot" while a 5 could be given to an Oscar winning movie with the "Oscar" tag. This would allow us to create a cumulative "rating", giving some weight-age to the tags a user provides along with the existing ratings.

Not every user who rates a movie tags it, so in that case, we considered only the rating.

2.3 Implementation

Our solution incorporates score obtained from user tags. There were two kinds of tags which were found while we observed the dataset.

1. **Objective ones like Actor names/Genres/Description of the movie**
2. **Opinionated ones like "bad plot" or "bad jokes".**

We Implemented Sentiment Analysis on these two kinds of user tags. The score which is obtained from first kind of tags is 0, because they do not contribute to user similarity in any manner. We will only be getting the tag score from second type of tags, we then take weighted average of the user ratings and the tag score obtained after sentiment analysis to compute score for each user.

$$rating_{new} = rating_{old} * w_{rating} + sentiment * w_{sentiment}$$

Fig 1: Weighted average of rating and tag sentiment score

2.3.1 Sentiment Analysis

To reiterate, we are performing Sentiment Analysis on the tags so that they complement the rating a user has given to a movie. For the objective tags, a sentiment is meaningless, since they are a fact and cannot be disputed. Opinionated tags like "poor acting", on the other hand would have a sentiment attached, so that they can affect the rating given to that movie.

We used the VaderSentiment library to compute the sentiment scores of the tags. This score lies between -1 and 1, where a negative score indicates a negative sentiment. A score of 0 would be assigned to a neutral term, like a name.

Since a user may assign multiple tags to the same movie, we averaged the sentiment scores of these tags to obtain a single numeric score, which we would use to computed the final weighted average of a movie's rating and tag score for each of its viewers.

2.3.2 Weighted Average

The purpose behind giving weights to score obtained from pure ratings and the score which is obtained through tags is because movie ratings capture the similar tastes of the users better than user tags which can be very subjective, but movie ratings by themselves are not sufficient measure of user similarity, hence we have experimented with many different weighting scheme for ratings and the user tag, giving more weight to user ratings over tags in each case and we selected the weight pair which reduced the MAE the most.

2.4 Results

To find the best values for the weights used in the weighted average, we tried several different values, and their MAE values are presented in the graph below. The solid blue line represents the MAE for each fold individually, while the dotted orange line is the running average of every 5 datapoints. It represents the average MAE over all five folds for a trial.

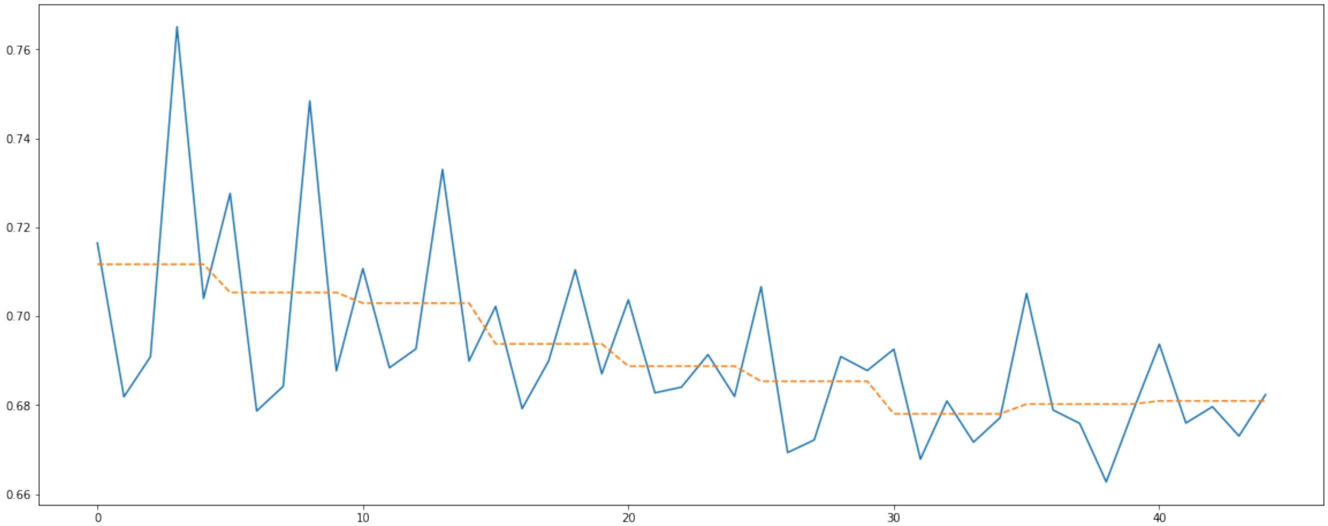

Fig 2: Weighted Average MAE values

The best results we got were for a weight of 0.7 for the rating and 0.3 for the mean tag sentiment score.

MAE Values					
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.69260	0.66789	0.68096	0.67170	0.677198	0.67807

Table 2: MAE Values.

Our final equation for the weighted average process is this:

$$rating_{new} = rating_{old} * 0.7 + sentiment * 0.3$$

Fig 3: Weighted average of rating and tag sentiment score

2.5 Questions

1. **What are the challenges in CF-RS built in part A?**
   One of the major issues of the model in part A is its simplicity. Only a portion of the entire dataset is being used and its unlikely that the model is sufficiently learning the features here.

2. **What improvements are you proposing?**
   Our suggested improvement of using a weighted average of tags and ratings instead of just using ratings should make the model more complex and allow it to better understand the data.

3. **How will the proposed improvement address the issues?**
   Our proposed improvement considers scores obtained from user tags along with user ratings score which was already being considered previously, since our improvement considers more features, the issue of the previous model being too simple is addressed.

4. **A corner case (if any) where this improvement might not work or can have an adverse effect.**
   Sentiment analysis may not be accurate enough which might lead to errors on few specific examples. As an example, a name may have a non zero rating, which is inaccurate since a name is objective and should have no sentiment.

5. **Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation**
   Please check Results for the specific MAE values.

2.6 Conclusion

This improvement is clearly better than the previous implementation as evident from the lower MAE values. The results can be explained Intuitively because, the previous implementation considered only user ratings, that is, it gave the weight for user ratings as 1 and weight for tags as 0, in our implementation we have given a non zero weight to user tags as well which can capture user similarity in a better way thus leading to lower MAE values and more accurate movie recommendation for target user.

Further improvements in this area could be obtained if one can use some unsupervised approach like clustering to deal with the tags (users in nearby or the same clusters would be treated as neighbours or their similarities would be given a boost).

3. Improvement No.2

3.1 Motivation

The existing model did not consider the importance of the number of common movies watched among users. It was common for the active user to have highly correlated neighbours that were based on a very small number of co-rated items. These neighbours that were based on tiny samples (often 3 - 5 co-rated items) frequently proved themselves to be terrible predictors for the active user and hence, should be penalized. The more movies that we have to compare the opinions of two users, the more we can trust the computed correlation as it would be representative of the true correlation between the users. So, a modification in the Pearson correlation co-efficient calculation is required to remedy this scenario.

3.2 Assumptions

It has been assumed that if the number of commonly rated movies among two users is more than H, the similarity calculated using the general Pearson correlation coefficient calculation is the correct indicator of the similarity among the two users. However, if the number of commonly rated movies is less than H, the similarity calculated using the general Pearson correlation coefficient calculation needs to be linearly reduced by a factor.

3.3 Implementation

We base our solution on an edited version of the Person correlation coefficient formula to calculate the similarity among two users:

$$sim\,(u,v)^{WPCC} = \begin{cases} sim\,(u,v)^{PCC} \cdot \frac{|I|}{H}, & |I| \leqslant H \\ sim\,(u,v)^{PCC}, & \text{otherwise} \end{cases}$$

Fig 4: Weighted Pearson

This weighted Pearson correlation coefficients linearly reduces the similarity among users when commonly rated movies are less than H. We experimented with different values of H to determine its optimal value.

3.4 Results

- MAE values

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
No weighting	0.70207	0.67016	0.6633	0.67545	0.68367	0.67893
H = 5	0.70229	0.66893	0.66309	0.67519	0.6833	0.67856
H = 10	0.70165	0.67041	0.66288	0.67413	0.68303	0.67842
H = 15	0.70046	0.67078	0.66288	0.67386	0.68239	0.678074
H = 20	0.70164	0.67385	0.66533	0.67369	0.68128	0.679158

Fig 5: MAE Values

- Graph

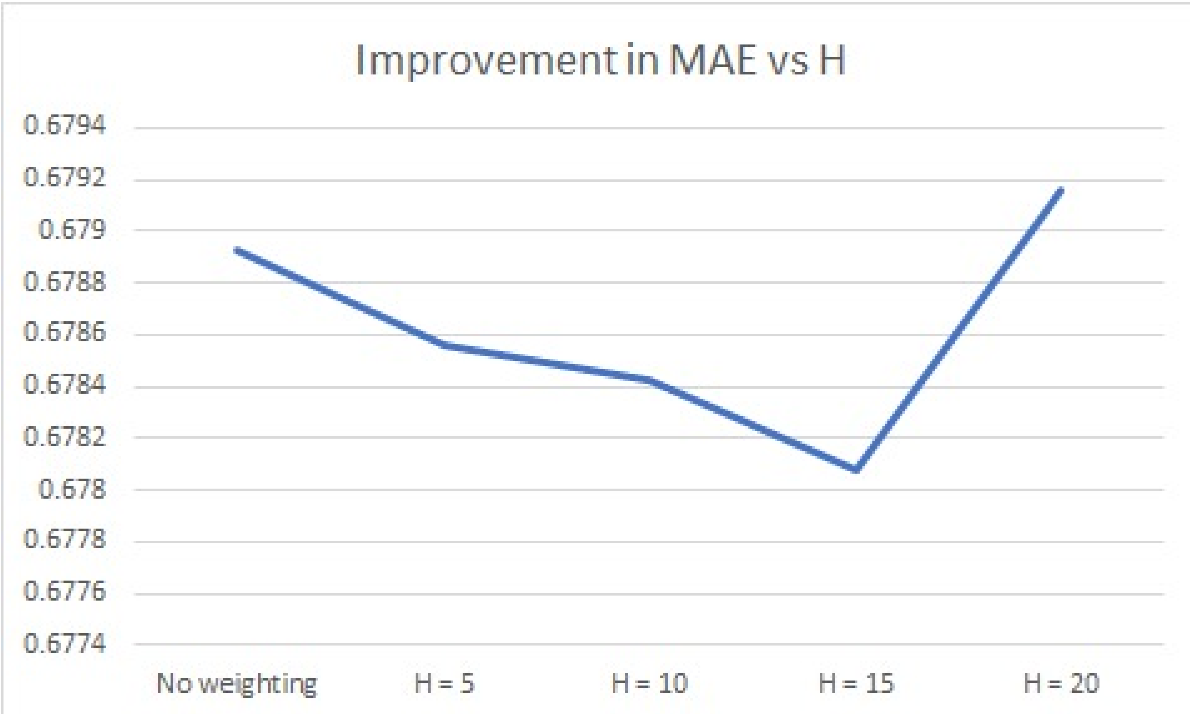

Fig 6: Improvement in MAE vs H

As indicated in Fig. 6, the ideal value of H is 15.

3.5 Questions

1. **What are the challenges in CF-RS built in part A?**
   It was common for the active users to have highly correlated neighbours that were based on a very small number of co-rated items. These similarities are based on limited data which could be misleading and hence, should be penalized.

2. **What improvements are you proposing?**
   We linearly reduce the calculated similarity if the number of commonly rated movies are less than a threshold value H.

3. **How will the proposed improvement address the issues?**
   Our proposed improvement penalizes the similarities among users with less commonly rated movies and hence, does not allow these users to be highly similar.

4. **A corner case (if any) where this improvement might not work or can have an adverse effect.**
   Users with the same taste, but not enough co-rated movies, may not be given high similarities.

5. **Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation.**
   We have addressed this question above in section 3.4.

3.6 Conclusion

This improvement is clearly better than the previous implementation as evident from the lower MAE values as weighting increases to 15 as indicated by the graph in the results. This shows that several users initially claimed as highly correlated were not due to a low number of co-rated movies.

4. Improvement No.3

4.1 Motivation and Implementation

The existing model considers the top 10 correlated neighbours per user. Increasing the number of neighbours considered may reduce the MAE value as more data considered could improve the reliability and accuracy of rating predictions made.

4.2 Results

- MAE Values

	MAE Values					
Neigbours	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
10	0.70208	0.67017	0.66331	0.67546	0.68368	0.6823
20	0.69767	0.66071	0.66017	0.67183	0.67183	0.67405
30	0.69648	0.65985	0.65765	0.67219	0.67219	0.67305
40	0.69443	0.66059	0.65904	0.67254	0.67981	0.67328
50	0.69551	0.66243	0.65776	0.67493	0.67936	0.674

Fig 7: MAE Values

- Graphs

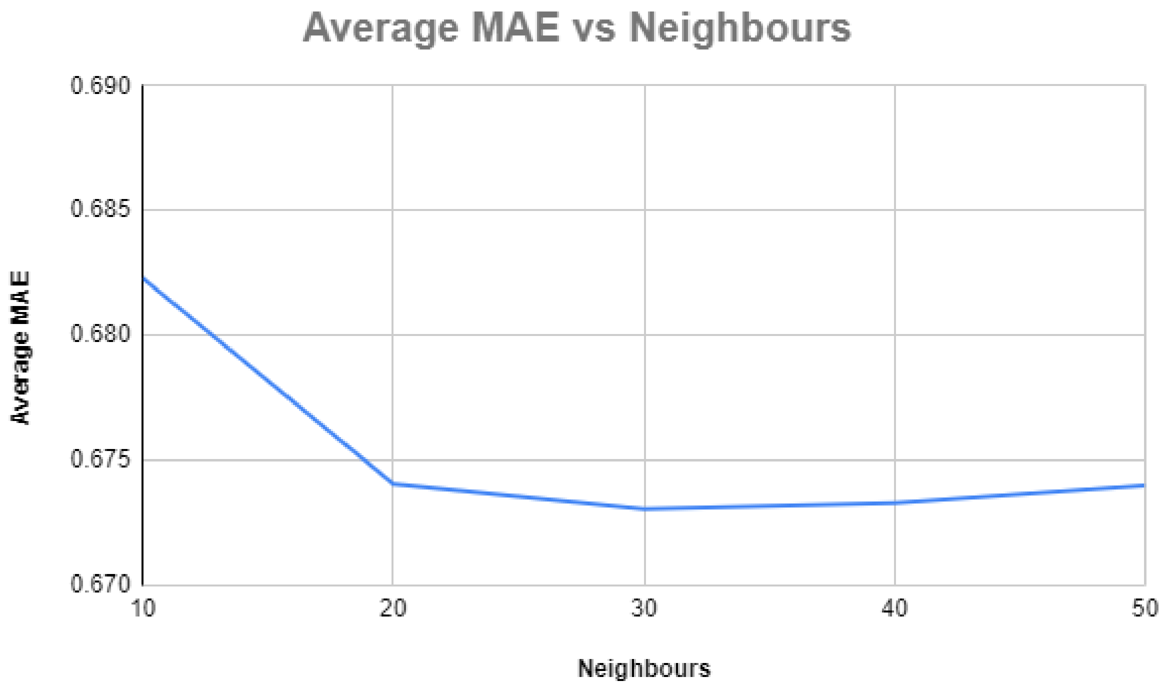

Fig 8: Average MAE vs Neighbours

As indicated in Fig. 8, the ideal number of neighbours considered for prediction is 30 as we obtain the lowest MAE.

4.3 Questions

1. **What are the challenges in CF-RS built in part A?**
   We base out our solution on the top 10 correlated neighbours per user. This value may be too less to capture enough similarities for predictions.
2. **What improvements are you proposing?**
   Increase the number of neighbours as long as the MAE keeps reducing.
3. **How will the proposed improvement address the issues?**
   It will consider more correlated neighbours for predictions and would ensure a lowered MAE.
4. **A corner case (if any) where this improvement might not work or can have an adverse effect.**

If other than the top 10 correlated users to a particular user, the rest of the neighbours may be highly dissimilar and lead to incorrect predictions.

5. **Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation.** We have addressed this question above in section 4.2.

4.4 Conclusion

Considering the top 30 correlated neighbours evidently gives lower MAE values as we rely our predictions on a greater amount of data of similar users.

5. Improvement No.4

5.1 Motivation

When calculating the similarity of users based on ratings of movies, the frequency of the ratings of the movie should also be taken into account. The idea is to reduce the weights of commonly rated movies, capturing the intuition that they are not as useful in predicting a user's tastes as a unique movie with few ratings. This is because similar ratings on a unique movie would be more indicative of alignment of interests than similar ratings on a popular movie.

5.2 Implementation

We use the same metric that was used in tf-idf transforms for inverse document frequency by using the inverse user frequency as a weight term to be multiplied with the movie ratings. We define fj as log (n/nj) where nj is the number of users who have rated movie j and n is the total number of users available in the database. Thus, if everyone has rated a movie, then its weight drops to zero.

$$w(a,i) = \frac{\sum_j f_j \sum_j f_j v_{a,j} v_{i,j} - (\sum_j f_j v_{a,j})(\sum_j f_j v_{i,j})}{\sqrt{UV}}$$

where

$$U = \sum_j f_j (\sum_j f_j v_{a,j}^2 - (\sum_j f_j v_{a,j})^2)$$

$$V = \sum_j f_j (\sum_j f_j v_{i,j}^2 - (\sum_j f_j v_{i,j})^2)$$

Fig 9: IUF Formula

Before using Pearson Correlation Coefficient on two vectors, we transform the input matrix by multiplying the ratings with their respective weights.

5.3 Results

- MAE Values

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.69069	0.68224	0.67108	0.67868	0.68458	0.68145

Fig 10: MAE for Inverse User Frequency

As indicated by the results, cosine similarity performs better than Pearson correlation to determine the similarity among users as it gives a lower MAE value.

5.4 Questions

1. **What are the challenges in CF-RS built in part A?**
   Similarity of users based on ratings of popular movies have the same weight as those based on ratings of unique movies. This may not capture the correlation between two users who have rated a controversial movie properly as a popular movie would reduce it's weightage in correlation calculation.

2. **What improvements are you proposing?**
   We add a factor of inverse user frequency for Correlation calculation which takes into account the number of users who have rated a particular movie.

3. **How will the proposed improvement address the issues?**
   The inverse user frequency penalizes similarity of movies that have many ratings, so that similarity on less rated or controversial movies can have a greater effect on calculating the correlation.

4. **A corner case (if any) where this improvement might not work or can have an adverse effect.**
   In case of a new user (cold start) or a user who has watched only popular movies, we would not be able to recommend any particular movie.

5. **Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation.**
   We have addressed this question above in section 5.3.

5.5 Conclusion

Inverse User Frequency gives a slightly lower MAE than simple Pearson Correlation Coefficient. However, there is probably not much improvement due to the fact that the number of unique co-rated movies where users have a high similarity are few.

6. Improvement No.5

6.1 Motivation

The existing model uses Pearson Correlation to calculate the similarity among users. This metric drops the movies that are not rated by both users. However, this has been proven to perform poorly when the number of co-rated items among users is very low. Hence, experimentation with a metric could be done that considers movies that are only rated by one of the users.

6.2 Implementation

We replace the Pearson Correlation metric with Cosine Similarity to to calculate the similarity among two users.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Fig 11: Cosine Similarity

The cosine approach provides a kind-of Bayesian regularization for the metric, ensuring that the similarity is not fully decided only by the (potentially small) subset of movies the two users have in common. This is not natively present in Pearson correlation , so in contexts where users tend to have very differing sets of items in their profiles, Pearson would perform worse, in principle.

6.3 Results

- MAE Values

As indicated by the results, cosine similarity performs better than Pearson correlation to determine the similarity among users as it gives a lower MAE value.

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.70197	0.67054	0.66629	0.67139	0.68193	0.67842

Fig 12: MAE values

6.4 Questions

1. **What are the challenges in CF-RS built in part A?**
   Pearson Correlation only considers commonly rated movies, and would perform poorly if the number of co-rated movies among users are very low.

2. **What improvements are you proposing?**
   We replace the Pearson correlation metric with cosine similarity among users.

3. **How will the proposed improvement address the issues?**
   The cosine approach provides a kind-of Bayesian regularization for the metric, ensuring that the similarity is not fully decided only by the (potentially small) subset of movies the two users have in common.

4. **A corner case (if any) where this improvement might not work or can have an adverse effect.**
   Pearson correlation would perform better in cases where the number of co-rated movies among the users are high.

5. **Demonstrate the actual impact of the improvement. Give three examples, where the improvement yields better results compared to the part A implementation.**
   We have addressed this question above in section 6.3.

6.5 Conclusion

Cosine Similarity gives a lower MAE value than Pearson Correlation, This shows that giving some importance to non co-rated movies lead to better results as co-rated movies among users are very few.

7. Conclusion

We have built a user-based collaborative filtering recommender system using Pearson Correlation to meaure similarity among users. Five improvements have been proposed, explained and interpreted which give a better performance at predicting user ratings.

8. References

1. HaifengLiu et.al. "A new user similarity model to improve the accuracy of collaborative filtering". Science Direct (Volume 56, January 2014, Pages 156-166.) For link click here

2. John S. Breese et.al. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". Microsoft Research Redmond, WA 98052-6399. For link click here

3. Johnathan L. herlocker et.al. "An Algorithmic Framework For Performing Collaborative Filtering". For link click here

4. Sarabjot S. Anand. "Intelligent Techniques for Web Personalization". Research Gate (January 1970 DOI: 10.1007/11577935) . For link click here

5. John Riedl et.al. "Explaining Collaborative Filtering Recommendations" . Research Gate (January 2001 DOI: 10.1145/358916.358995). For link click here

6. Andrea Isoni. Book : "Machine Learning For The Web ".