

Recommendation System for Movies

Parminder Kaur (V00820508), Navpreet Kaur (V00823334)

Department of Computer Science, University of Victoria

Abstract

Over the past few decades, internet has turned into vast global market place for exchange of goods and services. There are lot of things that can be done online like getting the latest news, to do socialising etc. These days though, what is truly thriving is online shopping. One can easily get the latest trend and buy it online. Convenience is one of the main reasons why more and more people are turning to online shopping. The explosion of growth in the ecommerce world has only made it tougher for ecommerce sites to attract the critical mass of visitors needed to stay in business. Recommender systems have become extremely popular in recent years, and are common in a variety of applications like movies, music, news, books, research articles, search queries. The recommender system that we made for this project is movie recommender system based on collaborative filtering technique. The data is amazon movie data. The system predict movie rating for users. This Project is consist of two different recommender systems one which is implementing basic formulas of recommendation system and do movie prediction while the other recommender system is working with one additional formula "Wilson Score". The Second recommendation system is implemented in order to find more stable and reliable results for recommendation. A detail study is done to observe the performance of both systems.

Contents

1. Introduction	3
2. Related Work	4
3. Data Prepossessing	4
3.1 Data Collection.....	5
3.2 Data Analysis.....	5
3.3 Data Preparation.....	6
3.4 Data.....	7
4. Recommendation System	8
4.1 Recommendation System1.....	8
4.2 Recommendation system2.....	9
5. Performance Experiment	10
5.1 Measuring Prediction Accuracy	11
5.1.1 RMSE.....	11
5.1.2 MAE.....	12
5.2 Measuring Usage Prediction	12
5.3 Observations	15
6. Conclusion	16

1. Introduction

There are hundreds of E-Commerce websites available over internet with thousands of choices over one product. There are various marketing strategies to keep going E-Commerce business and some popular websites like eBay, Netflix, Amazon are successfully implementing such strategies. It has been noticed that number of transactions carried out on eBay on Black Friday and Cyber Mondays crossed over a million. Online shopping environment does not include a real life interaction with customers and guide them with best products according to their choice and budgets. So it could be a major challenge for online retailers to gain attentions of customers in order to serve them with their choices and also maintain the profit for organisation.

Recommender Systems are boon for online retailers to keep gain interest of its customers. Recommender Systems uses data mining techniques to provide suggestion to users while shopping on web. The recommendations are based on many constraints. Recommendations are offered based on user's priority, purchase history, user's reviews on products, clicks, logs, ranked list of items etc. These techniques help sites to adapt itself according to user's choices. There are three approaches [8] for recommender systems Collaborative filtering, Content based filtering, Hybrid Recommender System.

Collaborative filtering

Collaborative filtering method includes collecting and analyzing large amount of information on user's behaviour , activities or preferences and based on this data system will predict the future choices that a user may like that similar users have liked. Two popular approaches KNN(K-nearest neighbour) and Pearson Correlation implements Collaborative filtering which works on assumption that "people who agree in the past will agree in future" that means if they like a product in past then they will like same product in future as well however this approach suffers through three major consequences Cold start, Scalability and Sparsity. This approach requires large amount of data on particular user to make good recommendations for a user. There are millions of users and products and large amount of computation is necessary to recommend product. Even the most active users will only rate a small set of overall database thus popular products have few ratings.

Content based filtering

As the name says this approach deals with type of content (product) that user have liked in past or examining in present. These methods create an item profile including features and attribute that characterize item and also a user profile (content based profile) that include importance of each feature to user that can be computed from variety of techniques (Bayesian Classifier, cluster analysis etc). However the major issue with this approach is, whether the machine will be able to learn user preferences from user's actions from one type of product and implement that information to recommend another type of product.

Hybrid based filtering

Hybrid based filtering is combination of both above mentioned approaches, collaborative based filtering and content based filtering. It is considered as best approach. Netflix is good example of Hybrid filtering it suggests movies on the basis of searching and user's previous choices that he has rated high. There are also few hybrid techniques example Weighted, Switching, Feature Combination, Cascade etc.

2. Related Work

Amazon.com [2] has over 29 million customers and several million catalogue items. For Amazon.com, does little or no offline computations scale with the number of customer or catalogue items. Algorithms are impractical on large datasets unless compromised with recommendation quality by using techniques of sampling or partitioning. Cluster models makes online user-segment classification expensive if we increase the segments without which recommendation quality remains poor. Search base models build keyword, category, and author indexes offline but failed to provide recommendations for numerous purchasing and ratings.

Focussing on User-User similarity is impractical for Amazon.com with huge data. For Amazon, Item-to-item collaborative filtering matches each user's purchased and rated items to similar items to recommendation list. The algorithm builds a similar-items (Cosine method) table by finding items that customers tend to purchase together. Given a similar item tables, algorithms work very fast by aggregating the items that user have purchased or rated and make a

computations very quick. Worst case offline computation $O(N^2 M)$ and practical $O(NM)$ for customer with few purchases.

We have explored a new factor to improve recommendations for movies while we are dealing with movies recommendation on Amazon. Usually users have ratings and reviews on movies they purchased. But, Amazon data also consider the helpfulness of reviews to other users that customer has given to movie. By considering the already existed data and knowing how helpful it is we can increase the improvement in recommendation. If a customer has to suggest a movie, we will look for other similar customers to him who have seen movie already and also look for customer's helpful reviews so that a more stable and reliable recommendation is made.

3. Data Preprocessing and Cleaning

3.1 Data Collection

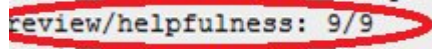
Data for Recommendation System was downloaded from Stanford Repository [6]. Data statics Aug 1997-Oct 2012, as given on their websites, number of reviews 7,911,684, number of users 889,176 and number of products(movies) rated are 253,059 . Data is initially in form of text file and size of file is 8GigaByte.

3.2 Data Analysis

Data was downloaded in form of text file and with various attribute in it like Product id, User id, Profile name, Score, Helpfulness, Summary, Review (text).Format of input was available in following manner as shown in figure.

```
product/productId: B00006HAXW
review/userId: A1RSDE90N6RSZF
review/profileName: Joseph M. Kotow
review/helpfulness: 9/9
review/score: 5.0
review/time: 1042502400
review/summary: Pittsburgh - Home of the OLDIES
review/text: I have all of the doo wop DVD's and this one is as good or better than the
1st ones. Remember once these performers are gone, we'll never get to see them again.
Rhino did an excellent job and if you like or love doo wop and Rock n Roll you'll LOVE
this DVD !!
```

As in figure we can see helpfulness has to be divided into two columns which is used later in project for recommendation purpose (explained later).



review/helpfulness: 9/9

3.3 Data Prepration

There is huge unnecessary data in text format which is need to be removed and the format is not in form the text as we are working on java code we require text input for recommendation is required. We use Microsoft SQL 2008 to pre-process data. Data was quite big so we first divided the whole data into small file using Emeditor/Notepad++.



First Unnecessary attribute names product id, user id etc. was removed in text file and then text file was imported in SQL by using Bulk import query which is used to import bulk data of text or CSV file into SQL. Once the data was imported in SQL and table as given below is formed we dropped the unnecessary columns (profile name, times, summary, text (review)) from table by using queries. We want user id to be in column one and product id to be in column two so query is made to switch columns as well.

productid	userid	profile	helpfulness	maxhelp	score	times	summary	texts
B003AI2VGA	A328S9RN3U5M68	Grady Harp	4	4	3.0	1181952000	Worthwhile and ...	THE VIRGIN OF ...
B003AI2VGA	A1I7QGUDP043DG	Chrissy K. McVa...	8	10	5.0	1164844800	This movie need...	The scenes in thi..
B003AI2VGA	A1M5405JH9THP9	golgotha.gov	1	1	3.0	1197158400	distantly based ...	THE VIRGIN OF ...
B003AI2VGA	ATXL536YX71TR	KerrLines ""...	1	1	3.0	1188345600	"What's going o...	Informationally, ...
B003AI2VGA	A3QYDL5CDNYN66	abra "a devoted...	0	0	2.0	1229040000	Pretty pointless ...	The murders in J..
B003AI2VGA	AQJVNDW6YZFQS	Charles R. Williams	8	11	1.0	1164153600	This is junk, stay...	Mexican men ar...
B00006HAXW	AD4CDZK7D31XP	Anthony Accordino	64	65	5.0	1060473600	A Rock N Roll Hi...	Over the past fe..
B00006HAXW	A3Q4S5DFVPB70D	Joseph P. Aiello	26	26	5.0	1041292800	A MUST-HAVE ...	I recvd this vide...
B00006HAXW	A2P7UB02HAVEPB	"bruce_from_la"	24	24	5.0	1061164800	If You Like Doo...	Wow! When I sa..

3.4 Data

Finally when data is in form of table we run the query to fetch random data from table to form input file for recommendation system which has statics 1 million tuples, 30k users and 29k movies. Table data fetched at end is saved as text file. Now data is of size 1GigaByte to use for input to recommendation system. The input file (.txt) has following format

File	Edit	Format	View	Help			
A5RA4HIR3S8EW		B008PZZND6			5.0	10	13
A3K7Z2KT2HOU8L		B008PZZND6			5.0	10	13
A38XU9MG6PIWHV		B008PZZND6			5.0	10	13
A29WTVAJAR03Q4		B008PZZND6			5.0	3	3
AQHJLAY31F5JL		B008PZZND6			5.0	19	26
AP0DQSDFSLP4P		B008PZZND6			4.0	19	26
A20MIBSYZBXI7S		B008PZZND6			5.0	19	26
ATDE9JYCP10L1		B008PZZND6			5.0	12	16
A3GAF5AQ6OZP2V		B008PZZND6			5.0	12	16
A38U2M9OAEJAXJ		B008PZZND6			4.0	28	39
A20TJ5P97W5MS1		B008PZZND6			5.0	5	6
A11VWUNAPEZZU7		B008PZZND6			5.0	5	6
A2GM9GNP2G8HO1		B008PZZND6			5.0	5	6

First column refers to user id, second column refers to product id (movies), third column is score (rating) given by user and 4 fourth and fifth column is fraction (10/13) which is helpfulness factor divided into two parts.

Helpfulness defines the usefulness of review for other users. If column four has value 10 and column five has value 13 that means 10 out of 13 people find the comment of user useful. Finally we have input file to use for recommendation System. However testing phase also required some different data preparation which is explained in section of evaluation of Recommendation System.

4. Recommendation System

The Recommender System is implemented using JAVA Programming. We have implemented recommender system in two different ways. The System1 in which basic recommender system is implemented. The System2 is implementing a new factor “Helpfulness”, the idea behind system2 is to make a more stable and reliable prediction.

4.1 Recommender System1

Implementing User-User recommendation system requires to calculate user-user similarity. The three approaches which are used for calculating user-user similarity are cosine distance, Euclidean Distance and Pearson distance. Pearson Distance is considered as so far the best user-user similarity. For system1, to calculate user-user similarity Pearson distance is used. Following is the formula to calculate User-User similarity using Pearson,

$$sim_{x,y} = \frac{\sum_{i=1}^m (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^m (y_i - \bar{y})^2}}$$

For calculating rating of movie

$$\hat{r}_{u,i} = \frac{\sum_{v \in U_i} r_{v,i} \cdot sim_{v,u}}{\sum_{v \in U_i} sim_{v,u}}$$

In formula above, u is a new user for which a new movie i is to be predicted and v is a user who is similar to user u and rated most of similar movies that user u has rated and if user u has rated a movie high then user v also rated same movie high and vice versa. “ $sim_{v,u}$ ” is similarity of user v to user u in formula.

As the matrix is sparse and in big data sets some users do not have similar users so it is hard to find rating for movie. If For some user, no similar users are found then to find rating for movie mean and biases is implemented.

$$\hat{r}_{u,i} = \mu + b_u + b_i$$

4.1 Recommender System2

In system2 the above system1 is implemented same. The additional implementation is helpfulness factor. When we observe the amazon movie data, the helpfulness factors seems to be providing more promising movie rating. On amazon when a user give rating and reviews to a movie, other users also rate that reviews of user for the movie. In this manner if we combine both the ratings of movie (user rating + rating provided to review of movie given by that particular user), we will have strong stability in rating. The similarity will be find in same manner but while recommended movies the additional “helpfulness” factor will be added in formula.

The challenge with this factor was how to find balance fraction. As If given helpfulness is 9/9 that means 9 people rated the rating of user and they all find it useful. 88/90 means 88 out of 90 find the rating of person helpful. Now we can see that fraction for products are not same if a rating is 9/9 helpful for a user it should not be given importance rather than helpfulness of 88/90 which is clearly unfairness to second rating which is highly helpful for more people.

We have real life example from amazon.com to demonstrate this problem

13.



SALTON HOUSEWARES, INC.
TR2500C ULTIMATE PLUS
BREAKMAKER

Buy new: ~~\$135.99~~

In Stock

★★★★★ (1)

14.



KitchenAid KP26M1XLC
Professional 600 Series 6-Quart
Stand Mixer, Licorice

Buy new: ~~\$499.99~~ **\$329.99**

10 Used & new from \$325.00

Get it by **Monday, Feb 9** if you order in the next **19 hours** and choose one-day shipping.

Eligible for **FREE** Super Saver Shipping.

★★★★★ (580)

As we can see in picture, a 5 star rated product by only 1 person is given priority than a product with 4.5 rating by 580 users.

Considering this challenge we have come up with idea of implementing Wilson score [1] on suggestion from Dr. Alex Thomo.

$$\left(\hat{p} + \frac{z_{\alpha/2}^2}{2n} \pm z_{\alpha/2} \sqrt{[\hat{p}(1 - \hat{p}) + z_{\alpha/2}^2/4n]/n} \right) / (1 + z_{\alpha/2}^2/n).$$

Wilson score worked out in 1927 by Edwin B. Wilson.

Here \hat{p} is the *observed* fraction of positive ratings, $z_{\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution, and n is the total number of ratings.

So with the Wilson score formula we find a balanced fraction. Now to find the rating for movie following formula is used,

$$\hat{r}_{u,i} = \frac{\sum_{v \in U_i} r_{v,i} \cdot sim_{v,u} \cdot h(r_{vi})}{\sum_{v \in U_i} sim_{v,u} \cdot h(r_{vi})}$$

Where $h(r_{vi})$ represents the helpfulness.

5. Performance Experiments

Before starting evaluation we have prepared some testing files. We worked on random 10 Million tuples of Amazon data. We are performing testing on 10% of data that we are working on. Extraction of random 10000 tuples from data was made via SQL. For recommendation system testing, basic rule of testing is “Hide and try to predict it”. So we have also follow same rule we hide the 10k tuples from 1 million tuples and predict movies for those 10k tuples.

We have applied each testing techniques [3] separately on both systems recommendation system1 and system2. So following are results of tests performed on both the Systems.

5.1 Measuring Rating Prediction accuracy

The recommended system have predicted the movie rating for 10k users on the scale of 1 to 5 star. So to check how accurately the ratings have been predicted by recommended system, RMSE (Root Mean Square Error) and MAE (Mean absolute error) are used.

5.1.1 RMSE (Root Mean Square Error)

RMSE is perhaps the most popular method in evaluating the accuracy of predictions generated by recommendation system.

$$RMSE = \sqrt{\frac{(r_{u,i} - \hat{r}_{u,i})^2}{N}}$$

Above is the formula to calculate RMSE where $r_{u,i}$ is original rating of the movie given by user and another $\hat{r}_{u,i}$ is prediction of the recommendation system and N is number of ratings. As discussed we have 10000 tuples for testing. So we had results from recommendation system1.

RMSE = 1.075648

Then we have recommendation system2 (Pearson similarity+ Wilson score) results, which resulting

RMSE = 1.085512

We can see error for both recommendation Systems almost perform similar. However the recommendation System2 (Pearson Similarity+ Wilson score) has .01 error more than recommendation system2.

We could say overall performance of recommendation system1 performed good according to RMSE testing.

Another testing technique for Recommendation System is

5.1.2 MAE (Mean absolute error)

$$MAE = \sqrt{\frac{1}{|\mathbf{N}|} \sum_{(u,i) \in \mathcal{R}} |\hat{r}_{ui} - r_{ui}|}$$

Above is the formula to calculate MAE where $r_{u,i}$ is original rating of the movie given by user and another $\hat{r}_{u,i}$ is prediction of the recommendation system and N is number of ratings.

we had results from recommendation system1(Pearson similarity).

MAE= 0.901222

Then we have recommendation system2 (Pearson similarity+ Wilson score) results

MAE= 0.899515

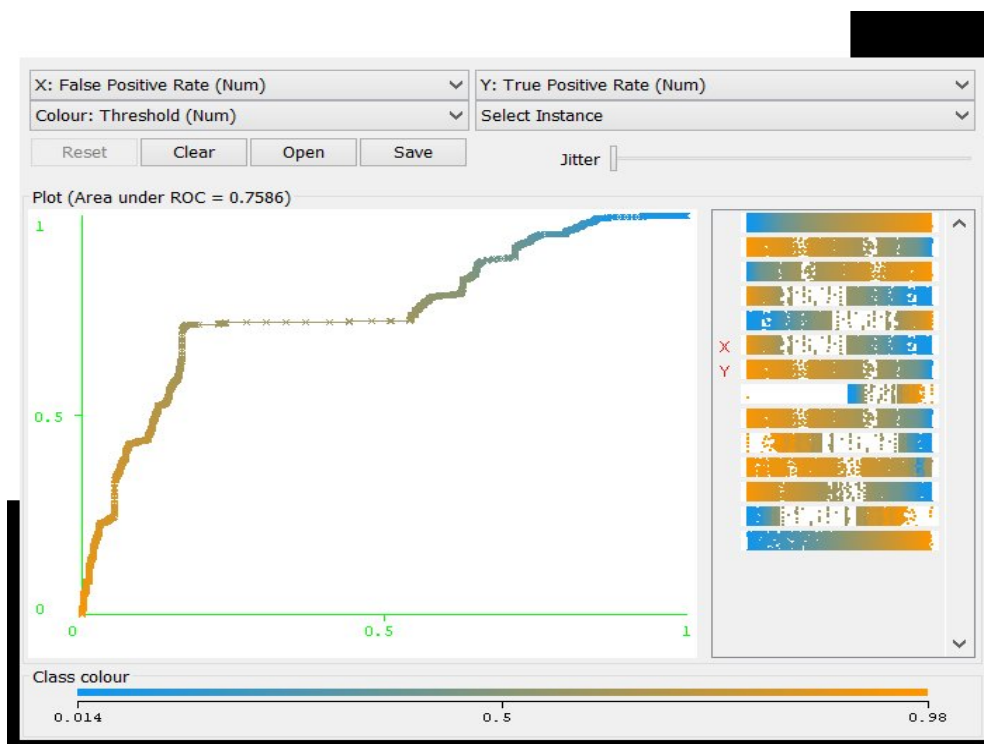
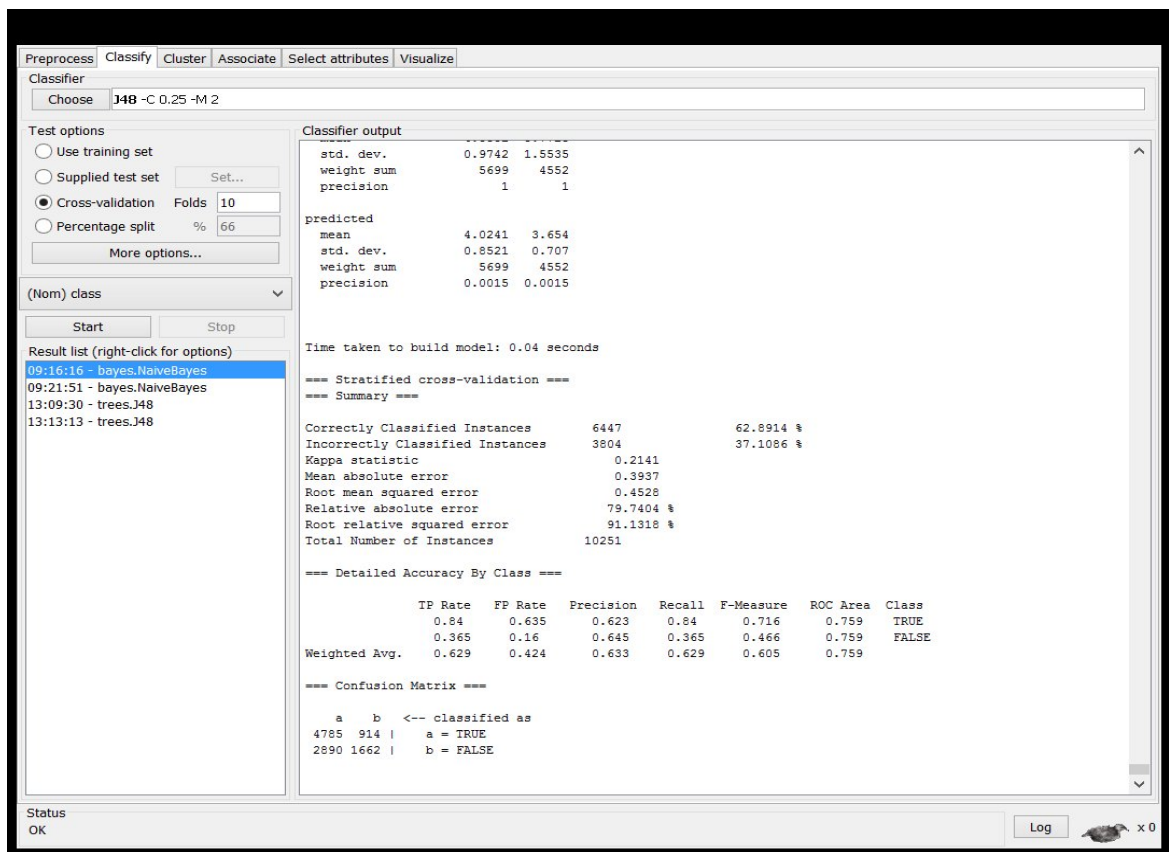
When MAE applied the system2 performed well than system1 as lower the MAE the system performed well, which is lower for system2.

5.2 Measuring usage prediction

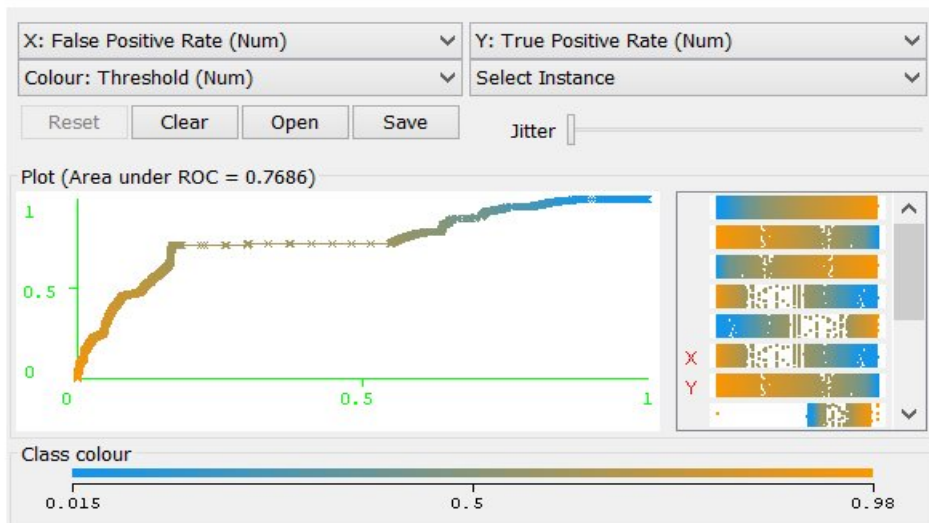
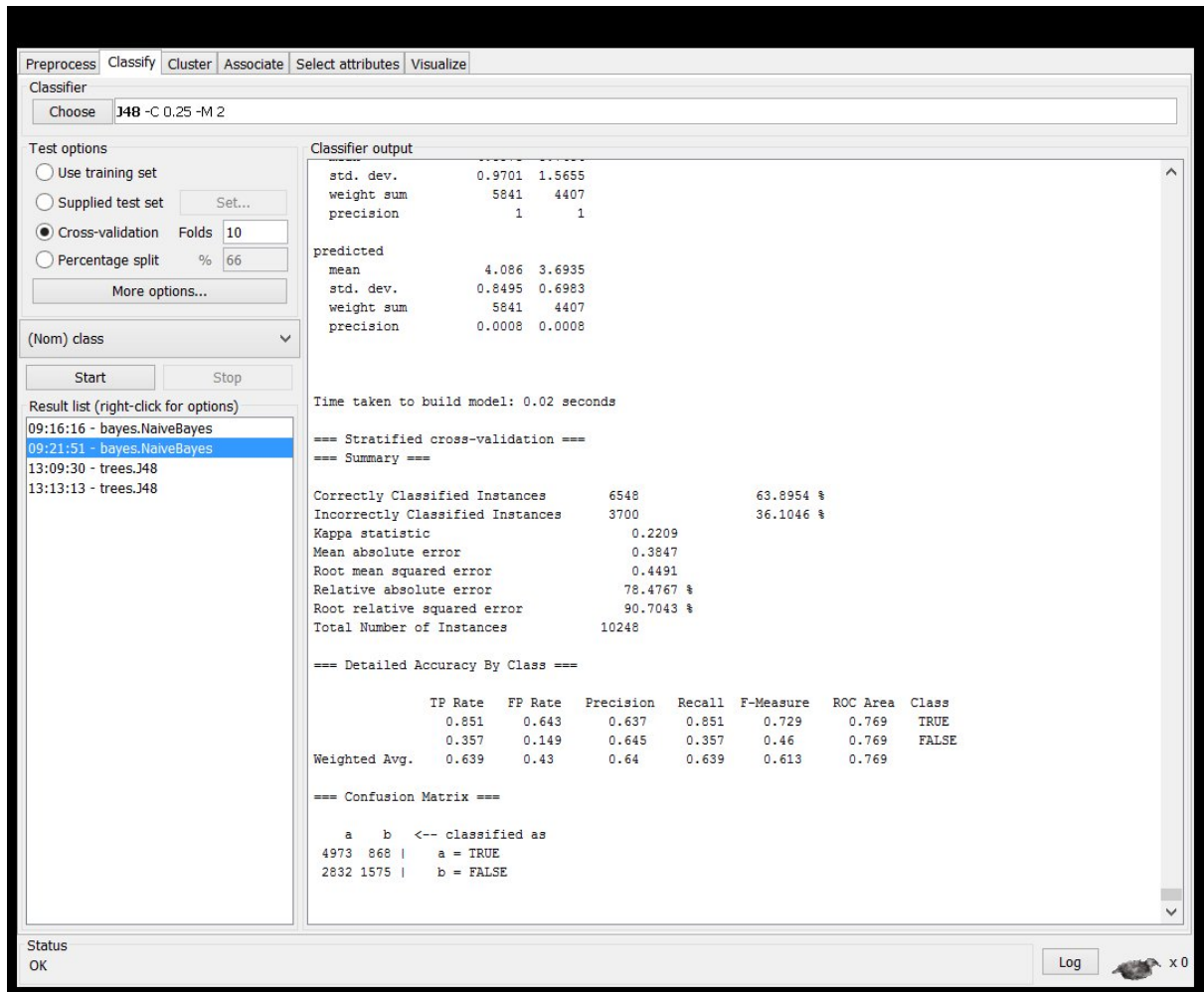
The above section presents the accuracy of recommendation system while suggesting a movie to the user we are not only interested in the correct rating of movie but also what are the top movie suggestions that can like by user and how much that list would be liked by the user. Both recommended systems are compared on the basis of their ROC Curves [5].

The ROC curves are made using WEKA [7] based on Naïve Bayes classifier.

System1 ROC = 0.759



System 2 (Pearson similarity + Wilson score) ROC = 0.769



5.3 Observations

The above testing of recommended system clearly shows that System1 (Pearson similarity) gives better accuracy while using RMSE formula. But usage prediction of system2 is better than system1. Our observation shows us that in many cases system2 outperforms system1 but for some cases it gives worst results because of which its RMSE error is greater than system1 as we calculate mean in RMSE formula.

While testing on 10K tuples system1 predict 5699 correct predictions whereas, system2 predict 5841 correct predictions. The rating is considered as correct if the error of predicted and original is less than 1. The above numbers clearly shows the outperformance of system2 and also system2 gives more balanced and reliable results.

User	Product	Original Rating	System 1	System 2	Performance better
A3R6ST52EBN2FK	B000ARIS52	4	4.2	4.172	System2
A6ADO7B6FUVN	B0008FXTB4	3	3.2	2.7	System1
A4A652FQLPVR0	B001PR0YGC	4	4	4	Both
A2LZBZZ6V3Q4CX	B00005RJ1U	1	4	4	Both
A11GO5VA74HD8K	B000VHU4CG	5	4.13	4.19	System2
ABSX5TGEGRH76	B000067JG2	5	3.818	4.42	System2
A2JC55M4BFADBK	B000QE1U9K	5	4.46	4.50	System2
A18Y3VRG7OP4J2	B00006FMFZ	5	4.64	4.64	Both

A2YTUSSQKJIZP K	B0001KU90U	5	4.08	4.08	Both
A2ORMBJU6V5UV 5	B004SIP8QQ	5	4.6	4.3	System1
A1FNUCH09U9YE 9	B009ITM7CS	5	4.80	4.78	System1
A3J7FI5VE7IEAC	B0007TKOA 0	5	4.5	4.38	System1
A14Q6R4481YFF G	B006H90TLI	4	4.2	4.41	System1

The above table clearly shows the list about how both systems works. We tried to observe some users where system1 performs better than system2 and reason behind it. We find that if the helpfulness factor is made better as in some cases where the helpfulness is just 1/1 or have some low values system1 always perform better. So to make system2 works better this factor should have some big values not just 1/1.

6. Conclusions

As mentioned in introduction, Recommender Systems are boon for online retailers to gain interest of their customers. As we have studies different recommendation system filtering we implemented collaborative filtering and make user-user recommendation system. We have implemented Basic recommendation techniques and tried out to test what new can be done with it to make it better. Hence we came up with the idea of new addition to recommendation formula what we call “helpfulness factor”.

We believed that if we have to find the better recommendation for users we have to depend on users which are more reliable and their reliability can be tested by the fact their reviews are rated by the other users. So we thought to make use of helpfulness in our project and use it in an efficient way. Helpfulness was divided into two column during data pre-processing and we find a challenge with the helpfulness that the fractions are not in the form, the way we wanted them. We noticed that few products are rated 2/2 that means two out of two people found the review useful and other product has 58/60 helpfulness which again say 58

out of 60 people found review useful. So to deal with this fraction problem we have implemented Wilson score formula which makes the fractions correct.

We have seen many E-Commerce websites doing this mistake they make recommendation preference to products which has been rated five star by only one user rather than a product which has 4.5 rating given by 58 users. This helpfulness factor performs good at various places and performs more likely to basic Pearson recommender most of times and at times it performs worst.

But if we look at the results that we have found after testing on 10K users, we find system2 works well and we think it have future scope if “helpfulness factor” is observed more carefully by making some experiments with its values.

Acknowledgement

This paper is part of Fall 2015 Data Mining Course and it was presented for final project of the course(CSC 587D). We thank Dr. Alex Thomo for their constant guidance during the course and project.

References

- [1] “Evanmiller.org”. [Online]. Available: <http://www.evanmiller.org/how-not-to-sort-by-average-rating.html>
- [2] Greg Linden, Brent Smith, Jeremy York. “Amazon.com recommendations”. [Online]. Available: <http://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf>
- [3] Guy Shani , Asela Gunawardana.” Evaluating Recommendation Systems”. [Online]. Available: <http://research.microsoft.com/pubs/115396/evaluationmetrics.tr.pdf>
- [4] R.Srinivasa Raju, I.Kali Pradeep, I.Bhagyasri, P. Praneetha, and P.S.S. Teja, “Recommender Systems for E-commerce: Novel Parameters and Issues,” in Proceedings of the 2013 International Journal of Advanced Research in Computer Science and Software Engineering, vol . 3.
- [5] “Receiver operating characteristics”. [Online]. Available: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

[6] "snap.stanford". [Online]. Available: <http://snap.stanford.edu/data/index.html>

[7] Machine Learning Group at the University of Waikato, "WEKA: Waikato Environment for Knowledge Analysis." [Online]. Available: <http://www.cs.waikato.ac.nz/ml/index.html>

[8] "wikipedia". [Online]. Available: http://en.wikipedia.org/wiki/Recommender_system