ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Proyecto Fin de Carrera

Estudio del rendimiento de los sistemas de recomendación percibido por usuarios individuales y grupos

AUTOR: Ana Fuster Pay

DIRECTOR: José María Molina García-Pardo

Junio / 2015

Permission of use on loan

Acknowledgements

I would like to thank my supervisor Toon De Pessemier for supporting me and for all the help that he has provided me during the development of my master's thesis. I would also like to thank Michael Ekstrand for helping me with all the problems that I have had using LensKit. Furthermore, I cannot forget to thank my Spanish promoter, the professor José María Molina, and my Belgian promoter, the professor Wout Joseph since they have made my stay in Ghent possible.

Furthermore, I have to thank all the people that have participated in my questionnaires because without them this research would not have been possible.

Last but not least, I would like to thank my family for their support, for their continuous encouragement during my degree and the developing of this thesis, and for give me the opportunity to live this experience. Mariloli and Beatriz, thank for your help with my language problems. Pablo, thank you for being my support, despite the distance, for always helping me and do not allow me to give up.

Estudio del rendimiento de los sistemas de recomendación percibido por usuarios individuales y grupos.

La característica más importante de un sistema de recomendación es el nivel de satisfacción de los usuarios con él. Numerosos estudios afirman que la medida de la precisión de un sistema no es suficiente para satisfacer a los usuarios. Por ello, son necesarias otras medidas cualitativas como la diversidad, novedad o confianza en el sistema para entender la percepción que los usuarios tienen sobre la calidad del sistema de recomendación. En este proyecto hemos analizado la importancia de estas medidas subjetivas en la percepción que los usuarios tienen del sistema, cómo éstas afectan a la satisfacción de los usuarios, a través de un estudio online en el dominio de las películas. Encontramos que para evaluar la calidad del sistema de recomendación es necesario tener en cuenta además de la precisión otras características como la novedad y la eficacia.

Por otro lado, cada día crece la necesidad de crear recomendaciones para grupos de usuarios. Normalmente una persona no acude sola al cine, es usual que vaya en pareja o con un grupo de amigos. Por ello, hemos llevado a cabo una evaluación online de grupos de usuarios para estudiar la viabilidad del uso del mismo sistema de recomendación para crear recomendaciones tanto para usuarios individuales como para grupos. Y los resultados muestran que es posible realizar recomendaciones a grupos sin necesidad de sistemas de recomendación más complejos.

1.1 INTRODUCCIÓN

La cantidad de información disponible en Internet ha crecido enormemente desde la aparición de nuevas tecnologías y redes sociales. Como consecuencia de este crecimiento, ha nacido un problema denominado 'sobrecarga de información'. La solución a dicho problema es el uso de sistemas de recomendación, de esta manera conseguimos filtrar aquella información que realmente es relevante para el usuario, se aligera el tiempo empleado para la toma de decisiones a la hora de realizar una compra, seleccionar un libro para leer o una película para ver. Sin embargo, ¿cómo podemos estar seguros de que estamos usando el mejor sistema de recomendación?

Muchos investigadores han analizado el empleo de nuevas medidas subjetivas para medir la percepción que los usuarios tienen del sistema, ya que la satisfacción de los usuarios asegura la optimización del sistema. Con el fin de encontrar la relación entre dichas medidas y la calidad del sistema, realizamos un estudio con usuarios reales en el dominio de las películas, con el propósito de analizar cómo afectan estas medidas a su satisfacción.

En este estudio se han examinado 6 algoritmos de recomendación diferentes (tres algoritmos de filtrado colaborativo, un algoritmo híbrido y dos algoritmos básicos) a través de una evaluación offline con el fin de identificar los mejores parámetros para configurar cada uno de los algoritmos, seguida por una evaluación online con usuarios reales. En dicha evaluación online, los usuarios tienen que comparar seis listas con películas recomendadas por cada uno de los algoritmos, además de responder a una serie de preguntas para medir la *Precisión*, *Novedad*, *Confianza*, *Variedad*, *Efectividad* y *Calidad* de cada uno de los algoritmos de recomendación empleados.

Además, también llevamos a cabo el análisis de recomendaciones grupales con el propósito de demostrar que no se necesitan sistemas complejos para crear buenas recomendaciones para grupos. También investigamos si las medidas subjetivas mencionadas con anterioridad influyen en la satisfacción del grupo con las recomendaciones recibidas.

1.2 EVALUACIÓN

La primera parte de nuestro estudio incluye una evaluación teórica de seis familias de algoritmos distintas con el objetivo de obtener los mejores parámetros para cada una de ellas. Una vez que tenemos la configuración óptima para cada algoritmo, llevamos a cabo una evaluación online a través de dos cuestionarios con el propósito de conocer la satisfacción de nuestros usuarios con cada uno de los algoritmos.

1.2.1 Evaluación offline

Hemos aprovechado la gran cantidad de bases de datos públicas disponibles en el dominio de las películas para llevar a cabo esta parte de la investigación. Concretamente hemos empleado tres bases de datos de MovieLens (100k, 1M, 10M).

- 100K Dataset: incluye 100000 puntuaciones, 963 usuarios, 1682 películas.
- 1M Dataset: incluye 1000000 puntuaciones, 6040 usuarios, 3900 películas.
- 10M Dataset: incluye 10000054 puntuaciones, 71567 usuarios, 10681 películas, 95580 tags.

En dichas bases de datos todas las puntuaciones están en una escala del 1 al 5, y todos los usuarios han puntuado al menos 20 películas.

También hemos empleado el software LensKit, el cual ha sido desarrollado por el grupo de investigación GroupLens de la Universidad de Minnesota. LensKit es una herramienta en Java creada con el propósito de facilitar la investigación en la evaluación de sistemas de recomendación. Para ello provee un framework con gran cantidad de librerías que permiten configurar los algoritmos deseados empleando implementaciones modulares o desarrollando nuestro nuevo código con muy poco esfuerzo.

- Algoritmos

Como ya hemos mencionado, hemos estudiado 6 familias de algoritmos diferentes, las cuales son:

1. **Lucene**: Es un algoritmo de recomendación híbrido, los resultados de salida de un algoritmo basado en contenido (Lucene) son tomados como entrada para el siguiente algoritmo que es de filtrado colaborativo basado en objeto (ItemItem). La salida de éste será la salida de nuestro sistema de recomendación híbrido. Hemos comparado dos versiones de este algoritmo, con y sin normalizar, y los mejores resultados se obtuvieron para Lucene Normalizado con un tamaño de vecindario de 100.
2. **SVD:** es un algoritmo de filtrado colaborativo basado en una descomposición matricial. Este algoritmo se ha configurado empleando cuatro bases de referencia distintas. Tras compararlas, la mejor base de referencia ha resultado ser Media Personalizada teniendo en cuenta 25 características.
3. **UserUser:** es un algoritmo de filtrado colaborativo basado en usuarios. Este algoritmo crea recomendaciones teniendo en cuenta la puntuación que usuarios con gustos similares al usuario para el que se crea la recomendación han dado a un objeto. Hemos configurado este algoritmo usando dos funciones de similitud diferentes: Pearson y Coseno. Finalmente, la mejor configuración fue UserUserCosine para un tamaño de vecindario de 50.
4. **ItemItem:** es un algoritmo de filtrado colaborativo basado en objeto. Este algoritmo guarda las puntuaciones que un usuario ha dado a diferentes objetos y de esta manera crea las recomendaciones teniendo en cuenta la puntuación que el usuario ha dado a un objeto similar al que se está recomendando. Los mejores resultados se obtienen con un tamaño de vecindario de 20 usando la función coseno como medida de la similitud entre objetos.
5. **Popular:** es un algoritmo básico. La popularidad de un objeto es una medida de cuán conocido es el objeto. Se calcula con la media de la gente que ha puntuado un objeto y las puntuaciones que ese objeto ha recibido.
6. **Personalized Mean** (PersMean): es un algoritmo básico, cada usuario recibe una recomendación adaptada a sus gustos.

- Resultados

Hemos analizado los algoritmos anteriores teniendo en cuenta tres medidas: error cuadrático medio (RMSE), ganancia acumulada descendente normalizada (nDCG) y entropía. En función de los valores obtenidos para cada una de estas medidas hemos seleccionado la mejor configuración para cada uno de los algoritmos. La siguiente tabla resume los resultados obtenidos para cada una de las medidas analizadas, teniendo en cuenta la mejor configuración de cada algoritmo:

	1. RMSE	2. nDCG	3. Entropía
1º	SVD	Popular	Popular
2º	ItemItem	ItemItem	Lucene
3º	UserUser	Lucene	ItemItem
4º	Lucene	SVD	SVD
5º	Persmean	UserUser	Persmean
6º	-	Persmean	UserUser

1.2.2 Evaluación Online

Para llevar a cabo la evaluación online, hemos creado dos formularios gracias a la tecnología de *Google Forms.*

En primer lugar, necesitamos obtener valoraciones a varias películas dadas por los usuarios para poder crear las recomendaciones. Para conseguir un mayor número de participantes hemos enviado los cuestionarios a través de redes sociales como Facebook o Twitter, lo que ha facilitado recoger los datos para llevar a cabo la evaluación así como el procesado de las respuestas recogidas.

El primer formulario está dividido en dos partes: la primera parte está diseñada para recoger los datos personales del usuario, y la segunda parte del formulario es la lista de películas que debe valorar. Todos los usuarios tienen que puntuar aquellas películas que hayan visto de la lista de 100 películas que aparecen en el formulario, estas películas se tomaron del ranking de películas más votadas de IMDB. Entre el 25 de noviembre de 2014 y el 7 de diciembre de 2014, rellenaron el primer formulario 158 usuarios, 138 usuarios individuales y 20 grupos de usuarios. Una vez que hemos recogido los datos, empezamos a procesarlos para obtener las recomendaciones para cada usuario.

El segundo formulario contiene las 6 listas de recomendaciones, y a continuación una serie de 17 preguntas encaminadas a evaluar cada una de las 6 diferentes características subjetivas de interés de los algoritmos. Estás preguntas fueron empleadas en dos estudios anteriores desarrollados por Ekstrand y Knijnenburg.

Debemos señalar que sólo 60 de los 158 usuarios que rellenaron el primer formulario, completaron este segundo formulario entre el 16 de febrero de 2015 y el 21 de marzo de 2015. De los cuales 50 eran usuarios individuales y 10 eran grupos. Entre los usuarios individuales podemos distinguir por género (29 mujeres y 21 hombres) y por edad (40 jóvenes y 10 mayores).

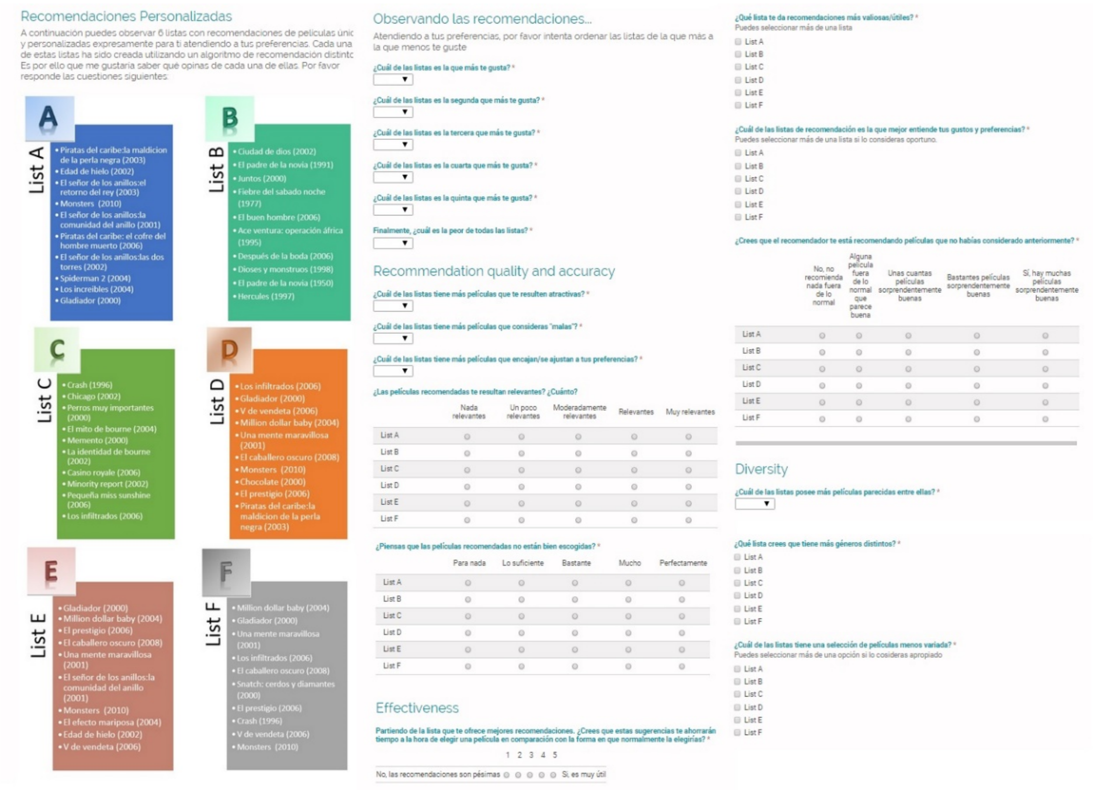


Figura 1: Muestra de una parte del segundo formulario.

1.3 RESULTADOS

En primer lugar, pedimos a nuestros usuarios que ordenaran las 6 listas de recomendación teniendo en cuenta su preferencia, de la que más le gustaba a la que menos, y los resultados obtenidos muestran que los algoritmos de filtrado colaborativo seguidos por el basado en popularidad son los que mejor satisfacen las necesidades de los usuarios. Sin embargo, el algoritmo híbrido Lucene y Persmean son los peores.

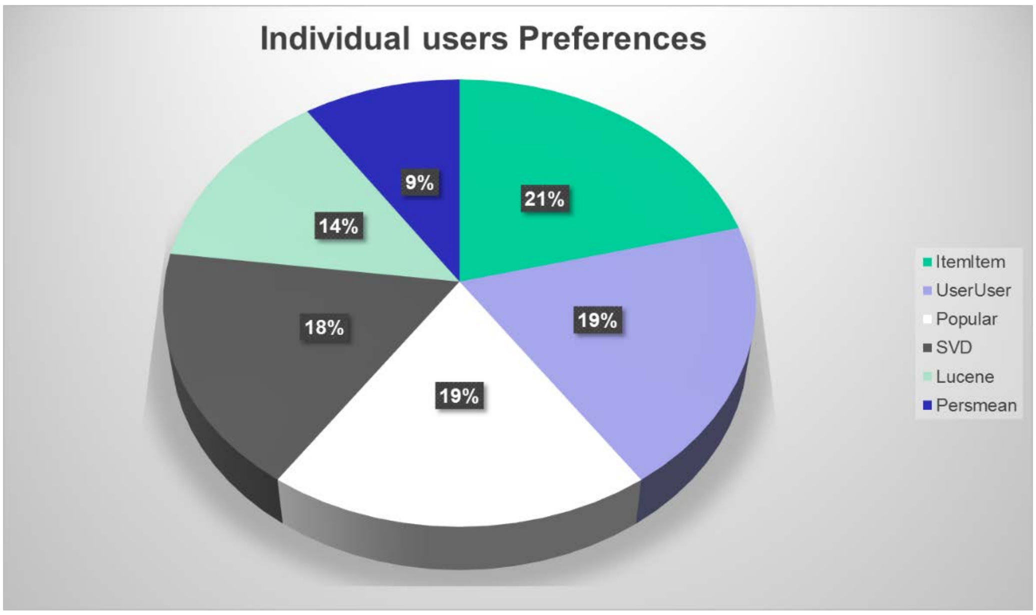


Figura 2: Preferencias de los usuarios individuales

Además, en nuestro estudio nos hemos centrado en la medida de la percepción que los usuarios tienen sobre ciertas características que poseen los sistemas de recomendación como son *Precisión*, *Confianza*, *Variedad*, *Novedad*, *Efectividad* y *Calidad*. A continuación vamos a comentar los resultados principales obtenidos evaluando cada una de ellas.

La *Precisión* de un algoritmo está fuertemente relacionada con la primera impresión que los usuarios tienen de él. La satisfacción de nuestros usuarios está unida a lo llamativas y buenas que encuentran las películas recomendadas.

La *Confianza* es un algoritmo también está fuertemente relacionada con la satisfacción que los usuarios tienen con el sistema de recomendación ya que como hemos visto en la primera impresión de nuestros usuarios, los algoritmos que mejor entienden sus gustos son los mejor considerados en su elección inicial. Esto nos lleva a deducir que es necesario generar confianza en el sistema. Los resultados muestran que los algoritmos en los que más confían los usuarios son ItemItem y Popular.

Los resultados obtenidos tras la evaluación de la *Variedad* de un algoritmo no son concluyentes, ninguna de las preguntas evaluadas han aportado resultados significativos estadísticamente, con lo cual no podemos extrapolar los resultados obtenidos.

En cambio, cabe señalar que los resultados obtenidos en nuestra evaluación muestran que la *Novedad* tiene un efecto negativo en la satisfacción de los usuarios con el sistema. Las listas de recomendación que incluyen más películas sorprendentes son aquellas que pertenecen a los algoritmos peor considerados en la primera impresión de los usuarios. Podemos afirmar que para asegurar buenas recomendaciones el diseñador del sistema debe garantizar que la lista de recomendación incluya alguna película conocida por el usuario para así incrementar su confianza en el sistema, ya que una lista de recomendación que sólo incluye películas desconocidas para el usuario (nuevas) le hace desconfiar en el sistema.

Para calificar un sistema de recomendación como *Efectivo*, no solamente se necesitan predicciones precisas y novedosas. Es muy importante que el sistema sea considerado como una herramienta muy útil en la vida del usuario, ya que le debe servir para ahorrar tiempo en la toma de decisiones.

La *Calidad* de un sistema de recomendación es una medida que está muy relacionada con características anteriores como son la Precisión y la Confianza. La opinión que los usuarios tienen acerca de estas características influye en su percepción de la calidad del sistema.

Podemos observar un resumen de los resultados obtenidos en la siguiente tabla:

	1. **Precisión**	2. **Calidad**	3. **Variedad**
1º	Popular	Popular	Popular
2º	ItemItem	ItemItem	Lucene
3º	UserUser	UserUser	Persmean
4º	SVD	Lucene	UserUser
5º	Lucene	SVD	SVD
6º	Persmean	Persmean	ItemItem

Si comparamos estos resultados con los obtenidos en la evaluación offline. Podemos apreciar que nDCG, con un coeficiente de correlación de 0.834, es la medida que mejor mide la calidad de un sistema de recomendación comparado con RMSE o Entropía.

Durante todo el estudio hemos ido analizando las diferencias observadas entre hombres y mujeres. Sólo teniendo en cuenta la medida de la Precisión y la Calidad, estas diferencias han sido significativas estadísticamente. En ambos casos, podemos señalar que los hombres prefieren en mayor medida las películas recomendadas por el algoritmo Popular, mientras que las mujeres prefieren en mayor medida las películas recomendadas por Lucene, UserUser y SVD. Este resultado nos lleva a señalar que las mujeres tienen gustos muy predefinidos, y aunque les gusta ver las películas más popular, siempre encuentran interesantes aquellas películas de los géneros que más le gustan, como pueden ser comedias románticas. Además el estudio anual del *Theatrical Market Statistics* demuestra que el número de mujeres que asisten al cine es mayor que el número de hombres. Esto concuerda con el hecho de que los hombres sólo van a ver las películas más populares mientras que las mujeres acuden a ver las películas populares y además aquellas de otros géneros que encuentran interesantes aunque su nivel de popularidad sea inferior.

Hemos realizado también el mismo análisis con las diferencias entre personas mayores de 25 años y menores, pero debido a que el tamaño de los grupos no es equitativo (10 mayores y 40 menores), es difícil extrapolar los resultados observados.

1.4 RECOMENDACIONES PARA GRUPOS

En esta sección del estudio, nuestro propósito es conocer mejor la satisfacción de los grupos con sus recomendaciones. Por ello en el formulario enviado a los grupos, se añadieron algunas preguntas de respuesta libre para conocer mejor su opinión. De sus respuestas hemos apreciado tres maneras distintas de llegar a un acuerdo para puntuar una película o elegir la mejor lista de recomendación, las cuáles son:

1. Por decisión democrática.
2. Dar una puntuación individual a la película y después calcular la media.
3. Discutir pros y contras de cada película o lista de recomendación.

Además también señalan que lo más complicado es decidir cuál es la mejor lista de recomendación para todo el grupo.

Por otro lado, debemos señalar que cuanto mayor es la similitud entre los gustos de los miembros del grupo mejor es la percepción que tienen de la recomendación dada y además más fácil les resulta llegar a un acuerdo.

Al evaluar cada una de las características de los algoritmos, los resultados son casi los mismos que los obtenidos con los usuarios individuales. La única diferencia que podemos apreciar es que los grupos prefieren ItemItem antes que Popular, pero aun así ambos algoritmos son los mejores en cuanto a la Precisión, Confianza y Calidad. Teniendo en cuenta los peores algoritmos en términos de Precisión, tanto grupos como usuarios individuales están de acuerdo en que son Persmean y Lucene. Además, ambos creen que dichos algoritmos son los que recomiendan más películas novedosas, o inesperadas con connotación negativa.

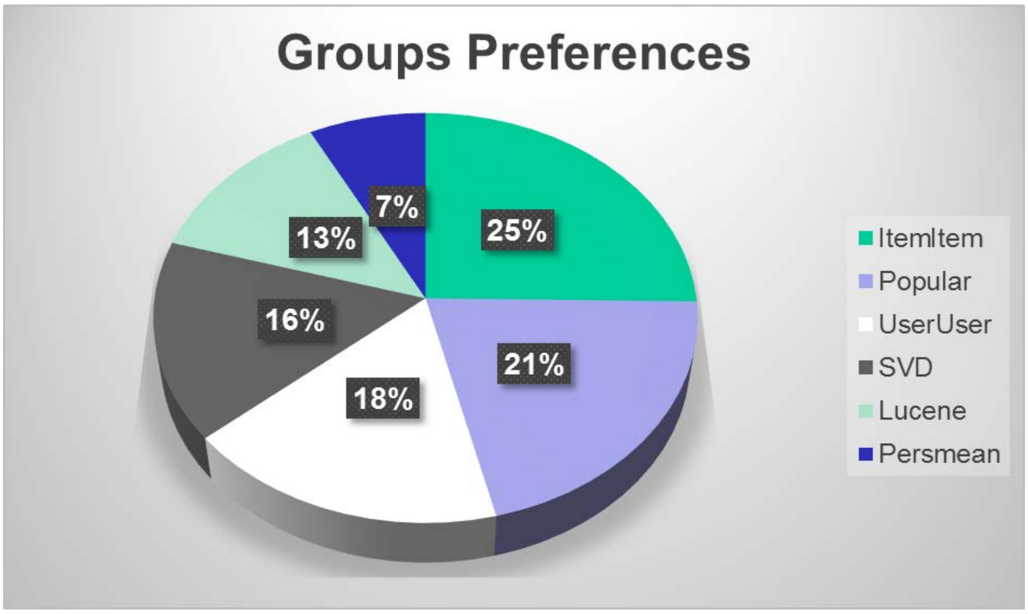


Figura 3: Preferencias grupos de usuarios

1.5 CONCLUSIONES Y ESTUDIOS FUTUROS

Con los resultados obtenidos en este estudio podemos concluir que es posible crear recomendaciones para grupos de usuarios sin necesidad de emplear sistemas complejos, ya que los resultados obtenidos empleando el mismo sistema que para recomendaciones individuales son muy similares.

Además, las medidas subjetivas estudiadas han demostrado que tienen gran influencia en la satisfacción de los usuarios con el sistema. Cabe señalar que la *Novedad* tiene una influencia negativa en la percepción de los usuarios acerca del algoritmo de recomendación. Y la *Efectividad* es importante para predecir la utilidad del sistema.

Los mejores resultados se han obtenido con los algoritmos ItemItem y Popular, mientras que los peores han sido Lucene y Persmean.

Otra conclusión importante son las diferencias apreciadas entre hombres y mujeres, mientras los hombres prefieren en mayor medida las recomendaciones del algoritmo

Popular, las mujeres prefieren en mayor medida que los hombres las recomendaciones dadas por UserUser, SVD y Lucene.

Próximos estudios pueden centrarse en el análisis de la *Diversidad*, empleando para ello más usuarios reales que los que han participado en nuestro estudio. Además se deben desarrollar nuevas medidas teóricas para evaluar otras características del sistema además de precisión y calidad con el fin de mejorar el sistema de recomendación. Y también sería interesante poder analizar la influencia del tamaño de los grupos en la satisfacción con las recomendaciones.

Study of individual users and groups: perceptions of recommender systems performance

Ana Fuster Pay

Supervisor: De Pessemier, Toon

Abstract−**The most important aspect of a recommender system is the users' satisfaction with it. Several studies affirm that the measure of Accuracy is not enough to fulfil users' satisfaction. Other qualitative metrics such as Diversity, Novelty or Trust are needed to understand users' perception of the quality of a recommender [5] [8]. We, therefore, explored how relevant are these subjective metrics in the users' satisfaction with the system through an online study in the movies domain. We found that, in addition to Accuracy, other aspects such as Novelty and Effectiveness are needed to evaluate the system in order to consider it successful. Additionally, there is a need of group recommendations growing every day. It is usual that you do not go to the cinema alone. For this reason, we carried out an online evaluation of groups in order to study the viability of using the same recommenders. We realized that group recommendations are possible without the need of complex systems.**

Keywords − **recommender system, users' satisfaction, subjective metrics, groups recommendations.**

I. INTRODUCTION

The amount of data available on the Internet has enormously increased since the apparition of new technologies and social networks. As a consequence, a problem has emerged called 'information overload'. The solution for this problem is the use of recommender systems. However, how can we be sure that we are using the best system to make recommendations?

Lots of researchers [1] [4] [5] [8] have discussed the use of new subjective metrics to measure the perception of the system that users have about it since users satisfaction ensures the goodness of the recommender. To figure out the relation among these metrics and the quality of a system, we offer a user study in the movie domain with the aim of analyzing how these metrics affect their satisfaction.

This paper examines six different algorithms (three common collaborative filtering, one hybrid, and two basics) through an offline evaluation to identify the best parameter for each of them, followed by the online evaluation with real users. In this online experiment, users have to compare six lists of recommendations produced by each algorithm regarding the measurements of Accuracy, Novelty, Understands Me, Diversity, Effectiveness and Quality.

Our study also covers the analysis of group recommendations with the purpose of proving that there is no need for complex systems in order to make good group recommendations. Moreover, we investigate whether the subjective metrics above mentioned influence in group satisfaction.

II. EVALUATION

The first part of our study covers a theoretical evaluation of six different families of algorithms aimed at obtaining the best parameters for each one. Once we have it, we conducted an online evaluation through two questionnaires with the goal of understanding users' satisfaction with each algorithm.

A. Offline Evaluation

We have taken advantage of the huge amount of publicly datasets available on the movie domain to carry out this part of our research. Concretely, the three MovieLens [3] datasets (100k, 1M, 10M). We have also taken benefit from a software tool (LensKit [7]) which was developed to support different algorithms by the GroupLens research group [2].

1) Algorithms

For this evaluation, we have made use of six families of algorithms:

1. Lucene: We have compared two versions of this algorithm, with and without normalization, and the best results were obtained with Lucene Normalized and a neighborhood size of 100.

2. SVD: collaborative filtering algorithm based on matrix decomposition. We have configured the FunkSVD using four different baselines. After comparing them, the best baseline was SVDPersMean with a feature count of 25.

3. UserUser: user-based collaborative filtering algorithm. We have configured it with two different similarity functions: Cosine and Persmean. Finally, the best configuration was UserUserCosine with a neighborhood size of 50.

4. ItemItem: item-based collaborative filtering algorithm. We have obtained the best results with a neighborhood size of 20.

5.Popular: basic algorithm. The popularity of a given item is a measure of how well known the item is

6.Personalized Mean: basic algorithm, each user receives a recommendation adapted to his tastes.

2) Results

We have analyzed these algorithms taken into account three metrics: RMSE, nDCG and Entropy. Table 1 summarizes the results obtained for the best configuration of each algorithm.

Table 1: Ranking based on objective metrics. Note that we cannot calculate the RMSE for Popular. That is why it does not appear on the first rank.

	1. RMSE	2. nDCG	3. Entropy
1st	SVD	Popular	Popular
2nd	ItemItem	ItemItem	Lucene
3rd	UserUser	Lucene	ItemItem
4th	Lucene	SVD	SVD
5th	Persmean	UserUser	Persmean
6th	-	Persmean	UserUser

B. Online Evaluation

To carry out the online evaluation, we have created two forms powered by the technology of *Google Forms.*

The first step in the evaluation is to collect users' rating to give them recommendations. To reach a larger number of participants we have sent it through social networks such as *Facebook* or *Twitter*, making it easier to collect the data and process their responses. This form is divided into two sections: the first one is designed to collect the personal data of the subject under study, and the second part of the form is the rating list. Users rated a list of 100 selected movies from the top of *IMDB*. Between 25th November 2014 and 7th December 2014 158 users filled the survey, 138 were individual users and 20 were groups. Once we have collected the data, we start to process it to obtain the recommendations to each user. The second form contains 6 recommendations' lists and 17 questions to know users perception of the algorithms used. These questions are taken from Ekstrand [1] and Knijnenburg et al. [6] since they have proved that these questions worked well in other similar studies.

We have to highlight that only 60 of the 158 users that filled the first form completed this second survey: 50 of them were individual users and 10 were groups. Among the individual users, we can make a distinction by gender (29 female and 21 male) and also by age (40 younger than 25 and 10 older than 25).

1) Results

We asked the users to order the lists taking into account their preferences, and the results obtained show that Collaborative filtering algorithm followed by Popular are the most satisfying ones for the users. However, Persmean and Lucene are the worst ones.

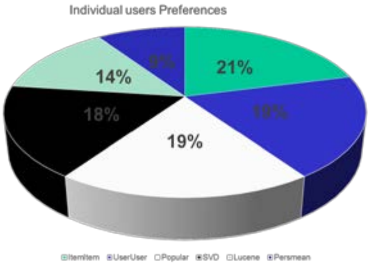


Figure 1: Percentages of individual users' preferences.

Furthermore, in this study, we have focused on measuring the users' perception of some recommender systems' features such as Accuracy, Understands Me, Novelty, Effectiveness and Quality. We are now going to explain some of the key findings.

Accuracy is strongly related to the users' first impression of an algorithm. The satisfaction of the users is tied to their perception of how appealing or good the recommended movies are.

Understands Me is also highly related to the user satisfaction since, the algorithms that best understand their tastes are the best considered ones in their initial choice. This suggests that it is necessary to generate trust. The results show that the algorithms on which more users rely are ItemItem and Popular.

We have to underline that Novelty has a negative effect on users' satisfaction. The recommendations with more surprising movies are made by the worst considered algorithms regarding the users' first impression. We can affirm that, to ensure good recommendations, the designer has to guarantee some known movies in order to increase the trust on the system since only novel items in a list makes the user beware of the system.

The Quality of a recommender system is a metric which is highly related to other metrics such as Accuracy and Understands Me. The opinion that the users have about these other metrics influence their perceptions of the system' Quality.

We can summarize the results in Table 2.

Table 2: Ranking of algorithms based on three subjective metrics.

	1. *Accuracy*	2. *Quality*	3. *Diversity*
1st	*Popular*	*Popular*	*Popular*
2nd	*ItemItem*	*ItemItem*	*Lucene*
3rd	*UserUser*	*UserUser*	*Persmean*
4th	*SVD*	*Lucene*	*UserUser*
5th	*Lucene*	*SVD*	*SVD*
6th	*Persmean*	*Persmean*	*ItemItem*

If we compare these results with the obtained from the offline evaluation. We can ensure that nDCG with a correlation coefficient of 0.834 is the metric that best measures the goodness of a recommender compared to the others.

III. Group Recommendations

In this section of the study, our purpose is to figure out the satisfaction of groups with their recommendations. Therefore, we have added some additional open questions to the groups' questionnaires. From their answers, we have appreciated three different ways to reach an agreement in order to rate movies or select the best recommendation list, which are:

1. Democratic decision.
2. Individual ratings and averaging.
3. Discuss pros and cons of each movie.

The biggest difficulty found by the group members is to select the best recommendations list. Furthermore, we can highlight the differences that they have appreciated between genders. Additionally, we can remark that a higher similarity in the group members tastes is reflected in a better perception of the recommender systems and also in the facility of reaching an agreement.

Evaluating each metric, the results are almost the same as for individual users. Nevertheless, it is notable that groups prefer ItemItem before Popular, but both are still the best algorithms in terms of Accuracy, Understands Me and Quality. Moreover, groups as well as individual users think that the algorithms with more novel movies recommended are Persmean and Lucene, whose are considered the worst in term of Accuracy.

IV. Conclusions and Future Research

From this study, we can conclude that group recommendations are possible without the need of complex systems since the results obtained are quite similar to the analysis of the individual users. Furthermore, the subjective metrics studied have demonstrated their influence in users' satisfaction. It's notable that Novelty has a huge negative influence on the user's perception of the recommender algorithm. Future research should focus on performing this study with more users to improve the online analysis of Diversity. Additionally, the development of new theoretical metrics to evaluate other aspects is needed to improve the recommender systems.

REFERENCES

[1] Ekstrand, M. (2014). *Towards Recommender Engineering: Tools and Experiments in Recommender Differences*. Ph.D. Thesis, University of Minnesota. Retrieved from http://elehack.net/research/thesis/

[2] GroupLens Research. (n.d. a) What is GroupLens. Retrieved October 6, 2014 from http://grouplens.org/about/what-is-grouplens/

[3] GroupLens Research. (n.d. b) Datasets. Retrieved October 6, 2014 from http://files.grouplens.org/datasets/movielens

[4] Herlocker, J., Konstan, J., Terveen,L., Riedl,J. (2004). *Evaluating collaborative filtering recommender systems*. ACM Transactions on Information Systems, 22(1), 5-53. [doi>10.1145/963770.963772]

[5] Knijnenburg, B., Willemsen, M., Kobsa, A. (2011, October). *A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems*. Proceedings of the fifth ACM conference on Recommender systems, Chicago: ACM. [doi>10.1145/2043932.2043993]

[6] Knijnenburg, B., Willemsen, M., Gantner, Z., Soncu, H., Newell, C., (2012), *Explaining the user experience of recommender systems.* User Modeling and User-Adapted Interaction, 22(4-5), 441-504. [doi>10.1007/s11257-011-9118-4]

[7] LensKit. (n.d.). Retrieved October 27, 2014, from http://www.recsyswiki.com/wiki/LensKit

[8] Pearl Pu, Li Chen (2011, October).*A User-Centric Evaluation Framework of Recommender Systems.* Proceedings of the fifth ACM conference on Recommender systems, Chicago: ACM. [doi>10.1145/2043932.2043962]

Index

List of Figures

List of Tables

1 INTRODUCTION

1.1 STATE OF THE ART

Nowadays, more than 2.5 billion gigabytes of data are created every day in multiple forms. On the internet, 72 hours of Youtube videos are uploaded, Google addresses 4 million search queries, 2.4 million posts on Facebook, 278 thousand tweets, 61141 hours of music are listened on Pandora, and 204 million emails are sent, Amazon makes 83000$ in sales and 17 thousand transactions take place at Walmart in one single minute [50].

By 1966, before the introduction of the personal computer, before the explosion of the World Wide Web, before the 'Information Age', Hubert Murray [38] said "every day, approximately 20 million words of technical information are recorded. A reader capable of reading 1000 words per minute would require 1.5 months, reading eight hours every day, to get through one day's output, and at the end of that period he would have fallen 5.5 years behind in his reading" (p. 1).

If there were such a huge amount of data 50 years ago, this amount has enormously increased nowadays. However, the positive issue is that it allows us to improve our knowledge and to enrich personally.

According to Yue [45], in the present time, new technologies have spread the usage of the Internet as a searching tool due to the fact that there is a huge amount of information that can be found on the Internet. Moreover, social networks where people can communicate and upload different materials have been used to the spread of this amount of information available.

A problem is emerging as a consequence of this, called 'information overload'. Due to this problem, recommendation services have gained great attention in the last years [3] [45].

However, sometimes the recommendations generated by recommender systems are not as good as expected. Research on evaluation of recommender system have previously focused on algorithm performance in terms of *Accuracy*. Herlocker et al. [19]

described it by saying that: It is believed that the measurement of accuracy is not enough to provide users with a useful tool which helps to meet their needs. Moreover, these authors [19] agree on the fact that a system should be useful for users although accuracy should also be part of that usefulness.

In recent years, it has been recognized by industry and academic researchers that the ultimate goal of recommenders is to help users make better decisions. For this reason, the measure of *Accuracy* is not enough to fulfil user satisfaction. Other qualitative metrics such as *Diversity*, *Novelty* or *Trust* are needed to understand the users' perception of the quality of a recommender [14] [19] [22] [33] [40].

But this qualitative metrics cannot be measured in an offline experiment; a user interaction with the system is required. Therefore, to determine the best algorithm and the best configuration of it, an online evaluation is needed.

1.2 Objective

The main aim of this piece of work is to understand the subjective differences that users perceive among different algorithms and how these differences affect their opinion about a recommender system. In this thesis, we shall analyse users' perception of recommender to improve their qualit*y*. In addition, in order to find the best performance of each family of algorithms used, a study of their parameters will be carried out through LensKit [29] and a survey will be filled by real users to develop the online evaluation. After that, a comparison between offline and online metrics will be elaborated.

In order to study group recommendations, we will ask users to fill the survey in groups. In this way, we will analyse how valuable our recommendations are for groups.

2 Theoretical Study

2.1 Algorithms

Regarding the algorithm used, the work domain or the kind of knowledge employed, we can find lots of different approaches of recommender systems [6] [23] [25] [41] [43].

In this thesis we are going to distinguish between four different types of recommender systems:

2.1.1 Content-based

The system is trained to make recommendations based on previous ones, which means that the system will make the recommendation to the user based on previous choices that this specific user has had on the past [4] [15] [30] [35][43].

2.1.2 Collaborative filtering

This system is prepared to make recommendations according to tastes. That is, it analyses your tastes and the ones in neighbourhoods so that it recommends you the items regarding what other users with similar tastes have enjoyed [19] [43] [56].

2.1.3 Knowledge-based

Knowledge-based systems store a series of items so that they create knowledge from the information that they are given. Moreover, they use that knowledge in order to make recommendations to different users. We can distinguish between two types of knowledge based recommender systems: case-based and constraint based [25] [43] [54].

2.1.4 Hybrid recommender systems

These systems are a mixture of the ones which we have previously explained. The main characteristic that we can observe is that they interconnect two types of systems and use their advantages so that the disadvantages of each of the systems are not taken into consideration for the recommendations [43] [47] [55].

In section 3, we will analyse each of them in depth.

2.2 METRICS

2.2.1 Objective Metrics

Traditionally, recommender algorithms 'goodness' have been judged based on a small set of coverage and accuracy metrics.

With regard to *Accuracy*, we can distinguish between decision-support or statistical, whose metrics compare the estimated ratings against the actual ratings [41].

We are going to use one of the statistical metrics to measure *Accuracy*, specifically root mean squared error (RMSE). And we are going also to study entropy and normalized cumulative discounted gain (nDCG) to have a better perception of the algorithm' performance.

2.2.1.1 Root Mean Squared Error (RMSE)

It is one of the most used metrics to measure *Accuracy*. It computes the differences between the predicted ratings and the true ratings known. We can find two variations depending on how it is calculated, averaging based on users or items [41].

2.2.1.2 Measuring Ranking Prediction

In order to measure the *Quality* of the recommendation lists, a metric called Normalized Cumulative Discounted Gain (nDCG) has been demonstrated to work well in the area of recommender systems [41].

This metric is based on the assumption that a user is going to read the movies recommended on a list using the top-down strategy, so that the accuracy of the recommendation list is the sum of the accuracy of each movie recommended but also influenced by the position of the movie in the list (as the movie is in a lower position, its accuracy decreases) [41].

2.2.1.3 Entropy

Entropy quantifies the level of consistency of the relationship between two items. Therefore, we use it to measure diversity in a recommendation list [10] [31].

2.2.2 Subjective Metrics

As we have discussed, *Accuracy* is not the only measure that can influence users' satisfaction since there are other characteristics that also have an influence on their perceptions [23] [24] [33] [43] [46].This is the reason why we need other measures to obtain a good evaluation, along with those mentioned above.

In this thesis, we have focused on how users perceive the algorithms used to make their recommendations, and how it influences their engagement with the recommender system.

We will study users' perceptions on the dimensions of *Novelty*, *Diversity*, *Serendipity*, degree of *Quality* and *Effectiveness* to understand how users perceive the different output from various recommender algorithms, and how those differences affect their opinion of an algorithm.

2.2.2.1 Novelty

Novelty is the measure of how many new and interesting recommended items are received by users. It is quite difficult to ensure *Novelty* at the same time than *Accuracy* because there may be items unknown by users but irrelevant for them.

'Serendipity' is sometimes used instead of *Novelty*, but this is not accurate since *Novelty* only implies items unknown by the users while serendipity refers to unknown items which are surprisingly good to users [24] [41] [52]. As Wen Wu, Liang He and Jing Yang [52] said: "Serendipity is a measure of how surprising the successful recommendations are".

2.2.2.2 Diversity

Diversity is generally defined as the opposite of similarity. In some cases, suggesting a set of similar items may not be as useful for the user because it may take longer to explore the range of items. Moreover, if there is not any similarities among the items recommended, users' satisfaction with the system could be affected too [23] [41] [46].

2.2.2.3 Effectiveness

Effectiveness is a measure of how useful a recommender system is in the life of a user. It refers to the fact of saving time in the process of looking for an item he is interested in by using a recommender system instead of searching it by himself [33] [41].

3 RECOMMENDER SYSTEMS EVALUATION TOOL

3.1 OVERVIEW OF THE TOOL: LENSKIT

As we can read on LensKit wiki page, "LensKit is a Java-based recommender toolkit from GroupLens. It provides a common API for recommender algorithms, an evaluation framework for offline evaluation of recommender performance, and highly modular implementations of standard algorithms for recommendation and rating prediction" [29]. Moreover, it offers extensive support code to allow developers with a minimum of new work to build extensions [13].

As Michael D. Ekstrand [14] said in his dissertation, the main aim of LensKit was to provide support for research on recommender systems and design a reliable platform useful for technique experimentation in several configurations of the system. Its purpose is to provide recommender systems with high quality and to be a useful tool for recommender researches. That is why we are going to use this framework in our research about users' perception of recommender systems.

The current version at the time of writing was 2.1. To demonstrate some of the implementation aspects of LensKit, we look at a common similarity method, the Pearson Correlation, but also at other methods such as the Cosine Correlation.

Additionally, LensKit contains an evaluator class which can perform cross validation and report evaluation results using a set of metrics such as RMSE, nDCG, etc.

As we have said, several recommendation techniques are implemented by LensKit [13]. These techniques differ in the item scorer they implement. This item scorer implementation configures the algorithm. An item scorer can be defined as an overall idea about the expected ability to generate personalized scores for every user. Moreover, LensKit also makes use of data access objects (DAOs) in order to access to all the components of the system [14].

3.2 Evaluation Scripts

LensKit uses Groovy in order to create the evaluation scripts, whose organization is carried out taking into account different configurations.

When we try to use LensKit to compare algorithms, our script has to specify three issues [14]:

1. The dataset we want to use.
2. The algorithms we want to test and compare.
3. The metrics used to make the comparison.

After that, we will be able to develop our recommenders.

3.3 ALGORITHMS IMPLEMENTED USING LENSKIT

3.3.1 *ItemItem*

It is an item-based collaborative filtering algorithm. This algorithm stores different user's rating of different items so that the recommendation is carried out regarding the rating that the user has given to an item which is similar to the one that is being recommended [12] [25] [41].

3.3.1.1 Parameters

At the time of implementing this algorithm with the help of LensKit, we have used some specific LensKit parameters to configure *ItemItem*:

NeighborhoodSize: this parameter allows us to establish the size of neighborhood of each prediction [26].

ItemSimilarity: with this parameter we stipulate the similarity function that we are going to use in the system in order to find out the relation between items [26]. We use *ItemVectorSimilarity*, using cosine similarity as *VectorSimilarity*.

Threshold: Can be defined as the measure that distinguishes the main similarities which should remain in order to make a good recommendation. In our case, we consider the main similarities as the ones that are positive, so that they are the ones that we keep [26].

UserVectorNormalizer: Before the similarity is computed, we use this parameter to apply a normalization to the vector of user rating [26].

3.3.2 *UserUser*

UserUser is a user-based on the nearest neighbour collaborative filtering recommendation. It makes recommendations with regard to the rating that an item has obtained from users with his similar tastes [12] [25] [41] [56].

Similarity between users can be measured using different ways. In our study, we will use two of them, the Pearson correlation and the Cosine similarity.

3.3.2.1 Parameters

At the time of implementing this algorithm with the help of LensKit, we have used some specific LensKit parameters to configure *UserUser*:

UserVectorNormalizer: Before giving prediction and computing the similarity, this parameter applies a normalization to the vector of user rating [27].

NeighborhoodFinder: this parameter is used to find the amount of neighbors which are specified to score the items and make the prediction [27].

UserSimilarity: this parameter is used to specify the similarity used to compare users [27]. We use the *CosineVectorSimilarity as UserVectorSimilarity*.

3.3.3 *SVD*

The Singular Value Decomposition (*SVD*) is a well-known and better performance matrix factorization technique. This technique uses three matrices that are factors from a matrix called R of size m by n.

$$R = U \cdot S \cdot V'$$

Where, *U* and *V* are two orthogonal matrices of size $m \times r$ and $n \times r$ respectively. *r* is the rank of the matrix *R* (the rank of a matrix is the number of linearly independent rows or columns in the matrix) [9] [41] [43]. The rows represent the users while the columns represent the movies. The matrix *S* is a diagonal matrix of size $r \times r$ containing the singular values of the matrix *R*. All these values of S (the specific ratings) are in a decreasing order.

3.3.3.1 Parameters

At the time of implementing this algorithm with the help of LensKit, we have used some specific LensKit parameters to configure Funk*SVD*:

The main step to use Funk*SVD* is to configure *FunkSVDItemScorer* as our *ItemScorer.*

BaselineScorer: this parameter is used to configure the baseline that we are going to use to configure the FunkSVD algorithm [28]. We will use four different baselines: GlobalMeanRatingItemScorer, UserMeanItemScorer, ItemMeanRatingItemScorer and PersonalizedMeanRatingItemScorer.

FeatureCount: the FunkSVD algorithm learn from the baseline a specific number of features that are stipulate by this parameter [28].

3.3.4 *Popular*

The popularity of a given item is a measure of how well known the item is. It is calculated by the average number of people who have chosen an item and the ratings that this item has been given. Moreover, it is worth mentioning that this algorithm gives the same recommendations to all the users regardless of their tastes [12] [25].

3.3.5 *Personalized Mean*

Personalization is an algorithm based on the difference found in the users' recommendations by the system. Therefore, each user receives a recommendation adapted to his tastes. The result is a production of different recommendation lists according to different users' preferences. Besides, there is at the same time a comparison among these recommendation lists in order to find the similarity among their items [34].

3.3.6 *Lucene*

Lucene, is an open library that can be used by all the public as a source of information in order to create tag based algorithms. This library is provided by some techniques used in inverse indexing and searching the index. The main aim of this algorithm is to simulate users' taste according to the results obtained from its search on the index. In this way, it is ensured a good recommendation list of movies to the user [8] [47] [54].

In our study, *Lucene* is used as hybrid recommendation algorithm. Therefore, the output results taken from *Lucene* are used as the input of a second recommender system, which is, in this case, a collaborative filtering algorithm based on item.

4 EVALUATION

Evaluating a recommender system can be carried out by using offline analysis through public datasets, online analysis where live users interact with the system, or a combination of both of them [5] [21] [36]. Through it all, much of the work in recommender evaluation is focused on offline analysis of predictive *Accuracy*.

When we try to evaluate a recommender algorithm, we cannot use only offline evaluation as we would not obtain good results. For this reason, it is important to use both an offline and an online evaluation to obtain better results. For example, most of the times we want to recommend items that the user has not rated yet, so we will not have enough information to evaluate the goodness of the recommended item just from the dataset used [19] [44].

It is clear that it is easier to carry out an offline evaluation with existing datasets than an online evaluation with real users. However, the estimation obtained through an offline evaluation is not as precise as the results collected from an online experiment [44] [53].

For this reason, we implemented two different evaluations to study the performance of the six algorithms above mentioned in the movies' domain. In the offline experiment, we will study the characteristics that best perform each algorithm. In the online experiment our purpose is to know the user' opinion about the recommendations given taking into account their perception of the *Diversity*, *Quality*, *Novelty* and *Effectiveness* of these recommendations made by our algorithms.

We will start with offline evaluation since, as we have mentioned, they are the easiest to perform [19] [21] [36] [44], and we want to use the results obtained from this evaluation to configure the size of neighbourhood, the number of features or the normalizer used among others before start the online evaluation with real users.

4.1 OFFLINE EVALUATION

To perform an offline experiment a pre-collected dataset is needed. This dataset must contain items rated by the users to evaluate the quality of the recommendations using the metrics explained before. One of the advantages of this evaluation is the quickness analysing large numbers of users with a low cost [19] [44].

In this thesis we use offline evaluation to find the parameters that characterise the algorithms to obtain the best recommendations.

4.1.1 Evaluation datasets

In this section, we talk in detail about three datasets that were used in the experimental part of this thesis. The datasets corresponds to the movies domain since it is an area with diverse data sources available. Consequently, we have different sources so that we only need to integrate one of the existing dataset into our system. Another positive point is the general knowledge that every user has about the film industry, which let them have a good knowledge about this domain without being an expert. This makes the use of the system and the evaluation of the results easy. In these datasets user preferences are provided in form of ratings [19] [39].

Below we will talk in detail about the configuration of our experiments and how we have carried them out.

The first stage of this research project was the analysis of six traditional groups of recommended algorithms in order to identify suitable characteristics for each one.

We have used one of the most popular publicly available datasets. This is from GroupLens and is called MovieLens dataset. “GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities specializing in recommender systems, online communities, mobile and ubiquitous technologies, digital libraries, and local geographic information systems” [16].

“GroupLens Research has collected and made available rating datasets from the MovieLens web site (http://movielens.org)” [17]. We are going to use these three dataset with different sizes.

- The 100K dataset has 100 000 ratings, 963 users and 1682 movies with a density of 6.30%. The data was collected between September 1997 and April 1998.

- The 1M dataset has one million ratings, 6040 users and 3900 movies with a density of 4.25%. The data was collected in 2000.

- The 10M dataset has 10 000 054 ratings, 71567 users and 10681 movies (with 95580 tags) with a density of 1.31%.

All the users of these datasets have rated a minimum of 20 movies. The ratings are on a 5-likert scale.

4.1.2 Offline Experiment

We began with the 100K dataset from MovieLens and then divided our dataset into test and training at an 80% to 20% ratio: 80% of the ratings were put into the training dataset. For the remaining 20% we removed one rating randomly.

The training data is given to the recommender as input for the algorithm and it generates recommendations. The test data (which is not seen by the recommender) is used as a ground truth to check the consistency of the recommendations with the ratings hidden to the recommender, calculating the metrics of the recommender´s output.

We performed a 5-fold cross validation for this experiment assigning data randomly to either the test or training datasets.

4.1.2.1 Offline Experiment Algorithms

The six groups of algorithms mentioned in the previous chapter were used in this experiment: Lucene, User-User, Item-Item, SVD, Personalized Mean and Popular.

4.1.2.2 Offline Experiment Metrics

We used three metrics: accuracy, rank, and entropy defining rank as the position of the rating in the filtered recommendation list.

“Rank is a proxy for user utility, since users prefer to find relevant results earlier” [32]. One of the measures of accuracy most used is the Root Mean Squared Error (RMSE). This understands ratings as interval data. That is to say a 5 star movie is rated higher than a 4 star movie which in turn is ranked higher than a 3 star movie. However, this assumption is not totally correct since our data is ordinal and the distance between two points is not always the same [1]. For this reason is not a suitable tool to measure the quality of the recommender.

As Xavier Amatriain said in his blog post [2], rank-based evaluations such as normalized discounted cumulative gain (nDCG) measure the ability of the recommender algorithms with the accurate model of user preferences more accurately than RMSE due to the fact that rank metrics do use interval data.

Entropy is used to understand the *Diversity* of the recommendations [31].

4.1.2.3 Offline Experiment Results

First of all we will evaluate the results obtained with the 100k dataset followed by the 1M dataset and finally the 10M dataset. With the latter we will work with our online evaluation.

The main aim of this offline evaluation is to find the best performance for each algorithm. In this way we look for the best neighbourhood size and correlation. We will focus on obtaining the highest possible value of the rank metric nDCG and also look for accuracy and diversity in terms of RMSE and entropy.

4.1.2.3.1 100k Dataset

Now, we will look at the performance of our six families of algorithms using the 100k dataset.

First of all, the hybrid filtering recommender *Lucene*. Then we will compare it also with the same algorithm but normalized using the 'BaselineSubtractingUserVectorNormalizer'.

- *Lucene*

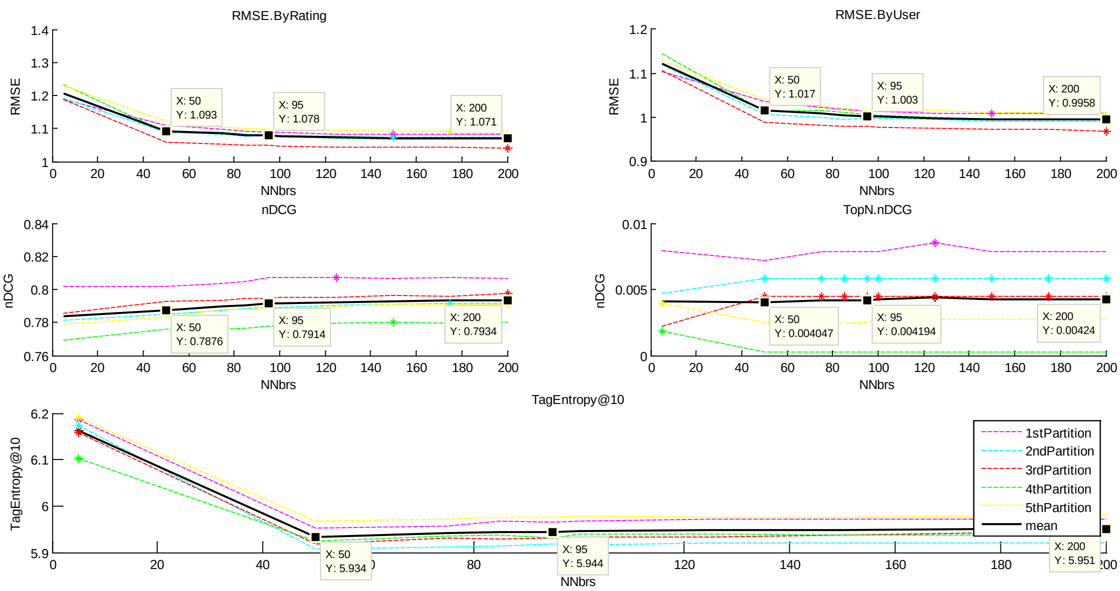


Figure 4-1: Lucene - 100k

- *Lucene Normalized*

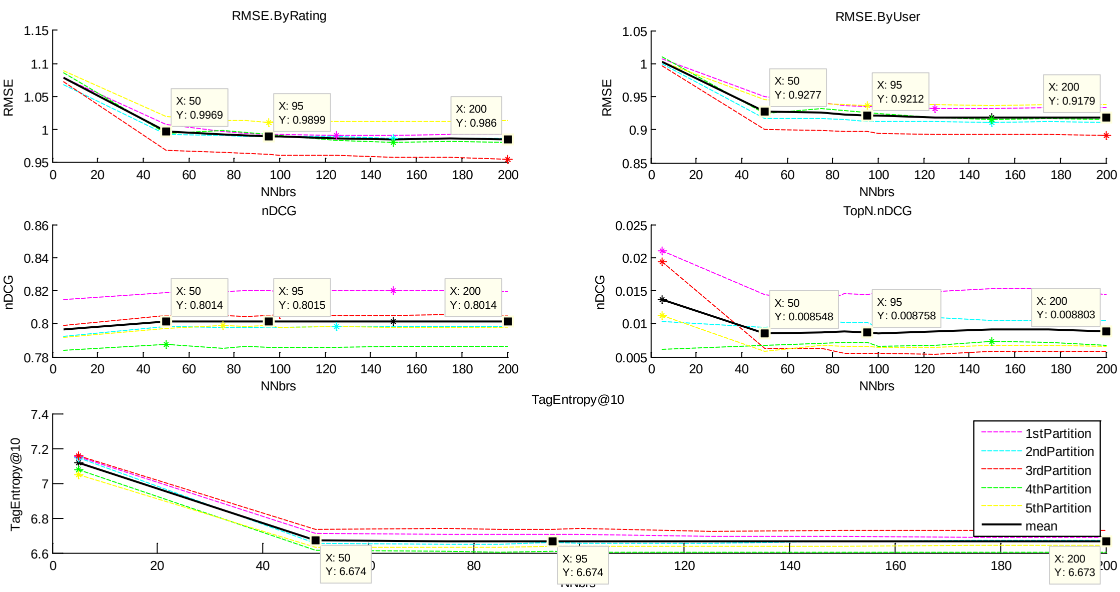


Figure 4-2: Lucene Normalized - 100k

4.1.2.3.1.1 Comparison between *Lucene* and *LuceneNormalized*:

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
LUCENE	50	1.093	1.017	0.7876	0.004047	5.934
***LUCENE*NORM**	50	0.9969	0.9277	0.8014	0.08548	6.674
LUCENE	95	1.078	1.003	0.7914	0.004194	5.944
***LUCENE*NORM**	95	0.9899	0.9212	0.8015	0.008758	6.674
LUCENE	200	1.071	0.9958	0.7934	0.00424	5.951
***LUCENE*NORM**	200	0.986	0.9179	0.8014	0.008803	6.673

Table 4-1: Comparison between Lucene and Lucene Normalized

We can see in Table 4-1 that *LuceneNormalized* gives us better results than *Lucene* across all the metrics and for every size of neighbourhood. The differences between each size of neighbourhood are very low. But looking at the normalized Discounted Cumulative Gain the best neighbourhood size could be 95.

Next, we will study the best performance of one of the collaborative filtering recommender families, *UserUser*. We will compare the results obtained using Pearson correlation (*UserUser*), then normalizing this algorithm (*UserUser Normalized*) and finally using Cosine correlation and normalizing (*UserUser Cosine*).

- *UserUser*

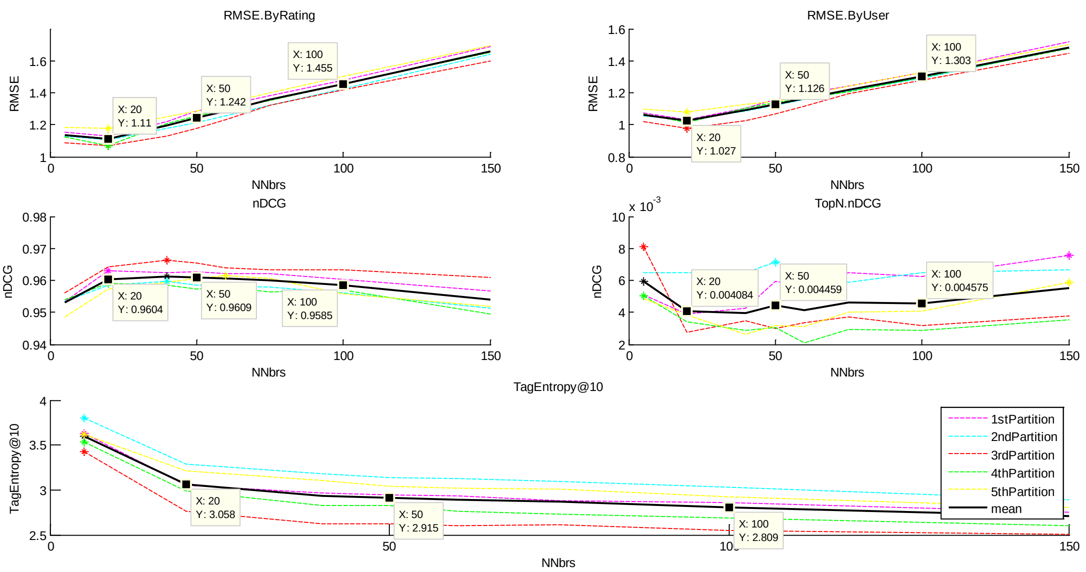


Figure 4-3: UserUser - 100k

- *UserUser Normalized*

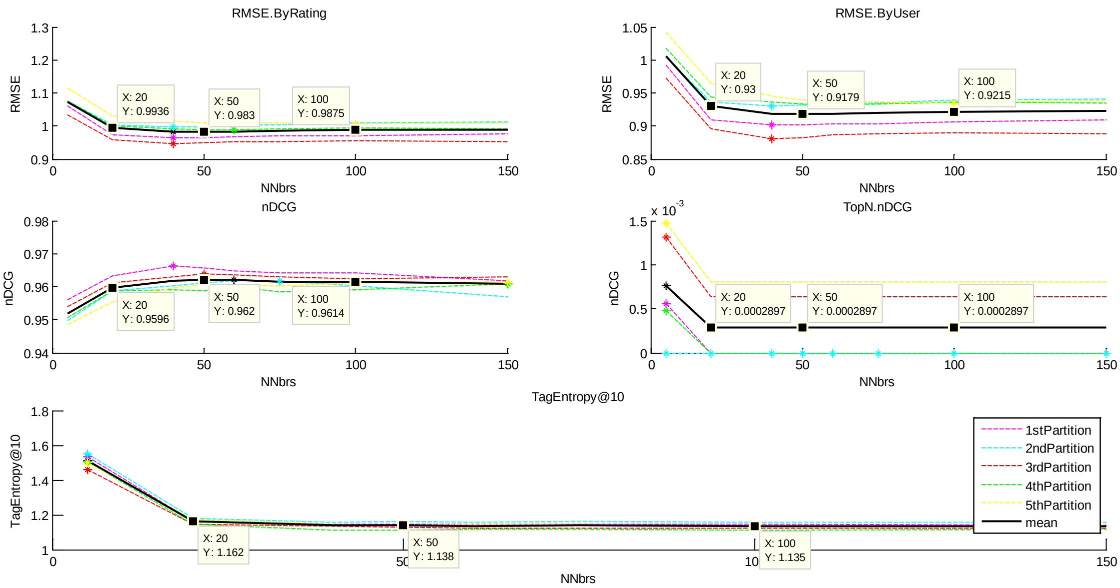


Figure 4-4: UserUser Cosine

- *UserUser Cosine*

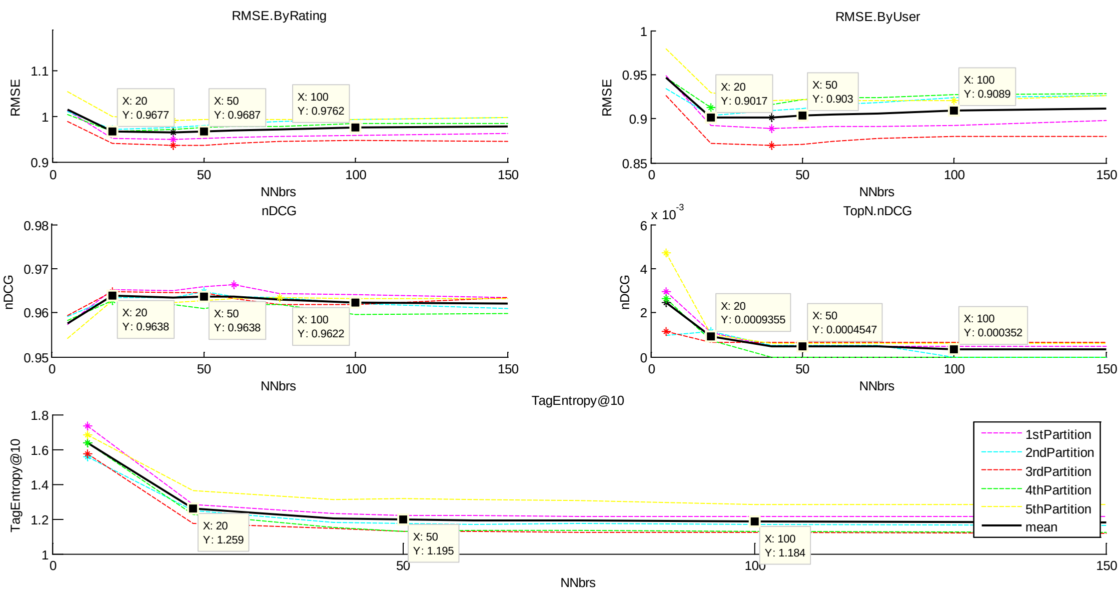


Figure 4-5: UserUser Cosine- 100k

4.1.2.3.1.2 Comparison between *UserUser*, *UserUserNorm* and *UserUserCosine*:

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
USERUSER	***20***	***1.11***	***1.027***	***0.9604***	***0.001084***	***3.058***
	50	***1.242***	***1.126***	***0.9609***	***0.004459***	***2.915***
	100	***1.455***	***1.303***	***0.9585***	***0.004575***	***2.809***
USERUSERNORM	***20***	***0.993***	***0.93***	***0.9596***	***0.0002897***	***1.162***
	50	***0.983***	***0.9179***	***0.962***	***0.0002897***	***1.138***
	100	***0.9875***	***0.9215***	***0.9614***	***0.0002897***	***1.135***
USERUSERCOSINE	***20***	***0.9677***	***0.9017***	***0.9638***	***0.0009355***	***1.259***
	50	***0.9687***	***0.903***	***0.9638***	***0.0004547***	***1.195***
	100	***0.9762***	***0.9089***	***0.9622***	***0.000352***	***1.184***

Table 4-2: Comparison among UserUser, UserUser Normalized and UserUser Cosine

Looking at Table 4-2, we can see that *UserUserCosine* gives us the best results in all the metrics, and the best neighbourhood size is 20.

If now we analyse the results obtained with the collaborative filtering by matrix factorization family algorithm, Single Value Decomposition. We will see the differences observed depending on the baseline taken into consideration.

- *SVD* Global Mean

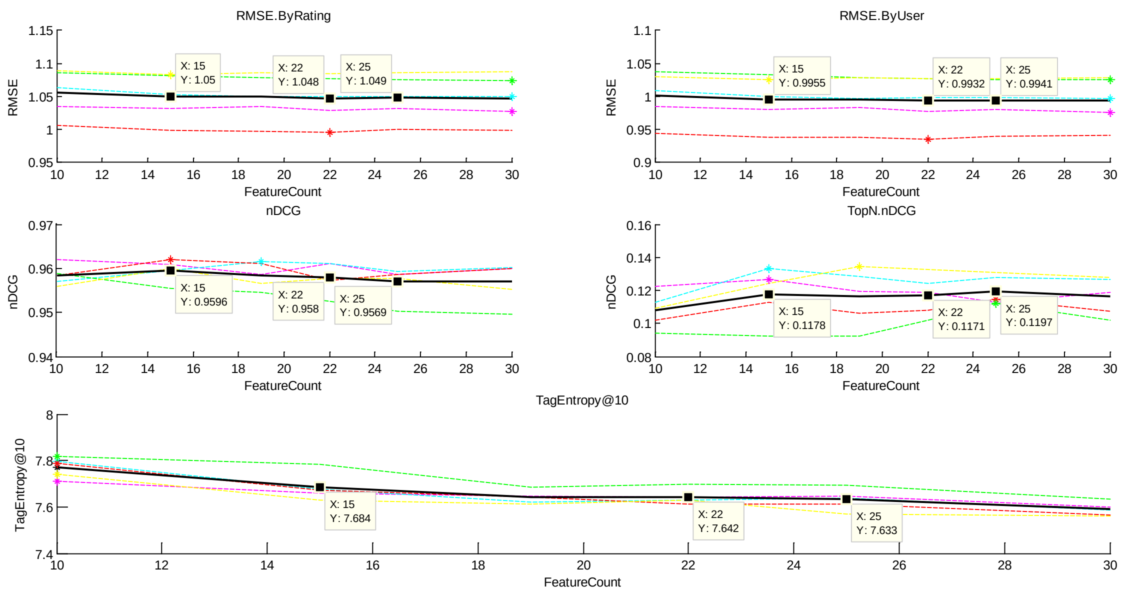


Figure 4-6: SVD Global Mean - 100k

- *SVD* Item Mean

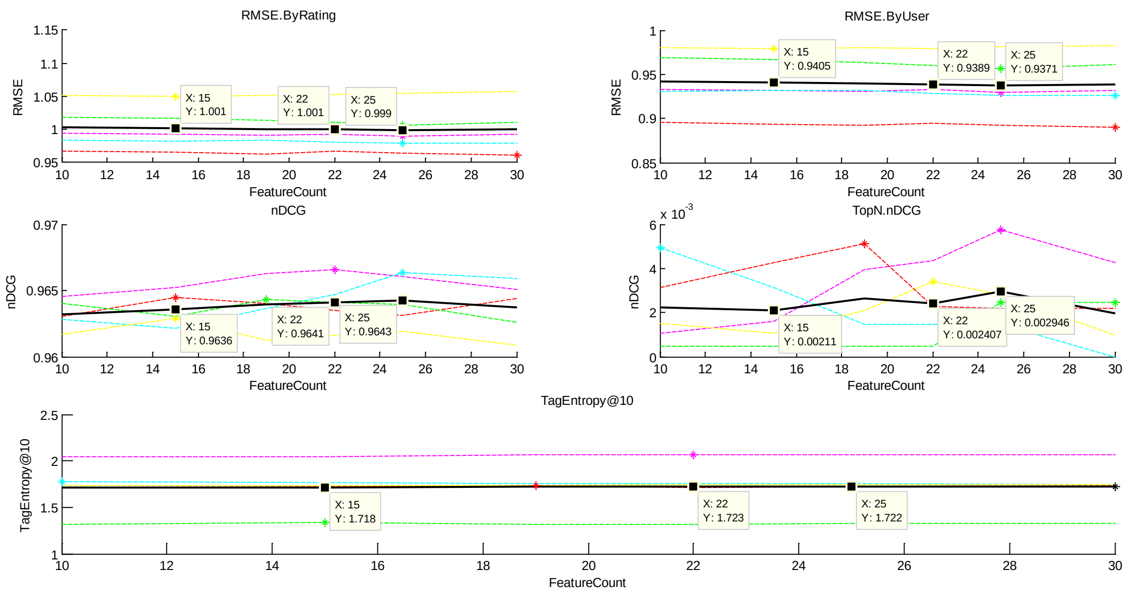


Figure 4-7: SVD Item Mean - 100k

- *SVD* Personalized *Mean*

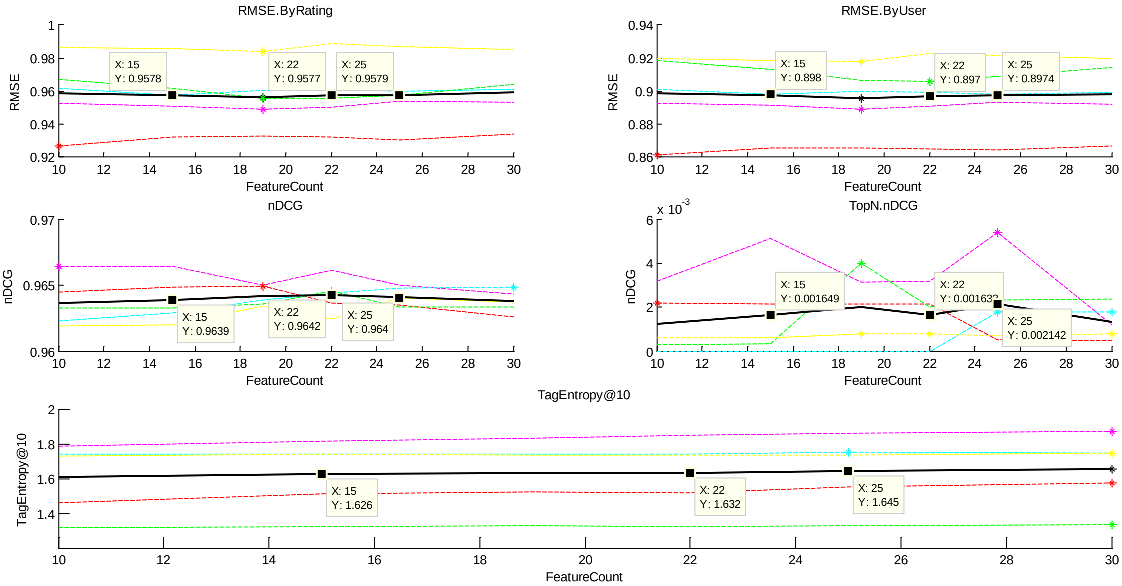


Figure 4-8: SVD Personalized Mean - 100k

- *SVD* User Mean

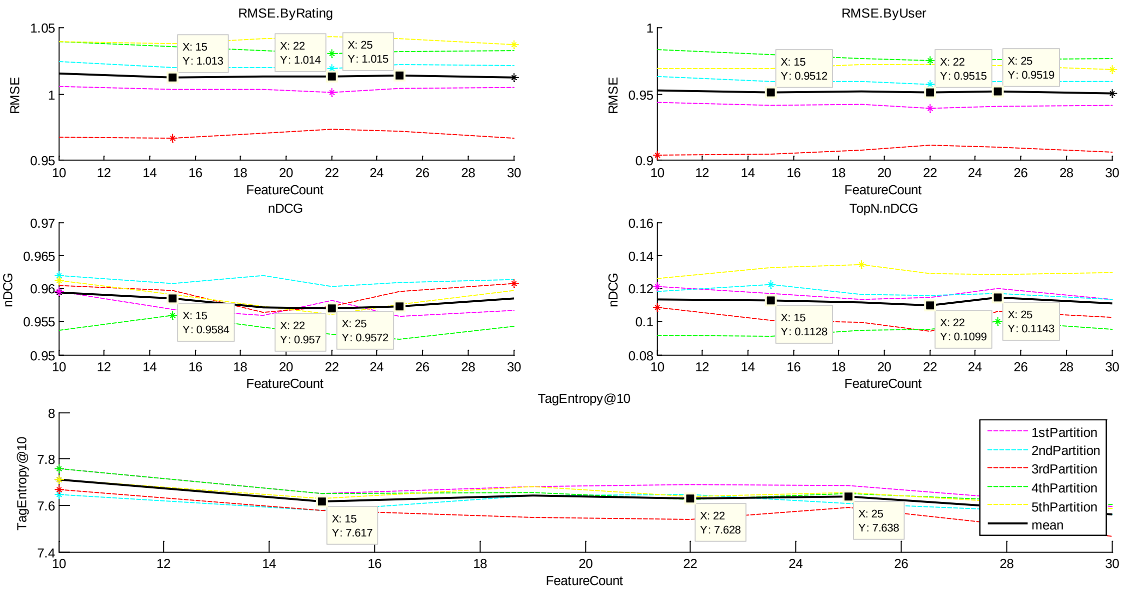


Figure 4-9: SVD User Mean - 100k

4.1.2.3.1.3 Comparison between *SVD*GlobalMean, *SVD*ItemMean, *SVDPersmean*, *SVD*UserMean:

ALGORITHM	FEATURE COUNT	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
SVDGLOBALMEAN	***15***	***1.05***	***0.9955***	***0.9596***	***0.1178***	***7.684***
SVDITEMMEAN	***15***	***1.001***	***0.9405***	***0.9636***	***0.00211***	***1.718***
SVDPERSMEAN	***15***	***0.9578***	***0.898***	***0.9639***	***0.001649***	***1.626***
SVDUSERMEAN	***15***	***1.013***	***0.9512***	***0.9584***	***0.1128***	***7.617***
SVDGLOBALMEAN	***22***	***1.048***	***0.9932***	***0.958***	***0.1171***	***7.642***
SVDITEMMEAN	***22***	***1.001***	***0.9389***	***0.9641***	***0.002407***	***1.723***
SVDPERSMEAN	***22***	***0.9577***	***0.897***	***0.9642***	***0.001633***	***1.632***
SVDUSERMEAN	***22***	***1.014***	***0.9515***	***0.957***	***0.1099***	***7.628***
SVDGLOBALMEAN	***25***	***1.049***	***0.9941***	***0.9569***	***0.1197***	***7.633***
SVDITEMMEAN	***25***	***0.999***	***0.9371***	***0.9643***	***0.002946***	***1.722***
SVDPERSMEAN	***25***	***0.9579***	***0.8974***	***0.964***	***0.002142***	***1.645***
SVDUSERMEAN	***25***	***1.015***	***0.9519***	***0.9572***	***0.1143***	***7.638***

Table 4-3: Comparison among SVDGlobalMean, SVDItemMean, SVDPersmean, and SVDUserMean

Table 4-3 shows the results obtained. We can see that *SVDPersmean* gives the best results in terms of RMSE and nDCG. The best neighbourhood size for this algorithm is 22, because the differences are very small and for this neighbourhood size we have obtained the best results for nDCG.

Next we are going to look at another family of collaborative filtering recommenders, *ItemItem*. In this case is normalized using the “MeanCenteringVectorNormalizer”.

- ItemItem:

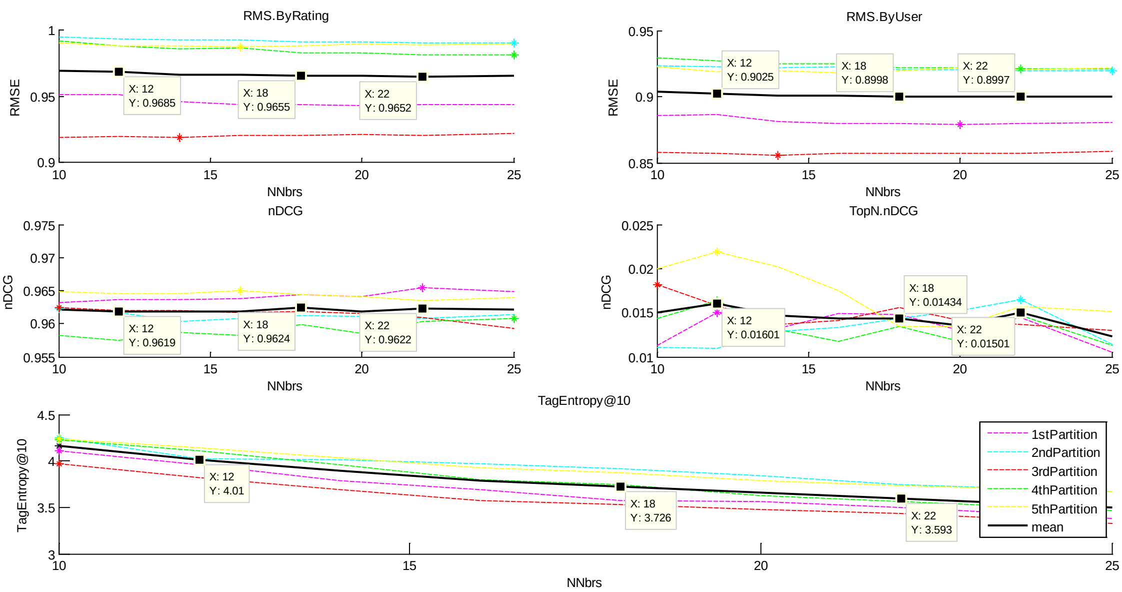


Figure 4-10: ItemItem - 100k

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
ITEMITEM	***12***	***0.9685***	***0.9025***	***0.9619***	***0.01601***	***4.01***
ITEMITEM	***18***	***0.9665***	***0.8998***	***0.9624***	***0.01434***	***3.726***
ITEMITEM	***22***	***0.9652***	***0.897***	***0.9622***	***0.01501***	***3.593***

Table 4-4: Comparison ItemItem for different Neighbourhood sizes

Looking at the results in Table 4-4, the best neighbourhood size for *ItemItem* is 18 because we have obtained the best results for nDCG and RMSE.

And finally we have analysed the performance of two basic algorithms: *Personalized Mean* and *Popular*.

- *Personalized Mean*

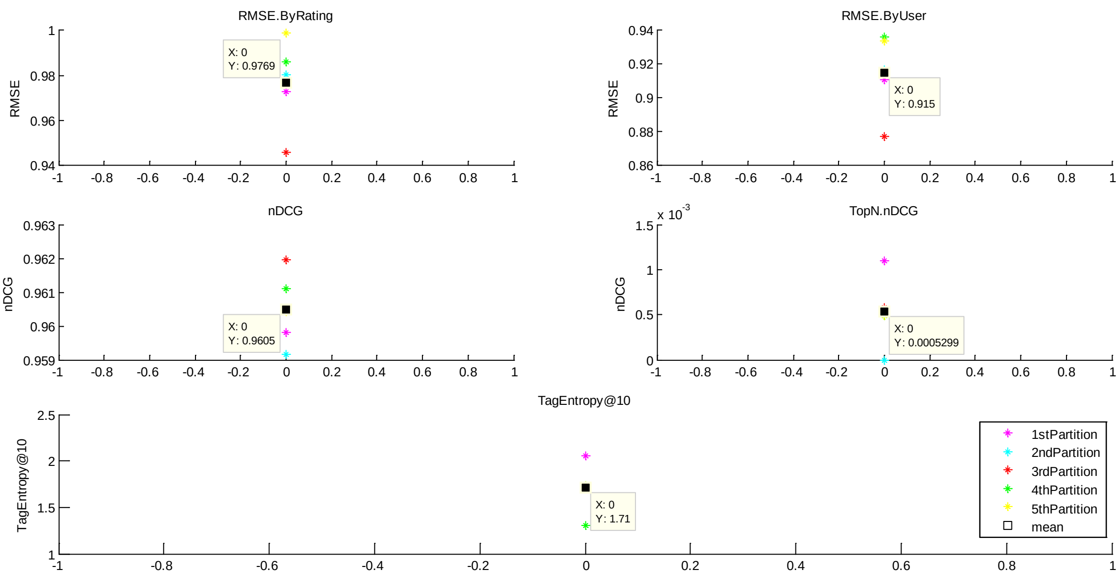


Figure 4-11: Personalized Mean - 100k

- *Popular*

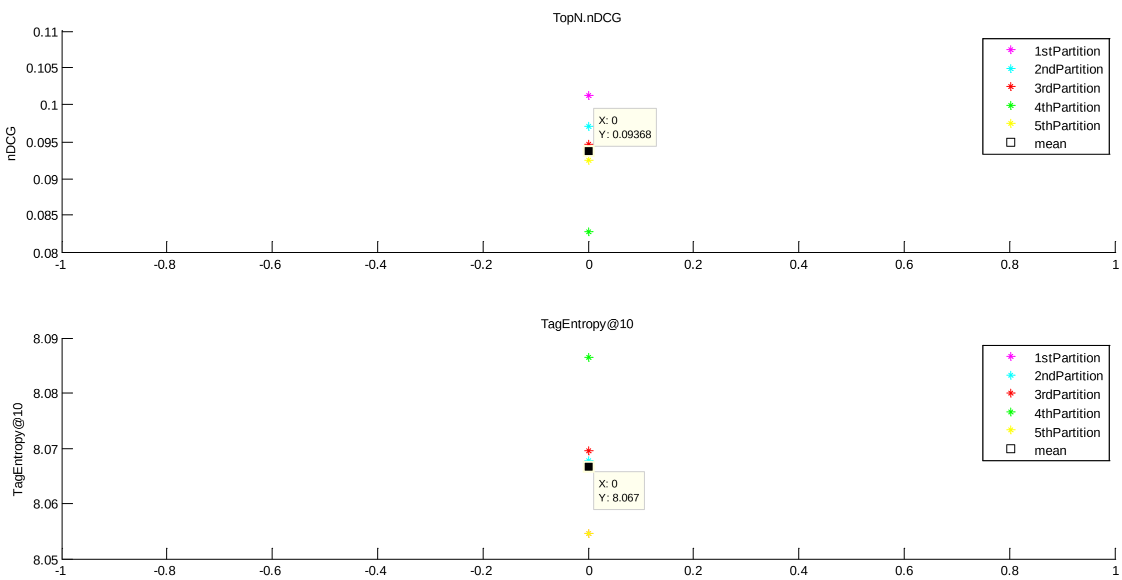


Figure 4-12: Popular - 100k

4.1.2.3.1.4 1 Million Dataset

We will follow the same structure as with the 100k dataset. We want to see whether the increased size of the dataset has influenced the algorithms performance, and whether the optimal neighbourhood size has changed.

First we are going to focus on the results of *Lucene* family.

- *Lucene*

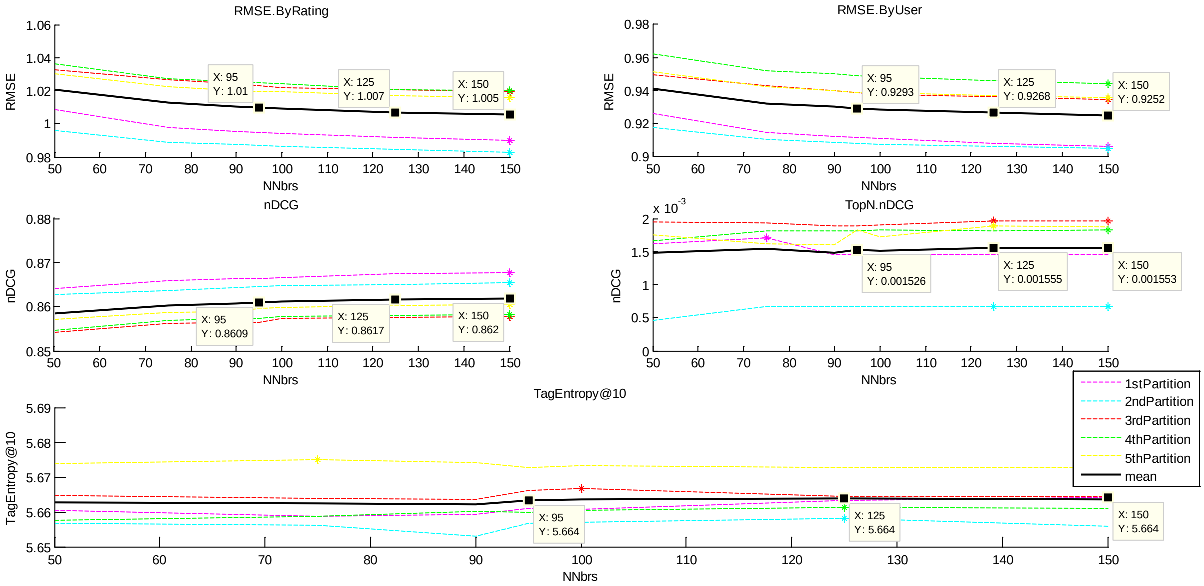


Figure 4-13: Lucene - 1M

- *Lucene Normalized*

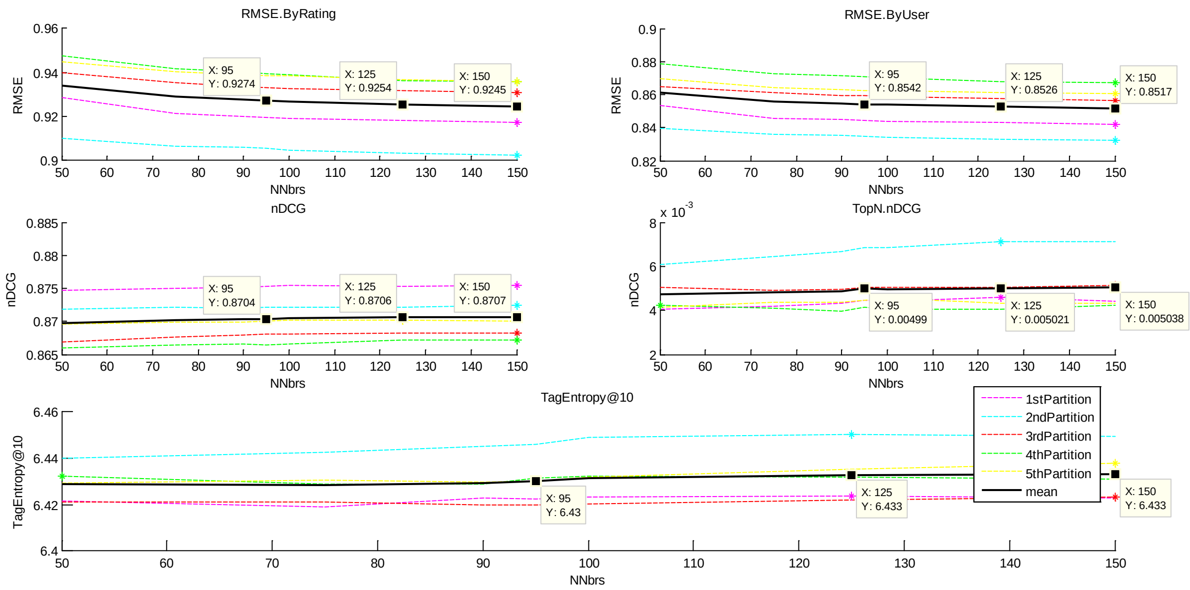


Figure 4-14: Lucene Normalized - 1M

4.1.2.3.1.5 Comparison between *Lucene* and *LuceneNorm*:

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
LUCENE	***95***	***1.01***	***0.9293***	***0.8609***	***0.001526***	***5.664***
LUCENENORM	***95***	***0.9274***	***0.8542***	***0.8704***	***0.00499***	***6.43***
LUCENE	***125***	***1.007***	***0.9268***	***0.8617***	***0.00155***	***5.664***
LUCENENORM	***125***	***0.9254***	***0.8526***	***0.8706***	***0.00502***	***6.433***
LUCENE	***150***	***1.005***	***0.9252***	***0.862***	***0.00155***	***5.664***
LUCENENORM	***150***	***0.9245***	***0.8517***	***0.8707***	***0.00503***	***6.433***

Table 4-5: Comparison between Lucene and Lucene Normalized

We can see in Table 4-5 that *LuceneNorm* gives us better results than *Lucene* across all metrics and for every size of neighbourhood.

The best neighbourhood size is 95. As we can see the neighbourhood size has increased from 50 to 95 in this 1M dataset compared to the 100k dataset.

Next, we compare the results from the User family of algorithms.

- *UserUser*

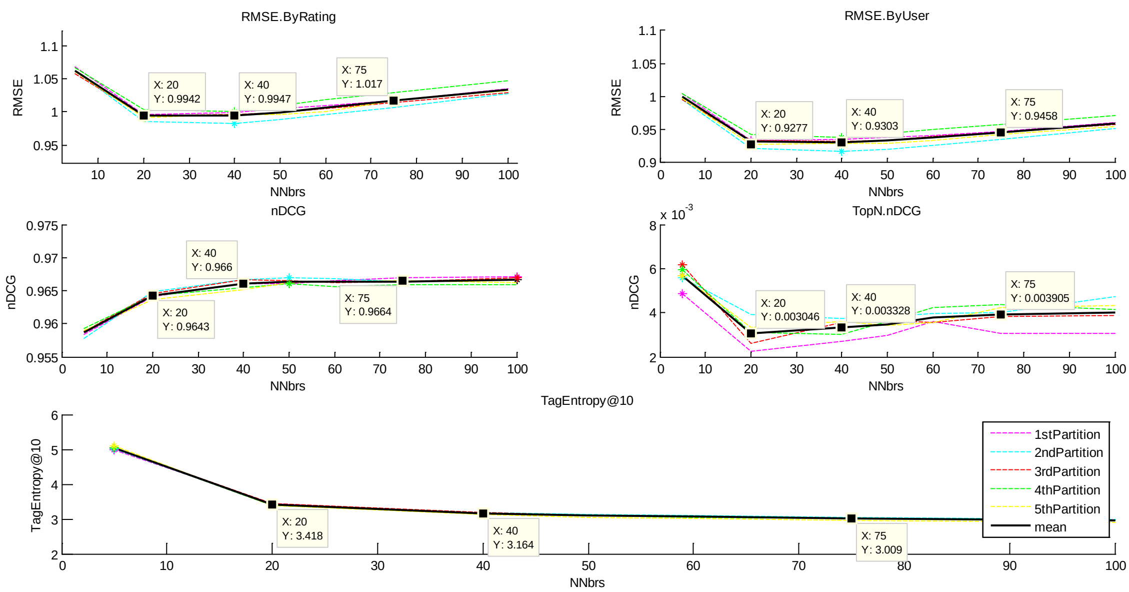


Figure 4-15: UserUser - 1M

- *UserUser Normalized*

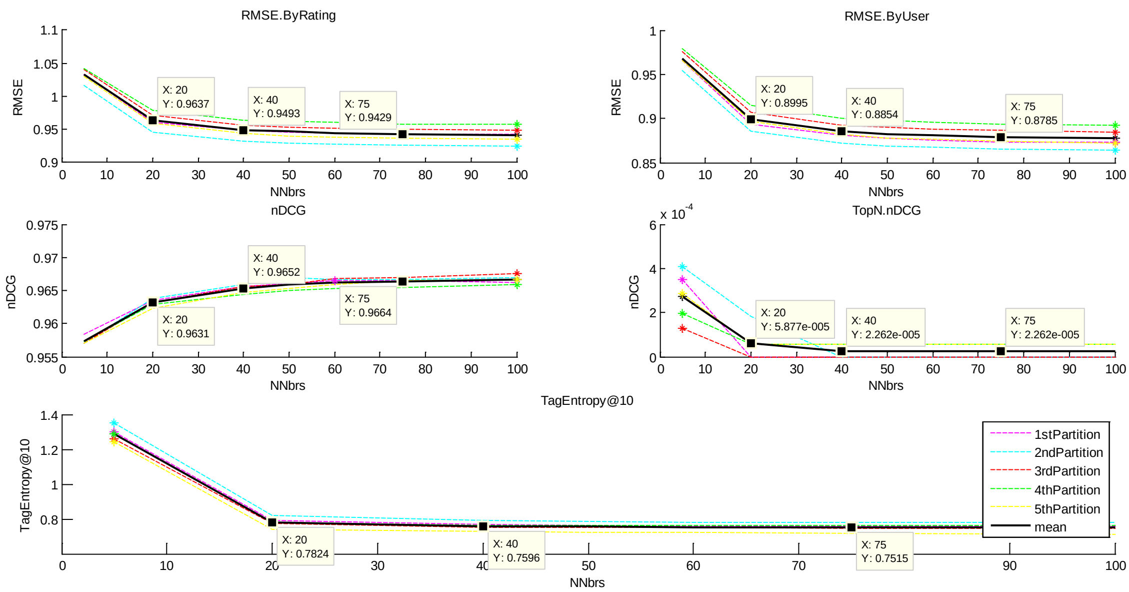


Figure 4-16: UserUser Normalized - 1M

- *UserUser Cosine*

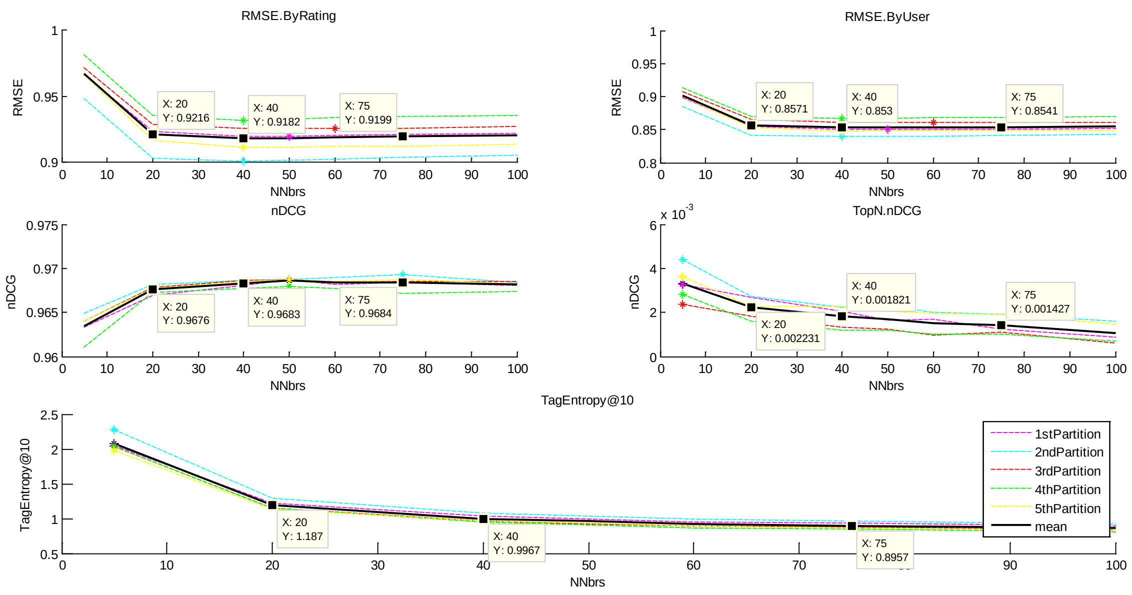


Figure 4-17: UserUser Cosine - 1M

4.1.2.3.1.6 Comparison between *UserUser*, *UserUserNorm* and *UserUserCosine*:

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
USERUSER	*20*	*0.9942*	*0.9277*	*0.966*	*0.003046*	*3.418*
USERUSER	*40*	*0.9947*	*0.9303*	*0.9643*	*0.003328*	*3.164*
USERUSER	*75*	*1.017*	*0.9458*	*0.9664*	*0.003905*	*3.009*
USERUSERNORM	*20*	*0.9637*	*0.8995*	*0.9631*	*5.877E-5*	*0.7824*
USERUSERNORM	*40*	*0.9493*	*0.8854*	*0.9652*	*2.262E-5*	*0.7596*
USERUSERNORM	*75*	*0.9429*	*0.8785*	*0.9664*	*2.262E-5*	*0.7515*
USERUSERCOSINE	*20*	*0.9216*	*0.8571*	*0.9676*	*0.002231*	*1.187*
USERUSERCOSINE	*40*	*0.9182*	*0.853*	*0.9683*	*0.001821*	*0.9967*
USERUSERCOSINE	*75*	*0.9199*	*0.8541*	*0.9684*	*0.001427*	*0.8957*

Table 4-6: Comparison among UserUser, UserUser Normalized and UserUser Cosine

Looking at Table 4-6 we can see that *UserUserCosine* gives us the best results in all the metrics except for TopN nDCG, where the best results are given by *UserUser*, but the differences are very small. The best neighbourhood size for *UserUserCosine* is 20. In this case, for this algorithm, the best neighbourhood size is the same as in the 100k dataset.

The following is the study of the performance of the family of the collaborative filtering by matrix factorization based algorithm *SVD*. Due to the fact of the high computational cost of this algorithm, we were forced to reduce the crossfold validation from five to only two partitions.

- *SVD* Global Mean

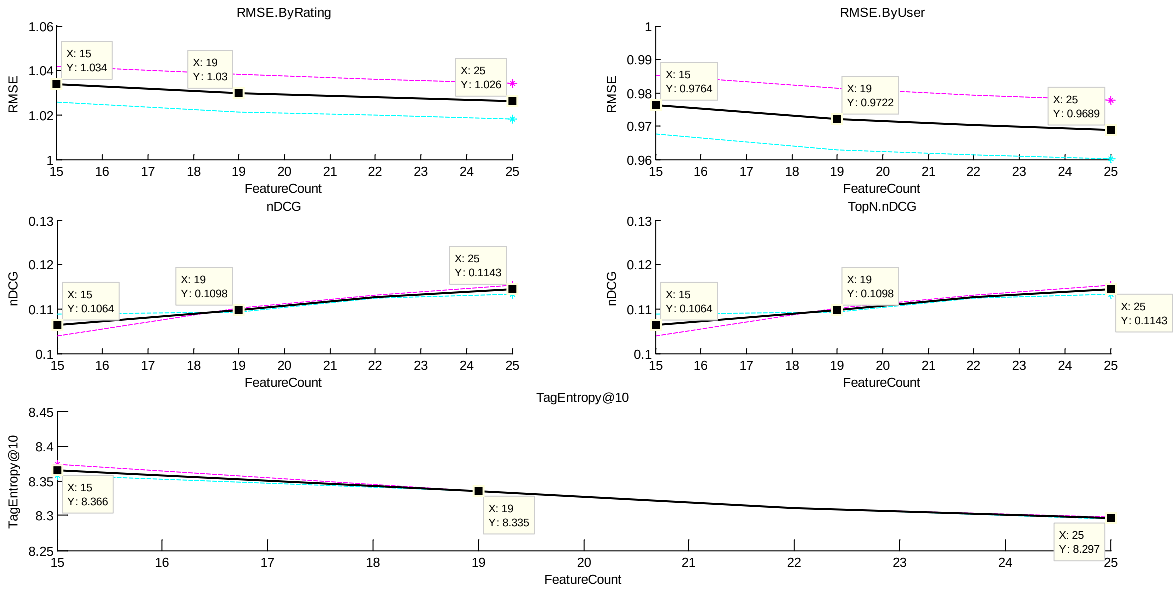


Figure 4-18: SVD Global Mean -1M

- *SVD* Item Mean

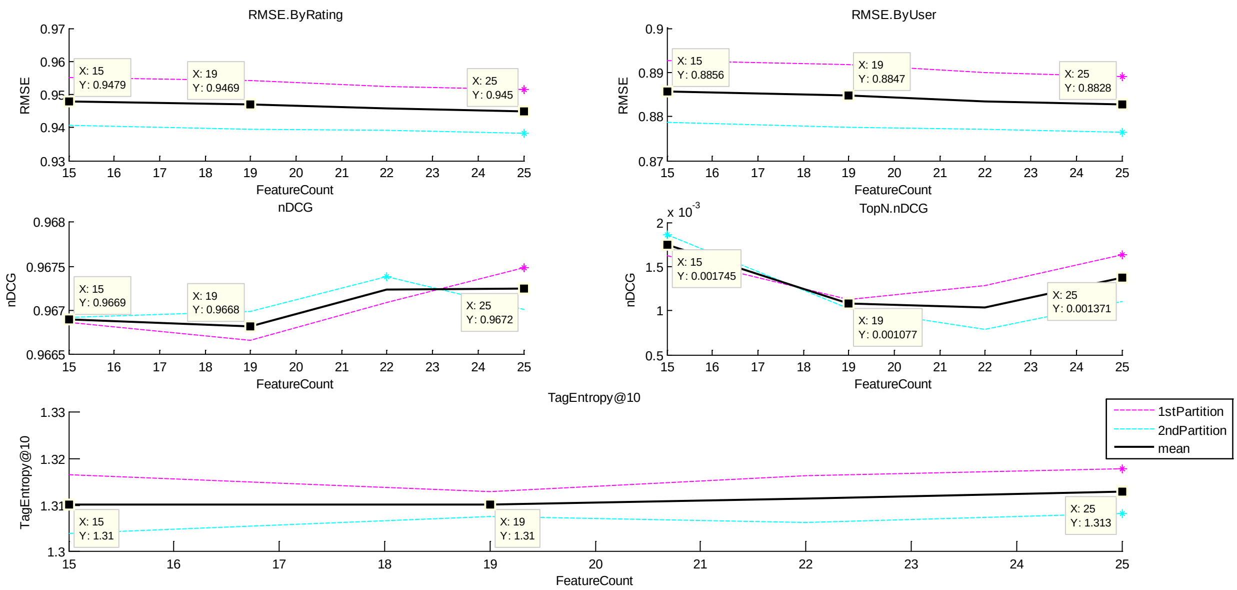


Figure 4-19: SVD Item Mean - 1M

- *SVD Personalized Mean*

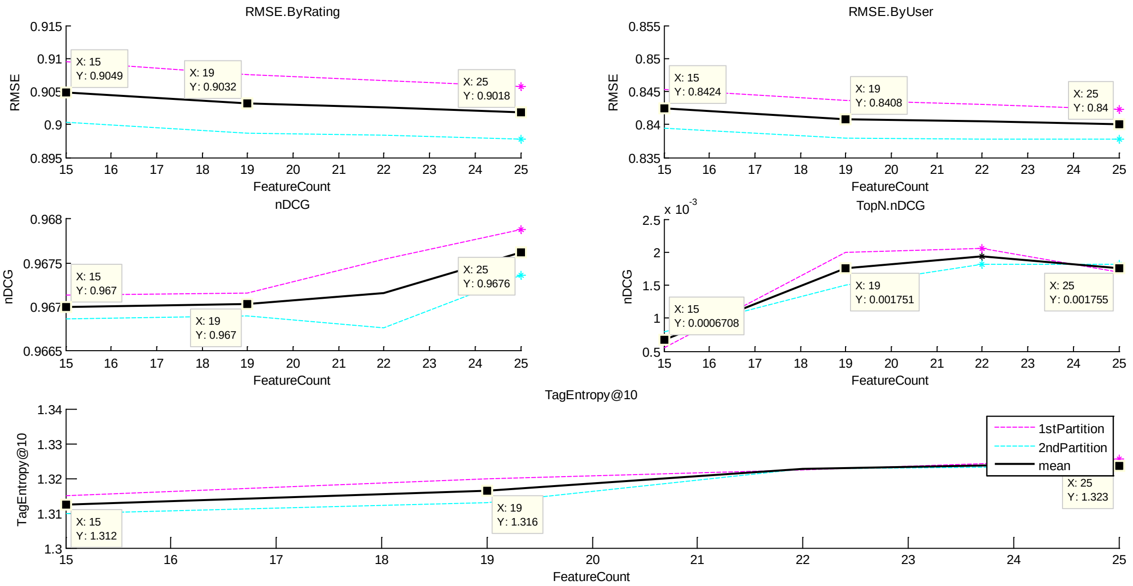


Figure 4-20: SVD Personalized Mean - 1M

- *SVD* User Mean

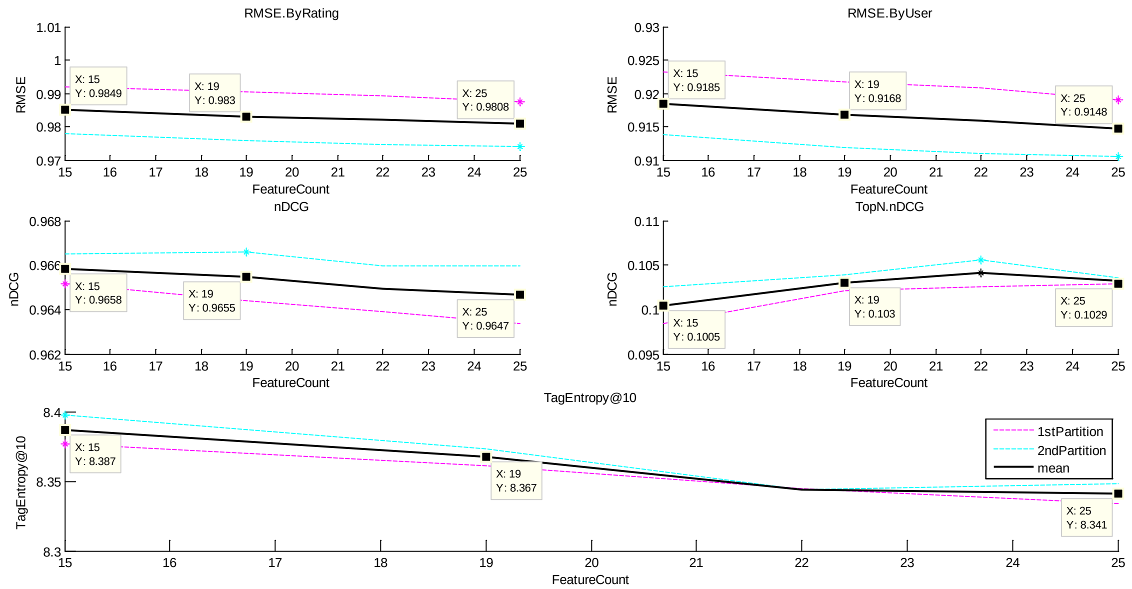


Figure 4-21: SVD User Mean - 1M

4.1.2.3.1.7 Comparison between *SVD*GlobalMean, *SVD*ItemMean, *SVDPersmean*, *SVD*UserMean:

ALGORITHM	FEATURE COUNT	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
SVDGLOBALMEAN	***15***	***1.034***	***0.9764***	***0.1064***	***0.1064***	***8.366***
SVDITEMMEAN	***15***	***0.9479***	***0.8855***	***0.9669***	***0.001077***	***1.31***
SVDPERSMEAN	***15***	***0.9049***	***0.8424***	***0.967***	***0.0006708***	***1.312***
SVDUSERMEAN	***15***	***0.9849***	***0.9185***	***0.9658***	***0.1005***	***8.387***
SVDGLOBALMEAN	***19***	***1.03***	***0.9722***	***0.1098***	***0.1098***	***8.335***
SVDITEMMEAN	***19***	***0.9469***	***0.8847***	***0.9668***	***0.001077***	***1.31***
SVDPERSMEAN	***19***	***0.9032***	***0.8408***	***0.967***	***0.001751***	***1.316***
SVDUSERMEAN	***19***	***0.983***	***0.9168***	***0.9655***	***0.103***	***8.367***
SVDGLOBALMEAN	***25***	***1.026***	***0.9689***	***0.1143***	***0.1143***	***8.297***
SVDITEMMEAN	***25***	***0.945***	***0.8828***	***0.9672***	***0.001371***	***1.313***
SVDPERSMEAN	***25***	***0.9018***	***0.84***	***0.9676***	***0.001755***	***1.323***
SVDUSERMEAN	***25***	***0.9808***	***0.9148***	***0.9647***	***0.1029***	***8.341***

Table 4-7: Comparison among SVDGlobalMean, SVDItemMean, SVDPersmean, and SVDUserMean

Table 4-7, shows the results obtained. We can see that *SVDPersmean* gives the best results in terms of RMSE and nDCG. The best neighbourhood size for this algorithm is 25. If we compare it with the result obtained on the 100k dataset, we will see that now the best neighbourhood size is almost the same.

Now we will discuss the results of the other collaborative filtering algorithm, *ItemItem*.

- ItemItem:

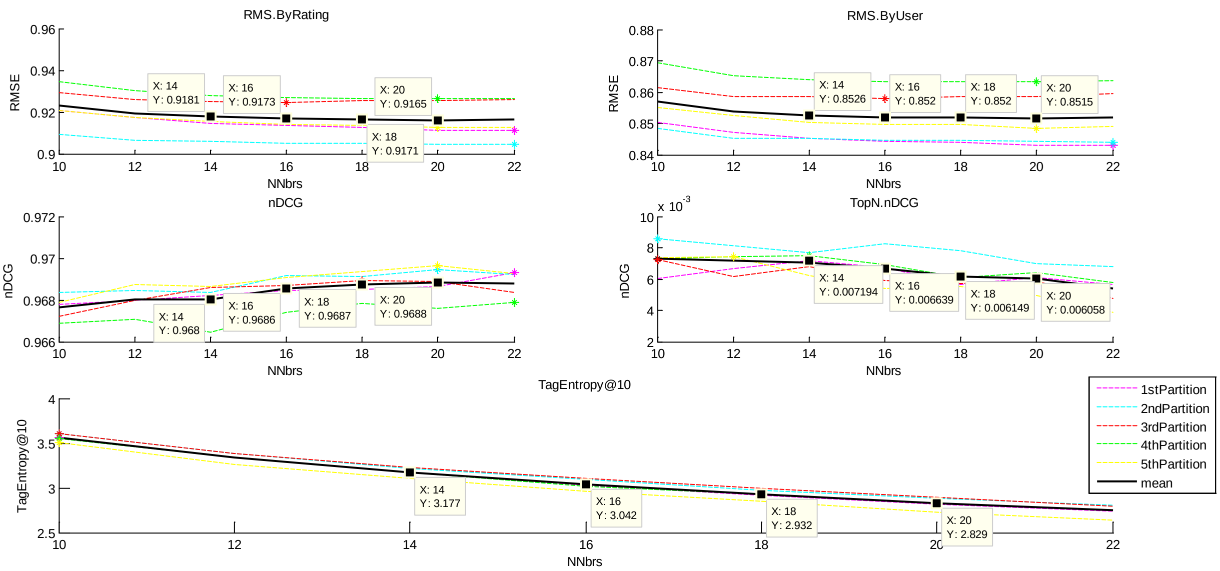


Figure 4-22: ItemItem - 1M

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
ITEMITEM	***14***	***0.9181***	***0.8526***	***0.968***	***0.007194***	***3.177***
ITEMITEM	***16***	***0.9173***	***0.852***	***0.9686***	***0.006639***	***3.042***
ITEMITEM	***18***	***0.9171***	***0.852***	***0.9687***	***0.006149***	***2.932***
ITEMITEM	***20***	***0.9165***	***0.8515***	***0.9688***	***0.006058***	***2.829***

Table 4-8: Comparison ItemItem for different sizes of neighbourhood

In this case (Table 4-8) the best results are obtained for a neighbourhood size of 20. The RMSE has the lowest value and the nDCG is higher than the other sizes. The results now are almost the same as in the 100k dataset.

And finally the two basics algorithms:

- *Personalized Mean*:

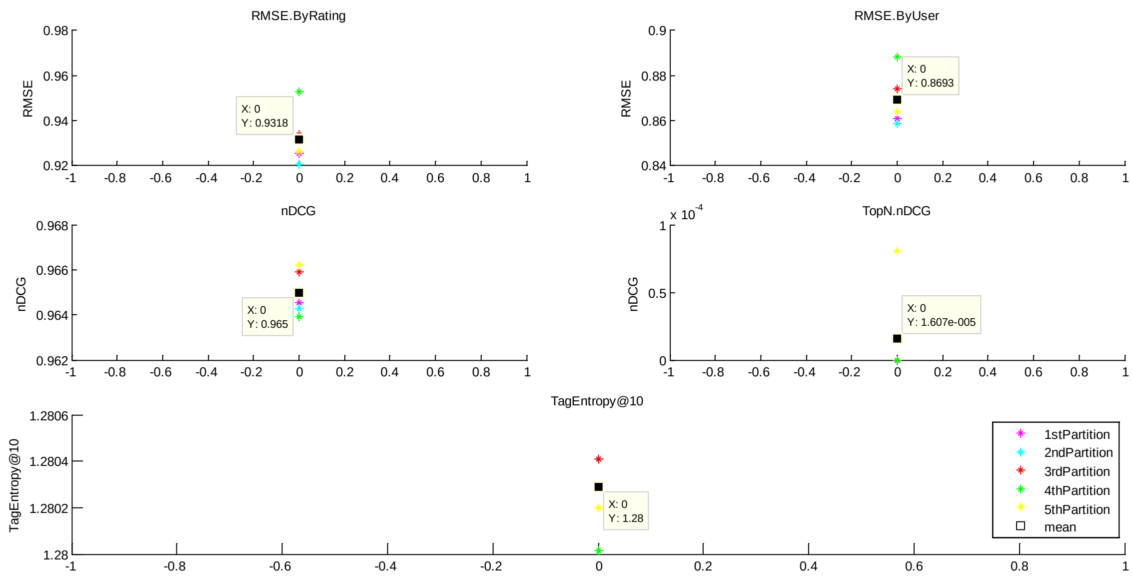


Figure 4-23: Personalized Mean - 1M

- *Popular*

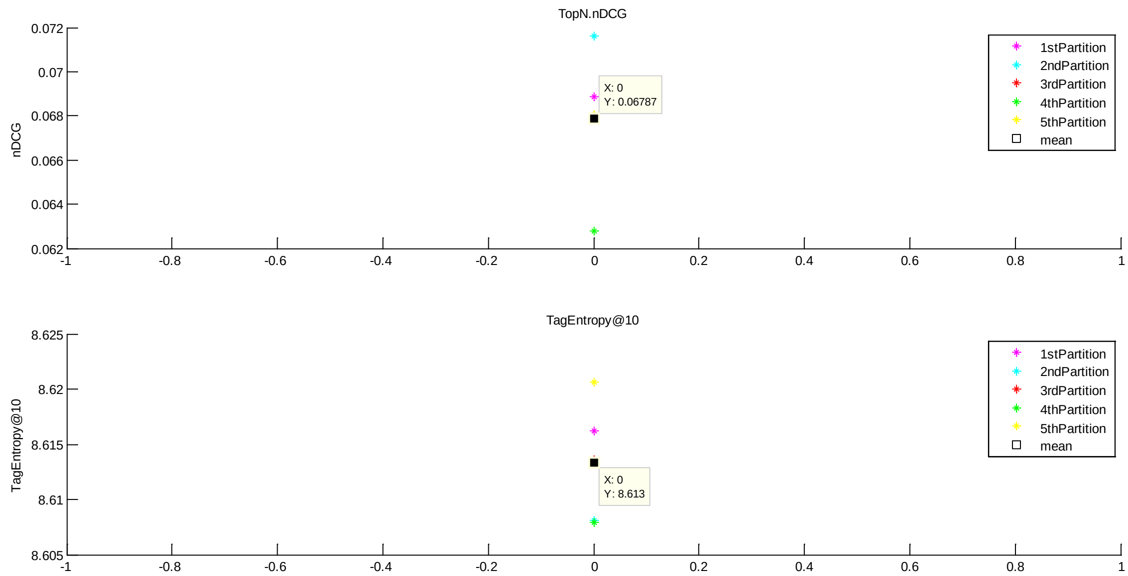


Figure 4-24: Popular - 1M

4.1.2.3.2 10 Million Dataset

Finally we have analysed the performance of our algorithms with the biggest dataset. The results obtained here will be extrapolated to the online evaluation. Once we know the optimum size of neighbourhood size or the optimum number of features, depending on the algorithm, we will add our users' ratings to this dataset to obtain recommendations for them. Looking at the hybrid filtering algorithms, we have obtained the next results:

- *Lucene*

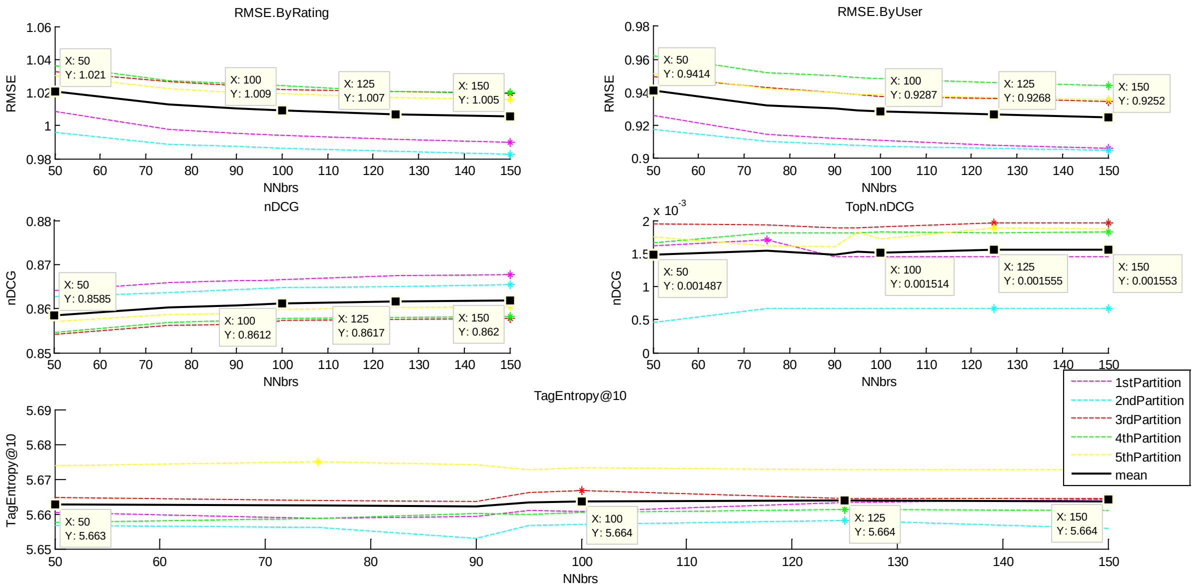


Figure 4-25: Lucene - 10M

- *Lucene Normalized*

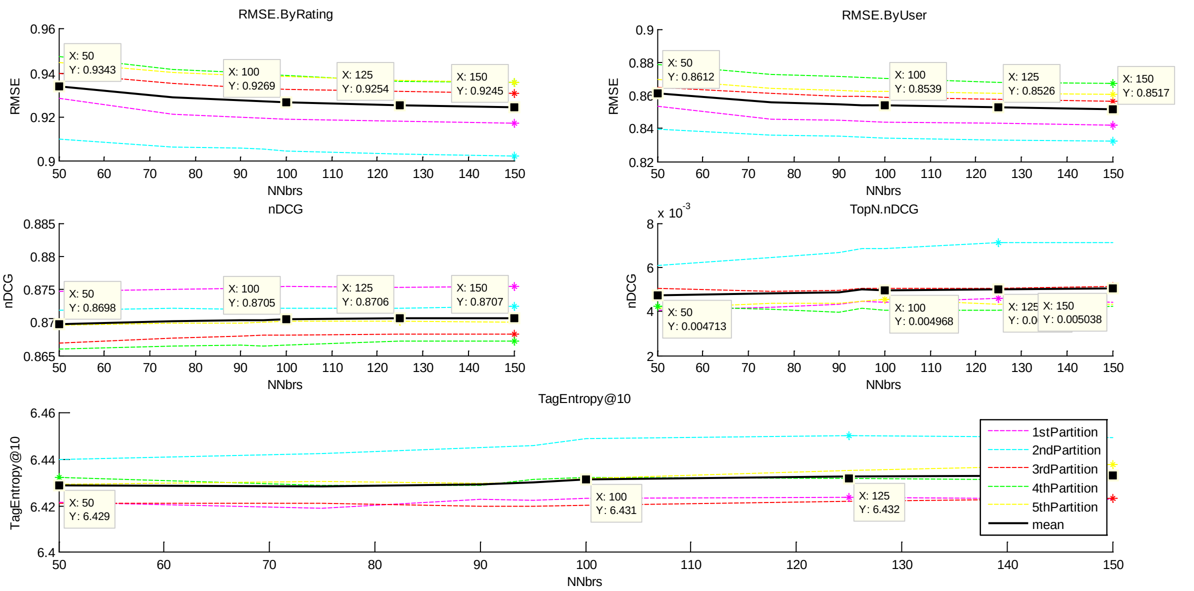


Figure 4-26: Lucene Normalized - 10 M

4.1.2.3.2.1 Comparison between *Lucene* and *LuceneNorm*:

ALGORITHM	NDCG	RMSE BY RATINGS	RMSE BY USERS	NDCG	TOPN NDCG	ENTROPY
LUCENE	***50***	***1.021***	***0.9414***	***0.8585***	***0.001487***	***5.663***
LUCENE NORM	***50***	***0.9343***	***0.8612***	***0.8698***	***0.004713***	***6.429***
LUCENE	***100***	***1.009***	***0.9287***	***0.8612***	***0.001514***	***5.664***
LUCENE NORM	***100***	***0.9269***	***0.8539***	***0.8705***	***0.004968***	***6.431***
LUCENE	***125***	***1.007***	***0.9268***	***0.8617***	***0.001555***	***5.664***
LUCENE NORM	***125***	***0.9254***	***0.8526***	***0.8706***	***0.005021***	***6.432***
LUCENE	***150***	***1.005***	***0.9252***	***0.862***	***0.001553***	***5.664***
LUCENE NORM	***150***	***0.9245***	***0.8517***	***0.8707***	***0.005038***	***6.433***

Table 4-9: Comparison between Lucene and Lucene Normalized

We can see in Table 4-9 the results for Lucene and Lucene Normalized, just looking at the mean of the five partitions when we use the 10M MovieLens dataset. We can find out that Lucene Normalized gives us the best results in regard to a higher nDCG and a lower RMSE than Lucene algorithm, also the entropy is higher what means a higher diversity.

The best neighbourhood size is 100, although the differences are very small, all the result are really close between 100 and 150. But just with 100 of neighbours we are obtaining good results.

Then looking at the results obtained with the collaborative filtering algorithm *UserUser*:

- *UserUser*

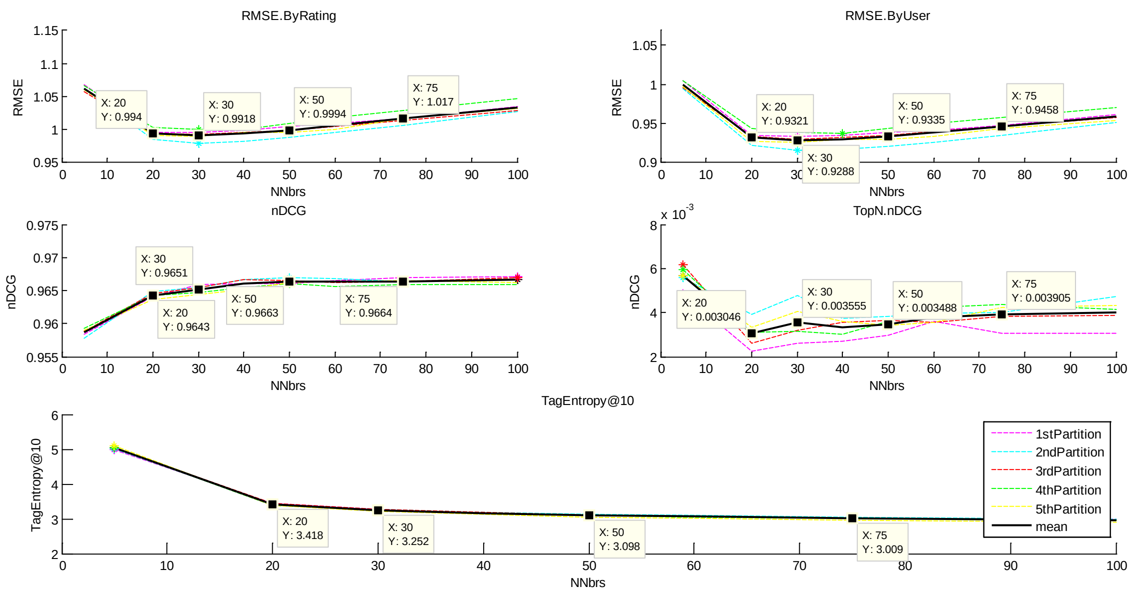


Figure 4-27: UserUser - 10M

- *UserUser Normalized*

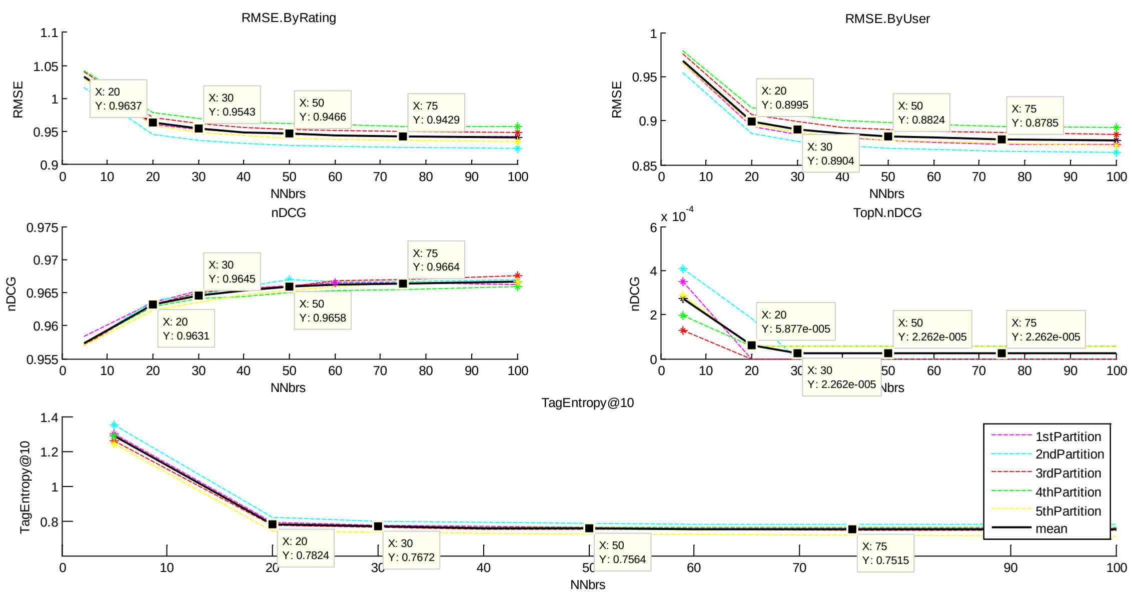


Figure 4-28: UserUser Normalized - 10M

- *UserUser Cosine*

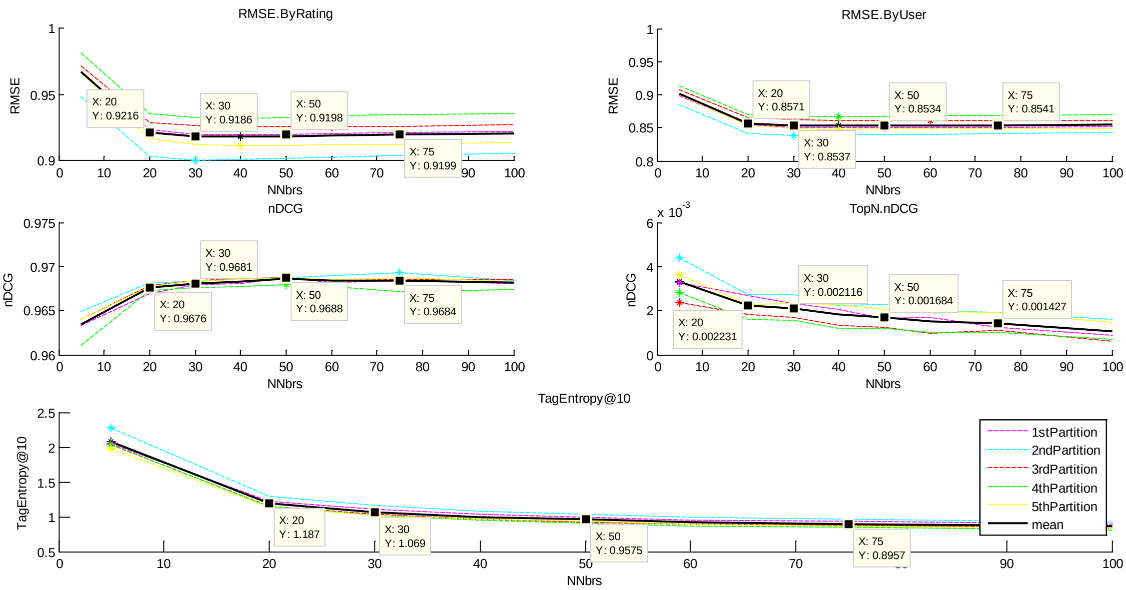


Figure 4-29: UserUser Cosine - 10M

4.1.2.3.2.2 Comparison among *UserUser*, *UserUserNorm* and *UserUserCosine*:

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
USERUSER	*20*	*0.994*	*0.9321*	*0.9643*	*0.003046*	*3.418*
USERUSERNORM	*20*	*0.9637*	*0.8995*	*0.9631*	*5.877E-5*	*0.7824*
USERUSERCOSINE	*20*	*0.9216*	*0.8571*	*0.9676*	*0.002231*	*1.187*
USERUSER	*30*	*0.9918*	*0.9288*	*0.9651*	*0.003555*	*3.252*
USERUSERNORM	*30*	*0.9543*	*0.8904*	*0.9645*	*2.262E-5*	*0.7672*
USERUSERCOSINE	*30*	*0.918*	*0.8537*	*0.9681*	*0.002116*	*1.069*
USERUSER	*50*	*0.9994*	*0.9335*	*0.9663*	*0.003488*	*3.098*
USERUSERNORM	*50*	*0.9466*	*0.8824*	*0.9658*	*2.262E-5*	*0.7564*
USERUSERCOSINE	*50*	*0.9198*	*0.8534*	*0.9688*	*0.001684*	*0.9575*
USERUSER	*75*	*1.017*	*0.9458*	*0.9664*	*0.003905*	*3.009*
USERUSERNORM	*75*	*0.9429*	*0.8785*	*0.9664*	*2.262E-5*	*0.7515*
USERUSERCOSINE	*75*	*0.9199*	*0.8541*	*0.9684*	*0.001427*	*0.8957*

Table 4-10: Comparison among UserUser, UserUser Normalized and UserUser Cosine

Looking at Table 4-10 we can see that *UserUserCosine* gives us the best results in all the metrics except for TopN nDCG, where the best results are given by *UserUser*, but the differences are very small. Once we have done the comparison, we can see that for a size of 50 neighbours we reach out the best results for the algorithm of UserUserCosine (looking at nDCG). So 50 is the best neighbourhood size.

Hereafter we can take a look at the results obtained from the *SVD* algorithms:

- *SVD* Global Mean

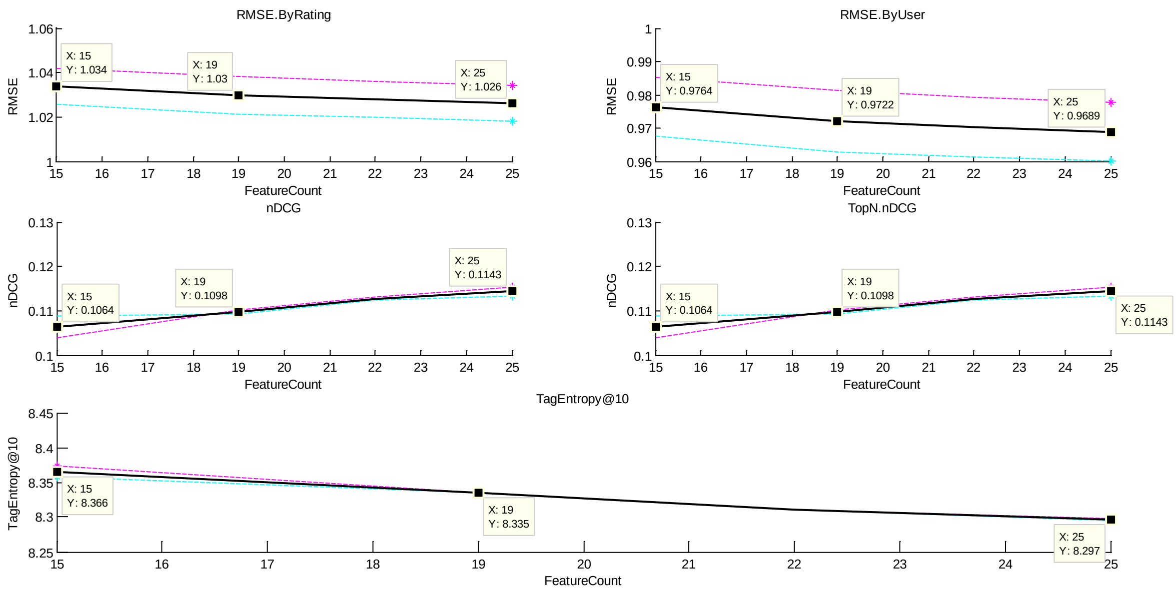


Figure 4-30: SVD Global Mean - 10M

- *SVD* Item Mean

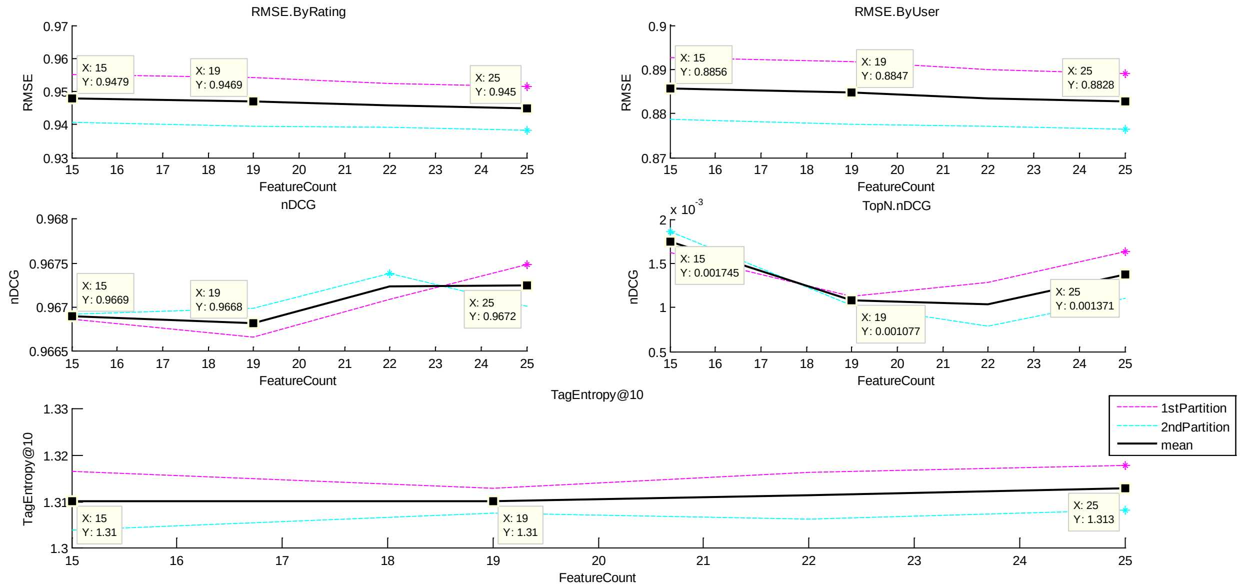


Figure 4-31: SVD Item Mean - 10M

- *SVD Personalized Mean*

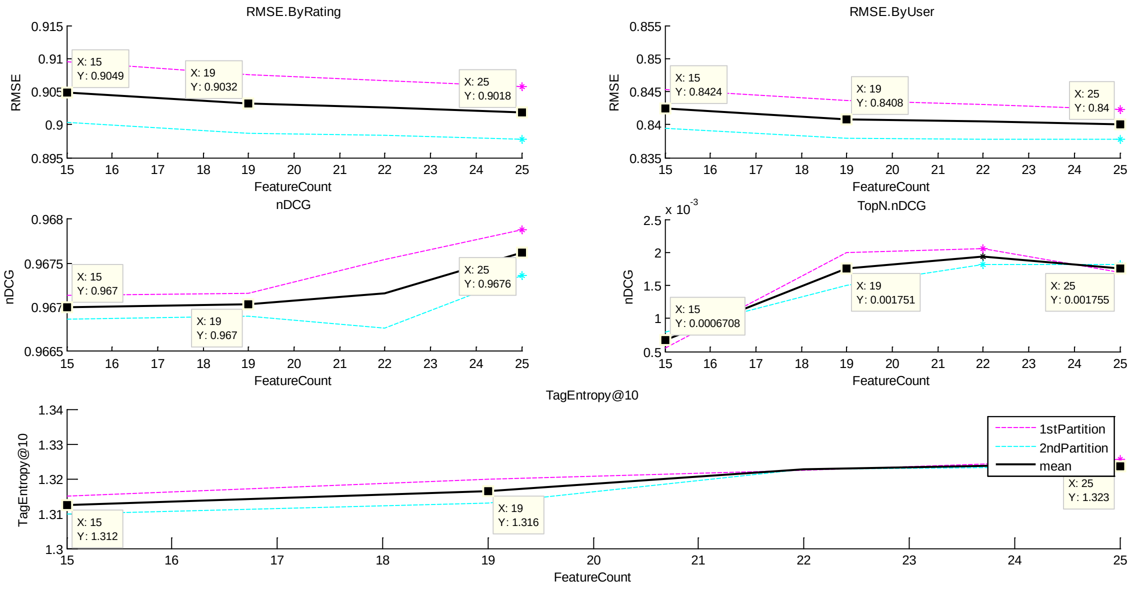


Figure 4-32: SVD Personalized Mean - 10M

- *SVD* User Mean

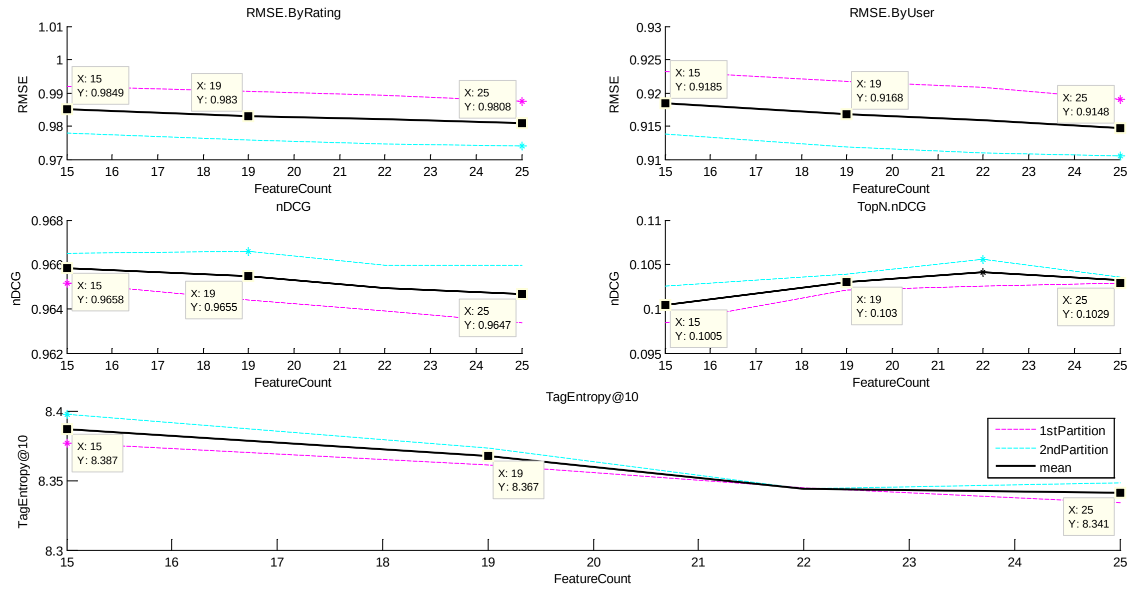


Figure 4-33: SVD User Mean - 10M

4.1.2.3.2.3 Comparison between *SVD*GlobalMean, *SVD*ItemMean, *SVDPersmean*, *SVD*UserMean:

ALGORITHM	FEATURE COUNT	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
SVDGLOBALMEAN	***15***	***1.034***	***0.9764***	***0.1064***	***0.1064***	***8.366***
SVDITEMMEAN	***15***	***0.9479***	***0.8855***	***0.9669***	***0.001077***	***1.31***
SVDPERSMEAN	***15***	***0.9049***	***0.8424***	***0.967***	***0.0006708***	***1.312***
SVDUSERMEAN	***15***	***0.9849***	***0.9185***	***0.9658***	***0.1005***	***8.387***
SVDGLOBALMEAN	***19***	***1.03***	***0.9722***	***0.1098***	***0.1098***	***8.335***
SVDITEMMEAN	***19***	***0.9469***	***0.8847***	***0.9668***	***0.001077***	***1.31***
SVDPERSMEAN	***19***	***0.9032***	***0.8408***	***0.967***	***0.001751***	***1.316***
SVDUSERMEAN	***19***	***0.983***	***0.9168***	***0.9655***	***0.103***	***8.367***
SVDGLOBALMEAN	***25***	***1.026***	***0.9689***	***0.1143***	***0.1143***	***8.297***
SVDITEMMEAN	***25***	***0.945***	***0.8828***	***0.9672***	***0.001371***	***1.313***
SVDPERSMEAN	***25***	***0.9018***	***0.84***	***0.9676***	***0.001755***	***1.323***
SVDUSERMEAN	***25***	***0.9808***	***0.9148***	***0.9647***	***0.1029***	***8.341***

Table 4-11: Comparison among SVDGlobalMean, SVDItemMean, SVDPersmean and SVDUserMean

Table 4-11 shows the results obtained. We can see that *SVDPersmean* gives the best results in terms of RMSE and nDCG. The best neighbourhood size for this algorithm is 25.

For the *ItemItem* collaborative filtering algorithm, the results obtained are the following:

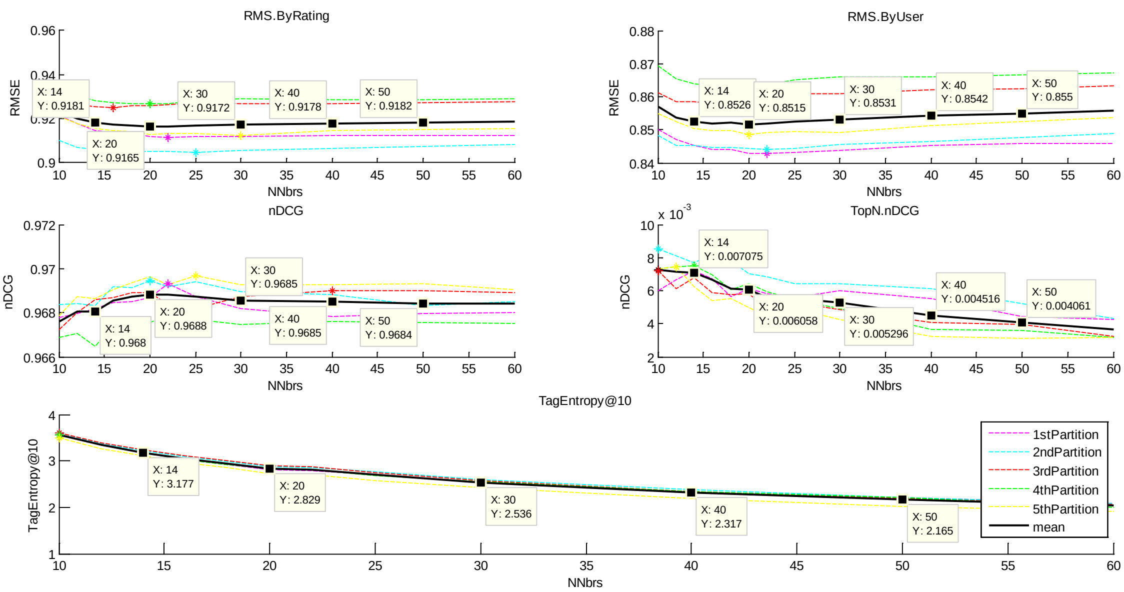


Figure 4-34: ItemItem - 10M

ALGORITHM	NNBRS	RMSE BY RATINGS	RMSE BY USER	NDCG	TOPN NDCG	ENTROPY
ITEMITEM	*14*	*0.9181*	*0.8526*	*0.968*	*0.007194*	*3.177*
ITEMITEM	*16*	*0.9173*	*0.852*	*0.9686*	*0.006639*	*3.042*
ITEMITEM	*18*	*0.9171*	*0.852*	*0.9687*	*0.006149*	*2.932*
ITEMITEM	*20*	*0.9165*	*0.8515*	*0.9688*	*0.006058*	*2.829*
ITEMITEM	*30*	*0.9172*	*0.8531*	*0.9685*	*0.005296*	*2.536*
ITEMITEM	*40*	*0.9178*	*0.8542*	*0.9685*	*0.004516*	*2.317*

Table 4-12: Comparison between different sizes of neighbourhood for ItemItem

The best neighbourhood size is 20, since we have obtained the highest value of nDCG with the lowest RMSE (Table 4-12).

And finally, the two basic algorithms:

- *Personalized Mean*

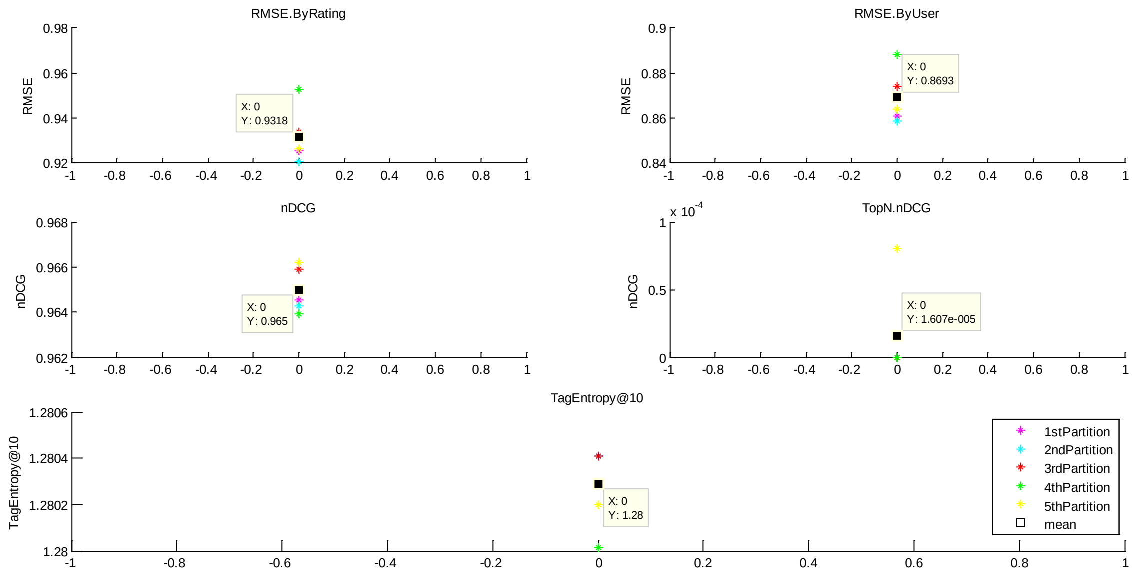


Figure 4-35: Personalized Mean - 10M

- *Popular*

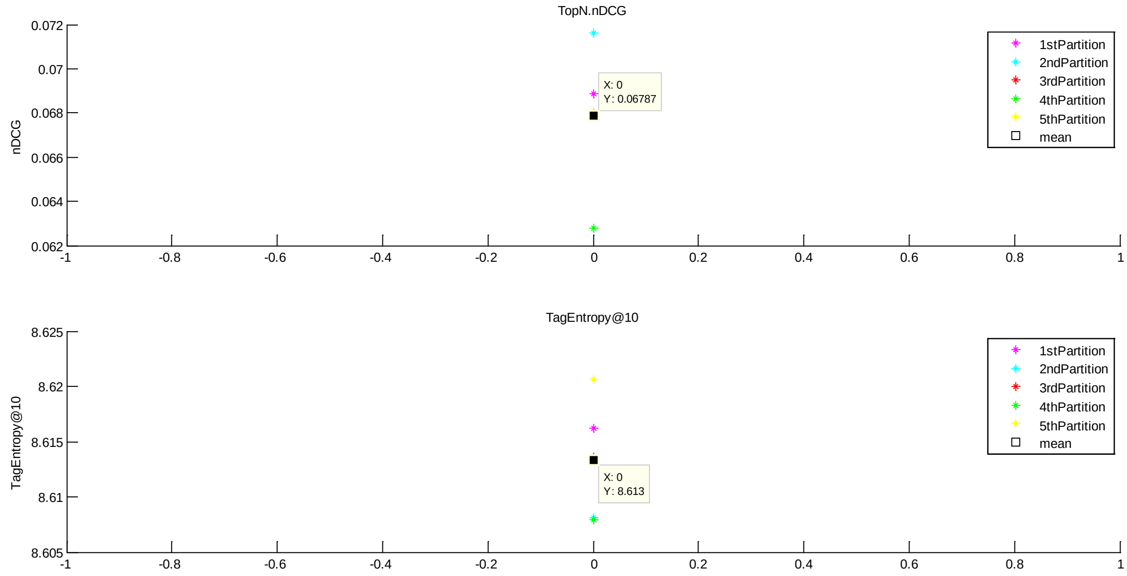


Figure 4-36: Popular - 10M

As we have seen with the 10M dataset, the results are almost the same for all the algorithms as with the 1M dataset. Only with *UserUser*Cosine we have noticed an increase in the best size of neighbourhood.

Table 4-78 summarizes the results of each algorithm with the best parameters. In the next section, we are going to make a comparison among these results and the ones collected through the online experiment.

4.2 Online Evaluation

4.2.1 Online Experiment

To carry out the online experiment, we have created two online forms powered by the technology of *Google Forms.*

The first form purpose is to collect users' ratings to give them recommendations. To reach a larger number of participants we have sent it through social networks such as Facebook or Twitter. And it made easier to collect the data and process their responses.

This form is divided into two sections: the first one is designed to collect the personal data of the subject under study. We ask for the Name, Gender, Age for a comparison between ages and genders of the algorithm performance and Email to get feedback. We encourage the users to fill all the ratings as precisely as possible, because we are going to recommend them a list of movies that they should enjoy (this was used as a hook to improve their motivation in the rating). Then, they have to select their general interests in movie genres: Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western; and the second part of the form is the rating list. Users rate a list of 100 selected movies from the top of IMDB. They only rate the films that they have seen.

As we wanted to collect data from individual users as well as for groups, when we asked for the personal data, we added a paragraph to encourage users to fill this form in group. If the check box of groups was selected, they were driven to another page of the form asking for the personal data of all the group members. Finally, they have to rate the same movies as the individual users. The aspect of this first questionnaire is show in Figure 4-37.

Nevertheless, in the case of the groups, we also have a third part on the form where the group members are going to write how they have decided which rate to give to each movie, if it was difficult or easy to reach an agreement, where they have found difficulties and how they have reached consensus.

People need to be together during the rating process. They have to make their decision in a conversation among all the participants of the group. As Cano [7] states *"Participants are subject to the process (changing), the converse, are generating changes in your talk and conversation".* (para. 6).

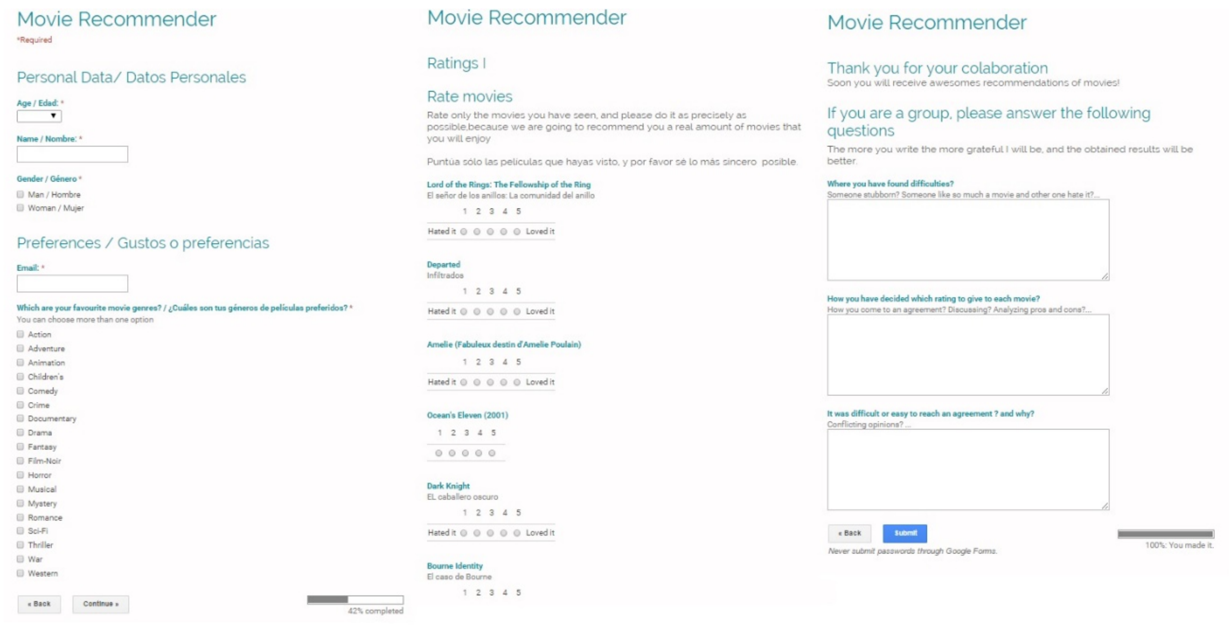


Figure 4-37: Aspect first questionnaire

Between 25th November 2014 and 7th December 2014, 158 users filled the survey, where 138 were individual users and 20 were groups (Figure 4-38).

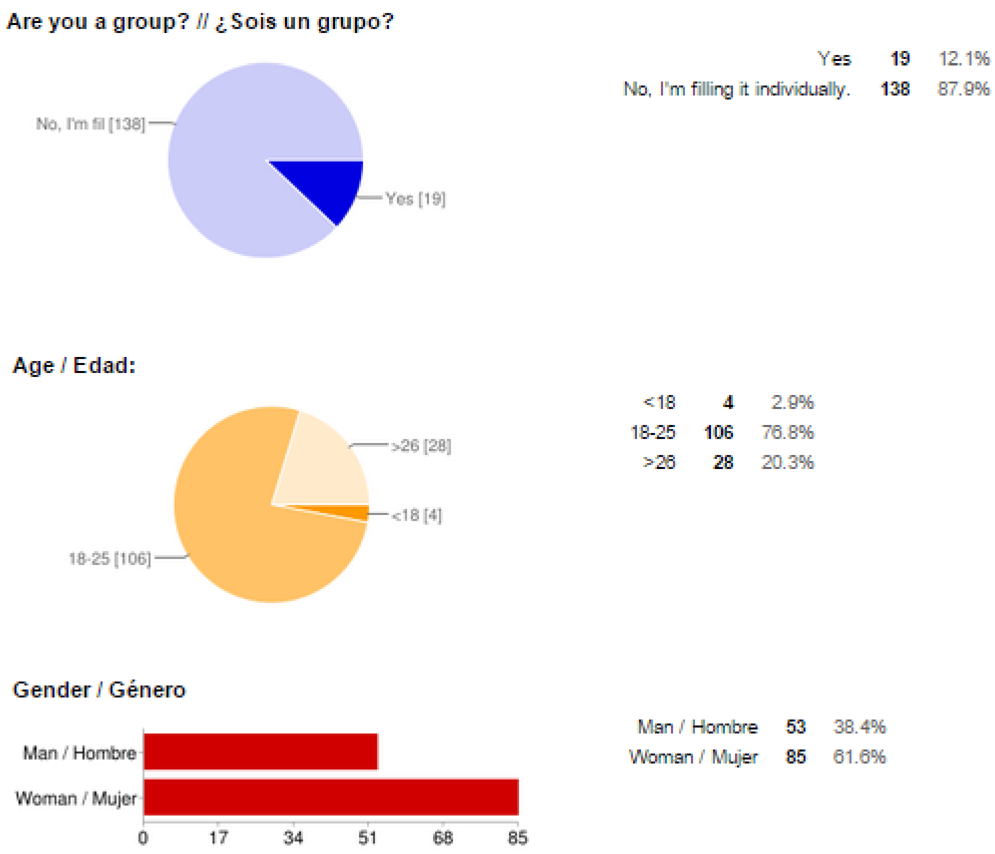


Figure 4-38: Summary of answers from the questionnaire

Once we have collected the data, we start to process it to obtain the recommendations to each user. Then, we make a form to each user with the 6 recommendations' lists and some questions to know their perception of the algorithms used. The aspect of this form is visible in Figure 4-39:

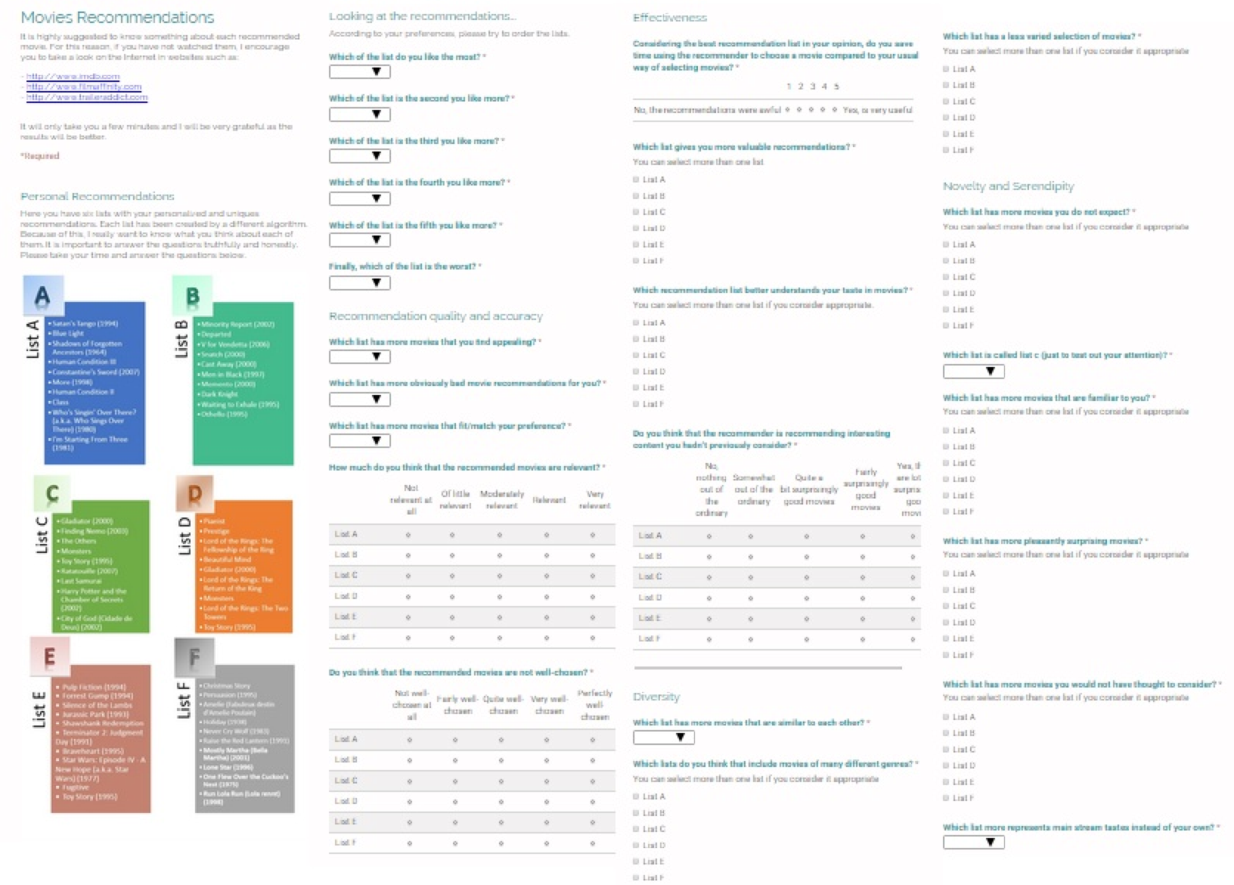


Figure 4-39: Aspect of the second questionnaire with the user recommendation lists

As in the case of the first form, we included some extra questions in the groups' questionnaire to have an idea of the difficulties found.

Looking at Figure 4-40, we have to highlight that only 60 of the 158 users that filled the first form, completed this survey between 16th February and 21th March. 50 of them were individual users and 10 were groups. Among the individual users, we can make a distinction by gender (29 female and 21 male) and also by age (40 younger than 25 and 10 older than 25).

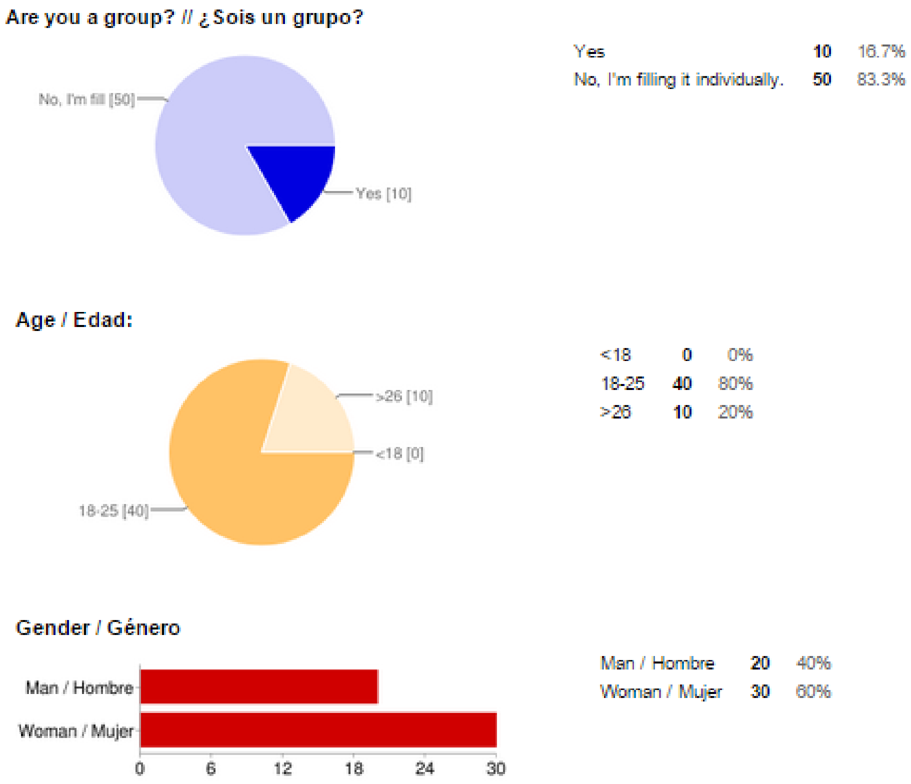


Figure 4-40: Summary of the answers from the second questionnaire

Taken into account the size of the groups (Figure 4-41), 7 of them were groups of two people and 3 of them were groups of 3 people. However, due to the small number of groups that answered the questionnaire, we are not going to make a distinction according to the size.

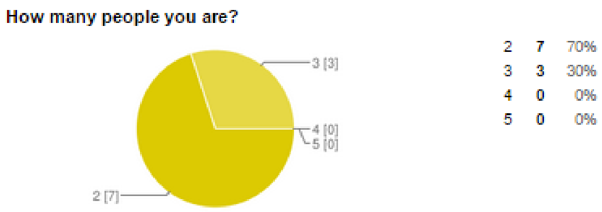


Figure 4-41: Size of the groups that filled the questionnaire

In order to create the questionnaire of this form, as we can see in Figure 4-39, we first asked the users to rank their initial preferences, followed by 17 questions to know the users' perceptions of the qualitative aspects we want to measure: *Accuracy*, *Understands Me*, *Diversity*, *Novelty*, *Effectiveness* and *Quality*. These questions are taken from Michael and Ekstrand [14] and Knijnenburg et al. [23] since they have proved that these questions work well in other studies which measured users' satisfaction of a recommender system.

4.2.2 Results

4.2.2.1 Preferences of individual users.

First of all, in order to know the first impression of our users we have asked them to order the displayed lists taking into account their preferences, from the best one to the worst according to their opinions. The distribution of their responses is displayed in Table 4-13 so that we can analyse whether or not their opinions differ significantly.

Algorithms	**1st Place**	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**
ItemItem	13	12	9	7	8	0
Lucene	5	6	3	12	12	12
Persmean	0	2	4	8	13	24
Popular	19	3	8	8	7	5
SVD	5	12	13	9	5	6
UserUser	8	15	13	6	5	3
Test Statistics						
Chi-Square	27,280	17,440	10,960	2,560	7,120	44,800
Df	5	5	5	5	5	5
Exact Sig.	0,000	0,004	0,053	0,789	0,221	0,000

Table 4-13: Users preferences answers

The results obtained from the chi-square test (Table 4-13) tell us that there are significant differences (p≈0,000) taking into account the number of times an algorithm is selected as the best by the users' opinion.

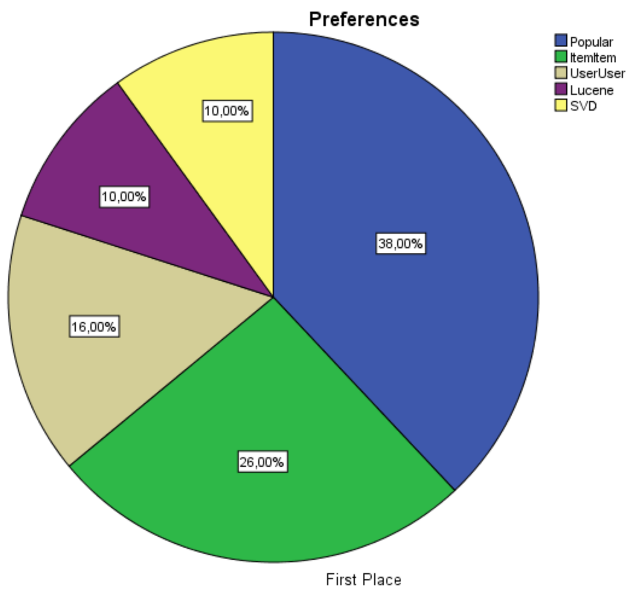


Figure 4-42: Algorithms selected in first place by the users.

In Figure 4-42, we can see that the algorithm which is best considered is *Popular* (38%), followed by the collaborative filtering algorithms by Item (26%) and by User (16%) respectively. Then, we can find *Lucene* and *SVD* (10%) and finally *Personalized Mean* (0%). However, as visible in Table 4-14, the difference appreciated between *Popular* and *ItemItem* are not significant (p=0.289), both are selected as the best algorithm by a huge number of users. We will have to take into account how many users have selected them in the second place to conclude which of them is the best.

Test Statistics

	Preference
Chi-Square	1,125[a]
df	1
Asymp. Sig.	,289

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 16,0.

Table 4-14: Study of the difference between Popular and ItemItem

Checking the ranking of the algorithms selected in second place (Table 4-13), we can find significant differences (p=0.004) among them. We can underline *Persmean* and *Lucene* since both algorithms are selected by a very low proportion of the users as the first options. However, collaborative filtering algorithms have the higher percentages here (more than 20% each one). It is notable that *Popular*, although is selected for a higher percentage of users in first place, is only selected by a 6% of the users in second place (Figure 4-43).

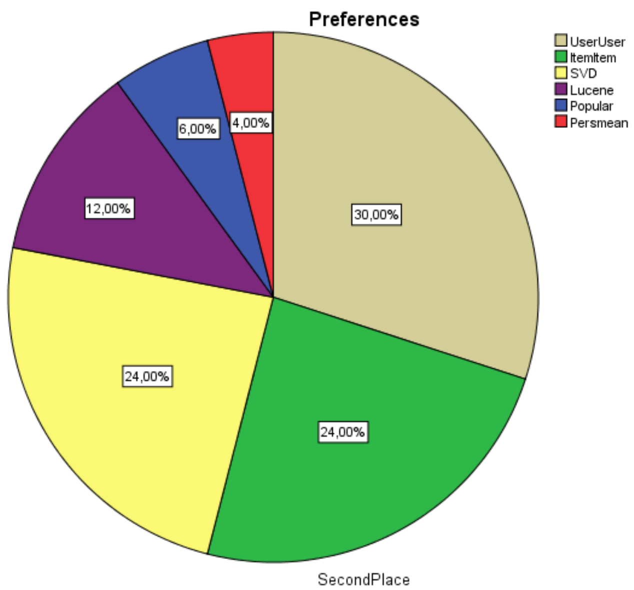


Figure 4-43: Algorithms selected in second place by the users.

Among the results collected from the third, fourth and fifth positions, we cannot extrapolate conclusions due to the high controversy found (Table 4-13). Nevertheless, on the last position, the results are clear because *Persmean* stands out from the others with 48%, followed by *Lucene* with 24%. Moreover, it is worth mentioning that *ItemItem* does not appear on the graph since nobody thinks that it is the worst algorithm. Nevertheless, it is important to say that *Popular* is selected as the worst algorithm by a 10% of the users (Figure 4-44). This demonstrate that recommendations based on *Popular*ity induce opposite views among the users.

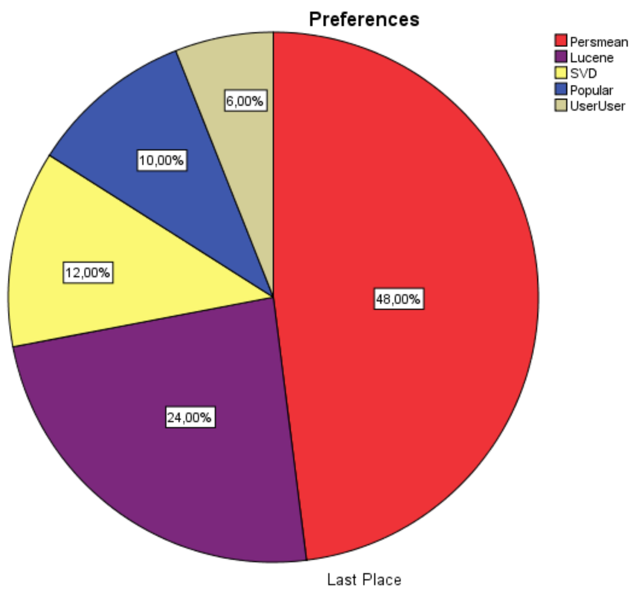


Figure 4-44: Algorithms selected in last place by the users.

To have a general overview of the results, we have weighted the data in such a way that we give 6 points to the algorithm selected in the first place, 5 points to the algorithm in the second place, 4 points to the algorithm in the third place, 3 points to the algorithm in the fourth place, and, finally, 1 point to the algorithm in the last place. This kind of ranking is called average ranking, according to the team of Survey Monkeys [49], which are one of the most important provider of web-based survey solutions. They [49] state in their webpage that this is the best way of analysing ranking questions on surveys. Therefore, we can illustrate this with a rank (Table 4-15):

1	*ItemItem*	223	21%
2	*UserUser*	206	19.4%
3	*Popular*	204	19.21%
4	*SVD*	185	17.42%
5	*Lucene*	144	13.55%
6	*Persmean*	100	9.41%

Table 4-15: Ranking of users preferences

As we have seen on the pie charts, collaborative filtering algorithms are clearly the best algorithms to the users' perception, followed by *Popular* with a high percentage, while *Persmean* is obviously the loser. Although *Popular* have been chosen by the majority of the users as the best algorithm in first place, we can see that it is not the best algorithm, since there are a considerate percentage of users that chosen it algorithm as the worst.

4.2.2.2 Preferences of groups.

We are now going to check the results obtained from the groups. Table 4-16 shows the distribution of their response to evaluate whether or not there are significant differences among algorithms.

As visible in Table 4-16, we can only extrapolate the differences observed among algorithms in first place (p=0.003) and in the last place (p=0.040).

Algorithms	**1st Place**	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**
ItemItem	6	2	1	1	0	0
Lucene	1	1	0	3	2	3
Persmean	0	0	0	1	4	5
Popular	3	2	3	1	0	1
SVD	0	2	3	2	2	1
UserUser	0	3	3	2	2	0
Test Statistics						
Chi-Square	17,600	3,200	6,800	2,000	6,800	11,600
Df	5	5	5	5	5	5
Exact. Sig.	0,003	0,782	0,270	0,944	0,270	0,040

Table 4-16: Groups preferences answers

Looking at Figure 4-45, we can note that the best algorithm is *ItemItem* (60%), followed by *Popular* (30%) and Figure 4-46 show us that the worst algorithm is *Persmean* (50%) although *Lucene* has also been bad considered (30%). The results do not show evidences in relation to *SVD* or *UserUser*.

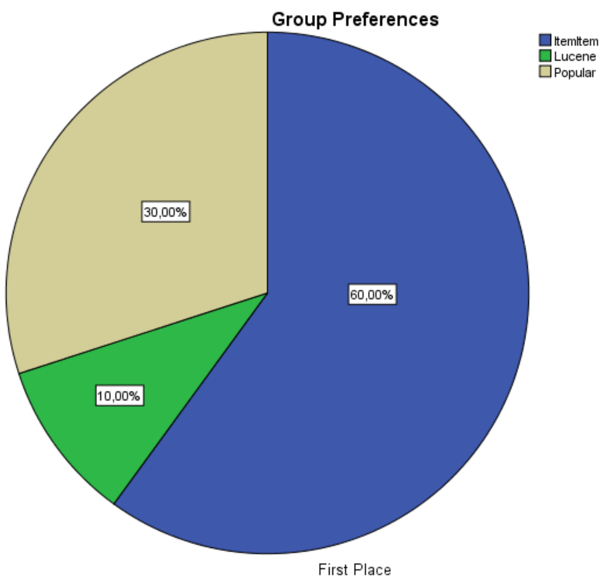


Figure 4-45: Groups preferences in first place

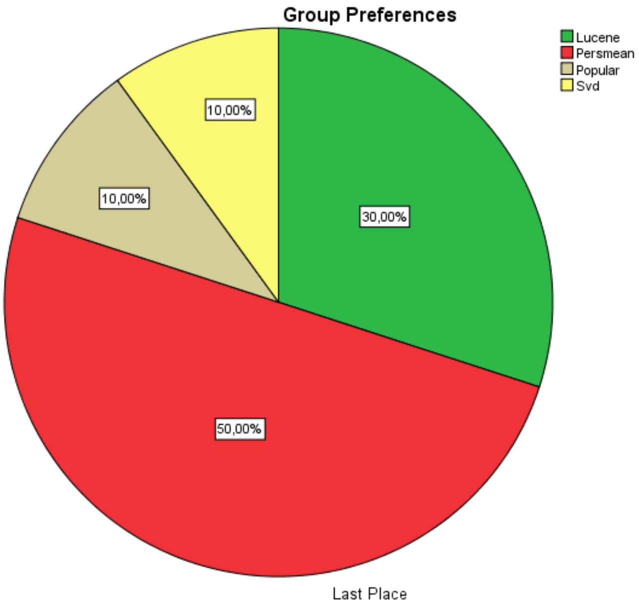


Figure 4-46: Groups preferences in last place

As we did on the analysis of the individuals users' preferences, to have a general overview of the results we have weighted the data (Table 4-17).

1	*ItemItem*	53	25.24%
2	*Popular*	44	20.95%
3	*UserUser*	37	17.62%
4	*SVD*	33	15.71%
5	*Lucene*	27	12.86%
6	*Persmean*	16	7.62%

Table 4-17: Ranking of the group preferences

In the case of the groups we can see that the winner is also *ItemItem*. *Popular* is now better considered than *UserUser*, although the difference between them is not huge. Moreover, it is clear that the worst considered are *Lucene* and *Persmean*, as it happens on the individuals users' analysis.

Although only 10 groups have filled our survey, the most striking issue is that their preferences are quite similar to the ones of the individual users. The best algorithms for the groups are *ItemItem* and *Popular*, and the worst are *Persmean* and *Lucene*. This result is the same as the one which was obtained by analysing the preferences of the individual users, which indicates that the use of traditional algorithms to make groups recommendations, once we have a group as a pseudo user, is a good way.

4.2.2.3 Preferences by Gender

Algorithms	**Women**						**Men**					
	1st Place	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**	**1st Place**	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**
ItemItem	8	6	8	2	4	0	5	6	1	5	4	0
Lucene	5	3	1	6	6	8	0	3	2	6	6	4
Persmean	0	1	1	6	9	13	0	1	3	2	4	11
Popular	5	3	3	8	6	4	14	0	5	0	1	1
SVD	5	7	7	4	2	4	0	5	6	5	3	2
UserUser	6	9	9	3	2	0	2	6	4	3	3	3

Table 4-18: Users preferences making a distinction by gender

After that, we want to check whether there are differences or not between men and women. Since we only have statistical significant differences (p=0.001) among the algorithms selected in first place (Table 4-19), we will only analyse these ones. Take into consideration that the assumption of the expected count is violated (it should have been less than 20% although the obtained value is 66, 7%), so we had had to look at the likelihood ratio to determine the p-value.

Algorithms * Gender Crosstabulation

			Gender		
			Woman	Man	Total
Algorithms	*ItemItem*	Count	8	5	13
		Expected Count	7,5	5,5	13,0
	Lucene	Count	5	0	5
		Expected Count	2,9	2,1	5,0
	Persmean	Count	0	0	0
		Expected Count	,0	,0	,0
	Popular	Count	5	14	19
		Expected Count	11,0	8,0	19,0
	SVD	Count	5	0	5
		Expected Count	2,9	2,1	5,0
	UserUser	Count	6	2	8
		Expected Count	4,6	3,4	8,0
Total		Count	29	21	50
		Expected Count	29,0	21,0	50,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	16,087[a]	5	,007
Likelihood Ratio	19,808	5	,001
Linear-by-Linear Association	,015	1	,904
N of Valid Cases	50		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is ,00.

Table 4-19: Statistical study of the differences observed in the preferences in first place between gender

Looking at Table 4-19, we can see how women rather than men prefer *Lucene* and *SVD* in their first place, since men had not selected these algorithms as the best ones. Men, in contrast, prefer *Popular* more than women prefer it. The annual report from the Theatrical Market Statistics [37] demonstrates that the majority of moviegoers are women. As they have described [37], "females have comprised a larger share of moviegoers (people who went to a movie at the cinema at least once in the year) consistently since 2010, this trend remains unchanged in 2014. In fact, the number of female moviegoers increased slightly in 2014, while the number of male moviegoers remained flat". This explains why more women prefer *Lucene*, since we can appreciate in the report that women not only go to the theatre to watch movies with high popularity but they also go to watch other movies such as movies with female film stars or romantic comedies. However, men only go to watch movies with high *Popular*ity.

4.2.2.4 Preferences by Age

Algorithms	**Younger 25**						**Older 25**					
	1st Place	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**	**1st Place**	**2nd place**	**3rd Place**	**4th Place**	**5th Place**	**6th Place**
ItemItem	12	10	8	5	4	0	1	2	1	2	4	0
Lucene	3	3	2	10	10	12	2	3	1	2	2	0
Persmean	0	1	2	6	12	20	0	1	2	2	1	4
Popular	13	3	7	6	7	4	6	0	1	2	0	1
SVD	5	9	11	8	4	3	0	3	2	1	1	3
UserUser	7	14	10	5	3	1	1	1	3	1	2	2

Table 4-20: Users preferences making a distinction by age

Making a distinction between people younger than 25 and older ones, we can only note statistical significant differences (Table 4-21) among the algorithms prefer in the last position (p=0.046) since the assumption of the expected count is violated (it should have been less than 20% although the obtained value is 83, 3%), so we had had to look at the likelihood ratio to determine the p-value.

Algorithms * Age Crosstabulation

			Young	Old	Total
Algorithms	*ItemItem*	Count	0	0	0
		Expected Count	,0	,0	,0
	Lucene	Count	12	0	12
		Expected Count	9,6	2,4	12,0
	Persmean	Count	20	4	24
		Expected Count	19,2	4,8	24,0
	Popular	Count	4	1	5
		Expected Count	4,0	1,0	5,0
	SVD	Count	3	3	6
		Expected Count	4,8	1,2	6,0
	UserUser	Count	1	2	3
		Expected Count	2,4	,6	3,0
Total		Count	40	10	50
		Expected Count	40,0	10,0	50,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	10,625[a]	5	,059
Likelihood Ratio	11,272	5	,046
Linear-by-Linear Association	9,945	1	,002
N of Valid Cases	50		

a. 10 cells (83,3%) have expected count less than 5. The minimum expected count is ,00.

Table 4-21: Statistical study of the differences observed in the preferences between age

Based on the opinion of people younger than 25, *Lucene* is in the last position much often in comparison to people older than 25. While *SVD* is much often in the last position for older people rather than younger. However, note that the sample size is quite small. Therefore, we should not extrapolate these results.

4.2.2.5 Comparison with the offline results

If we have a look at the offline ranking of algorithms' performance (Table 4-22), we find that one of the difference with the ranking made by users' first impression is the algorithm *UserUser*. The users' perception of this algorithm is better than the expected by the results of the offline evaluation. *Popular*, which has the best result in the offline evaluation is on the 3rd position on the online results. However, *ItemItem* has gained a position in the online evaluation, where is the algorithm best considered. Taking into account *Lucene* we can appreciate some differences between the offline and online results. In the offline evaluation is on the 3rd position while in the online evaluation is on the 5th. Users has a worst opinion about *Lucene* than expected. Nonetheless, we can note similarities between the offline results evaluation and the online results. The predictions based on topN nDCG are quite good, and they can be used to measure the goodness of the recommender systems.

Offline			**Online**		
Based on topN nDCG	Results	Results normalized to unity	**Based on users' preferences**	Number of users that select each algorithm by average ranking	Results normalized to unity
1st *Popular*	0.06787	1	**1st *ItemItem***	223	1
2nd *ItemItem*	0.006058	0.08925	**2nd *UserUser***	206	0.9237
3rd *Lucene*	0.004968	0.07319	**3rd *Popular***	204	0.9147
4th *SVD*	0.001695	0.02497	**4th *SVD***	185	0.8295
5th *UserUser*	0.001684	0.02481	**5th *Lucene***	144	0.6457
6th *Persmean*	0.00001607	0.00023	**6th *Persmean***	100	0.4484

Table 4-22: Comparison between the offline results and the online preferences

4.2.3 Analysis Subjective Metrics

4.2.3.1 Accuracy

In order to measure *Accuracy*, we have asked our users two different questions. The first of them has a positive connotation and the second one has a negative connotation. Therefore, we need to take it into account in order to make a good interpretation of the results.

Q1- WHICH LIST HAS MORE MOVIES THAT YOU FIND APPEALING?

Table 4-23 shows the choice of algorithm by each user, making a distinction between gender and age.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	18	6	12	13	5
ItemItem	12	7	5	11	1
UserUser	9	7	2	8	1
Lucene	6	5	1	3	3
SVD	4	4	0	4	0
Persmean	1	0	1	1	0

Table 4-23: Data collected from the questionnaire Q1

We can observe in Table 4-24 that there is statistical significance ($p \approx 0.000$) between our algorithms. In figure 1, we can see that *Popular* is the algorithm which is preferred by most of the users with a 36%, even though it is a basic algorithm. The reason is that everybody has watched and enjoyed *Popular* movies. Then, collaborative filtering by Item has a significant relevance ($p \approx 0.000$) above the rank matrix algorithm *SVD*, which is only selected by the 8% of the users. And finally, the most inaccurate algorithm for the users is *Personalized Mean* with only 2%.

Q1*ACCURACY*

	Observed N	Expected N	Residual
ItemItem	12	8,3	3,7
Lucene	6	8,3	-2,3
Persmean	1	8,3	-7,3
Popular	18	8,3	9,7
SVD	4	8,3	-4,3
UserUser	9	8,3	,7
Total	50		

Test Statistics

	Q1*ACCURACY*
Chi-Square	22,240[a]
df	5
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 8,3.

Table 4-24: Chi squared test Q1 with α=0.05.

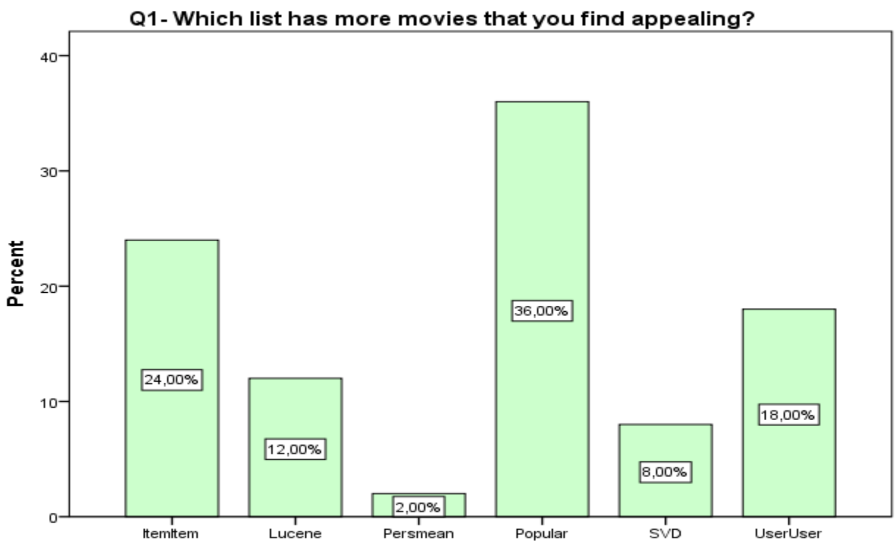


Figure 4-47: Bar diagram representing the data collected

In order to test if the results are dependent of users' gender and age, we have computed the chi-squared test. In the case of gender (Table 4-25), the assumption of the expected count cell is violated (it should have been less than 20% although the obtained value is 66,7%), so we have to look at the likelihood ratio to determine if our results are dependent on it. The Asymptotic Significance in this case is p = 0.033, lower than α. Therefore, we can consider that our results depend on gender. If we take a look at the differences between gender, the most notable feature is that men preferred *Popular* while women preferred *Lucene*. We also found some differences with *SVD* and *UserUser* since both algorithms are preferred by women. Although men and women agree with *ItemItem* and *Personalized Mean*, this last algorithm does not mean *Accuracy* for the users.

Q1*ACCURACY* * Gender Crosstabulation

			Gender		
			Woman	Man	Total
Q1*ACCURACY*	*ItemItem*	Count	8	4	12
		Expected Count	7,4	4,6	12,0
	Lucene	Count	5	1	6
		Expected Count	3,7	2,3	6,0
	Persmean	Count	0	1	1
		Expected Count	0,6	0,4	1,0
	Popular	Count	7	11	18
		Expected Count	11,2	6,8	18,0
	SVD	Count	4	0	4
		Expected Count	2,5	1,5	4,0
	UserUser	Count	7	2	9
		Expected Count	5,6	3,4	9,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	10,385[a]	5	,065
Likelihood Ratio	12,132	5	,033
Linear-by-Linear Association	,014	1	,905
N of Valid Cases	50		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is 0,38.

Table 4-25: Chi squared test Q1 by Gender with α=0.05

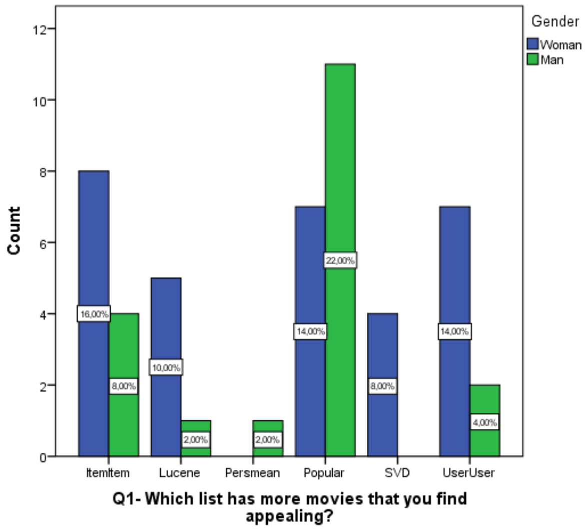


Figure 4-48: Bar diagram representing the results by gender. Note that all the percentages are expressed taking into account the total number of users (N=50).

Moreover, in the case of age (Table 4-26), the assumption of the expected cell count is also violated, (it should have been less than 20% although the obtained value is 75%). Looking then at the likelihood ratio, the asymptotic significance is p=0.123, which is higher than α, so that the results are independent of age.

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	6,250a	5	,283
Likelihood Ratio	8,674	5	,123
Linear-by-Linear Association	,818	1	,366
N of Valid Cases	50		

a. 9 cells (75,0%) have expected count less than 5. The minimum expected count is ,20.

Table 4-26: Chi squared test Q1 by Age with α=0.05

Q2- WHICH LIST HAS MORE OBVIOUSLY BAD MOVIE RECOMMENDATIONS FOR YOU?

With this question, we are measuring the inaccuracy of our algorithms. Thus, we want to approximately obtain results which are opposite to the ones that were obtained with the previous question since a list can both contain some very good recommendations but also very bad ones.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	6	4	2	4	2
ItemItem	0	0	0	0	0
UserUser	3	0	3	1	2
Lucene	13	10	3	13	0
SVD	5	3	2	3	2
Persmean	23	12	11	19	4

Table 4-27: Data collected from the questionnaire

We prove in Table 4-28 that there is statistical significance (p≈0.000) between our algorithms. In Figure 4-49, we can see that the worst algorithm for 46% of the users is *Personalized Mean*. *Lucene* is also bad considered by 26% of the users. Moreover, there is not a big difference between *Popular* and *SVD* since both algorithms are inaccurate for a 10% of the users approximately. The remarkable issue is that collaborative filtering algorithms by Item and by User are now the best considered by the users. This means that *Popular* recommendation is good in general but it has more notable bad movies while *ItemItem* or *UserUser* recommendations are worse than *Popular* in general but all the movies recommended are good.

Frequencies

	Q2*ACCURACY*			
	Category	Observed N	Expected N	Residual
1	*ItemItem*	0	8,3	-8,3
2	*Lucene*	13	8,3	4,7
3	*Persmean*	23	8,3	14,7
4	*Popular*	6	8,3	-2,3
5	*SVD*	5	8,3	-3,3
6	*UserUser*	3	8,3	-5,3
Total		50		

Test Statistics

	Q2*ACCURACY*
Chi-Square	42,160[a]
df	5
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 8,3.

Table 4-28: Chi squared test Q2 with α=0.05

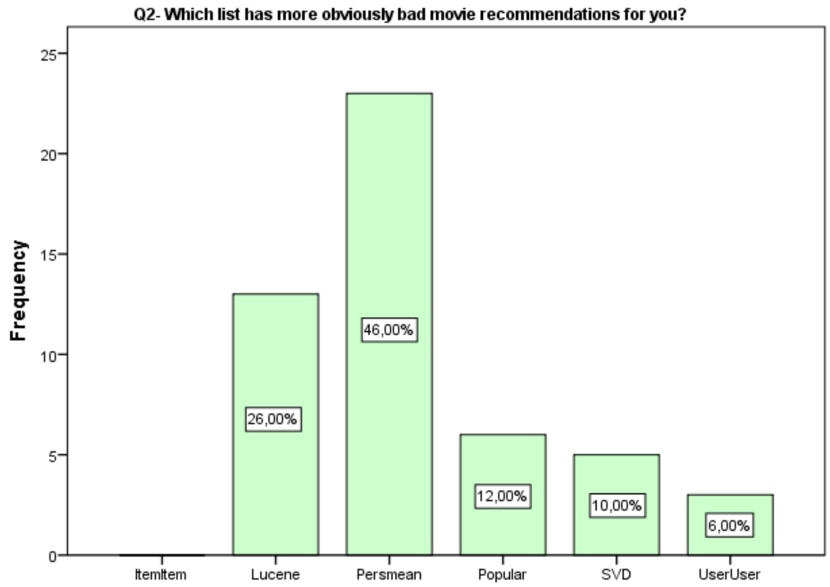


Figure 4-49: Bar diagram representing the data collected.

In this question, there is not dependence with neither gender (p=0.169) nor age (p= 0.06). In both cases the assumption of the expected count is violated, so we had had to look at the likelihood ratio to determine the p-value (Table 4-29).

Now, we shall analyse the results as a combination of Q1 and Q2. For this purpose, we will consider the answers obtained in the first question as positive ones for the algorithm +1 and the answers obtained in the second question as negative ones for the algorithm -1 (Figure 4-50).

In conclusion, Collaborative Filtering along with *Popular* are the best algorithms in terms of *Accuracy* while *Personalized Mean* is the worst.

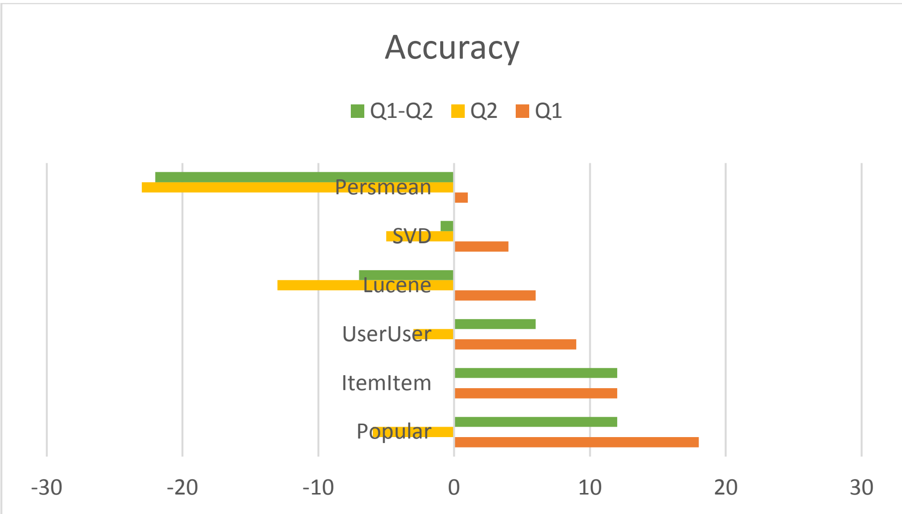


Figure 4-50: Combination of the two questions that measure Accuracy. The green bar is the result of the combination.

Algorithms * Gender Crosstabulation

			Gender		
			Women	Men	Total
Algorithms	*ItemItem*	Count	0	0	0
		Expected Count	,0	,0	,0
	Lucene	Count	10	3	13
		Expected Count	7,5	5,5	13,0
	Persmean	Count	12	11	23
		Expected Count	13,3	9,7	23,0
	Popular	Count	4	2	6
		Expected Count	3,5	2,5	6,0
	SVD	Count	3	2	5
		Expected Count	2,9	2,1	5,0
	UserUser	Count	0	3	3
		Expected Count	1,7	1,3	3,0
Total		Count	29	21	50

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	6,568[a]	5	,255
Likelihood Ratio	7,774	5	,169
Linear-by-Linear Association	3,087	1	,079
N of Valid Cases	50		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is ,00.

Algorithms * Age Crosstabulation

			Age		
			Young	Old	Total
Algorithms	*ItemItem*	Count	0	0	0
		Expected Count	,0	,0	,0
	Lucene	Count	13	0	13
		Expected Count	10,4	2,6	13,0
	Persmean	Count	19	4	23
		Expected Count	18,4	4,6	23,0
	Popular	Count	4	2	6
		Expected Count	4,8	1,2	6,0
	SVD	Count	3	2	5
		Expected Count	4,0	1,0	5,0
	UserUser	Count	1	2	3
		Expected Count	2,4	,6	3,0
Total		Count	40	10	50
		Expected Count	40,0	10,0	50,0

Chi-Square Tests by Age

	Value	df	Asymp. Sig (2-sided)
Pearson Chi-Square	9,348[a]	5	,096
Likelihood Ratio	10,599	5	,060
Linear-by-Linear Association	8,943	1	,003
N of Valid Cases	50		

a. 10 cells (83,3%) have expected count less than 5. The minimum expected count is ,00.

Table 4-29: Chi squared test Q2 by Gender and Age with α=0.05. Both cases violate the assumption of the expected cell count so we look at the likelihood ratio to evaluate the results.

4.2.3.2 Understands Me

With the following questions, our intention is to know which algorithm best understands users' taste. The third question Q3 has a negative connotation since we are looking for the algorithm with more popular movies. In contrast, the fourth question Q4 looks for the algorithm with more movies which match the user' taste.

Q3-WHICH LIST MORE REPRESENTS MAIN STREAM TASTES INSTEAD OF YOUR OWN?

Table 4-30 shows the choice of algorithm by each user, making a distinction between gender and age.

Algorithms	Users	By Gender		By Age	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Lucene	3	2	1	3	0
UserUser	12	8	4	10	2
SVD	11	7	4	8	3
ItemItem	6	3	3	4	2
Persmean	3	1	2	3	0
Popular	15	8	7	12	3

Table 4-30: Data collected from the questionnaire

As we can see looking at Table 4-31, there is statistical significance (p=0.009) among our algorithms. In view of the following figure, the algorithms with more popular movies are *Popular*, as expected, but also *UserUser* and *SVD* with a high percentage (more than 20%). Users think that *Lucene* and *Personalized Mean* do not represent main stream tastes, and it has sense since both algorithms are based on the users' taste.

UNDERSTAND ME

	Observed N	Expected N	Residual
ItemItem	6	8,3	-2,3
Lucene	3	8,3	-5,3
Persmean	3	8,3	-5,3
Popular	15	8,3	6,7
SVD	11	8,3	2,7
UserUser	12	8,3	3,7
Total	50		

Table 4-31: Chi squared test Q3 with α=0.05

Test Statistics

	UNDERSTAND ME
Chi-Square	15,280[a]
df	5
Asymp. Sig.	,009

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 8,3.

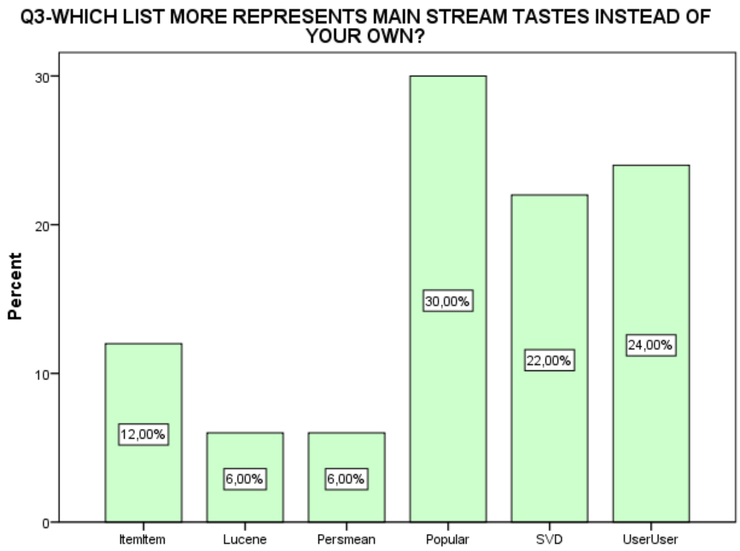


Figure 4-51: Bar diagram representing the data collected for Q3.

In this question, there is not dependence with gender (p=0.481) nor age (p=0.426). In both cases the assumption of the expected count is violated, so we had had to look at the likelihood ratio to determine the p-value (Table 4-32).

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	4,250[a]	5	,514
Likelihood Ratio	4,489	5	,481
Linear-by-Linear Association	2,122	1	,145
N of Valid Cases	50		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is 1,14.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,939[a]	5	,558
Likelihood Ratio	4,914	5	,426
Linear-by-Linear Association	,278	1	,598
N of Valid Cases	50		

a. 9 cells (75,0%) have expected count less than 5. The minimum expected count is ,60.

Table 4-32: Chi squared test Q3 by Gender and Age with α=0.05. Both cases violate the assumption of the expected cell count so we look at the likelihood ratio to evaluate the results.

Q4-WHICH RECOMMENDATION LIST BETTER UNDERSTANDS YOUR TASTE IN MOVIES?

Table 4-33 shows the answers of our users for this question, divided by age and gender. What we firstly see is that *Popular* is the algorithm with more votes. But we are going to analyse first whether these differences appreciated are significant or not.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Lucene	8	6	2	4	4
UserUser	14	10	4	13	1
SVD	12	8	4	11	1
ItemItem	16	9	7	15	1
Persmean	2	1	1	2	0
Popular	23	7	16	17	6

Table 4-33: Data collected from the questionnaire

Q4UNDERSTANDME

	Observed N	Expected N	Residual
ItemItem	16	12,5	3,5
Lucene	8	12,5	-4,5
Persmean	2	12,5	-10,5
Popular	23	12,5	10,5
SVD	12	12,5	-,5
UserUser	14	12,5	1,5
Total	75		

Test Statistics

	Q4UNDERSTANDME
Chi-Square	20,440[a]
df	5
Asymp. Sig.	,001

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 12,5.

Table 4-34: Chi-squared test Q4 with α=0.05

As we can see in Table 4-34, there is statistical significance (p=0.001) among our algorithms. It can be noted in Figure 4-52 that most users think that *Popular* is the algorithm that best fits their tastes although, as we have seen on the question before, it is at the same time the algorithm that best represents the main stream tastes. This leads us to understand that our users' taste is strongly correlated with the movies' popularity. Moreover, collaborative filtering algorithms by Item and by User are also algorithms that represent users' taste. In contrast, *Lucene* and *Personalized Mean* do not match users' taste. People think that this algorithms do not understand their taste, which means that these algorithms do not work well, as we have seen in the question above.

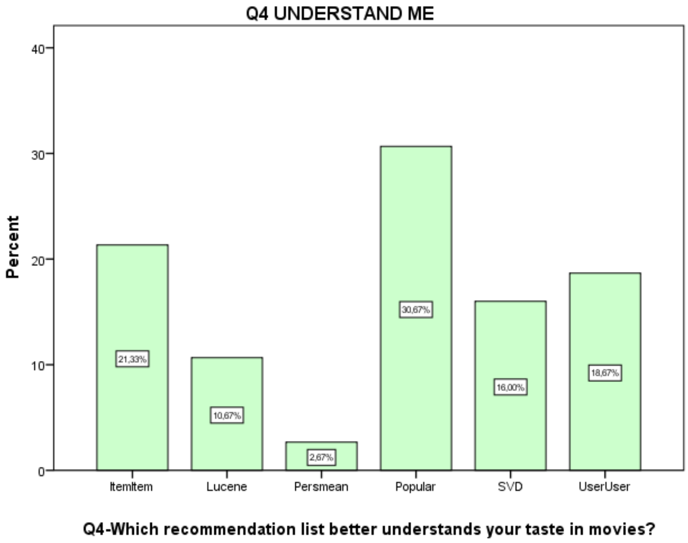


Figure 4-52: Bar diagram representing the data collected for Q4

Taking into account the genderof the users (Table 4-35) the assumption of the expected count is violated, so we have to look at the likelihood ratio to determine the p-value (p=0.176), it shows no dependence on the results.

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	7,585[a]	5	,181
Likelihood Ratio	7,665	5	,176
Linear-by-Linear Association	,000	1	,990
N of Valid Cases	75		

a. 4 cells (33, 3%) have expected count less than 5. The minimum expected count is, 85.

Table 4-35: Chi-squared test to analyse the differences between gender with α=0.05

If we now take into account age (Table 4-36), the assumption of the expected count is violated again, looking at the likelihood ratio it is remarkable (p=0.01) that people older than 25 opt for *Lucene* and *Popular* more than younger people. However, *Popular* in both ranges is the algorithm that best understands their taste, although collaborative filtering algorithms is highlighted too. However, note that the sample size is quite small. Therefore, we should not extrapolate these results.

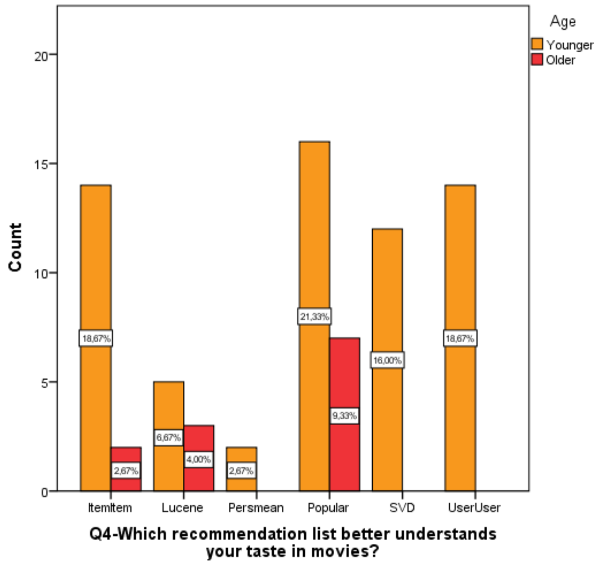


Figure 4-53: Distribution of the answers of Q4 by Age. Note that all the percentages are expressed taking into account the total number of users (N=50).

Q4UNDERSTANDME * Age Crosstabulation

			Age	
			Younger	Older
Q4UNDERSTANDME	*ItemItem*	Count	14	2
		Expected Count	13,4	2,6
	Lucene	Count	5	3
		Expected Count	6,7	1,3
	Persmean	Count	2	0
		Expected Count	1,7	,3
	Popular	Count	16	7
		Expected Count	19,3	3,7
	SVD	Count	12	0
		Expected Count	10,1	1,9
	UserUser	Count	14	0
		Expected Count	11,8	2,2
Total		Count	63	12
		Expected Count	63,0	12,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	11,796[a]	5	,038
Likelihood Ratio	15,042	5	,010
Linear-by-Linear Association	1,904	1	,168
N of Valid Cases	75		

a. 7 cells (58,3%) have expected count less than 5. The minimum expected count is 0,32.

Table 4-36: Chi-squared test to analyse the differences between age with α=0.05

In conclusion, without taking into account *Popular*, collaborative filtering algorithms are considered the algorithms that best understand users' taste. Although it is noted that people older than 25 have a better opinion about *Lucene* and *Personalized Mean* than young people. The reason is that young people are more influenced by the opinion of friends, family and other users while people older than 25 have their own taste more defined. Nevertheless, this result shows that even though these two algorithms create their recommendations based on user taste, the user does not perceive this.

Now, we shall analyse the results as a combination of Q3 and Q4. For this purpose, we will consider the answers obtained in the first question as negative ones for the algorithm -1 and the answers obtained in the second question as positive ones for the algorithm +1.

We can note (Figure 4-54) that *ItemItem* is the algorithm best considered in terms of understanding users' taste. Some users tend to like the same kind of movies, and this is why *ItemItem* works well with them. *Popular* is good considered, although is notable that there is some controversy in the results. Many people think that this is the algorithm that best understands them but we can also find a large group of people that think that this algorithm does not understand them. It has sense since, as we have seen, there are people who only like the same kind of movies, and *Popular* recommend movies of all different genres. Moreover, this also happens with *UserUser* and *SVD*. It is clear that *Persmean* is the worst considered regarding users' opinion.

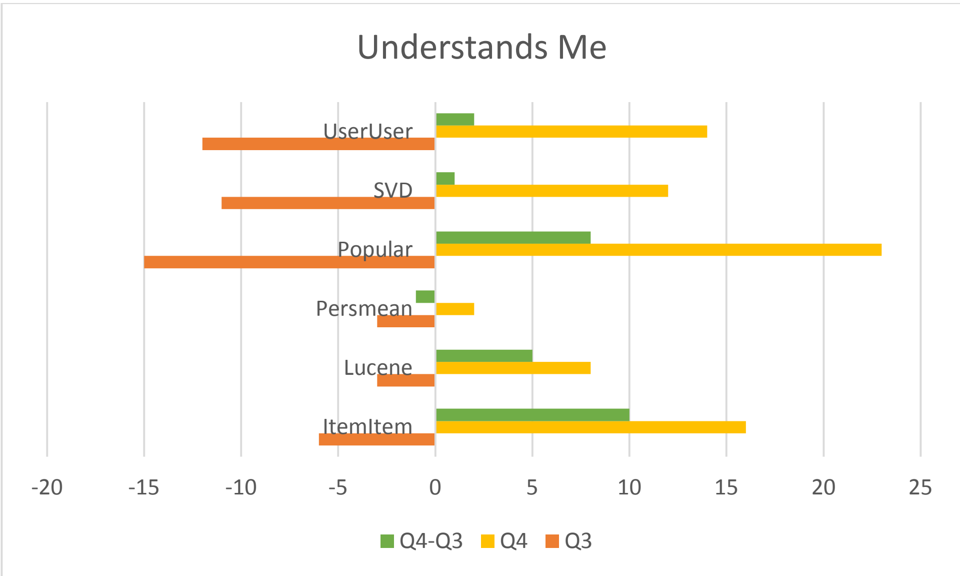


Figure 4-54: Combination of the two questions that measure Understands Me. The green bar is the result of the combination.

4.2.3.3 Variety / Diversity

In order to know which algorithm is really the one that recommends more diverse movies, we have asked our users three questions. Firstly, (Q5) we have asked for the algorithm that recommends more similar movies; then, (Q6) we have asked for the same issue but in an opposite manner; and finally, (Q7) we want to know the algorithm that recommends movies with more types of genres.

Q5- WHICH LIST HAS MORE MOVIES THAT ARE SIMILAR TO EACH OTHER?

Table 4-37 shows the choice of algorithms by each user, making a distinction between gender and age.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	12	7	5	10	2
ItemItem	11	6	5	10	1
UserUser	6	4	2	6	0
Lucene	8	4	4	5	3
SVD	7	5	2	4	3
Persmean	6	3	3	5	1

Table 4-37: Data collected from the questionnaire Q5

The chi-squared test (Table 4-38) tells us that there are not significant differences (p=0.549) among our algorithms. Users' opinion is highly divided among them. Thus, we can conclude nothing consistent from this question.

VARIETY/ *DIVERSITY*

	Observed N	Expected N	Residual
ItemItem	11	8,3	2,7
Lucene	8	8,3	-,3
Persmean	6	8,3	-2,3
Popular	12	8,3	3,7
SVD	7	8,3	-1,3
UserUser	6	8,3	-2,3
Total	50		

Table 4-38: Chi- squared test Q5 with α=0.05.

Test Statistics

	VARIETY/ *DIVERSITY*
Chi-Square	4,000[a]
df	5
Asymp. Sig.	,549

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 8,3.

Looking at Figure 4-55, we can observe, as we have said, that users' answers are highly matched.

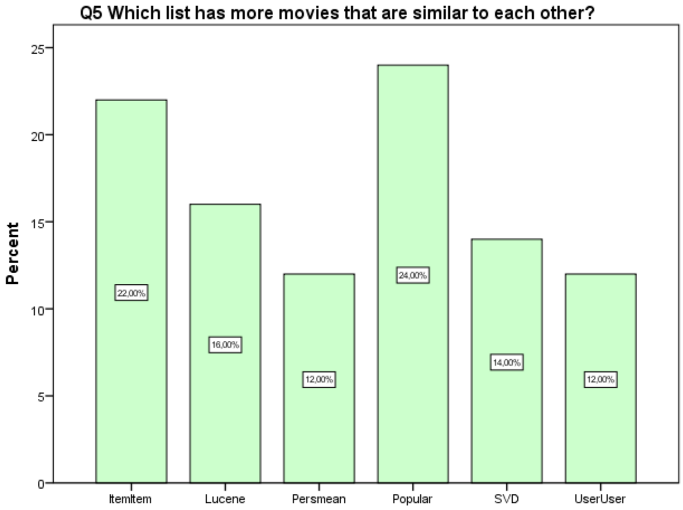


Figure 4-55: Bar diagram representing the data collected from the questionnaire Q5

Even if we look at the differences between gender and age (Table 4-39), the results are not significant. In both cases the assumption of the expected count is violated, so we have to look at the likelihood ratio to determine the p-value. In the case of the gender, the result is p=0.979 and in the case of the age, the result is p =0.165. Thus, we have to conclude that this question does not provide any information to us.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	,768[a]	5	,979
Likelihood Ratio	,768	5	,979
Linear-by-Linear Association	,051	1	,821
N of Valid Cases	50		

a. 10 cells (83,3%) have expected count less than 5. The minimum expected count is 2,28.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	7,407[a]	5	,192
Likelihood Ratio	7,852	5	,165
Linear-by-Linear Association	1,450	1	,228
N of Valid Cases	50		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is 1,20.

Table 4-39: Chi square test to analyse the differences between gender and age with α=0.05.

Q6- WHICH LIST HAS A LESS VARIED SELECTION OF MOVIES?

Table 4-40 shows the answers collected for this question separated by gender and age. We can appreciate that all the algorithms are almost equally voted. Therefore, we are going to check whether the differences are significant or not.

Algorithms	Users	By Gender		By Age	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	9	5	4	7	2
ItemItem	13	4	9	12	1
UserUser	11	6	5	8	3
Lucene	9	4	5	5	4
SVD	11	8	3	8	3
Persmean	10	4	6	9	1

Table 4-40: Data collected from the questionnaire Q6

As in the previous question, the chi-squared test (Table 4-41) tells us that there are not significant differences (p= 0.955) among users' opinions. All the algorithms are elected by approximately the same number of users, so that we cannot extrapolate the results.

Q6*DIVERSITY*

	Observed N	Expected N	Residual
ItemItem	13	10,5	2,5
Lucene	9	10,5	-1,5
Persmean	10	10,5	-,5
Popular	9	10,5	-1,5
SVD	11	10,5	,5
UserUser	11	10,5	,5
Total	63		

Test Statistics

	Q6*DIVERSITY*
Chi-Square	1,095[a]
df	5
Asymp. Sig.	,955

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 10,5.

Table 4-41: Chi- squared test Q6 with α=0.05

We can check it by looking at Figure 4-56, since all the bars approximately have the same height.

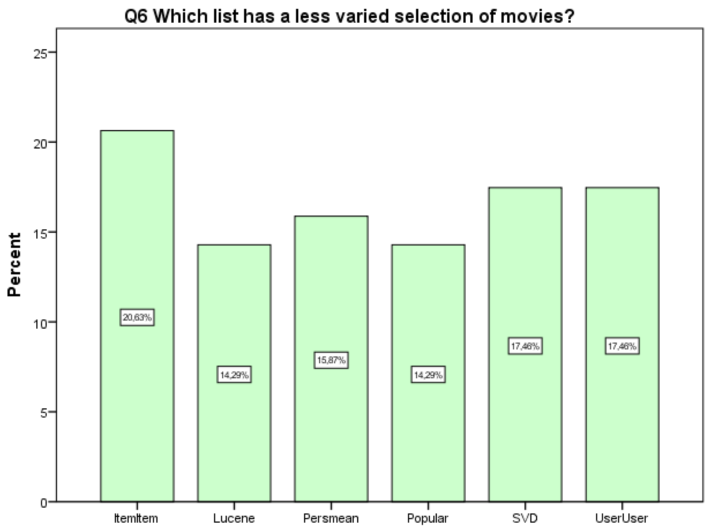


Figure 4-56: Bar diagram representing the data collected for Q6.

As with the previous question, in both cases the assumption of the expected count is violated, so we had had to look at the likelihood ratio to determine the p-value (Table 4-42). There are neither significant differences between men and women (p= 0.649) nor between old people and young people (p=0.241). Therefore, it is difficult to draw main conclusions departing from this results.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,268[a]	5	,659
Likelihood Ratio	3,329	5	,649
Linear-by-Linear Association	2,367	1	,124
N of Valid Cases	63		

a. 5 cells (41,7%) have expected count less than 5. The minimum expected count is 4,00.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	7,132[a]	5	,211
Likelihood Ratio	6,732	5	,241
Linear-by-Linear Association	,010	1	,918
N of Valid Cases	63		

a. 6 cells (50,0%) have expected count less than 5. The minimum expected count is 2,29.

Table 4-42: Chi square test to analyse the differences between gender and age with α=0.05.

Q7- WHICH LISTS DO YOU THINK THAT INCLUDE MOVIES OF MANY DIFFERENT GENRES?

Users' answers collected can be seen in Table 4-43:

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	17	10	7	12	5
ItemItem	3	2	1	2	1
UserUser	8	4	4	7	1
Lucene	11	6	5	10	1
SVD	9	5	4	7	2
Persmean	11	7	4	10	1

Table 4-43: Data collected from the questionnaire Q7.

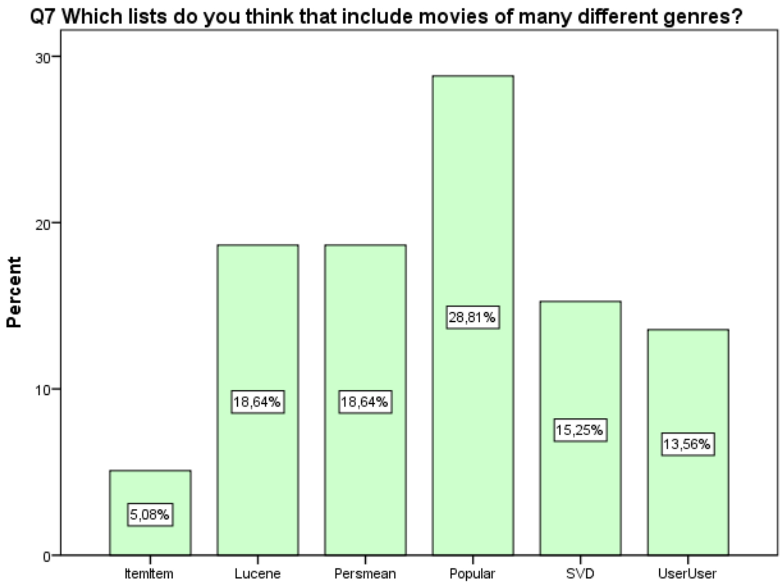


Figure 4-57: Bar diagram representing the data collected for Q7

As we can see in Figure 4-57, all the bars approximately have the same percentage of votes, what means that the users' opinion is divided among the six algorithms and this prevents us from extrapolating the results because they are not conclusive (p=0.059).

Q7*DIVERSITY*

	Observed N	Expected N	Residual
ItemItem	3	9,8	-6,8
Lucene	11	9,8	1,2
Persmean	11	9,8	1,2
Popular	17	9,8	7,2
SVD	9	9,8	-,8
UserUser	8	9,8	-1,8
Total	59		

Table 4-44: Chi squared test Q7

Test Statistics

	Q7*DIVERSITY*
Chi-Square	10,661[a]
df	5
Asymp. Sig.	,059

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 9,8.

As happened in previous questions, there are not significant differences between gender (p=0.959) nor age (p=0.735). We have looked again at the likelihood, since the assumption of the count cell is violated. (Table 4-45)

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	1,032[a]	5	,960
Likelihood Ratio	1,043	5	,959
Linear-by-Linear Association	,004	1	,949
N of Valid Cases	59		

a. 6 cells (50,0%) have expected count less than 5. The minimum expected count is 1,12.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,647[a]	5	,754
Likelihood Ratio	2,772	5	,735
Linear-by-Linear Association	,109	1	,741
N of Valid Cases	59		

a. 7 cells (58,3%) have expected count less than 5. The minimum expected count is ,61.

Table 4-45: Chi square test to analyse the differences between gender and age with α=0.05

Unfortunately, we cannot extrapolate the results obtained measuring the *Diversity* of algorithms because the answers are not conclusive.

Taking into account the results obtained from the chi-square formula in all algorithms, we realize that all of them are higher than 0.05, which means that they are not significant. Therefore, we conclude that those algorithms are equally diverse and all of them have the same function in terms of variety without any difference since users' opinion is randomly divided among them. Future studies should try to repeat these measures with a bigger amount of users.

4.2.3.4 Novelty

We try to measure the perception that the user has of *Novelty* in order to know whether it influences their opinion of the algorithm or not. Therefore, we have asked them four questions.

Q8 - WHICH LIST HAS MORE MOVIES YOU DO NOT EXPECT?

Table 4-46 shows the choice of algorithms by each user, making a distinction between gender and age.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	9	4	5	9	0
ItemItem	4	2	2	4	0
UserUser	5	2	3	3	2
Lucene	19	11	8	17	2
SVD	8	4	4	6	2
Persmean	23	12	11	18	5

Table 4-46: Data collected from the questionnaire Q8

If we have a look at Figure 4-58, we can see two algorithms that stand out from the rest ($p \approx 0.000$). These are *Persmean* with a 37.5% and *Lucene* with a 30.36%. Both algorithms are the less preferred by users. Moreover, these are the ones that recommend more movies that do not fit users' taste. In contrast, the collaborative filtering algorithms by *Item* and by *User* have the smaller percentage, what means that their recommendations do not surprise the users since they match their preferences.

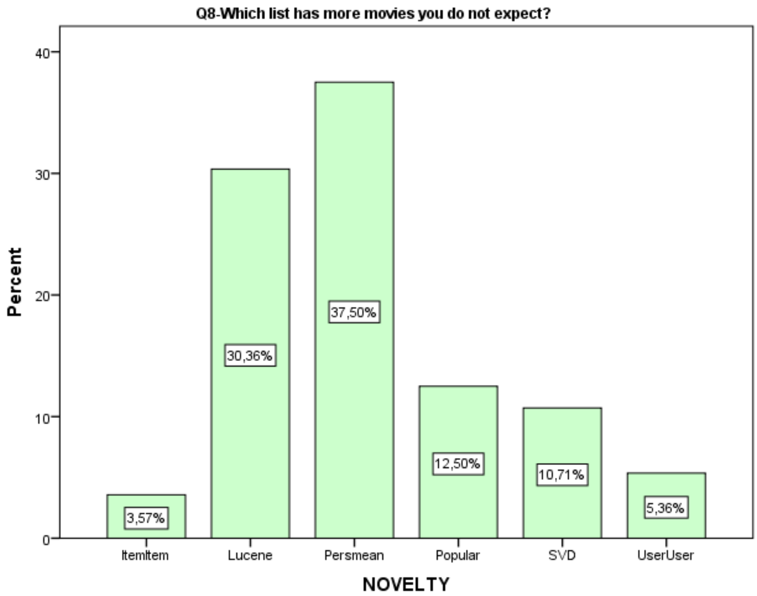


Figure 4-58: Bar diagram representing the data collected for Q8

Q8*NOVELTY*

	Observed N	Expected N	Residual
ItemItem	2	9,3	-7,3
Lucene	17	9,3	7,7
Persmean	21	9,3	11,7
Popular	7	9,3	-2,3
SVD	6	9,3	-3,3
UserUser	3	9,3	-6,3
Total	56		

Test Statistics

	Q8*NOVELTY*
Chi-Square	32,714[a]
df	5
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 9,3.

Table 4-47: Chi square test Q8 with α=0.05.

Nevertheless, we cannot make distinctions based on gender (p=0.850) since the results are not significant. Moreover, we can neither make distinctions based on the age (p=0.253). Thus, we cannot extrapolate the results taking into account gender or age (Table 4-48). Take into account that in both cases the assumption of the expected count is violated, so we have looked at the likelihood ratio to determine the p-value.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig (2-sided)
Pearson Chi-Square	1,342[a]	5	,931
Likelihood Ratio	1,994	5	,850
Linear-by-Linear Association	,117	1	,732
N of Valid Cases	56		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is ,71.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	4,998[a]	5	,416
Likelihood Ratio	6,586	5	,253
Linear-by-Linear Association	,996	1	,318
N of Valid Cases	56		

a. 9 cells (75,0%) have expected count less than 5. The minimum expected count is ,39.

Table 4-48: Chi square test to analyse the differences between gender and age with α=0.05

Q9 - WHICH LIST HAS MORE MOVIES THAT ARE FAMILIAR TO YOU?

The answers collected from the users are shown in Table 4-49.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	20	9	11	14	6
ItemItem	15	7	8	13	2
UserUser	14	8	6	13	1
Lucene	6	5	1	4	2
SVD	12	6	6	9	3
Persmean	4	1	3	4	0

Table 4-49: Data collected from the questionnaire Q9

In Figure 4-59, we can see that *Popular* is the algorithm that recommends more movies that are familiar to the user (p=0.011), followed by the collaborative filtering algorithms by Item and by User respectively with nearly a 20%. However, with less than a 10%, *Lucene* and *Personalized Mean* recommend the less familiar movies to the users. *Lucene* and *Persmean* are based on user taste, without taking into account what other users with similar taste like. On the contrary *ItemItem*, *UserUser* and *Popular* only have taken into account other users' preferences to make the recommendations.

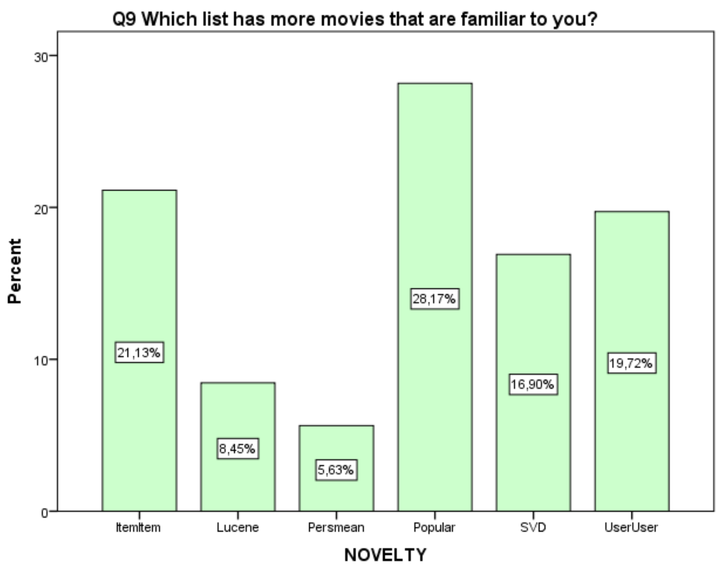


Figure 4-59: Bar diagram representing the data collected for Q9

Q9*NOVELTY*

	Observed N	Expected N	Residual
ItemItem	15	11,8	3,2
Lucene	6	11,8	-5,8
Persmean	4	11,8	-7,8
Popular	20	11,8	8,2
SVD	12	11,8	,2
UserUser	14	11,8	2,2
Total	71		

Test Statistics

	Q9*NOVELTY*
Chi-Square	14,944[a]
df	5
Asymp. Sig.	,011

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 11,8.

Table 4-50: Chi square test Q9 with α=0.05

Looking at the selections made by men and women, we see that there are not significant differences (p=0.481) in the results, so we cannot highlight the differences in gender. This also happens between people younger than 25 and older people (p=0.641). As in previous questions, we have looked at the likelihood ratio to determine the p-value, since the assumption of the count cell is violated, Table 4-51.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	4,227[a]	5	,517
Likelihood Ratio	4,492	5	,481
Linear-by-Linear Association	,005	1	,943
N of Valid Cases	71		

a. 4 cells (33,3%) have expected count less than 5. The minimum expected count is 1,80.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,601[a]	5	,761
Likelihood Ratio	3,387	5	,641
Linear-by-Linear Association	,005	1	,944
N of Valid Cases	71		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is ,85.

Table 4-51: Chi square test to analyse the differences between gender and age with α=0.0

Q10 - WHICH LIST HAS MORE PLEASANTLY SURPRISING MOVIES?

Looking at Table 4-53, we can see the users' opinion. There are three algorithms that are outstanding since more than 10 users have selected them. The chi-squared test (Table 4-52) proves that these differences are significant (p=0.01).

Q10*NOVELTY*

	Observed N	Expected N	Residual
ItemItem	8	11,5	-3,5
Lucene	10	11,5	-1,5
Persmean	3	11,5	-8,5
Popular	20	11,5	8,5
SVD	15	11,5	3,5
UserUser	13	11,5	1,5
Total	69		

Test Statistics

	Q10*NOVELTY*
Chi-Square	15,087[a]
df	5
Asymp. Sig.	,010

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 11,5.

Table 4-52: Chi square test Q10 with α=0.05

Algorithms	Users	By Gender		By Age	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	20	9	11	16	4
ItemItem	8	3	5	7	1
UserUser	13	8	5	13	0
Lucene	10	5	5	6	4
SVD	15	10	5	12	3
Persmean	3	2	1	3	0

Table 4-53: Data collected from the questionnaire Q10.

Popular with 28.99% is the algorithm with more pleasantly surprising movies for the users, followed by *SVD* with 21.74% and *UserUser* with 18.84% (Figure 4-60). As we have seen previously, people have a very good opinion of *Popular*, the movies recommended meet users expectations, and sometimes it might happen that some popular movies have been overlooked by the user and when he reads the title of the movie he realises that he likes it and he wants to watch it. *SVD* and *UserUser* are both collaborative filtering algorithms, both have taken into account what other users with similar taste like to make recommendations and this is why some of this recommendations can be surprising. In contrast, the algorithm with less pleasantly surprising movies is *Persmean*, although in Q9 it was ranked by the users as the algorithm with more movies that they do not expect to be there. It means that *Persmean* has a high level of *Novelty* but in a negative way.

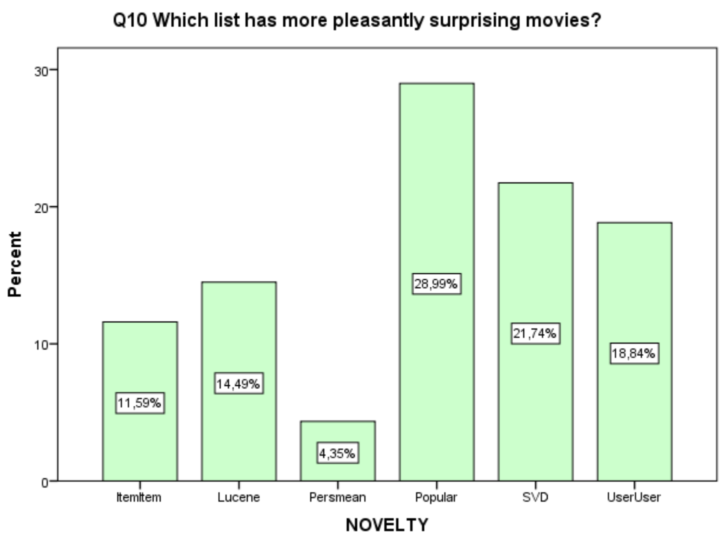


Figure 4-60: Bar diagram representing the data collected for Q10.

When we try to check the differences between gender and age (Table 4-54) the chi-squared test (looking at the likelihood ratio since the assumption of the count cell is violated) shows that the results are not significant. However, we can note a subtle disagreement between men and women in the opinion about *SVD* and *ItemItem* since more women than men prefer *SVD* while more men than women prefer *ItemItem*, although the differences are nearly unnoticeable (p=0.811). Between people older than 25 and younger the differences are apparently insignificant (p= 0.155) although, as in the case of gender, we can note a subtle difference with *Popular* that is preferred by the old people and *UserUser* by the young people. Nevertheless, we cannot extrapolate these results.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,257[a]	5	,813
Likelihood Ratio	2,266	5	,811
Linear-by-Linear Association	1,556	1	,212
N of Valid Cases	69		

a. 5 cells (41,7%) have expected count less than 5. The minimum expected count is 1,30.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	5,881[a]	5	,318
Likelihood Ratio	8,022	5	,155
Linear-by-Linear Association	1,480	1	,224
N of Valid Cases	69		

a. 7 cells (58,3%) have expected count less than 5. The minimum expected count is ,48.

Table 4-54: Chi square test to analyse the differences between gender and age with α=0.05

Q11 - WHICH LIST HAS MORE MOVIES YOU WOULD NOT HAVE THOUGHT TO CONSIDER?

The data collected is shown in Table 4-55, making distinctions both between the opinion of men and women and between young and old people.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	6	5	1	5	1
ItemItem	5	4	1	4	1
UserUser	6	5	1	5	1
Lucene	15	5 11	4	12	3
SVD	8	7	1	6	2
Persmean	25	11	14	18	7

Table 4-55: Data collected from the questionnaire Q11

Looking at Table 4-56, we can see that there are huge differences among algorithms (p≈0.000), underscoring *Persmean* with 38.46% and *Lucene* with 23.08% while *ItemItem*, *UserUser* and *Popular* are the algorithms with the lower percentage. In this question, users have remarked these algorithms that recommend movies that do not match their preferences since, as we have already seen in other questions, *Persmean* and *Lucene* are the algorithms least valued by the users. The data shows that they have understood the question with a negative connotation, opting for those algorithms that recommend movies that they would not have considered because they do not like these type of movies.

Q11*NOVELTY*

	Observed N	Expected N	Residual
ItemItem	5	10,8	-5,8
Lucene	15	10,8	4,2
Persmean	25	10,8	14,2
Popular	6	10,8	-4,8
SVD	8	10,8	-2,8
UserUser	6	10,8	-4,8
Total	65		

Table 4-56: Chi square test Q11 with α=0.05.

Test Statistics

	Q11*NOVELTY*
Chi-Square	28,323[a]
df	5
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 10,8.

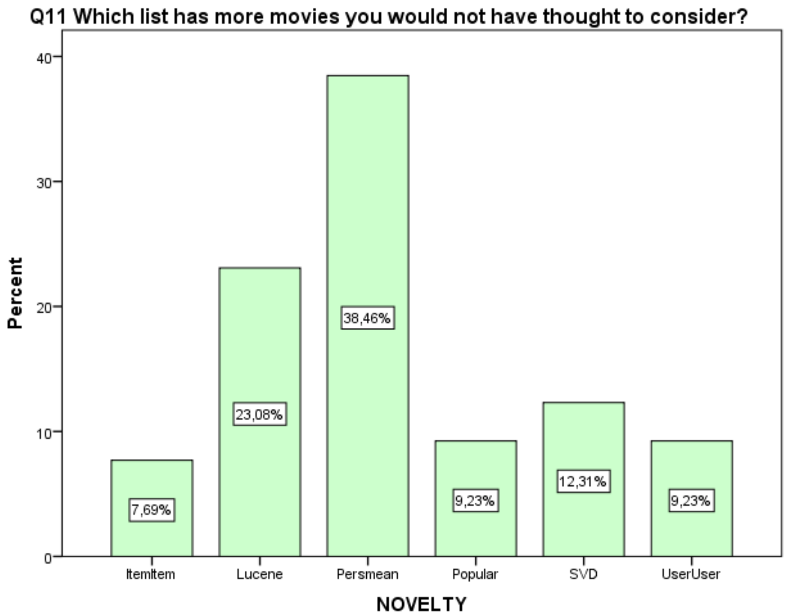


Figure 4-61: Bar diagram representing the data collected for Q11.

As with the previous questions, we cannot make distinctions between men and women options (p= 0.271) since the chi-squared test (Table 4-57) shows that the results are not significant. This also happens when we try to check age (p=0.485). Take into account that in both cases the assumption of the expected count is violated (66.2% of the cells have expected count less than 5), so we had had to look at the likelihood ratio to determine the p-value.

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	6,249[a]	5	,283
Likelihood Ratio	6,381	5	,271
Linear-by-Linear Association	,797	1	,372
N of Valid Cases	65		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is 1,54.

Chi-Square Tests by Age

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,184[a]	5	,672
Likelihood Ratio	4,463	5	,485
Linear-by-Linear Association	,096	1	,757
N of Valid Cases	65		

a. 8 cells (66,7%) have expected count less than 5. The minimum expected count is 1,15.

Table 4-57: Chi square test to analyse the differences between gender and age with α=0.05

4.2.3.5 Effectiveness

Q12 - WHICH LIST GIVES YOU MORE VALUABLE RECOMMENDATIONS?

The results obtained are shown in Table 4-58, where we can also see the opinion divided by gender and age.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	20	6	14	14	6
ItemItem	22	14	8	21	1
UserUser	17	12	5	15	2
Lucene	8	5	3	4	4
SVD	14	8	6	12	2
Persmean	4	2	2	3	1

Table 4-58: Data collected from the questionnaire Q12

Observing the data, Table 4-59, we can see two well differentiated groups (p=0.004). The first group includes *ItemItem*, *Popular*, *UserUser* and *SVD* while the second group includes *Lucene* and *Persmean*.

Q12*EFFECTIVENESS*

	Observed N	Expected N	Residual
ItemItem	22	14,2	7,8
Lucene	8	14,2	-6,2
Persmean	4	14,2	-10,2
Popular	20	14,2	5,8
SVD	14	14,2	-,2
UserUser	17	14,2	2,8
Total	85		

Test Statistics

	Q12*EFFECTIVENESS*
Chi-Square	17,282[a]
df	5
Asymp. Sig.	,004

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 14,2.

Table 4-59: Chi square test Q12 with α=0.05.

Within the first group, *ItemItem* with 25.88% is the algorithm that gives the users more valuable recommendations. Although the difference with *Popular* (23.53%) is not huge, both of them are algorithms with a high *Effectiveness*. The reason is that people find effectiveness in those movies which are similar to the ones that they have watched or the ones that are well-known. In contrast, the algorithms with less valuable recommendations are *Persmean* and *Lucene*, since the movies recommended by these algorithms are not always familiar to the users.

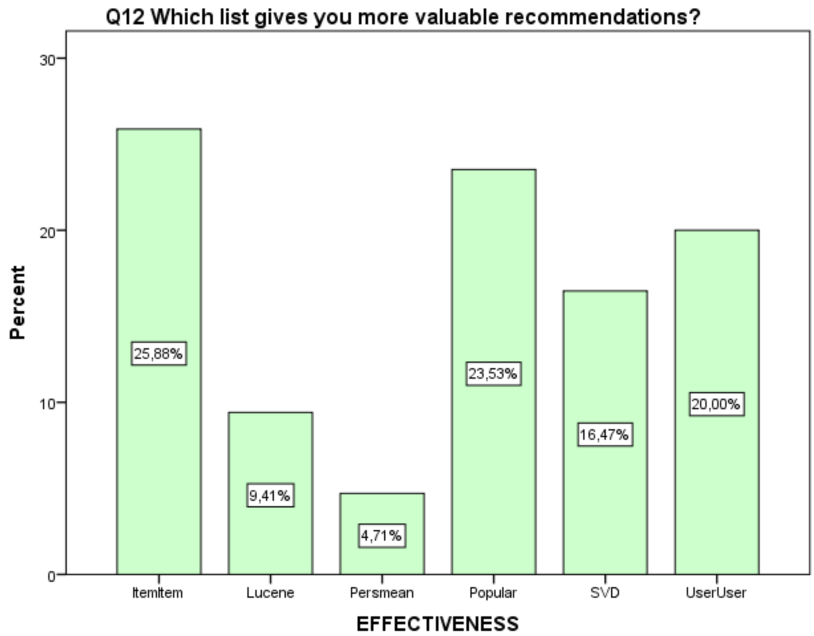


Figure 4-62: Bar diagram representing the data collected for Q12

The differences between women and men are not significant (p =0.159). However, we can make significant distinctions between young people and old people (p=0.028). In both cases the assumption of the expected count is violated, so we had had to look at the likelihood ratio to determine the p-value, Table 4-60. People younger than 25 find more valuable the recommendations made by *ItemItem*, *UserUser* and *SVD*, which are collaborative filtering algorithms. The reason is that young people are more influenced by their surroundings than old people who have their own taste more defined and this is why old people find more valuable the recommendations made by *Lucene* or *Popular*. However, note that the sample size is quite small. Therefore, we should not extrapolate these results.

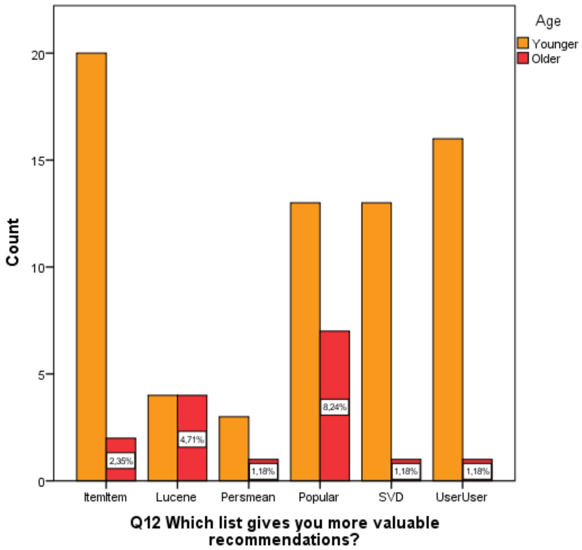


Figure 4-63: Users answers making a distinction by age

Q1*EFFECTIVENESS* * Gender Crosstabulation

			Gender	
			Woman	Man
Q1*EFFECTIVENESS*	*ItemItem*	Count	15	7
		Expected Count	12,9	9,1
	Lucene	Count	5	3
		Expected Count	4,7	3,3
	Persmean	Count	2	2
		Expected Count	2,4	1,6
	Popular	Count	7	13
		Expected Count	11,8	8,2
	SVD	Count	8	6
		Expected Count	8,2	5,8
	UserUser	Count	13	4
		Expected Count	10,0	7,0
Total		Count	50	35
		Expected Count	50,0	35,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	7,857[a]	5	,164
Likelihood Ratio	7,953	5	,159
Linear-by-Linear Association	,006	1	,940
N of Valid Cases	85		

a. 4 cells (33,3%) have expected count less than 5. The minimum expected count is 1,65.

Q1*EFFECTIVENESS* * Age Crosstabulation

			Age	
			Younger	Older
Q1*EFFECTIVENESS*	*ItemItem*	Count	20	2
		Expected Count	17,9	4,1
	Lucene	Count	4	4
		Expected Count	6,5	1,5
	Persmean	Count	3	1
		Expected Count	3,2	,8
	Popular	Count	13	7
		Expected Count	16,2	3,8
	SVD	Count	13	1
		Expected Count	11,4	2,6
	UserUser	Count	16	1
		Expected Count	13,8	3,2
Total		Count	69	16
		Expected Count	69,0	16,0

Chi-Square Tests

	Value	df	Asymp. Sig (2-sided)
Pearson Chi-Square	13,091[a]	5	,023
Likelihood Ratio	12,519	5	,028
Linear-by-Linear Association	,508	1	,476
N of Valid Cases	85		

a. 7 cells (58,3%) have expected count less than 5. The minimum expected count is ,75.

Table 4-60: Chi square test to analyse the differences between gender and age with α=0.05

Q13 - DO YOU THINK THAT THE RECOMMENDER IS RECOMMENDING INTERESTING CONTENT YOU HADN'T PREVIOUSLY CONSIDER?

Table 4-62 shows the data collected from the questionnaire. We have run a Friedman Test to obtain the rank of our algorithms. We can see, Table 4-61, that there is an overall statistically significant difference (p≈0.000) depending on which algorithm we evaluate. With this, we can only know that there are overall differences, but we do not know which particular algorithm differs from each other.

Ranks

	Mean Rank
Q13Item*EFFECTIVENESS*	3,90
Q13*LuceneEFFECTIVENESS*	2,95
Q13Pers*EFFECTIVENESS*	2,45
Q13*PopularEFFECTIVENESS*	3,76
Q13*SVDEFFECTIVENESS*	4,12
Q13User*EFFECTIVENESS*	3,82

Test Statistics[a]

N	50
Chi-Square	42,695
df	5
Asymp. Sig.	,000

a. Friedman Test

Table 4-61: Friedman Test to analyse the differences observed in users answers

	ItemItem	***Lucene***	***Persmean***	***Popular***	***SVD***	***UserUser***
No, nothing out of the ordinary	5	12	18	10	5	7
Somewhat out of the ordinary	16	20	21	13	18	16
Quite a bit surprisingly good movies	19	10	7	11	12	17
Fairly surprisingly good movies	7	8	3	12	11	7
Yes, there are lots of surprisingly good movies	3	0	1	4	4	3

Table 4-62: Data collected from the questionnaire for Q13

To find out which algorithms differ from the other we have to look at the results obtain by the post hoc analysis with Wilcoxon Signed Rank test. Looking at Table 4-63 we can see that only there are statistically significant differences with *Lucene* and *Persmean*. *Lucene* recommend more interesting content than *Persmean* (p=0.048). *SVD*, *ItemItem*, *UserUser* and *Popular* are above *Lucene*, recommending more interesting movies to the user but we cannot make a rank with these algorithms since the noted differences are not significant. To clarify this we can take a look at Figure 4-64.

Test Statistics[a]

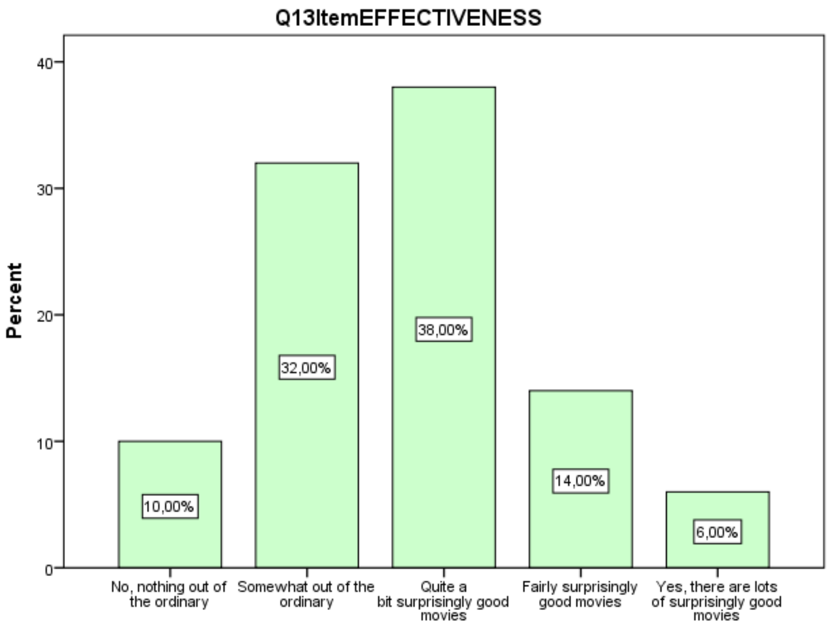
	Lucene - Item	Persmean-Item	Popular-Item	SVD-Item	User-Item	Persmean-Lucene	Popular-Lucene	SVD-Lucene	User-Lucene	Popular-Persmean	SVD-Persmean	User-Persmean	SVD-Popular	User-Popular	User-SVD
Z	-2,386[b]	-3,851[b]	-,143[c]	-,520[c]	-,494[b]	-1,973[b]	-2,283[c]	-2,623[c]	-1,834[c]	-3,582[c]	-4,375[c]	-3,710[c]	-,364[c]	-,621[b]	-1,452[b]
Asymp.Sig.(2-tailed)	,017	,000	,886	,851	,621	,048	,022	,009	,067	,000	,000	,000	,716	,535	,146

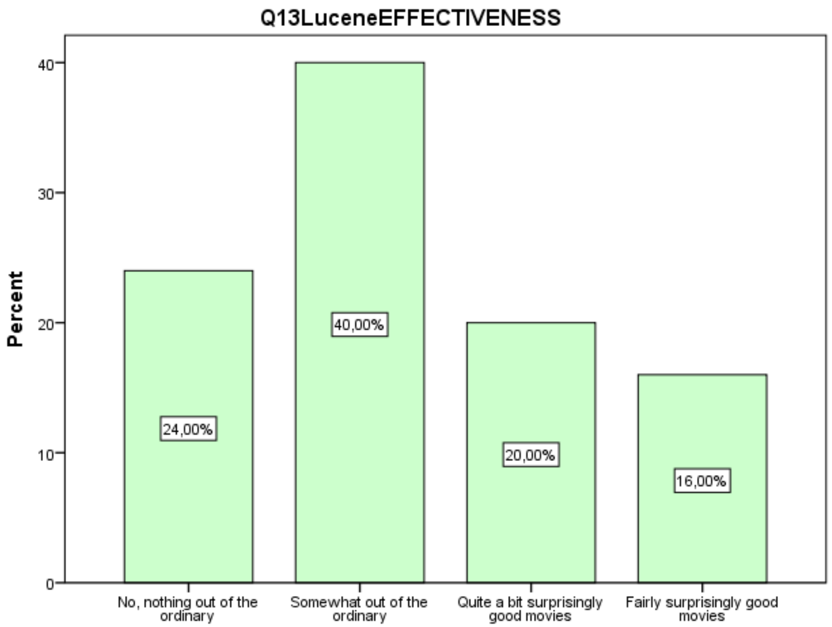
a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 4-63: Wilcoxon signed Rank Test to measure how different is each algorithm from the others





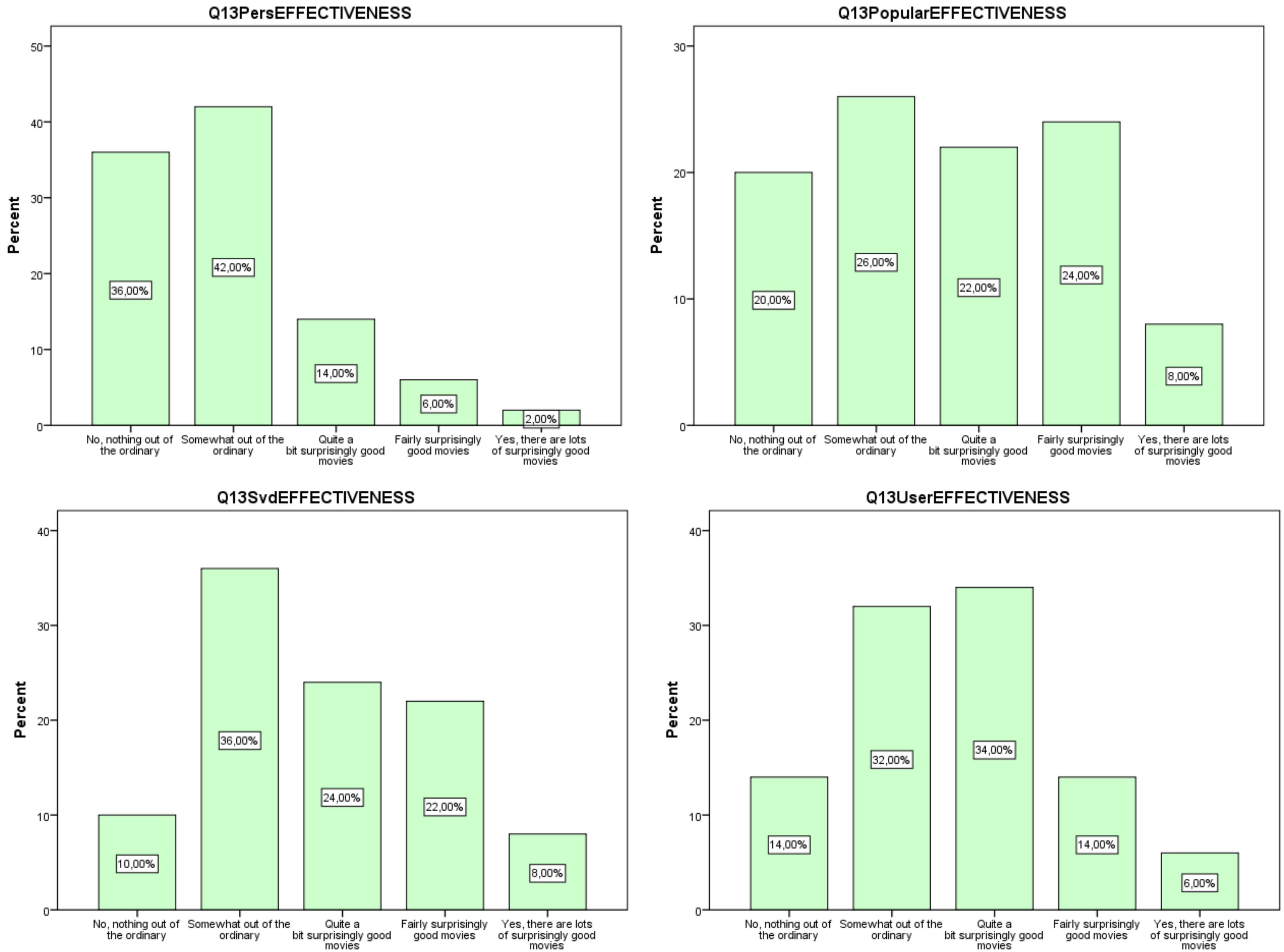


Figure 4-64: Users answers for each algorithm.

Q14 - CONSIDERING THE BEST RECOMMENDATION LIST IN YOUR OPINION, DO YOU SAVE TIME USING THE RECOMMENDER TO CHOOSE A MOVIE COMPARED TO YOUR USUAL WAY OF SELECTING MOVIES?

Rank	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
1:No, nothing	0	0	0	0	0
2: Not so much	7	6	1	6	1
3: I don't know	18	13	5	14	4
4: Yes, is a bit useful	19	8	11	16	3
5: Yes is very useful	6	2	4	4	2

Table 4-64: Data collected from the questionnaire Q14

First of all, looking at Table 4-64, we should underline that nobody think that the recommender is useless. Applying the chi-squared test, Table 4-65, we have seen the results are significant (p=0.000). Once we know it we can take a look at the bar diagram.

Q14*EFFECTIVENESS*

	Ranking Category	Observed N	Expected N	Residual
No, nothing	1	0	10,0	-10,0
Not so much	2	7	10,0	-3,0
I don't know	3	18	10,0	8,0
A bit useful	4	19	10,0	9,0
Yes, is very useful	5	6	10,0	-4,0
Total		50		

Test Statistics

	Ranking
Chi-Square	27,000[a]
df	4
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 10,0.

Table 4-65: Chi square test Q14 with α=0.05

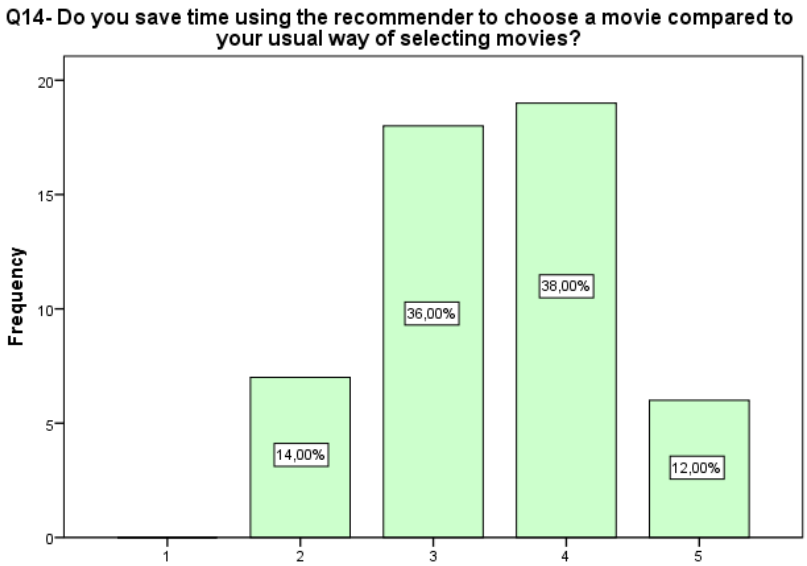


Figure 4-65: Bar diagram representing the data collected for Q14

It is clear that users' opinion is divided between option 3 and 4, what means that near a 40% find a bit useful the recommender to select a movie to watch since they save time using it, while other 40% of the users do not know if the recommender is useful to save time or not because they spend the same time using the recommender or looking for a movie by themselves.

In this question, we have not found significant differences between men and women (p=0.111) neither between young and old people (p=0.907). In both cases we have looked at the likelihood ratio to determine the p-value, since the number of cells that have an expected count lower than 5 is higher than 20%, Table 4-66.

Rank * Gender Crosstabulation

			Gender		
			Women	Men	Total
Rank	1	Count	0	0	0
		Expected Count	,0	,0	,0
	2	Count	6	1	7
		Expected Count	4,1	2,9	7,0
	3	Count	13	5	18
		Expected Count	10,4	7,6	18,0
	4	Count	8	11	19
		Expected Count	11,0	8,0	19,0
	5	Count	2	4	6
		Expected Count	3,5	2,5	6,0
Total		Count	29	21	50
		Expected Count	29,0	21,0	50,0

Chi-Square Tests by Gender

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	7,171[a]	4	,127
Likelihood Ratio	7,515	4	,111
Linear-by-Linear Association	6,558	1	,010
N of Valid Cases	50		

a. 6 cells (60,0%) have expected count less than 5. The minimum expected count is ,00.

Rank * Age Crosstabulation

			Age		
			Young	Old	Total
Rank	1	Count	0	0	0
		Expected Count	,0	,0	,0
	2	Count	6	1	7
		Expected Count	5,6	1,4	7,0
	3	Count	14	4	18
		Expected Count	14,4	3,6	18,0
	4	Count	16	3	19
		Expected Count	15,2	3,8	19,0
	5	Count	4	2	6
		Expected Count	4,8	1,2	6,0
Total		Count	40	10	50
		Expected Count	40,0	10,0	50,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	1,076[a]	4	,898
Likelihood Ratio	1,017	4	,907
Linear-by-Linear Association	,229	1	,632
N of Valid Cases	50		

a. 7 cells (70,0%) have expected count less than 5. The minimum expected count is ,00.

Table 4-66: Chi square test to analyse the differences between gender and age with α=0.05

4.2.3.6 Quality

To measure the *Quality* of our algorithms, we have asked our users three questions: the first of them (Q15) looks for the algorithm that recommends more movies that fit their preferences; the second question (Q16) looks for the relevance of the movies recommended by each algorithm; and the last question (Q17) tries to find out whether the recommended movies by each algorithm are well-chosen or not.

Q15 - WHICH LIST HAS MORE MOVIES THAT FIT/MATCH YOUR PREFERENCE?

Table 4-67 shows the data collected from the questionnaire. We can firstly see that there is a huge difference between *Popular* and *Persmean* since one of them is the algorithm that best matches the preferences of the users while the other one does not fit any preference at all.

Algorithms	*Users*	*By Gender*		*By Age*	
	Total (N)	Women (N)	Men (N)	Younger 25 (N)	Older 25 (N)
Popular	21	6	15	15	6
ItemItem	12	8	4	11	1
UserUser	10	8	2	9	1
Lucene	3	3	0	2	1
SVD	4	4	0	3	1
Persmean	0	0	0	0	0

Table 4-67: Data collected from the questionnaire Q15

Furthermore, the chi-squared test, Table 4-68, tells us that the results are significant (p≈0.000). Thus, we can see two differentiated groups. There are three algorithm that do not fit users' preferences, *Persmean*, *Lucene* and *SVD*. Contrarily, *Popular* is the one which best does it (42%), followed by the collaborative filtering algorithms *ItemItem* and *UserUser* with more than 20%.

Q15*QUALITY*

	Category	Observed N	Expected N	Residual
1	*ItemItem*	12	8,3	3,7
2	*Lucene*	3	8,3	-5,3
3	*Persmean*	0	8,3	-8,3
4	*Popular*	21	8,3	12,7
5	*SVD*	4	8,3	-4,3
6	*UserUser*	10	8,3	1,7
Total		50		

Test Statistics

	Algorithm
Chi-Square	35,200[a]
df	5
Asymp. Sig.	,000

a. 0 cells (0,0%) have expected frequencies less than 5. The minimum expected cell frequency is 8,3.

Table 4-68: Chi square test Q14 with α=0.05

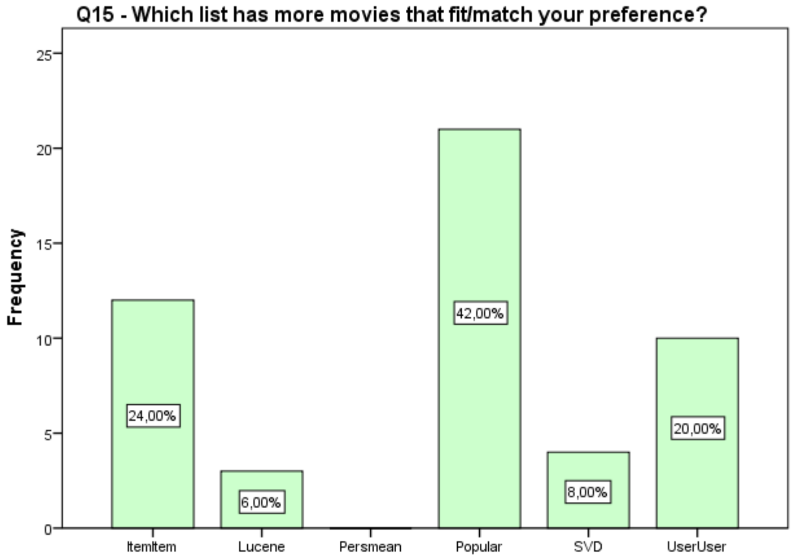


Figure 4-66: Bar diagram representing the data collected for Q15

As it can be seen on Table 4-69, there are significant differences (p=0.003) between men and women. Due to the fact that the 58.3% of cells have a count lower of 5, we have looked at the likelihood ratio to determine the p-value.

With regard to *Popular*, we can see that a higher percentage of men have chosen it. In contrast, a higher percentage of women have chosen *UserUser* and *Lucene*, which indicates that women have predefined preferences and they value the algorithms which recommend more similar movies. It has to be noted that no one values *Persmean* as an algorithm that fits their taste.

Algorithms * Gender Crosstabulation

			Gender		
			Women	Men	Total
Algorithms	*ItemItem*	Count	8	4	12
		Expected Count	7,0	5,0	12,0
	Lucene	Count	3	0	3
		Expected Count	1,7	1,3	3,0
	Persmean	Count	0	0	0
		Expected Count	,0	,0	,0
	Popular	Count	6	15	21
		Expected Count	12,2	8,8	21,0
	SVD	Count	4	0	4
		Expected Count	2,3	1,7	4,0
	UserUser	Count	8	2	10
		Expected Count	5,8	4,2	10,0
Total		Count	29	21	50
		Expected Count	29,0	21,0	50,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	14,892[a]	5	,011
Likelihood Ratio	17,617	5	,003
Linear-by-Linear Association	,005	1	,944
N of Valid Cases	50		

a. 7 cells (58,3%) have expected count less than 5. The minimum expected count is ,00.

Table 4-69: Chi square test to analyse the differences between gender with α=0.05

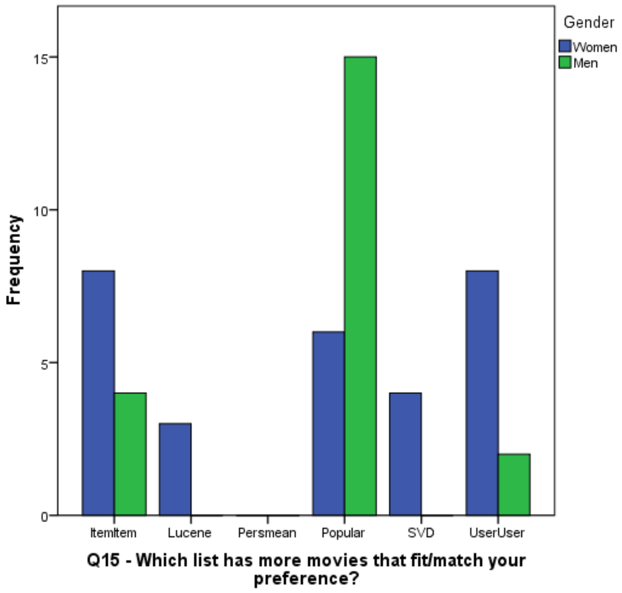


Figure 4-67: Answers Q15 making a distinction by gender

Taking into account the age, as in previous questions, we have to look at the likelihood ratio to determine the p-value, Table 4-70. We cannot make a distinction since the results are not significant (p= 0.668). The observed differences are very small.

Algorithms * Age Crosstabulation

			Age		
			Young	Old	Total
Algorithms	*ItemItem*	Count	11	1	12
		Expected Count	9,6	2,4	12,0
	Lucene	Count	2	1	3
		Expected Count	2,4	,6	3,0
	Persmean	Count	0	0	0
		Expected Count	,0	,0	,0
	Popular	Count	15	6	21
		Expected Count	16,8	4,2	21,0
	SVD	Count	3	1	4
		Expected Count	3,2	,8	4,0
	UserUser	Count	9	1	10
		Expected Count	8,0	2,0	10,0
Total		Count	40	10	50
		Expected Count	40,0	10,0	50,0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,006[a]	5	,699
Likelihood Ratio	3,209	5	,668
Linear-by-Linear Association	,100	1	,752
N of Valid Cases	50		

a. 9 cells (75,0%) have expected count less than 5. The minimum expected count is ,00.

Table 4-70: Chi square test to analyse the differences between age with α=0.05

Q16 - HOW MUCH DO YOU THINK THAT THE RECOMMENDED MOVIES ARE RELEVANT?

Table 4-71 shows the data collected and the results from the chi-squared test for each algorithm.

Test Statistics		***ItemItem***	***Lucene***	***Persmean***	***Popular***	***SVD***	***UserUser***
Not relevant at all	Observed N	2	10	13	5	3	2
	Expected N	10	10	10	10	10	10
	Residual	-8	0	3	-5	-7	-8
Of little relevant	Observed N	8	12	23	9	14	14
	Expected N	10	10	10	10	10	10
	Residual	-2	2	13	-1	4	4
Moderately relevant	Observed N	14	20	11	13	16	16
	Expected N	10	10	10	10	10	10
	Residual	4	10	1	3	6	6
Relevant	Observed N	23	5	2	13	14	15
	Expected N	10	10	10	10	10	10
	Residual	13	-5	-8	3	4	5
Very relevant	Observed N	3	3	1	10	3	3
	Expected N	10	10	10	10	10	10
	Residual	-7	-7	-9	0	-7	-7
Chi-Square		30.2	17.8	32.4	4.4	16.6	19
df		4	4	4	4	4	4
Asymp.Sig.		0.000	0.001	0.000	0.355	0.002	0.001

Table 4-71: Data collected from the questionnaire Q16 and chi square test

We have run a Friedman test, Table 4-72, to obtain the rank of our algorithms. We can see that there is an overall statistically significant difference (p≈0.000), depending on the algorithm which we evaluate. With this measure, we only know that there are overall differences, but we do not know which particular algorithm differs from the other.

Ranks

	Mean Rank
Q16Item*QUALITY*	4,38
Q16*LuceneQUALITY*	2,91
Q16*PersmeanQUALITY*	2,17
Q16*PopularQUALITY*	4,03
Q16*SVDQUALITY*	3,69
Q16User*QUALITY*	3,82

Test Statistics[a]

N	50
Chi-Square	56,460
df	5
Asymp. Sig.	,000

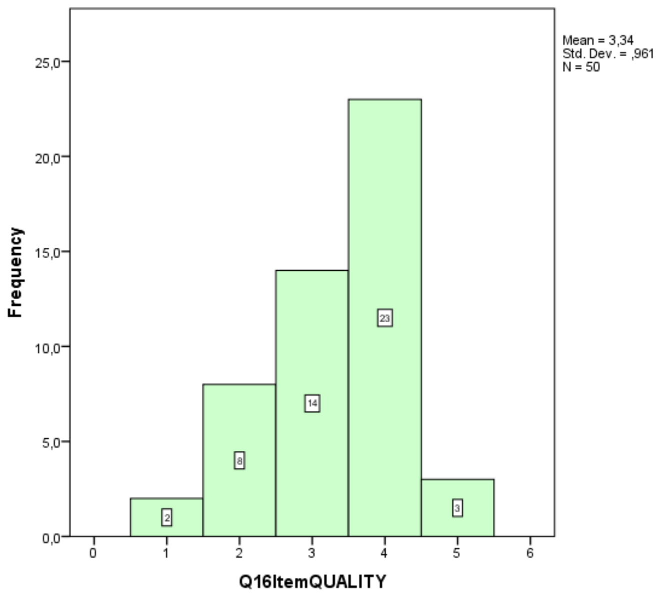
a. Friedman Test

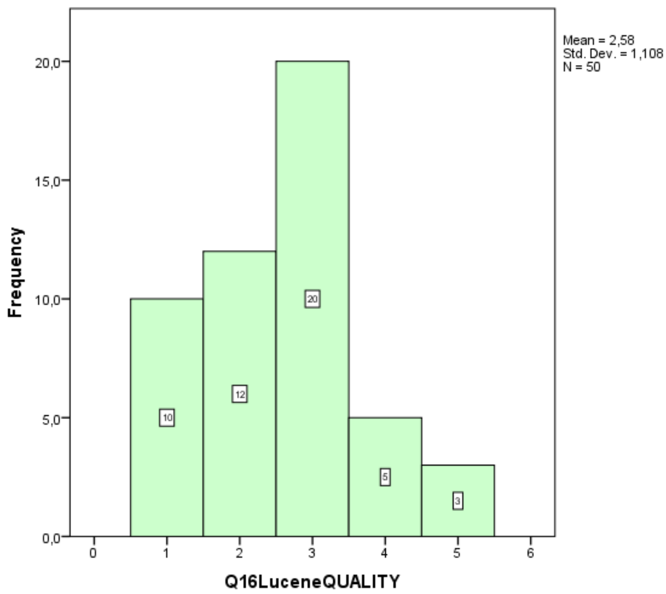
Table 4-72: Friedman Test to analyse the differences observed in users answers

To find out which algorithms differ from each other, we have to look at the results obtained by the post hoc analysis with Wilcoxon Signed Rank test. Looking at Table 4-73, we can see that there are neither significant differences between *Popular* and *ItemItem* (p=0.618), nor between *SVD* and *ItemItem* (p=0.054), nor between *UserUser* and *ItemItem* (0.071), nor between *Popular* and *SVD* (0.229), nor between *Popular* and *UserUser* (p=0.290), nor between *UserUser* and *SVD* (p=0.688). However, there are statistically significant differences between the other pairs of algorithms.

ItemItem, *Popular*, *UserUser* and *SVD* recommend more relevant movies than *Lucene* and *Persmean*, although we cannot determine which of these four algorithms recommend the most relevant movies since the comparison among them is not significant.

To clarify it, we can take a look at Figure 4-68, where we can graphically see the collected data for each algorithm.





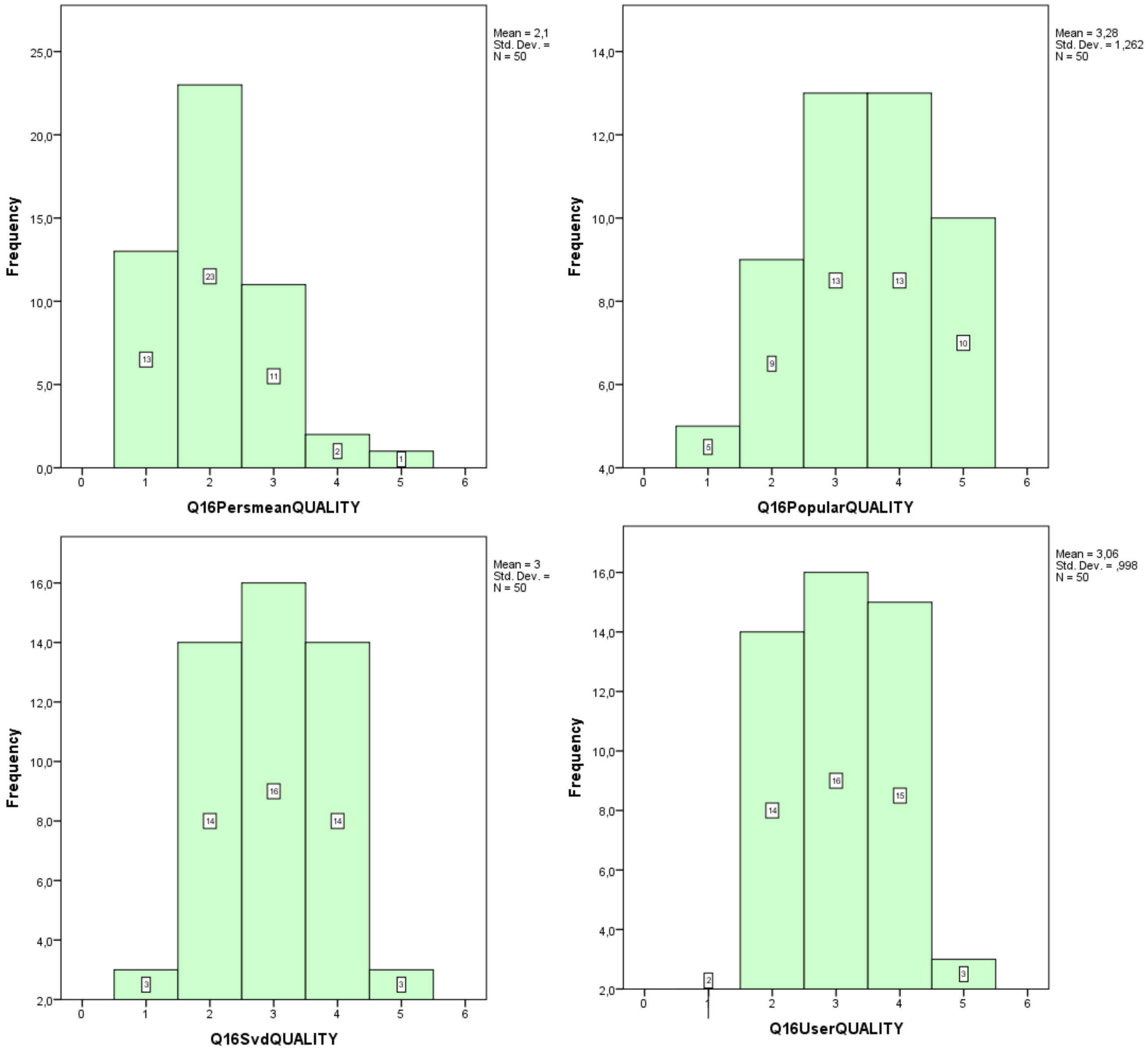


Figure 4-68: Users answers to each algorithm

Test Statistics[a]

	Lucene - Item	Persmean-Item	Popular-Item	SVD-Item	User-Item	Persmean-Lucene	Popular-Lucene	SVD-Lucene	User-Lucene	Popular-Persmean	SVD-Persmean	User-Persmean	SVD-Popular	User-Popular	User-SVD
Z	**-3,181[b]**	**-4,941[b]**	**-,498[b]**	**-1,9[b]**	**-1,8[b]**	**-2,516[b]**	**-3,181[c]**	**-2,248[c]**	**-2,109[c]**	**-4,547[c]**	**-3,968[c]**	**-4,586[c]**	**-1,202[b]**	**-1,058[b]**	**-,4[c]**
Asymp. Sig.	**,001**	**,000**	**,618**	**,054**	**,071**	**,012**	**,001**	**,025**	**,035**	**,000**	**,000**	**,000**	**,229**	**,290**	**,688**

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 4-73: Wilcoxon signed Rank Test to measure how different is each algorithm from the others.

Q17 - DO YOU THINK THAT THE RECOMMENDED MOVIES ARE NOT WELL-CHOSEN?

The data collected from the questionnaire is shown in Table 4-74.

	ItemItem	***Lucene***	***Persmean***	***Popular***	***SVD***	***UserUser***
Not well-chosen at all	6	13	21	13	4	4
Fairly well-chosen	17	14	18	7	23	22
Quite well-chosen	15	13	6	10	11	9
Very well-chosen	9	8	5	14	9	13
Perfectly well-chosen	3	2	0	6	3	2

Table 4-74: Data collected from the users answers

As we did in the previous question, we have run a Friedman Test, Table 4-75, to obtain the rank of our algorithms. We can see that there is an overall statistically significant difference (p≈0.000) depending on which algorithm we evaluate. Then, we are going to look at the ad hoc analysis with the Wilcoxon Signed Rank test, to find out the significant differences among algorithms.

Ranks

	Mean Rank
Q17Item*QUALITY*	3,81
Q17*LuceneQUALITY*	3,17
Q17*PersmeanQUALITY*	2,52
Q17*PopularQUALITY*	3,93
Q17*SVDQUALITY*	3,76
Q17User*QUALITY*	3,81

Test Statistics[a]

N	50
Chi-Square	27,975
df	5
Asymp. Sig.	,000

a. Friedman Test

Table 4-75: Friedman Test to analyse the differences observed in users answers.

Looking at Table 4-76, we can see that *Persmean* is the only algorithm that differs from the rest with a statistical significance. In addition, *Persmean* is the algorithm which recommends more not well-chosen movies. Contrarily, it is worth mentioning that we cannot determine which of the other algorithms recommend the best well-chosen movies since the comparison among them is not significant.

Test Statistics[a]

	Lucene - Item	Persmean-Item	Popular-Item	SVD-Item	User-Item	Persmean-Lucene	Popular-Lucene	SVD-Lucene	User-Lucene	Popular-Persmean	SVD-Persmean	User-Persmean	SVD-Popular	User-Popular	User-SVD
Z	**-1,804[b]**	**-3,374[b]**	**-1,059[c]**	**-,09[b]**	**-,18[c]**	**-2,991[b]**	**-1,744[c]**	**-1,082[c]**	**-1,269[c]**	**-3,573[c]**	**-3,565[c]**	**-3,453[c]**	**-,772[b]**	**-,873[b]**	**-,22[c]**
Asymp. Sig.	**,192**	**,001**	**,290**	**,928**	**,851**	**,003**	**,081**	**,279**	**,204**	**,000**	**,000**	**,001**	**,440**	**,382**	**,822**

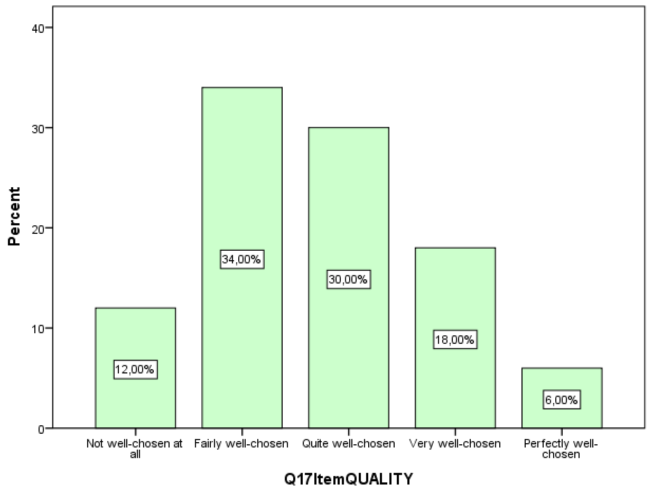
a. Wilcoxon Signed Ranks Test

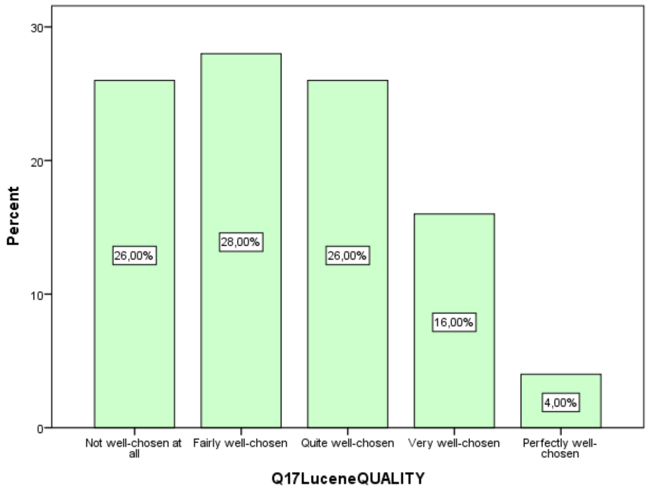
b. Based on positive ranks.

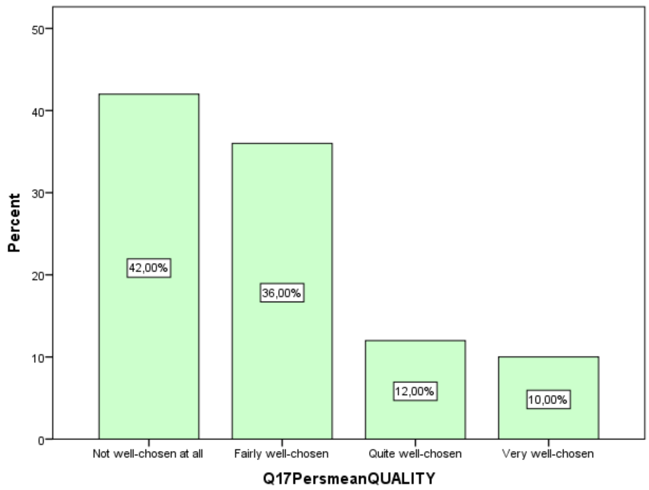
c. Based on negative ranks.

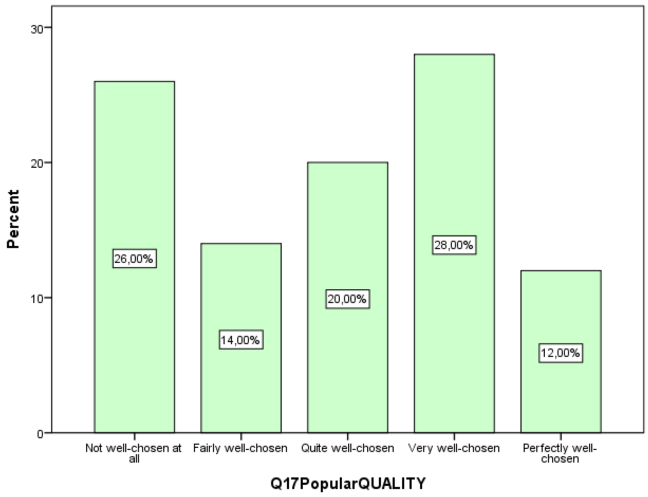
Table 4-76: Wilcoxon signed Rank Test to measure how different is each algorithm from the others.

To clarify it, we can take a look at Figure 4-69, where we can graphically see the collected data for each algorithm.









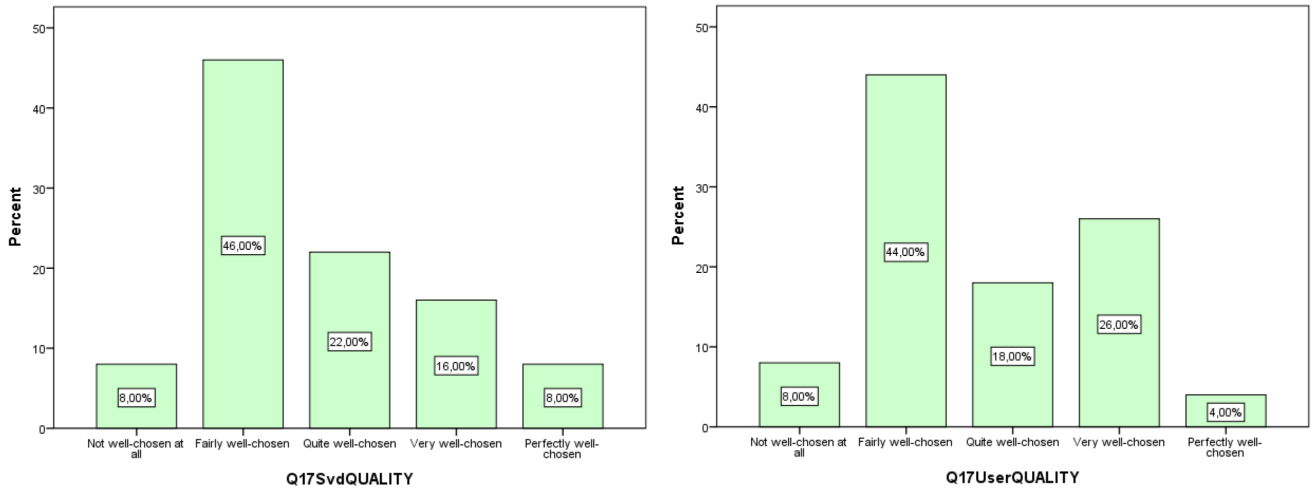


Figure 4-69: Users Answers to each algorithm

In conclusion, in terms of the recommender *Quality*, users are sure that *Persmean* is the worst, as we have been able to note in the results of Q16 and Q17. In contrast, *Popular* is the best algorithm by users' perception, followed by the collaborative filtering algorithms *ItemItem*, *UserUser* and *SVD* respectively.

4.2.3.7 Comparison among subjective metrics

In order to note the existing relationship among the subjective metrics, we have selected the most representative question of each metric: Q1 for *Accuracy*, Q4 for *Understands Me*, Q10 for *Novelty*, Q12 for *Effectiveness* and Q15 for *Quality*. However, to determine the relation among these metrics, we cannot calculate a correlation since we are comparing nominal qualitative variables that can only be classified but not ordered [52]. Therefore, we need a different statistic to measure the relationship among our metrics, the contingency coefficient (C), whose expression is:

$C = \sqrt{\frac{X^2}{n+X^2}}$, where: $n = number\ of\ votes$; $X^2 = coeff.\ chi - square$

The results obtained, Table 4-77, point out that all the metrics are related except *Novelty*, which is the only one with a lower value of C when it is compared with the other metrics.

	Accuracy	***Understands Me***	***Novelty***	***Effectiveness***	***Quality***
Accuracy	-	0.804	0.563	0.806	0.821
Understands Me	0.804	-	0.681	0.770	0.762
Novelty	0.563	0.681	-	0.674	0.604
Effectiveness	0.806	0.770	0.674	-	0.742
Quality	0.821	0.762	0.604	0.742	-

Table 4-77: Correlation among subjective metrics, using the contingency coefficient.

4.3 DISCUSSION

In this study, we have focused on measuring users' perception of some recommender systems' features such as *Accuracy*, *Understands Me*, *Novelty*, *Effectiveness* and *Quality*. We are now going to explain some of the key findings.

4.3.1 Effect of *Accuracy*

As we have demonstrated before, *Accuracy* is strongly related to the users' first impression of an algorithm. The satisfaction of the users is tied to their perception of how appealing or good the recommended movies are. This is not surprising since, for many years, the offline measure of *Accuracy* through RMSE has been the most extended metric to know how good the performance of an algorithm is.

4.3.2 Effect of *Understands Me*

Another important issue about users' satisfaction is the perception they have about how well the recommender can adapt to their preferences and tastes. As we have seen, *Understands Me* is also highly related to the satisfaction of the user with a recommender since, as seen on the users' first impression, the algorithms that best understand their tastes are the best considered ones in their initial choice.

This suggests that it is necessary to generate trust. The recommender should understand users' taste as it is crucial to give the user a good first impression of the system. The designers of systems have to take it into account, although it is difficult to inspire trust. The results show that the algorithms on which more users rely are *ItemItem* and *Popular*. To build trust on a system, users need to know some of the recommended items.

4.3.3 Effect of *Novelty*

The results of our experiment lead us to underline that *Novelty* has a negative effect on users' satisfaction. The recommendations with more surprising movies are made by the worst considered algorithms regarding the users' first impression. Moreover, we have seen that this metric significantly differs from the others. We can affirm that, to ensure good recommendations, the designer has to guarantee some known movies in order to increase the trust on the system, since only novel items in a list makes the user beware of the system.

4.3.4 Effect of *Effectiveness*

As the results show, the *Effectiveness* of the system is also highly related to the user satisfaction. The most valuable recommendations are made by the most accurate algorithms which were perceived by the users as the ones that best perform and the ones they trust on.

To qualify a system as effective, neither only accurate predictions nor novel recommendations are needed. It is also important to turn the system into a valuable tool in the users' life.

4.3.5 Effect of *Quality*

The *Quality* of a recommender system is a metric which is highly related to other metrics such as *Accuracy* and *Understands Me*. The opinion that the users have about these other metrics influences their perceptions of the system' *Quality*.

On their first impression of an algorithm, the *Quality* perceived is also noted. Thus, it is important to ensure a good *Quality* of the algorithm if we want it to obtain the best performance.

4.4 OBJECTIVE METRICS VS SUBJECTIVE METRICS

4.4.1 Offline Results

Algorithms	Neighbourhood size/Features	RMSE By Ratings	RMSE By Users	nDCG	topN nDCG	Entropy
Lucene Norm	100	0.9269	0.8539	0.8705	0.004968	6.431
UserUserCosine	50	0.9198	0.8534	0.9688	0.001684	0.9575
SVDPersmean	25	0.9058	0.8423	0.9679	0.001695	1.325
ItemItem	20	0.9165	0.8515	0.9688	0.006058	2.829
Persmean	-	0.9318	0.8693	0.965	0.00001607	1.28
Popular	-	-	-	-	0.06787	8.613

Table 4-78: Results of the objective metrics obtained through LensKit

Looking at the results obtained from the offline experiment (Table 4-78), we can rank the algorithms taking into account the different objective metrics. Thus, we have three rankings:

1. Based on RMSE, to compare it with the online measure of *Accuracy*.
2. Based on topN nDCG, to compare it with the online measure of *Quality*.
3. Based on Entropy, to compare it with the online measure of *Diversity*.

	1. Based on RMSE	**2. Based on topN nDCG**	**3. Based on Entropy**
1st	*SVD*	*Popular*	*Popular*
2nd	*ItemItem*	*ItemItem*	*Lucene*
3rd	*UserUser*	*Lucene*	*ItemItem*
4th	*Lucene*	*SVD*	*SVD*
5th	*Persmean*	*UserUser*	*Persmean*
6th		*Persmean*	*UserUser*

Table 4-79: Ranking based on objective metrics. Note that we cannot calculate the RMSE for Popular. That is why it does not appear on the first rank.

4.4.2 Online results

Although the data collected from the online questionnaire has hampered making a ranking based on *Diversity* since the differences observed on the questions that measure

Diversity were not significant, we have tried to make a rank taking into account the variety of the recommendations to see if there is a correlation between online and offline.

The rankings based on the subjective measures *Accuracy* and *Quality* are displayed in Table 4-80.

	1. *Accuracy*	**2. *Quality***	**3. *Diversity***
1st	*Popular*	*Popular*	*Popular*
2nd	*ItemItem*	*ItemItem*	*Lucene*
3rd	*UserUser*	*UserUser*	*Persmean*
4th	*SVD*	*Lucene*	*UserUser*
5th	*Lucene*	*SVD*	*SVD*
6th	*Persmean*	*Persmean*	*ItemItem*

Table 4-80: Ranking based on the subjective metrics.

4.4.3 Comparison

One of the most striking issue we find when we try to compare the results between *Accuracy* and RMSE is that the best algorithm taking into account RMSE (*SVD)* is one of the worst for *Accuracy* in the subjective measure. However, *ItemItem* is on the same position in both rankings.

This suggests that the basic assumption that the best algorithm is the most accurate is not completely true. As we have seen, users perceive *Accuracy* in a different manner. It seems that *SVD* does not work that well theoretically as in real life.

We have calculated the correlation between *Accuracy* and RMSE (Table 4-81) to see if the results are statistically significant. However, we found that they are not significant (p=0,245).

Correlations

		RMSE	*Accuracy* Q1-Q2
RMSE	Pearson Correlation	1	-,639
	Sig. (2-tailed)		,245
	N	5	5
Accuracy Q1-Q2	Pearson Correlation	-,639	1
	Sig. (2-tailed)	,245	
	N	5	5

Table 4-81: Correlation between Accuracy and RMSE

Additionally, if we look at Figure 4-70 where the data is represented as cluster, we can appreciate that there is an outlier that corresponds to *SVD*. Moreover, if we eliminate *SVD* from the correlation, the results (Table 4-82) show that the correlation is now significant (p=0.008) and both metrics are highly related (Pearson correlation =-0.992). This confirms that RMSE is a good metric to measure the accuracy for all the algorithms except for *SVD*.

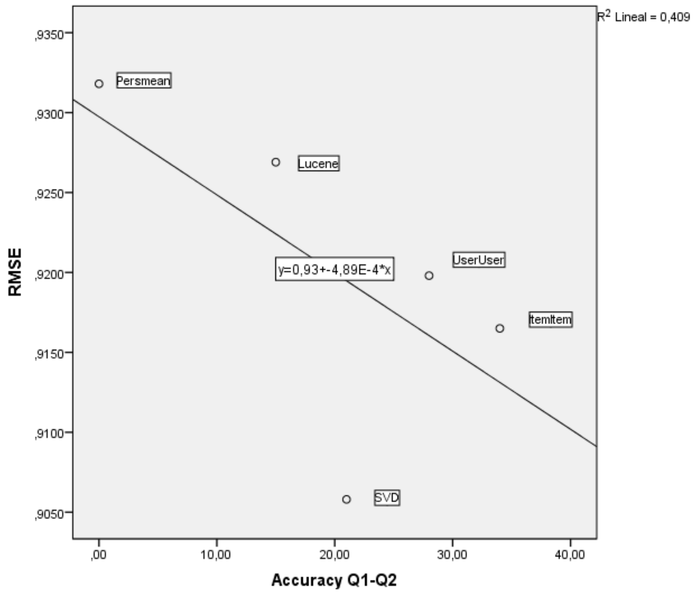


Figure 4-70: Cluster diagram Accuracy vs RMSE

Correlations

		RMSE	*Accuracy* Q1-Q2
RMSE	Pearson Correlation	1	-,992**
	Sig. (2-tailed)		,008
	N	4	4
Accuracy Q1-Q2	Pearson Correlation	-,992**	1
	Sig. (2-tailed)	,008	
	N	4	5

**. Correlation is significant at the 0.01 level (2-tailed).

Table 4-82: Correlation between Accuracy and RMSE without take into account SVD

Checking the differences between the results obtained by nDCG and by *Quality*, we can note that both have *ItemItem* and *Popular* as the best algorithms. Moreover, both have *Persmean* as the worst algorithm. The only difference is that by nDCG, *UserUser* is the 5th and *Lucene* the 3rd while by *Quality*, oppositely, *UserUser* is the 3rd and *Lucene* the 5th. This highlights that users do not have a perception of *Lucene* as good as expected. The reason could be, as noted with *Persmean*, the high level of *Novelty* found in its recommendations. Apart from that, nDCG has proven to be a useful tool to measure the *Quality* of a recommender system.

The results of the correlation between nDCG and *Quality* (Table 4-83) show that they are strongly related (Pearson correlation = 0.834). Moreover, this correlation is significant (p=0.039). This highlights that nDCG is a good metric to measure the *Quality* of a recommender system, and it works well with all the algorithms used in our research.

Correlations

		topN nDCG	*Quality*Q15
topN nDCG	Pearson Correlation	1	,834*
	Sig. (2-tailed)		,039
	N	6	6
*Quality*Q15	Pearson Correlation	,834*	1
	Sig. (2-tailed)	,039	
	N	6	6

*. Correlation is significant at the 0.05 level (2-tailed).

Table 4-83: Correlation between Quality and topN nDCG

Looking now at the differences between the results obtained by Entropy and by *Diversity*, it is notable that *ItemItem* is the worst for users, although theoretically is on the 3rd position. However, *Lucene* and *Popular* are both the best algorithms by Entropy and by *Diversity*. Moreover, *Persmean* is better considered by the users than theoretically. Taking into account *UserUser* and *SVD* the differences are not huge, both algorithms are considered as the algorithms with scant variety in the online experiment and in the offline one.

To ensure these conclusions, we have calculated the correlation between Entropy and *Diversity* (Table 4-84) and we have seen that the results are no significant (p=0.152). Moreover, there is no correlation between them since the coefficient of Pearson correlation is equal to 0.662.

Correlations

		Entropy	*Diversity*
Entropy	Pearson Correlation	1	,662
	Sig. (2-tailed)		,152
	N	6	6
Diversity	Pearson Correlation	,662	1
	Sig. (2-tailed)	,152	
	N	6	6

Table 4-84: Correlation between Entropy and Diversity

Looking at Figure 4-71, we can see that all the values are almost equidistant to the line that describes the correlation between both metrics, so we cannot underline any outlier. This highlights that entropy is not the best metric to measure the *Diversity* of a recommender system.

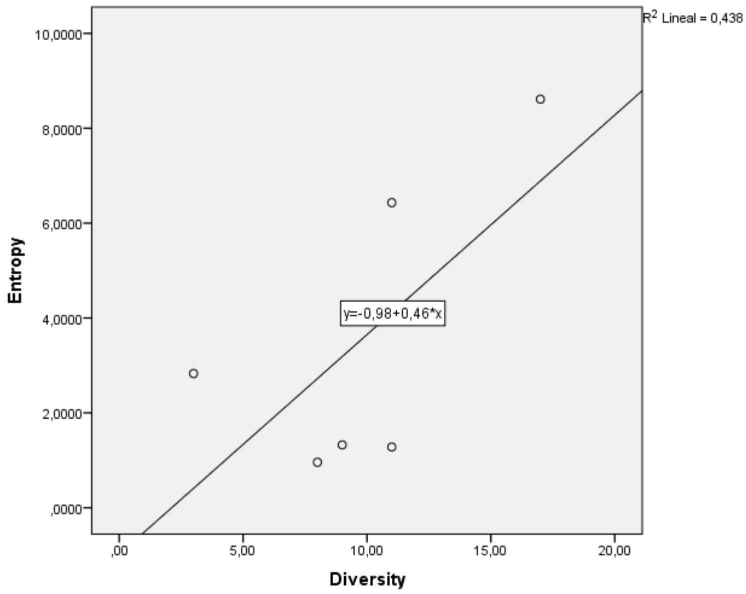


Figure 4-71: Cluster diagram Diversity vs Entropy

In view of the conclusions derived from the results of the comparison, it could be better to use the topN nDCG to measure the goodness of a recommender system than RMSE or Entropy.

4.5 GROUP RECOMMENDATIONS

In this section, we want to evaluate the effectiveness of the group recommendations. We have asked our users to fill the questionnaire in groups imagining they are going to watch a movie together. They had to reach an agreement to rate the top 100 movies, combining their preferences, and then, taking into account the preferences of each group as a pseudo user, we generate group recommendations using six traditional recommendation algorithms.

Once we had the recommendations lists for each group, we asked them again to answer the survey together to know the perception of all the group' members about the recommendations given.

As we did on the evaluation of individual users, we are going to evaluate not only *Accuracy* but also other qualitative metrics as *Understands Me*, *Novelty*, *Diversity*, *Effectiveness* and *Quality*.

Furthermore, to have a good understanding of their preferences, we asked them some additional questions in order to know if they found difficulties evaluating the recommendations lists, and the viability of this kind of group recommendations.

We would have liked to have been able to make a distinction taking into account the size of the groups. However, only 10 groups have filled our questionnaire and for this reason it is difficult to take into account the size of the groups. Furthermore, most of them are groups of 2 members. Only three groups have 3 members (Figure 4-41).

4.5.1 Analysis Subjective Metrics

4.5.1.1 Accuracy

Q1- WHICH LIST HAS MORE MOVIES THAT YOU FIND APPEALING?

Table 4-85 shows the frequency analysis of the data collected, where we can see the observed count and the expected count. Furthermore, we have run the chi-squared test to prove that the differences among algorithms are significant, p= 0.003. Note that we have look at the exact significance since the sample size is very small.

The 60% of our groups are sure that the algorithm that recommends more appealing movies is *ItemItem*, followed by *Popular* with a 30%. Only one of our groups chose *Lucene*, while nobody opted for *UserUser*, *SVD* nor *Persmean* (Figure 4-72).

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	Item	6	1,7	4,3
2	*Lucene*	1	1,7	-,7
3	*Persmean*	0	1,7	-1,7
4	*Popular*	3	1,7	1,3
5	*SVD*	0	1,7	-1,7
6	*UserUser*	0	1,7	-1,7
Total		10		

Test Statistics

	Q1*Accuracy*
Chi-Square	17,600[a]
df	5
Asymp. Sig.	,003
Exact Sig.	,003
Point Probability	,002

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-85: Chi square test to measure the differences observed in Q1 for groups with α=0.05.

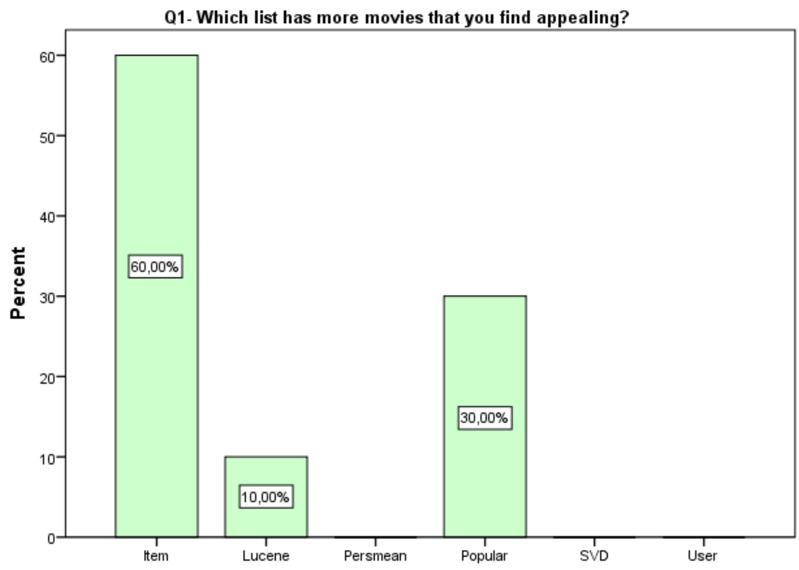


Figure 4-72: Bar diagram with the collected data from groups Q1

Q2- WHICH LIST HAS MORE OBVIOUSLY BAD MOVIE RECOMMENDATIONS FOR YOU?

The answers of this question show a significant difference (p≈0.000) among algorithms (Table 4-86). *Persmean*, with a 70%, is the one which made more obvious bad recommendations to the groups, followed by *Lucene* with a 30%, while the other four algorithms are not selected by any group; therefore, the remaining algorithms do not recommend bad movies to the users.

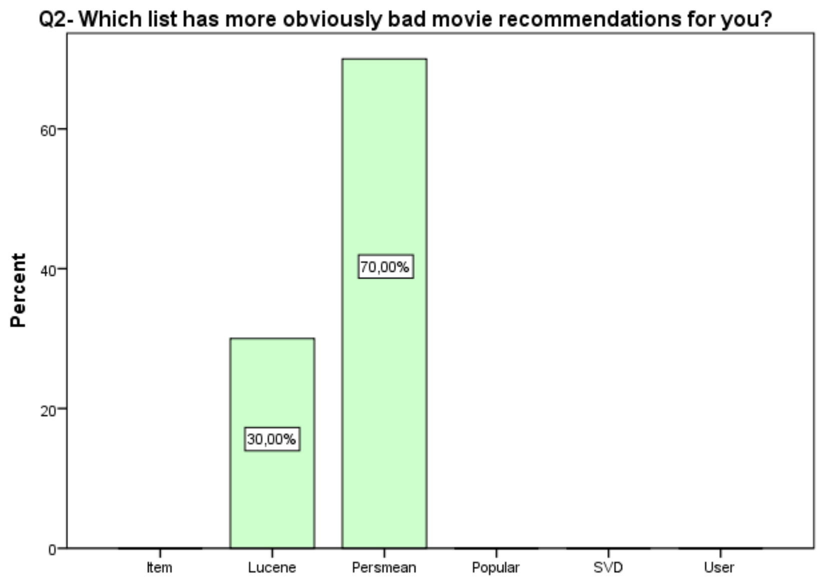


Figure 4-73: Bar diagram with the collected data from groups Q2

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	0	1,7	-1,7
2	*Lucene*	3	1,7	1,3
3	*Persmean*	7	1,7	5,3
4	*Popular*	0	1,7	-1,7
5	*SVD*	0	1,7	-1,7
6	*UserUser*	0	1,7	-1,7
Total		10		

Test Statistics

	Q2*Accuracy*
Chi-Square	24,800[a]
df	5
Asymp. Sig.	,000
Exact Sig.	,000
Point Probability	,000

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-86: Chi square test to measure the differences observed in Q2 for groups with α=0.05.

To have a global result we make a combination of both questions since Q1 has a positive connotation while Q2 has a negative connotation in terms of *Accuracy*. In Figure 4-74, we can see that the more accurate algorithms are *ItemItem* and *Popular* while the less accurate are *Lucene* and *Persmean*. However, *UserUser* and *SVD* are not noted by the users.

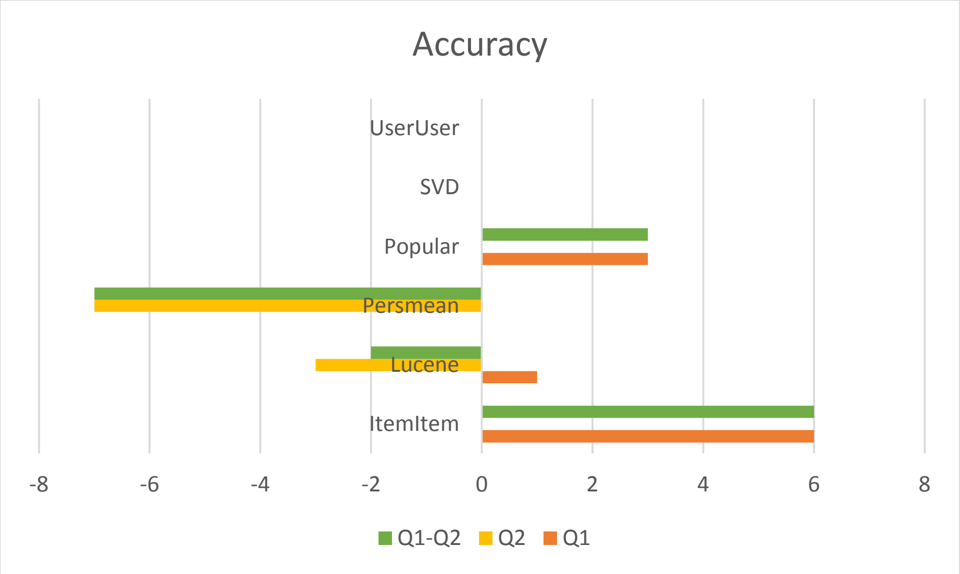


Figure 4-74: Combination of Q1-Q2 to have a global result for Accuracy

4.5.1.2 Understands Me

Q3-WHICH LIST MORE REPRESENTS MAIN STREAM TASTES INSTEAD OF YOUR OWN?

In Figure 4-75, we can see a diversification of the groups' opinion. Furthermore, the chi-squared test (Table 4-87) tells us that there are not significant differences between the results (p=0.065). We can only underline that half of the groups have elected *Popular* as the algorithm that more represents main stream tastes, which is obvious.

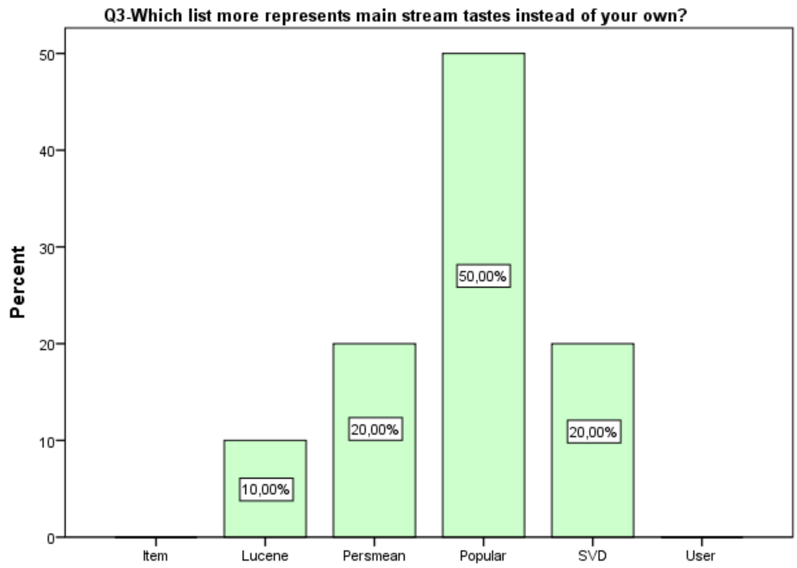


Figure 4-75: Bar diagram with the collected data from groups Q3

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	0	1,7	-1,7
2	*Lucene*	1	1,7	-,7
3	*Persmean*	2	1,7	,3
4	*Popular*	5	1,7	3,3
5	*SVD*	2	1,7	,3
6	*UserUser*	0	1,7	-1,7
Total		10		

Test Statistics

	Q3*Understands Me*
Chi-Square	10,400[a]
df	5
Asymp. Sig.	,065
Exact Sig.	,076
Point Probability	,036

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-87: Chi square test to measure the differences observed in Q3 for groups with α=0.05.

Q4-WHICH RECOMMENDATION LIST BETTER UNDERSTANDS YOUR TASTE IN MOVIES?

The answers show that a 55% of the groups think that *ItemItem* is the algorithm that better understands their test, followed by *Popular* with a 30% of the votes. This percentages differ significantly (p=0.003) from the other algorithms, which are not as good understanding users' taste. We can depreciate *Lucene* and *SVD* since only one group has opted for them.

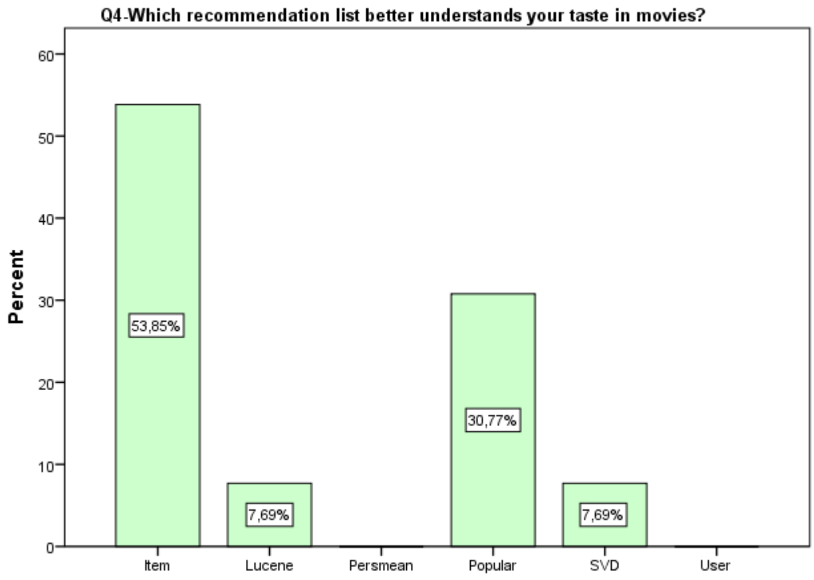


Figure 4-76: Bar diagram with the collected data from groups Q4

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	7	2,2	4,8
2	*Lucene*	1	2,2	-1,2
3	*Persmean*	0	2,2	-2,2
4	*Popular*	4	2,2	1,8
5	*SVD*	1	2,2	-1,2
6	*UserUser*	0	2,2	-2,2

Table 4-88: Chi square test to measure the differences observed in Q4

Test Statistics

	Q4*Understands Me*
Chi-Square	17,923[a]
df	5
Asymp. Sig.	,003
Exact Sig.	,003
Point Probability	,001

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 2,2.

The combination of Q3 and Q4 gives us a global overview of *Understands Me*, taking into account the groups' opinion.

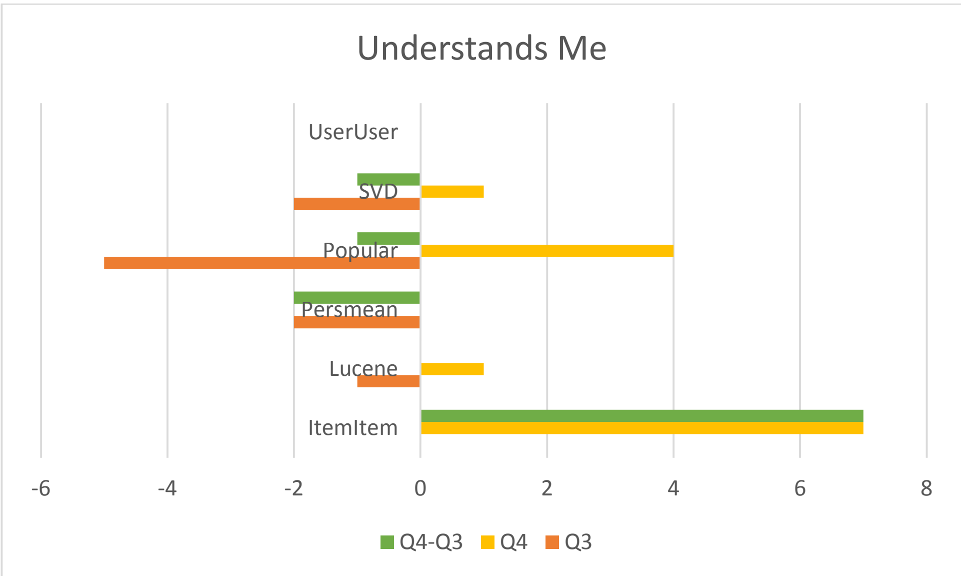


Figure 4-77: Combination of Q4-Q3 to have a global result for Understands Me

In conclusion, *ItemItem* is the best algorithm understanding the groups' taste. With *Popular*, we have seen a big controversy, because although the majority of the groups think that it is the algorithm that best represents main stream tastes, it is also chosen by a considerable number of groups as the algorithm that best understands them. The reason is that people appreciate well-known movies. Contrarily, the differences are very small to extrapolate results among the other algorithms.

4.5.1.3 Diversity

Q5- WHICH LIST HAS MORE MOVIES THAT ARE SIMILAR TO EACH OTHER?

The results obtained on this question are not conclusive since the data is almost equally distributed among the algorithms (p=0.270), so we cannot extrapolate the results.

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	0	1,7	-1,7
2	*Lucene*	4	1,7	2,3
3	*Persmean*	2	1,7	,3
4	*Popular*	2	1,7	,3
5	*SVD*	2	1,7	,3
6	*UserUser*	0	1,7	-1,7
Total		10		

Test Statistics

	Q5*Diversity*
Chi-Square	6,800[a]
df	5
Asymp. Sig.	,236
Exact Sig.	,270
Point Probability	,085

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-89: Chi square test to measure the differences observed in Q5 for groups with α=0.05

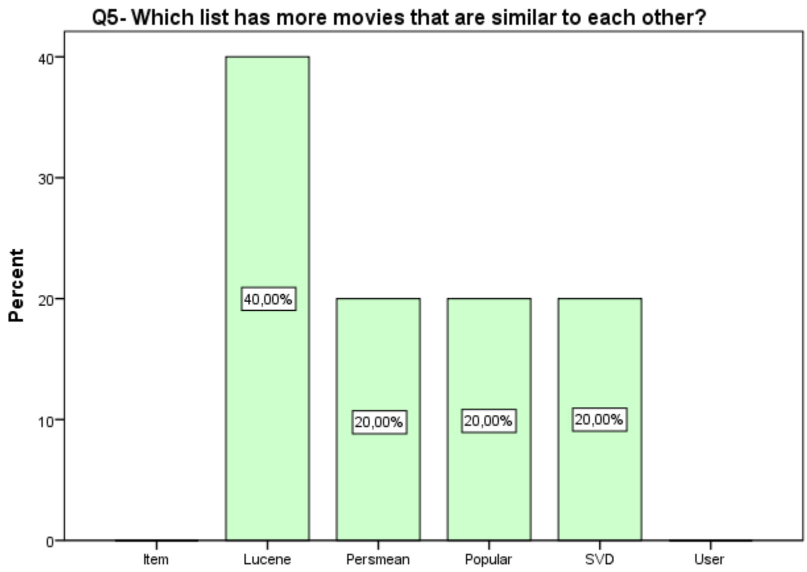


Figure 4-78: Bar diagram with the collected data from groups Q5

Q6- WHICH LIST HAS A LESS VARIED SELECTION OF MOVIES?

Looking at Figure 4-79, we can note that three algorithms are explicitly chosen by the groups (p=0.041), which are *Persmean*, *Lucene* and *SVD*, while the other three algorithms are not chosen. Therefore, the less diverse algorithms are *Persmean*, *Lucene* and *SVD* with equal significance.

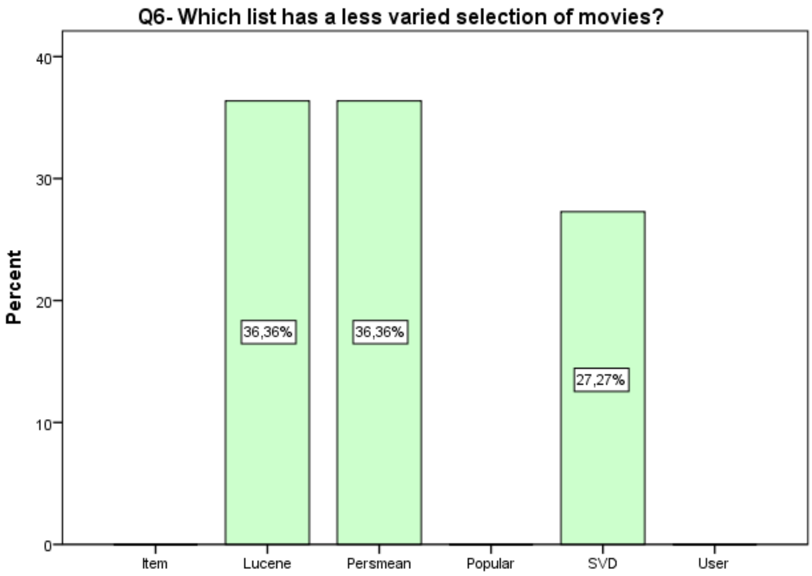


Figure 4-79: Bar diagram with the collected data from groups Q6

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	0	1,8	-1,8
2	*Lucene*	4	1,8	2,2
3	*Persmean*	4	1,8	2,2
4	*Popular*	0	1,8	-1,8
5	*SVD*	3	1,8	1,2
6	*UserUser*	0	1,8	-1,8
Total		11		

Test Statistics

	Q6*Diversity*
Chi-Square	11,364[a]
df	5
Asymp. Sig.	,045
Exact Sig.	,041
Point Probability	,003

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,8.

Table 4-90: Chi square test to measure the differences observed in Q6 for groups with α=0.05

Q7- WHICH LISTS DO YOU THINK THAT INCLUDE MOVIES OF MANY DIFFERENT GENRES?

The results obtained are not consistent (p=0.420) so that we cannot extrapolate them. The groups' opinion is highly divided among *ItemItem*, *Popular*, *SVD* and *UserUser*. However, none of the groups has chosen *Lucene* nor *Persmean*.

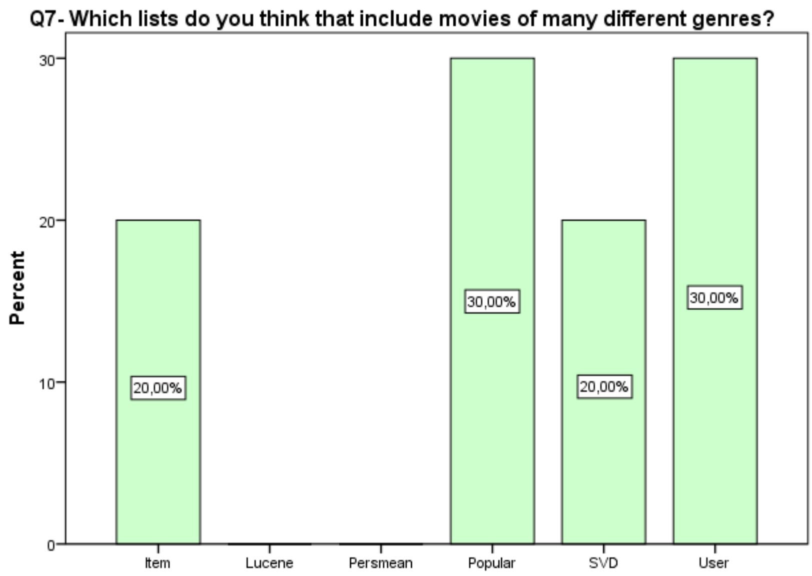


Figure 4-80: Bar diagram with the collected data from groups Q7

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	2	1,7	,3
2	*Lucene*	0	1,7	-1,7
3	*Persmean*	0	1,7	-1,7
4	*Popular*	3	1,7	1,3
5	*SVD*	2	1,7	,3
6	*UserUser*	3	1,7	1,3
Total		10		

Test Statistics

	Q7*Diversity*
Chi-Square	5,600[a]
df	5
Asymp. Sig.	,347
Exact Sig.	,420
Point Probability	,150

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-91: Chi square test to measure the differences observed in Q7 for groups with α=0.05

In conclusion, taking into account *Diversity*, we can only note that the algorithms that recommend a less varied selection of movies are *Lucene* and *Persmean*.

4.5.1.4 Novelty

Q8 - WHICH LIST HAS MORE MOVIES YOU DO NOT EXPECT?

Although the differences on the results, Table 4-92, are not significant (p=0.102), we can underline that *Persmean* and *Lucene* are the algorithms with more surprising movies.

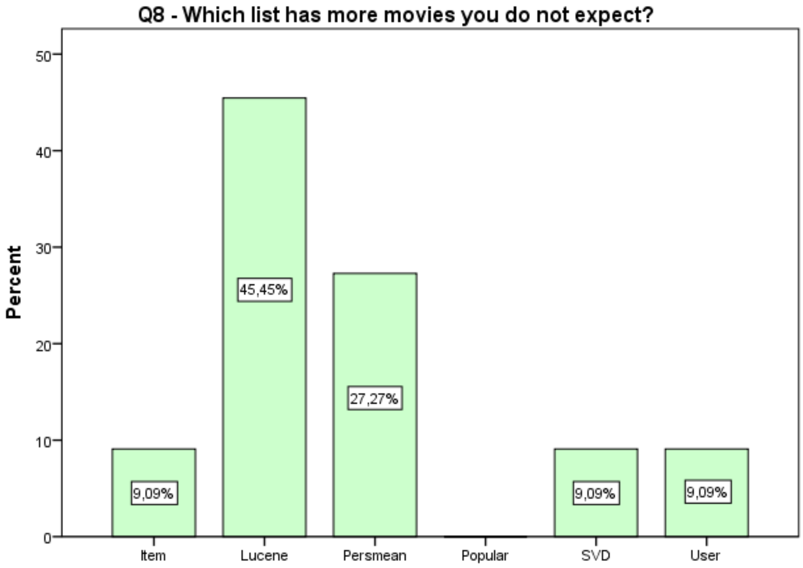


Figure 4-81: Bar diagram with the collected data from groups Q8

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	1	1,8	-,8
2	*Lucene*	5	1,8	3,2
3	*Persmean*	3	1,8	1,2
4	*Popular*	0	1,8	-1,8
5	*SVD*	1	1,8	-,8
6	*UserUser*	1	1,8	-,8
Total		11		

Test Statistics

	Q8*Novelty*
Chi-Square	9,182[a]
df	5
Asymp. Sig.	,102
Exact Sig.	,111
Point Probability	,042

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,8.

Table 4-92: Chi square test to measure the differences observed in Q8 for groups with α=0.05

Q9 - WHICH LIST HAS MORE MOVIES THAT ARE FAMILIAR TO YOU?

In this question, the differences among the algorithms are neither significant (p= 0.083). However, it could be said that *Popular*, *ItemItem* and *UserUser* are the ones that recommend more familiar movies (Figure 4-82).

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	3	2,0	1,0
2	*Lucene*	1	2,0	-1,0
3	*Persmean*	0	2,0	-2,0
4	*Popular*	5	2,0	3,0
5	*SVD*	0	2,0	-2,0
6	*UserUser*	3	2,0	1,0
Total		12		

Test Statistics

	Q9*Novelty*
Chi-Square	10,000[a]
df	5
Asymp. Sig.	,075
Exact Sig.	,083
Point Probability	,023

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 2,0.

Table 4-93: Chi square test to measure the differences observed in Q9 for groups with α=0.05

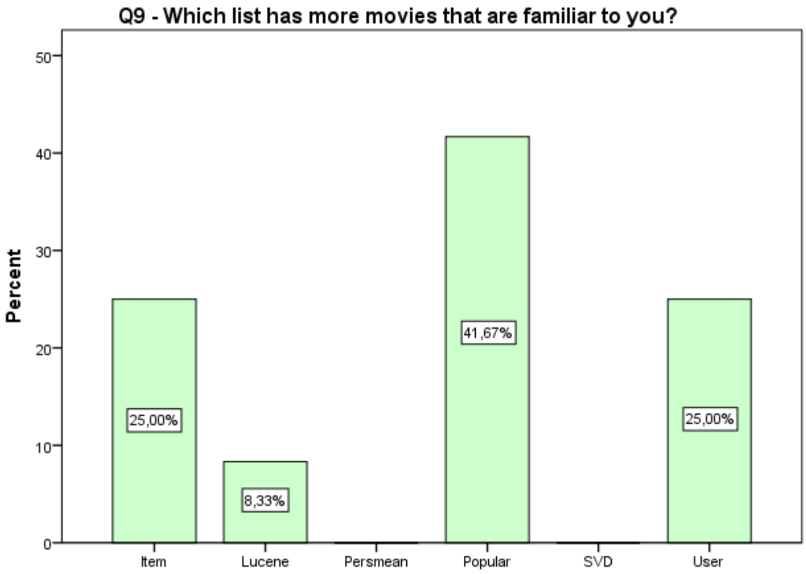


Figure 4-82: Bar diagram with the collected data from groups Q9

Q10 - WHICH LIST HAS MORE PLEASANTLY SURPRISING MOVIES?

At first sight, Figure 4-83, we can see that *ItemItem* is the algorithm which is more chosen by the groups, which means that this is the one with more pleasantly surprising movies. However, the chi-squared test tells us that the observed differences are not significant (p=0.083) (Table 4-94)

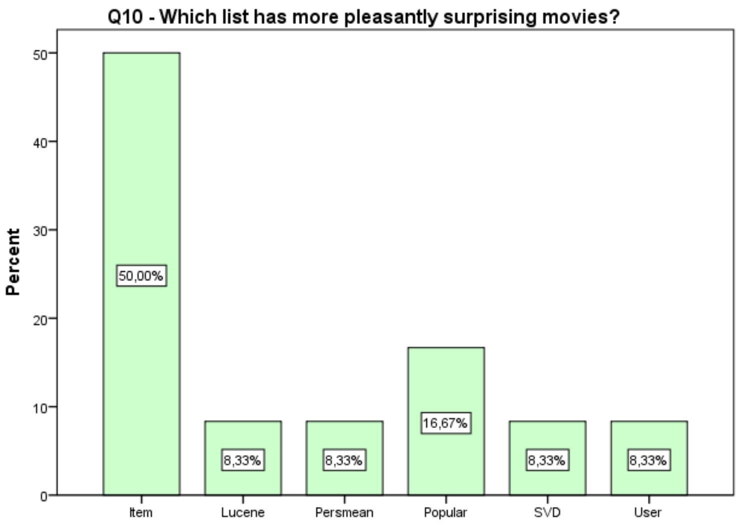


Figure 4-83: Bar diagram with the collected data from groups Q10

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	6	2,0	4,0
2	*Lucene*	1	2,0	-1,0
3	*Persmean*	1	2,0	-1,0
4	*Popular*	2	2,0	,0
5	*SVD*	1	2,0	-1,0
6	*UserUser*	1	2,0	-1,0
	Total		13	

Test Statistics

	Q10*Novelty*
Chi-Square	10,000[a]
df	5
Asymp. Sig.	,075
Exact Sig.	,083
Point Probability	,023

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 2,0.

Table 4-94: Chi square test to measure the differences observed in Q10 for groups with α=0.05

Q11 - WHICH LIST HAS MORE MOVIES YOU WOULD NOT HAVE THOUGHT TO CONSIDER?

The results obtained show, Figure 4-84, that the groups' opinion is divided between *Persmean* and *Lucene*, which are the algorithms that recommend more surprising movies (p=0.038) (Table 4-95).

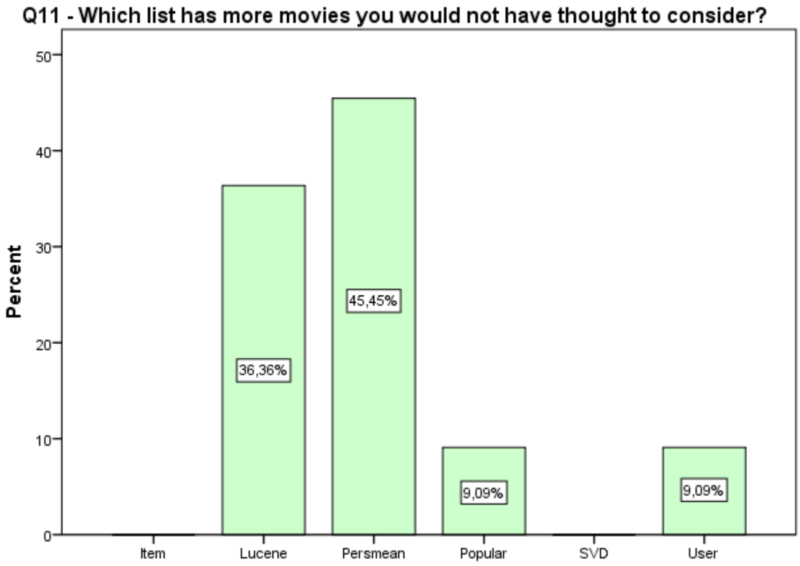


Figure 4-84: Bar diagram with the collected data from groups Q11

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	0	1,8	-1,8
2	*Lucene*	4	1,8	2,2
3	*Persmean*	5	1,8	3,2
4	*Popular*	1	1,8	-,8
5	*SVD*	0	1,8	-1,8
6	*UserUser*	1	1,8	-,8
Total		11		

Test Statistics

	Q11*Novelty*
Chi-Square	12,455[a]
df	5
Asymp. Sig.	,029
Exact Sig.	,038
Point Probability	,018

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,8.

Table 4-95: Chi square test to measure the differences observed in Q11 for groups with α=0.05

We can conclude that *Popular*, *ItemItem* and *UserUser* are the algorithms that recommend more familiar movies to the groups. In contrast, *Persmean* and *Lucene* recommend more novel movies. However, these movies do not fit groups' tastes.

4.5.1.5 Effectiveness

Q12 - WHICH LIST GIVES YOU MORE VALUABLE RECOMMENDATIONS?

Looking at Table 4-96, it is clear that *ItemItem* is the algorithm whose recommendations are the most priceless (p= 0.030). Moreover, it is difficult to find out differences among the other algorithms. It is only notable that *Persmean* has not been chosen by any group.

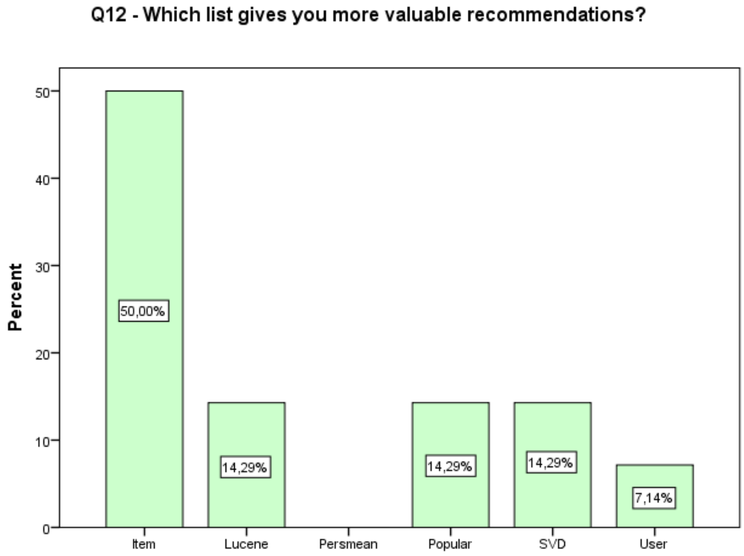


Figure 4-85: Bar diagram with the collected data from groups Q12

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	7	2,3	4,7
2	*Lucene*	2	2,3	-,3
3	*Persmean*	0	2,3	-2,3
4	*Popular*	2	2,3	-,3
5	*SVD*	2	2,3	-,3
6	*UserUser*	1	2,3	-1,3
Total		14		

Test Statistics

	Q12*Effectiveness*
Chi-Square	12,571[a]
df	5
Asymp. Sig.	,028
Exact Sig.	,030
Point Probability	,008

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 2,3.

Table 4-96: Chi square test to measure the differences observed in Q12 for groups with α=0.05

Q13 - DO YOU THINK THAT THE RECOMMENDER IS RECOMMENDING INTERESTING CONTENT YOU HADN'T PREVIOUSLY CONSIDER?

The answers from this question are summarized in Table 4-97:

	ItemItem	*Lucene*	*Persmean*	*Popular*	*SVD*	*UserUser*
No, nothing out of the ordinary	0	1	1	1	1	0
Somewhat out of the ordinary	1	6	7	4	3	4
Quite a bit surprisingly good movies	6	2	1	1	3	3
Fairly surprisingly good movies	3	1	1	4	3	3
Yes, there are lots of surprisingly good movies	0	0	0	0	0	0

Table 4-97: Data collected from groups' questionnaire Q13

To check whether the differences are significant or not, we have run a Friedman Test. Moreover, it allows us to know the mean rank of our algorithms. As we can see in Table 4-98, the differences among algorithms are relevant (p=0.034). The next step is to realize among which algorithms we can appreciate these differences, through the Wilcoxon signed rank test.

Friedman Test Ranks

	Mean Rank
Q13Item*Effectiveness*	4,55
Q13*LuceneEffectiveness*	2,85
Q13*PersmeanEffectiveness*	2,45
Q13*PopularEffectiveness*	3,60
Q13*SVDEffectiveness*	3,75
Q13User*Effectiveness*	3,80

Test Statistics[a]

N	10
Chi-Square	12,026
df	5
Asymp. Sig.	,034

a. Friedman Test

Table 4-98: Friedman Test Q13

Test Statistics[a]

	Lucene - Item	*Persmean-* Item	*Popular* -Item	*SVD-* Item	User- Item	*Persmean- Lucene*	*Popular -Lucene*	*SVD- Lucene*	User- *Lucene*	*Popular- Persmean*	*SVD- Persmean*	User- *Persmean*	*SVD- Popular*	User- *Popular*	User- *SVD*
Z	-2,251[b]	-2,640[b]	-1,190[b]	-1,414[b]	-1,134[b]	-,447[b]	-1,225[c]	-1,155[c]	-1,511[c]	-1,857[c]	-1,667[c]	-1,823[c]	-,106[b]	-,141[c]	-,322[c]
Asymp. Sig.	,024	,008	,234	,157	,257	,655	,221	,248	,131	,063	,096	,068	,915	,888	,748

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 4-99: Wilcoxon signed rank test Q13 to analyse the differences observed in users' answers

Looking at Table 4-99, we can determine that there are only significant differences between *ItemItem* and *Lucene* (p=0.024) and between *ItemItem* and *Persmean* (p=0.008), where *ItemItem* is the algorithm with more surprisingly good movies, while *Persmean* and *Lucene* are the ones with less.

Q14 - CONSIDERING THE BEST RECOMMENDATION LIST IN YOUR OPINION, DO YOU SAVE TIME USING THE RECOMMENDER TO CHOOSE A MOVIE COMPARED TO YOUR USUAL WAY OF SELECTING MOVIES?

The general opinion about the usefulness of the recommender is not very clear. None of the groups considers it as very useful, but neither as completely usefulness (Figure 4-86). This means that they can use it to select movies, but they are not bothered about it so that if they cannot use it for any reason, it will not be a problem.

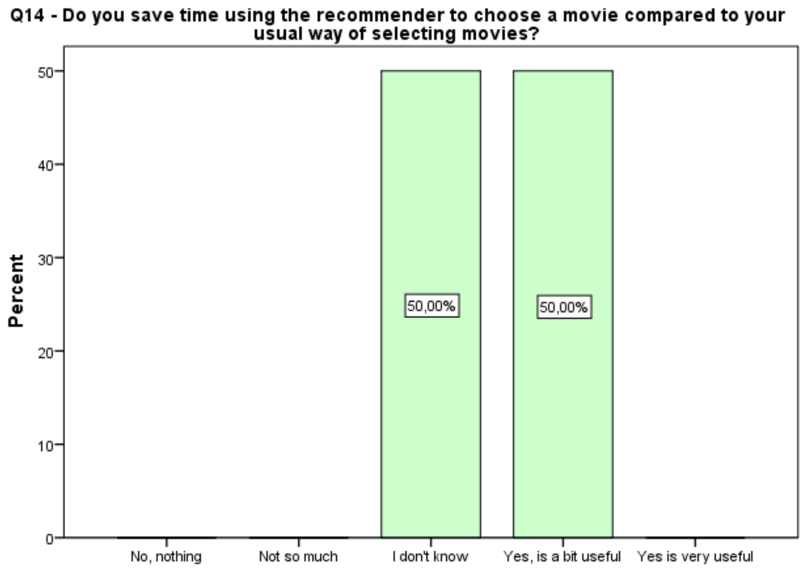


Figure 4-86: Bar diagram with the collected data from groups Q14

Frequencies

	RankQ14 Category	Observed N	Expected N	Residual
1	No, nothing	0	2,0	-2,0
2	Not so much	0	2,0	-2,0
3	I don't know	5	2,0	3,0
4	Yes, is a bit useful	5	2,0	3,0
5	Yes, is very useful	0	2,0	-2,0
Total		10		

Test Statistics

	Q14*Effectiveness*
Chi-Square	15,000[a]
df	4
Asymp. Sig.	,005
Exact Sig.	,005
Point Probability	,000

a. 5 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 2,0.

Table 4-100: Chi square test to measure the differences observed in Q14 for groups with α=0.05

4.5.1.6 Quality

Q15 - WHICH LIST HAS MORE MOVIES THAT FIT/MATCH YOUR PREFERENCE?

The data collected from the questionnaire, shows that *ItemItem* and *Popular* are the ones that best match the groups' preferences. However, the chi-squared test gives us a p-value of 0.184 (Table 4-101); therefore, we cannot extrapolate these results.

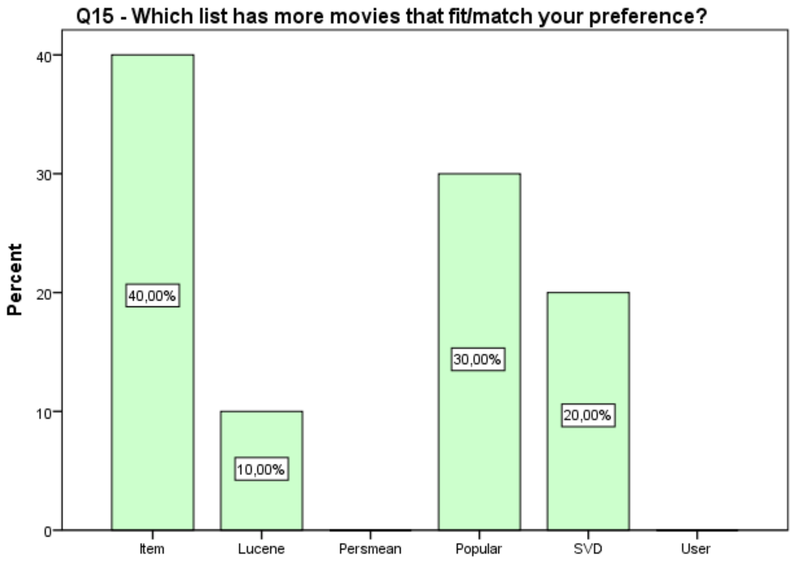


Figure 4-87: Bar diagram with the collected data from groups Q15

Frequencies

	Algorithm Category	Observed N	Expected N	Residual
1	*ItemItem*	4	1,7	2,3
2	*Lucene*	1	1,7	-,7
3	*Persmean*	0	1,7	-1,7
4	*Popular*	3	1,7	1,3
5	*SVD*	2	1,7	,3
6	*UserUser*	0	1,7	-1,7
Total		10		

Test Statistics

	Q15*Quality*
Chi-Square	**8,000[a]**
df	**5**
Asymp. Sig.	**,156**
Exact Sig.	**,184**
Point Probability	**,078**

a. 6 cells (100,0%) have expected frequencies less than 5. The minimum expected cell frequency is 1,7.

Table 4-101: Chi square test to measure the differences observed in Q15 for groups with α=0.05

Q16 - HOW MUCH DO YOU THINK THAT THE RECOMMENDED MOVIES ARE RELEVANT?

The opinion of the groups is reflected in Table 4-102:

	ItemItem	Lucene	Persmean	Popular	SVD	UserUser
Not relevant at all	0	1	2	1	0	0
Of little relevant	0	6	6	2	3	3
Moderately relevant	3	3	2	3	4	4
Relevant	5	0	0	2	3	3
Very relevant	2	0	0	2	0	0

Table 4-102: Data collected from groups' questionnaire Q16

Looking at the results obtained with the Friedman test, Table 4-103, we can see some differences among our algorithms in terms of mean rank (p=0.000), where *ItemItem* recommends the most relevant movies and *Persmean* recommends the most irrelevant ones.

Ranks

	Mean Rank
Q16Item*Quality*	5,25
Q16*LuceneQuality*	2,30
Q16*PersmeanQuality*	2,00
Q16*PopularQuality*	4,05
Q16*SVDQuality*	3,85
Q16User*Quality*	3,55

Test Statistics[a]

N	10
Chi-Square	23,312
df	5
Asymp. Sig.	,000

a. Friedman Test

Table 4-103: Friedman Test Q16

To find out which algorithms differ from each other, we have to look at the results obtained by the post hoc analysis with Wilcoxon Signed Rank test. Looking at Table 4-104, we can see that there are neither significant differences between *ItemItem* and *Popular* (p=0.222), nor between *Popular* and *UserUser* (p=0.713), nor between *Popular* and *SVD* (0.668), nor between *UserUser* and *SVD* (0.931), nor between *UserUser* and *Lucene* (p=0.054), nor between *Lucene* and *Persmean* (p=0.414). However, there are statistically significant differences between the other pairs of algorithms.

ItemItem, *Popular*, *UserUser* and *SVD* recommend more relevant movies than *Lucene* and *Persmean*. Moreover, the movies recommended by *ItemItem* are more relevant than the ones recommended by *SVD* or *UserUser*.

Test Statistics

	Lucene - Item	*Persmean-Item*	*Popular-Item*	*SVD-Item*	*User-Item*	*Persmean-Lucene*	*Popular-Lucene*	*SVD-Lucene*	*User-Lucene*	*Popular-Persmean*	*SVD-Persmean*	*User-Persmean*	*SVD-Popular*	*User-Popular*	*User-SVD*
Z	-2,850[b]	-2,850[b]	-1,222[b]	-2,251[b]	-2,081[b]	-,816[b]	-1,983[c]	-1,999[c]	-1,930[c]	-2,220[c]	-2,456[c]	-2,197[c]	-,428[b]	-,368[b]	-,087[c]
Asymp. Sig	,004	,004	,222	,024	,037	,414	,047	,046	,054	,026	,014	,028	,668	,713	,931

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 4-104: Wilcoxon signed rank test Q16

Q17 - DO YOU THINK THAT THE RECOMMENDED MOVIES ARE NOT WELL-CHOSEN?

Groups' answers are summarized on Table 4-105:

	ItemItem	*Lucene*	*Persmean*	*Popular*	*SVD*	*UserUser*
Not well-chosen at all	0	2	3	1	0	0
Fairly well-chosen	0	5	4	2	4	4
Quite well-chosen	4	1	2	4	3	3
Very well-chosen	4	1	0	1	2	1
Perfectly well-chosen	2	1	1	2	1	2

Table 4-105: Data collected from groups' questionnaire Q17

Looking at the mean rank of our algorithms (Table 4-106), we can see some significant differences among them (p=0.002). *ItemItem* is clearly the algorithm that best chooses the movies recommended, followed by *Popular*, *UserUser* and *SVD*, without a big difference among them, and we find *Lucene* and *Persmean* in the last position.

Ranks

	Mean Rank
Q17Item*Quality*	5,10
Q17*LuceneQuality*	2,70
Q17*PersmeanQuality*	2,20
Q17*PopularQuality*	3,80
Q17*SVDQuality*	3,50
Q17User*Quality*	3,70

Table 4-106: Friedman Test Q17

Test Statistics[a]

N	10
Chi-Square	18,731
df	5
Asymp. Sig.	,002

a. Friedman Test

To ensure these differences among algorithms, we will take a look at the results of the post hoc analysis completed (Table 4-107).

Test Statistics[a]

	Lucene - Item	Persmean-Item	Popular-Item	SVD-Item	User-Item	Persmean-Lucene	Popular-Lucene	SVD-Lucene	User-Lucene	Popular-Persmean	SVD-Persmean	User-Persmean	SVD-Popular	User-Popular	User-SVD
Z	-2,360[b]	-2,714[b]	-1,667[b]	-2,530[b]	-1,933[b]	-,412[b]	-1,403[c]	-1,200[c]	-1,276[c]	-2,041[c]	-2,111[c]	-1,983[c]	-,345[b]	,000[d]	-,276[c]
Asymp. Sig.	,018	,007	,096	,011	,053	,680	,161	,230	,202	,041	,035	,047	,730	1,000	,783

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 4-107: Wilcoxon signed rank test Q17

We can appreciate that *ItemItem* is clearly the algorithm that recommend the best movies, although it does not have statistically significant differences with *UserUser* and *Popular*. Moreover, *Popular*, *SVD* and *UserUser* are better than *Persmean*.

4.5.2 Group members' opinion

To know if it is possible to make recommendations for groups, treating a group as a single pseudo user, we have asked some questions to the group members. We have done it twice: one time after asking them to rate the top 100 movies, and another time after giving them the recommendations of ours algorithms.

4.5.2.1 Pre-Recommendations

The first question asked after they rate the top 100 movies together was: *How have you decided the rating of each movie?*

Looking at the answers, we can distinguish three different ways used by the groups to reach an agreement. One option is to give individual rating to the movies by each group member and then averaging these ratings to obtain a final rating (aggregating ratings); the second option is by democratic decision; and the last option is to examine pros and cons of each movie and reach an agreement.

The second question was: *Was it difficult or easy to reach an agreement? Why?*

We found differences in the answers depending on individual preferences of the group' members. When they have similar tastes, they find it easy to reach an agreement. In contrast, when they have some dissimilarities, it is more difficult. Some of the groups highlight the different preferences expressed by gender since, in their opinion, men prefer Sci-Fi while women prefer animated movies.

Finally, the last question was: *Where have you found difficulties?*

The answers to this question were quite similar since all of the groups indicate rating a movie when two members have opposite opinion about it as the biggest difficulty. Moreover, they have also found difficulty when some of the group members had not seen a movie. Nevertheless, they point out that one of the solutions for the opposite preferences was to average the ratings.

4.5.2.2 Post-Recommendations

The first question asked after giving them the recommendations by each algorithm was*: How have you decided which the best list is?*

The answers were quite similar. Nearly all of the groups said that they have decided it talking among them and reasoning their argumentations to reach an agreement, although most of the group members had similar tastes and it made the decision easy.

The second question was: Was *it easy or difficult to reach an agreement? Why?*

Only one of the ten groups that completed the questionnaire found it difficult. The other groups told us that it was easy since in their case all the group members like the same kind of movies.

This question points out the importance of the similarity among group members, being the better similarity, the better perception that the group members have of the recommendations.

Finally, the last question was: *Where have you found difficulties?*

Only a minority of the groups indicated that they found no difficulties, while a huge number of the groups indicated that the highest difficulty found was to decide the best list. This difficulty lies in the fact of having opposite preferences. Although users say they have a huge similarity among the other group members, there can be discrepancies about some movies. One user can love “The butterfly effect” while other can hate it. However, both of them can like Sci-Fi and Thriller movies although they do not agree on this particular movie.

4.5.3 Discussion

Due to the small number of groups that have participated in our survey, it is difficult to extrapolate conclusions taking into account the qualitative metrics. Nevertheless, we can compare the results obtained with the individual users' conclusions.

Although we have a high risk extrapolating the results in relation to groups, we can do it since they are almost the same as the obtained in the analysis of the individual users. Taking *Accuracy* into account, the best algorithms are *ItemItem* and *Popular* while the worst are *Persmean* and *Lucene*. We only know that *SVD* and *UserUser* are in the 3rd and 4th position of the ranking but we do not know which one is better.

The perception of the groups is that they can trust in *ItemItem* since this collaborative filtering algorithm is the one that best understands their taste, followed by *Popular* and *SVD*. Regarding the other three algorithms, we can only say that groups think that these algorithms do not understands them.

In this case, *Novelty* has again a negative influence on the group' perception of the algorithm. The algorithms with more surprising movies but at the same time with more movies that the groups will not consider are *Persmean* and *Lucene*.

Thus, once we have analysed all the metrics, we can underline that *ItemItem* is the algorithm that best satisfies the groups' perception of the recommendations, followed by *Popular*. However, the worst algorithm is *Persmean* due to the high number of novel movies that are included on its recommendations, followed by *Lucene* for the same reason.

As we have seen with the individual users' analysis, *Novelty* has a great negative effect on the satisfaction of the groups with the recommender system, since novel items still have to be evaluated and introduce some kind of doubt. In contrast, known items introduce trust in the system, and therefore satisfaction. Therefore, lists without known items have a negative effect on the satisfaction.

5 CONCLUSION

In this dissertation we have seen two evaluations of recommender systems with the aim of understanding users' perceptions of the quality of a recommender system, particularly concentrating on the quality of an algorithm.

First of all, an evaluation of six different groups of algorithms has been carried out using LensKit with the purpose of achieving the highest performance in each algorithm. Furthermore, to theoretically analyse the quality of these algorithms we have focused on the compute of objective metrics such as RMSE, nDCG and Entropy.

Following this a questionnaire was created to obtain ratings from real users that allowed us to work out an online evaluation. These ratings were then added to the 10M dataset from MovieLens and LensKit using six lists of recommendations for each user. A second survey was created and sent to the users that filled out the first questionnaire. The main aim of this survey was to evaluate, through 17 questions, the users' perception about different metrics such as *Accuracy*, *Understands Me, Diversity*, *Novelty*, *Effectiveness* and *Quality*.

In addition to this, the same process was applied to analyse group recommendations. The only difference was that in the groups´ questionnaire some additional open questions were added with the purpose of letting the groups give their opinion concerning any difficulties found. In this way an analysis of the viability of these simple ways in which to create group recommendations has been carried out.

Finally, a comparison between objective and subjective metrics was conducted.

This study allow us to highlight nDCG as the best metric in which to measure the quality of the systems. However, our online evaluation shows that users perceive the weaknesses present at each algorithm It is therefore important to take into account other metrics in addition to *Accuracy* such as *Novelty* or *Effectiveness* that could have a negative effect on users' perception of the system.

In this way, collaborative filtering algorithm by Item has proven to be the best algorithm as perceived by users. Saying this, it still has some weaknesses which need improvement.

Another important conclusion derived from this dissertations is that it is possible to make easily recommendations to groups. Once we have the group' ratings, the results highlight that the recommendations of the algorithms work with groups as well as with individual users.

6 FUTURE RESEARCH

One of the most striking issues is that it has been proved that the quality of a recommender system is strongly related to the perception that users have of it. More research is needed to improve the weaknesses appreciated on the algorithms. Therefore, new metrics have to been developed to measure other qualitative aspects in addition to accuracy.

Moreover, due to the number of groups that filled our survey, we could not make a study of the influence of the size of the groups in the results. Future research can include the evaluation of a higher number of groups to investigate if the size of the groups can influence their perception of the system. Furthermore, the gender of the groups' members can be analysed to highlight the differences appreciated between men and women taking into account their perception of the system.

7 REFERENCES

[1] Amatrian, X. (2011, April 7) .*Recommender Systems: We're doing it (all) wrong.* [Web log post]. Retrieved from http://technocalifornia.blogspot.com.es/2011/04/recommender-systems-were-doing-it-all.html

[2] Amatriain, X. (2013, July 23). Recommendations as Personalized Learning to Rank [Web log post]. Retrieved from http://technocalifornia.blogspot.com.es/2013/07/recommendations-as personalized.html

[3] Arekar, T., Sonar, R., Uke, N. (2015). *A Survey on Recommendation System.* International Journal of Innovative Research in Advanced Engineering (IJIRAE), 2(1).

[4] Baltrunas, L. (2011). *Context-Aware Collaborative filtering Recommender Systems.* (Doctoral Dissertation). Retrieved from http://baltrunas.info/research-menu/research-thesiss

[5] Bellogín, A. (2012). *Recommender System Performance Evaluation and Prediction: An Information Retrieval Perspective*. (Dissertation). Retrieved from http://ir.ii.uam.es/~alejandro/thesis/thesis-bellogin.pdf

[6] Burke, R., Felfering, A., Göker, M. (2011). *Recommender Systems: An Overview*. Association for the Advancement of Artificial Intelligence, 32(3).

[7] Cano, A. (2008, July). Técnicas conversacionales para la recogida de datos en investigación cualitativa: El grupo de discusión (I). Nure Investigación. Retrieved from http://www.nureinvestigacion.es/FICHEROS_ADMINISTRADOR/F_METODOLOGICA/formetod_35116200811150.pdf

[8] Cantador, I. (2008). *Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach.* (Disertation). Retrieved from https://repositorio.uam.es/handle/10486/1271

[9] Carleton College. Dimensionality Reduction and the Singular Value Decomposition. Retrieved November 24, 2014 from http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/svd.html

[10] Chandrashekhar, H., Bhasker, B. (2011) Personalized recommender system using entropy based collaborative filtering technique. Journal of Electronic Commerce Research, 12(3), 214-237. Retrieved from: http://www.researchgate.net/profile/Bharat_Bhasker/publication/267806087_PERSONALIZED_RECOMMENDER_SYSTEM_USING_ENTROPY_BASED_COLLABORATIVE_FILTERING_TECHNIQUE/links/00b7d5355442c82080000000.pdf

[11] Chen, Y., Wu, C., Xie, M., Guo, X. (2011). *Solving the Sparsity Problem in Recommender Systems Using Association Retrieval.* JCP, 6(9), 1896-1902.

[12] De Pessemier, T., Dooms, S., Martens, L. (2014). *Comparison of group recommendation algorithms.* Multimedia Tools and Applications, 72(3), 2497-2541. [doi>10.1007/s11042-013-1563-0]

[13] Ekstrand, M., Ludwig, M., Konstan, J., Riedl, J. (2011). *Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit.* Proceedings of the eleventh ACM conference on Recommender systems, Chicago: ACM. [doi>10.1145/2043932.2043958]

[14] Ekstrand, M. (2014). *Towards Recommender Engineering: Tools and Experiments in Recommender Differences*. Ph.D Thesis, University of Minnesota. Retrieved from http://elehack.net/research/thesis/

[15] Felfernig, A., Burke, R. (2008, August) *Constraint-based recommender systems: technologies and research issues.* Proceedings of the tenth international conference on Electronic commerce, Innsbruck, Austria: ACM. [doi>10.1145/1409540.1409544]

[16] Grouplens Research. (n.d. a) What is GroupLens. Retrieved October 6, 2014 from http://grouplens.org/about/what-is-grouplens/

[17] Grouplens Research. (n.d. b) Datasets. Retrieved October 6, 2014 from http://files.grouplens.org/datasets/movielens

[18] Grouplens Research. (n.d. c) MovieLens. Retrieved December 11, 2014 from http://movielens.org

[19] Herlocker, J., Konstan, J., Terveen, L., Riedl, J. (2004). *Evaluating collaborative filtering recommender systems*. ACM Transactions on Information Systems (TOIS), 22(1), 5-53. [doi>10.1145/963770.963772]

[20] Hingston, M. (2006). User Friendly Recommender Systems. (Dissertation). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.136.515

[21] Kluver, D., Konstan, J. (2014, October). *Evaluating Recommender Behaviour for New Users.* Proceedings of the eighth ACM conference on Recommender systems, Silicon Valley: ACM. [doi>10.1145/2645710.2645742]

[22] Knijnenburg, B., Willemsen, M., Kobsa, A. (2011, October). *A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems*. Proceedings of the fifth ACM conference on Recommender systems, Chicago: ACM. [doi>10.1145/2043932.2043993]

[23] Knijnenburg, B., Willemsen, M., Gantner, Z., Soncu, H., Newell, C., (2012), *Explaining the user experience of recommender systems.* User Modeling and User-Adapted Interaction, 22(4-5), 441-504. [doi>10.1007/s11257-011-9118-4]

[24] Konstan, J., Riedl, J., (2012). R*ecommender systems: from algorithms to user experience*. User Modeling and User-Adapted Interaction, 22(1-2), 101-123. [doi>10.1007/s11257-011-9112-x]

[25] Lei Li (2014).*Next Generation of Recommender Systems: Algorithms and Applications*. (Doctoral Dissertation). Retrieved from http://digitalcommons.fiu.edu/etd/1446/. (FIU Electronic Theses and Dissertations. Paper 1446)

[26] LensKit Contributors (2010a). Item-Item Collaborative Filtering. Retrieved September 29, 2014 from http://LensKit.org/documentation/algorithms/item-item/

[27] LensKit Contributors (2010b). User-User Collaborative Filtering. Retrieved September 29, 2014 from http://LensKit.org/documentation/algorithms/item-item/

[28] LensKit Contributors (2010c). Matrix Factorization CF. Retrieved September 29, 2014 from http://LensKit.org/documentation/algorithms/svd/

[29] LensKit. (n.d.). Retrieved October 27, 2014, from http://www.recsyswiki.com/wiki/LensKit

[30] Lops, P., de Gemmis, M., Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. Ricci, F., Rokach, L., Shapira, B., Kantor, P., Recommender Systems Handbook. (pp. 73-105). New York, United States: Springer US.

[31] Masisi, L., Nelwamondo, V., Marwala, T. (2008, November). *The use of entropy to measure structural diversity.* Proceedings of the fourth IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Orlando: IEEE [doi>10.1109/ICCCYB.2008.4721376]

[32] McNee, S., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S., Rashid, Al M., Konstan, J. (2002, November). *On the recommending of citations for research papers*. Proceedings of the 2002 ACM Conference on Computer supported cooperative work, New Orleans: ACM [doi>10.1145/587078.587096]

[33] McNee, S., Riedl, J. & Konstan, J. (2006, April).*Making Recommendations Better: An Analytic Model for Human-Recommender Interaction.* Proceeding of the premier international conference for human-computer interaction: CHI 2006, Montréal: ACM. [doi>10.1145/1125451.1125660]

[34] McNee, S. (2006*). Meeting User Information Needs in Recommender Systems*. (Doctoral Dissertation). Retrieved from http://dl.acm.org/citation.cfm?id=1237125

[35] Middleton, S. (2003). *Capturing knowledge of user preferences with recommender systems.* (Dissertation). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.298.3862&rep=rep1&type=pdf

[36] Mortensen, M. (2007). *Design and Evaluation of a Recommender System*. (Dissertation). Retrieved from http://munin.uit.no/handle/10037/762

[37] Motion Picture Association of America (MPAA). (2014) Theatrical Market Statistics. Retrieved April 30, from http://www.mpaa.org/wp-content/uploads/2014/03/MPAA-Theatrical-Market-Statistics-2013_032514-v2.pdf

[38] Murray, H. (1966). Methods for Satisfying the Needs of the Scientist and the Engineer for Scientific and Technical Communication. Redstone Scientific Information Center. p1. Retrieved December 2014 from http://www.dtic.mil/dtic/tr/fulltext/u2/627845.pdf

[39] Özgöbek Ö., Shabib, N., Atle, J. (2014, July) *Data sets and news recommendation*. Proceedings of 2nd International Workshop on News Recommendation and Analytics, Denmark: CEUR-WS

[40] Pearl Pu, Li Chen (2011, October).*A User-Centric Evaluation Framework of Recommender Systems.* Proceedings of the fifth ACM conference on Recommender systems, Chicago: ACM. [doi>10.1145/2043932.2043962]

[41] Pearl Pu, Li Chen, Rong Hu. (2012). *Evaluating recommender systems from the user's perspective: survey of the state of the art*. User Modeling and User-Adapted Interaction, 22(4-5), 317-355. [doi>10.1007/s11257-011-9115-7]

[42] Quijano, L. (2010). *Impacto de los factores y organizaciones sociales en los procesos de recomendación para grupos*. (Dissertation). Retrieved from http://eprints.ucm.es/11321/

[43] Ricci, F., Rokach, L. Shapira, B., Kantor, P. (2011) *Recommender Systems Handbook*. New York, United States: Springer US.

[44] Shani, G. & Gunawardana, A. (2011). Evaluating Recommendation Systems. In Ricci, F., Rokach, L. Shapira, B., Kantor, P. (Eds.), *Recommender Systems Handbook*. (pp. 257-297). New York, United States: Springer US.

[45] Shi, Yue. (2013).*Ranking and Context-awareness in Recommender Systems*. (Dissertation). Retrieved from http://repository.tudelft.nl/view/ir/uuid%3Af7d3977e-f191-40d4-8f27-784a32902a55. [doi>10.4233/uuid:f7d3977e-f191-40d4-8f27-784a32902a55]

[46] Sinha, R., Swearingen, K. (2002, April). *The role of transparency in recommender systems*, CHI '02 extended abstracts on Human factors in computing systems, Minnesota: ACM. [doi>10.1145/506443.506619]

[47] Soni, R. (2012, December 1) *Hybrid recommender system and why you should know* [Web log post]. Retrieved from http://sonirajan.com/hybrid-recommender-system-and-why-you-should-know-sean-owen/

[48] Subramaniam, V. (2008). *Programming Groovy.* Texas, United States: The Pragmatic Bookshelf.

[49] Survey Monkey, Retrieved April 23, from http://help.surveymonkey.com/articles/en_US/kb/What-is-the-Rating-Average-and-how-is-it-calculated

[50] Team Gwava. (2014). How Much Data is Created on the Internet Each Day?. *Gwava.* Retrieved 20, February 2015 from http://www.gwava.com/blog/internet-data-created-daily-2014/

[51] Thuy Ngoc Nguyen, An Te Nguyen. (2013, October) *Towards Context-aware Recommendations: Strategies for Exploiting Multi-criteria Communities.* Proceedings of the ninth IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Texas: IEEE

[52] Universidad de Sevilla. (2008, September). Análisis de datos en la investigación educativa. Retrieved April, 2015, from: http://ocwus.us.es/metodos-de-investigacion-y-diagnostico-en-educacion/analisis-de-datos-en-la-investigacion-educativa/Bloque_I/page_100.htm.

[53] Wen Wu, Liang He, Jing Yang (2012, August). Evaluating Recommender Systems. Proceedings on the seventh international conference on Digital Information Management: ICDIM 2012, Macau: IEEE. [doi>10.1109/ICDIM.2012.6360092]

[54] Willis, J. (2011). *Tag-based Recommender System*. (Dissertation). Retrieved from http://repository.tudelft.nl/view/ir/uuid:93c50b3a-cb16-4d1b-b8cf-e2145a42128a/

[55] Zanardi, V., Capra, L. (2011). A Scalable Tag-Based Recommender System for New Users of the Social Web. Hameurlain, A., Liddle, S., Schewe, K., Zhou, X. (Eds.), *Database and Expert Systems Applications* (pp.542-557). Springer Berlin Heidelberg.

[56] Zhao, T., Shang, M. (2010, July). *User-based Collaborative-Filtering Recommendation Algorithms on Hadoop*. International Conference on Knowledge Discovery and Data Mining, Washington: IEEE. [doi> 10.1109/WKDD.2010.54]

8 APPENDIX A

In this appendix some additional tables from the Wilcoxon signed rank test are shown.

8.1 INDIVIDUAL USERS

8.1.1 Q13 - Do you think that the recommender is recommending interesting content you hadn't previously consider?

Ranks

		N	Mean Rank	Sum of Ranks
Q13*LuceneEFFECTIVENESS* - Q13Item*EFFECTIVENESS*	Negative Ranks	23[a]	17,85	410,50
	Positive Ranks	10[b]	15,05	150,50
	Ties	17[c]		
	Total	50		
Q13Pers*EFFECTIVENESS* - Q13Item*EFFECTIVENESS*	Negative Ranks	30[d]	17,22	516,50
	Positive Ranks	4[e]	19,63	78,50
	Ties	16[f]		
	Total	50		
Q13*PopularEFFECTIVENESS* - Q13Item*EFFECTIVENESS*	Negative Ranks	16[g]	15,06	241,00
	Positive Ranks	15[h]	17,00	255,00
	Ties	19[i]		
	Total	50		
Q13*SVDEFFECTIVENESS* - Q13Item*EFFECTIVENESS*	Negative Ranks	13[j]	14,96	194,50
	Positive Ranks	16[k]	15,03	240,50
	Ties	21[l]		
	Total	50		
Q13User*EFFECTIVENESS* - Q13Item*EFFECTIVENESS*	Negative Ranks	15[m]	13,87	208,00
	Positive Ranks	12[n]	14,17	170,00
	Ties	23[o]		
	Total	50		
Q13Pers*EFFECTIVENESS* - Q13*LuceneEFFECTIVENESS*	Negative Ranks	16[p]	14,56	233,00
	Positive Ranks	9[q]	10,22	92,00
	Ties	25[r]		
	Total	50		
Q13*PopularEFFECTIVENESS* - Q13*LuceneEFFECTIVENESS*	Negative Ranks	8[s]	13,06	104,50
	Positive Ranks	20[t]	15,08	301,50
	Ties	22[u]		
	Total	50		
Q13*SVDEFFECTIVENESS* - Q13*LuceneEFFECTIVENESS*	Negative Ranks	8[v]	19,88	159,00
	Positive Ranks	27[w]	17,44	471,00
	Ties	15[x]		
	Total	50		
Q13User*EFFECTIVENESS* - Q13*LuceneEFFECTIVENESS*	Negative Ranks	6[y]	24,25	145,50
	Positive Ranks	24[z]	13,31	319,50
	Ties	20[aa]		
	Total	50		
Q13*PopularEFFECTIVENESS* - Q13Pers*EFFECTIVENESS*	Negative Ranks	5[ab]	9,70	48,50
	Positive Ranks	23[ac]	15,54	357,50
	Ties	22[ad]		
	Total	50		
Q13*SVDEFFECTIVENESS* - Q13Pers*EFFECTIVENESS*	Negative Ranks	2[ae]	21,75	43,50
	Positive Ranks	31[af]	16,69	517,50
	Ties	17[ag]		
	Total	50		
Q13User*EFFECTIVENESS* - Q13Pers*EFFECTIVENESS*	Negative Ranks	4[ah]	19,75	79,00
	Positive Ranks	29[ai]	16,62	482,00
	Ties	17[aj]		

	Total	50		
Q13*SVDEFFECTIVENESS* - Q13*PopularEFFECTIVENESS*	Negative Ranks	14[ak]	16,43	230,00
	Positive Ranks	17[al]	15,65	266,00
	Ties	19[am]		
	Total	50		
Q13User*EFFECTIVENESS* - Q13*PopularEFFECTIVENESS*	Negative Ranks	16[an]	18,50	296,00
	Positive Ranks	16[ao]	14,50	232,00
	Ties	18[ap]		
	Total	50		
Q13User*EFFECTIVENESS* - Q13*SVDEFFECTIVENESS*	Negative Ranks	16[aq]	12,25	196,00
	Positive Ranks	8[ar]	13,00	104,00
	Ties	26[as]		
	Total	50		

a. Q13*LuceneEFFECTIVENESS* < Q13Item*EFFECTIVENESS*
b. Q13*LuceneEFFECTIVENESS* > Q13Item*EFFECTIVENESS*
c. Q13*LuceneEFFECTIVENESS* = Q13Item*EFFECTIVENESS*
d. Q13Pers*EFFECTIVENESS* < Q13Item*EFFECTIVENESS*
e. Q13Pers*EFFECTIVENESS* > Q13Item*EFFECTIVENESS*
f. Q13Pers*EFFECTIVENESS* = Q13Item*EFFECTIVENESS*
g. Q13*PopularEFFECTIVENESS* < Q13Item*EFFECTIVENESS*
h. Q13*PopularEFFECTIVENESS* > Q13Item*EFFECTIVENESS*
i. Q13*PopularEFFECTIVENESS* = Q13Item*EFFECTIVENESS*
j. Q13*SVDEFFECTIVENESS* < Q13Item*EFFECTIVENESS*
k. Q13*SVDEFFECTIVENESS* > Q13Item*EFFECTIVENESS*
l. Q13*SVDEFFECTIVENESS* = Q13Item*EFFECTIVENESS*
m. Q13User*EFFECTIVENESS* < Q13Item*EFFECTIVENESS*
n. Q13User*EFFECTIVENESS* > Q13Item*EFFECTIVENESS*
o. Q13User*EFFECTIVENESS* = Q13Item*EFFECTIVENESS*
p. Q13Pers*EFFECTIVENESS* < Q13*LuceneEFFECTIVENESS*
q. Q13Pers*EFFECTIVENESS* > Q13*LuceneEFFECTIVENESS*
r. Q13Pers*EFFECTIVENESS* = Q13*LuceneEFFECTIVENESS*
s. Q13*PopularEFFECTIVENESS* < Q13*LuceneEFFECTIVENESS*
t. Q13*PopularEFFECTIVENESS* > Q13*LuceneEFFECTIVENESS*
u. Q13*PopularEFFECTIVENESS* = Q13*LuceneEFFECTIVENESS*
v. Q13*SVDEFFECTIVENESS* < Q13*LuceneEFFECTIVENESS*
w. Q13*SVDEFFECTIVENESS* > Q13*LuceneEFFECTIVENESS*
x. Q13*SVDEFFECTIVENESS* = Q13*LuceneEFFECTIVENESS*
y. Q13User*EFFECTIVENESS* < Q13*LuceneEFFECTIVENESS*
z. Q13User*EFFECTIVENESS* > Q13*LuceneEFFECTIVENESS*
aa. Q13User*EFFECTIVENESS* = Q13*LuceneEFFECTIVENESS*
ab. Q13*PopularEFFECTIVENESS* < Q13Pers*EFFECTIVENESS*
ac. Q13*PopularEFFECTIVENESS* > Q13Pers*EFFECTIVENESS*
ad. Q13*PopularEFFECTIVENESS* = Q13Pers*EFFECTIVENESS*
ae. Q13*SVDEFFECTIVENESS* < Q13Pers*EFFECTIVENESS*
af. Q13*SVDEFFECTIVENESS* > Q13Pers*EFFECTIVENESS*
ag. Q13*SVDEFFECTIVENESS* = Q13Pers*EFFECTIVENESS*
ah. Q13User*EFFECTIVENESS* < Q13Pers*EFFECTIVENESS*
ai. Q13User*EFFECTIVENESS* > Q13Pers*EFFECTIVENESS*
aj. Q13User*EFFECTIVENESS* = Q13Pers*EFFECTIVENESS*
ak. Q13*SVDEFFECTIVENESS* < Q13*PopularEFFECTIVENESS*
al. Q13*SVDEFFECTIVENESS* > Q13*PopularEFFECTIVENESS*
am. Q13*SVDEFFECTIVENESS* = Q13*PopularEFFECTIVENESS*
an. Q13User*EFFECTIVENESS* < Q13*PopularEFFECTIVENESS*
ao. Q13User*EFFECTIVENESS* > Q13*PopularEFFECTIVENESS*
ap. Q13User*EFFECTIVENESS* = Q13*PopularEFFECTIVENESS*
aq. Q13User*EFFECTIVENESS* < Q13*SVDEFFECTIVENESS*
ar. Q13User*EFFECTIVENESS* > Q13*SVDEFFECTIVENESS*
as. Q13User*EFFECTIVENESS* = Q13*SVDEFFECTIVENESS*

8.1.2 Q16 - How much do you think that the recommended movies are relevant?

Ranks

		N	Mean Rank	Sum of Ranks
Q16*LuceneQUALITY* - Q16Item*QUALITY*	Negative Ranks	31[a]	19,79	613,50
	Positive Ranks	8[b]	20,81	166,50
	Ties	11[c]		
	Total	50		
Q16*PersmeanQUALITY* - Q16Item*QUALITY*	Negative Ranks	41[d]	24,91	1021,50
	Positive Ranks	6[e]	17,75	106,50
	Ties	3[f]		
	Total	50		
Q16*PopularQUALITY* - Q16Item*QUALITY*	Negative Ranks	23[g]	18,46	424,50
	Positive Ranks	16[h]	22,22	355,50
	Ties	11[i]		
	Total	50		
Q16*SVDQUALITY* - Q16Item*QUALITY*	Negative Ranks	21[j]	16,31	342,50
	Positive Ranks	10[k]	15,35	153,50
	Ties	19[l]		
	Total	50		
Q16User*QUALITY* - Q16Item*QUALITY*	Negative Ranks	21[m]	15,05	316,00
	Positive Ranks	9[n]	16,56	149,00
	Ties	20[o]		
	Total	50		
Q16*PersmeanQUALITY* - Q16*LuceneQUALITY*	Negative Ranks	23[p]	18,09	416,00
	Positive Ranks	10[q]	14,50	145,00
	Ties	17[r]		
	Total	50		
Q16*PopularQUALITY* - Q16*LuceneQUALITY*	Negative Ranks	9[s]	17,39	156,50
	Positive Ranks	29[t]	20,16	584,50
	Ties	12[u]		
	Total	50		
Q16*SVDQUALITY* - Q16*LuceneQUALITY*	Negative Ranks	12[v]	18,38	220,50
	Positive Ranks	26[w]	20,02	520,50
	Ties	12[x]		
	Total	50		
Q16User*QUALITY* - Q16*LuceneQUALITY*	Negative Ranks	11[y]	19,59	215,50
	Positive Ranks	26[z]	18,75	487,50
	Ties	13[aa]		
	Total	50		
Q16*PopularQUALITY* - Q16*PersmeanQUALITY*	Negative Ranks	5[ab]	12,40	62,00
	Positive Ranks	33[ac]	20,58	679,00
	Ties	12[ad]		
	Total	50		
Q16*SVDQUALITY* - Q16*PersmeanQUALITY*	Negative Ranks	7[ae]	18,93	132,50
	Positive Ranks	34[af]	21,43	728,50
	Ties	9[ag]		
	Total	50		
Q16User*QUALITY* - Q16*PersmeanQUALITY*	Negative Ranks	4[ah]	15,75	63,00
	Positive Ranks	34[ai]	19,94	678,00
	Ties	12[aj]		
	Total	50		
Q16*SVDQUALITY* - Q16*PopularQUALITY*	Negative Ranks	24[ak]	19,75	474,00
	Positive Ranks	15[al]	20,40	306,00
	Ties	11[am]		
	Total	50		
Q16User*QUALITY* - Q16*PopularQUALITY*	Negative Ranks	20[an]	21,00	420,00
	Positive Ranks	17[ao]	16,65	283,00
	Ties	13[ap]		
	Total	50		
Q16User*QUALITY* - Q16*SVDQUALITY*	Negative Ranks	12[aq]	13,33	160,00
	Positive Ranks	14[ar]	13,64	191,00
	Ties	24[as]		

	Total	50

a. Q16*LuceneQUALITY* < Q16Item*QUALITY*
b. Q16*LuceneQUALITY* > Q16Item*QUALITY*
c. Q16*LuceneQUALITY* = Q16Item*QUALITY*
d. Q16*PersmeanQUALITY* < Q16Item*QUALITY*
e. Q16*PersmeanQUALITY* > Q16Item*QUALITY*
f. Q16*PersmeanQUALITY* = Q16Item*QUALITY*
g. Q16*PopularQUALITY* < Q16Item*QUALITY*
h. Q16*PopularQUALITY* > Q16Item*QUALITY*
i. Q16*PopularQUALITY* = Q16Item*QUALITY*
j. Q16*SVDQUALITY* < Q16Item*QUALITY*
k. Q16*SVDQUALITY* > Q16Item*QUALITY*
l. Q16*SVDQUALITY* = Q16Item*QUALITY*
m. Q16User*QUALITY* < Q16Item*QUALITY*
n. Q16User*QUALITY* > Q16Item*QUALITY*
o. Q16User*QUALITY* = Q16Item*QUALITY*
p. Q16*PersmeanQUALITY* < Q16*LuceneQUALITY*
q. Q16*PersmeanQUALITY* > Q16*LuceneQUALITY*
r. Q16*PersmeanQUALITY* = Q16*LuceneQUALITY*
s. Q16*PopularQUALITY* < Q16*LuceneQUALITY*
t. Q16*PopularQUALITY* > Q16*LuceneQUALITY*
u. Q16*PopularQUALITY* = Q16*LuceneQUALITY*
v. Q16*SVDQUALITY* < Q16*LuceneQUALITY*
w. Q16*SVDQUALITY* > Q16*LuceneQUALITY*
x. Q16*SVDQUALITY* = Q16*LuceneQUALITY*
y. Q16User*QUALITY* < Q16*LuceneQUALITY*
z. Q16User*QUALITY* > Q16*LuceneQUALITY*
aa. Q16User*QUALITY* = Q16*LuceneQUALITY*
ab. Q16*PopularQUALITY* < Q16*PersmeanQUALITY*
ac. Q16*PopularQUALITY* > Q16*PersmeanQUALITY*
ad. Q16*PopularQUALITY* = Q16*PersmeanQUALITY*
ae. Q16*SVDQUALITY* < Q16*PersmeanQUALITY*
af. Q16*SVDQUALITY* > Q16*PersmeanQUALITY*
ag. Q16*SVDQUALITY* = Q16*PersmeanQUALITY*
ah. Q16User*QUALITY* < Q16*PersmeanQUALITY*
ai. Q16User*QUALITY* > Q16*PersmeanQUALITY*
aj. Q16User*QUALITY* = Q16*PersmeanQUALITY*
ak. Q16*SVDQUALITY* < Q16*PopularQUALITY*
al. Q16*SVDQUALITY* > Q16*PopularQUALITY*
am. Q16*SVDQUALITY* = Q16*PopularQUALITY*
an. Q16User*QUALITY* < Q16*PopularQUALITY*
ao. Q16User*QUALITY* > Q16*PopularQUALITY*
ap. Q16User*QUALITY* = Q16*PopularQUALITY*
aq. Q16User*QUALITY* < Q16*SVDQUALITY*
ar. Q16User*QUALITY* > Q16*SVDQUALITY*
as. Q16User*QUALITY* = Q16*SVDQUALITY*

8.1.3 Q17 - Do you think that the recommended movies are not well-chosen?

Ranks

		N	Mean Rank	Sum of Ranks
Q17*LuceneQUALITY* - Q17Item*QUALITY*	Negative Ranks	23[a]	17,98	413,50
	Positive Ranks	13[b]	19,42	252,50
	Ties	14[c]		
	Total	50		
Q17*PersmeanQUALITY* - Q17Item*QUALITY*	Negative Ranks	29[d]	20,64	598,50
	Positive Ranks	9[e]	15,83	142,50
	Ties	12[f]		
	Total	50		
Q17*PopularQUALITY* - Q17Item*QUALITY*	Negative Ranks	14[g]	12,14	170,00
	Positive Ranks	15[h]	17,67	265,00
	Ties	21[i]		
	Total	50		
Q17*SVDQUALITY* - Q17Item*QUALITY*	Negative Ranks	16[j]	15,78	252,50
	Positive Ranks	15[k]	16,23	243,50
	Ties	19[l]		
	Total	50		
Q17User*QUALITY* - Q17Item*QUALITY*	Negative Ranks	14[m]	12,96	181,50
	Positive Ranks	13[n]	15,12	196,50
	Ties	23[o]		
	Total	50		
Q17*PersmeanQUALITY* - Q17*LuceneQUALITY*	Negative Ranks	20[p]	16,50	330,00
	Positive Ranks	8[q]	9,50	76,00
	Ties	22[r]		
	Total	50		
Q17*PopularQUALITY* - Q17*LuceneQUALITY*	Negative Ranks	10[s]	22,45	224,50
	Positive Ranks	26[t]	16,98	441,50
	Ties	14[u]		
	Total	50		
Q17*SVDQUALITY* - Q17*LuceneQUALITY*	Negative Ranks	13[v]	21,65	281,50
	Positive Ranks	24[w]	17,56	421,50
	Ties	13[x]		
	Total	50		
Q17User*QUALITY* - Q17*LuceneQUALITY*	Negative Ranks	13[y]	17,31	225,00
	Positive Ranks	21[z]	17,62	370,00
	Ties	16[aa]		
	Total	50		
Q17*PopularQUALITY* - Q17*PersmeanQUALITY*	Negative Ranks	6[ab]	12,50	75,00
	Positive Ranks	26[ac]	17,42	453,00
	Ties	18[ad]		
	Total	50		
Q17*SVDQUALITY* - Q17*PersmeanQUALITY*	Negative Ranks	5[ae]	18,90	94,50
	Positive Ranks	29[af]	17,26	500,50
	Ties	16[ag]		
	Total	50		
Q17User*QUALITY* - Q17*PersmeanQUALITY*	Negative Ranks	7[ah]	16,71	117,00
	Positive Ranks	29[ai]	18,93	549,00
	Ties	14[aj]		
	Total	50		
Q17*SVDQUALITY* - Q17*PopularQUALITY*	Negative Ranks	19[ak]	16,03	304,50
	Positive Ranks	13[al]	17,19	223,50
	Ties	18[am]		
	Total	50		
Q17User*QUALITY* - Q17*PopularQUALITY*	Negative Ranks	18[an]	21,50	387,00
	Positive Ranks	18[ao]	15,50	279,00
	Ties	14[ap]		
	Total	50		
Q17User*QUALITY* - Q17*SVDQUALITY*	Negative Ranks	13[aq]	14,88	193,50
	Positive Ranks	15[ar]	14,17	212,50

	Ties	22[as]
	Total	50

a. Q17*LuceneQUALITY* < Q17Item*QUALITY*
b. Q17*LuceneQUALITY* > Q17Item*QUALITY*
c. Q17*LuceneQUALITY* = Q17Item*QUALITY*
d. Q17*PersmeanQUALITY* < Q17Item*QUALITY*
e. Q17*PersmeanQUALITY* > Q17Item*QUALITY*
f. Q17*PersmeanQUALITY* = Q17Item*QUALITY*
g. Q17*PopularQUALITY* < Q17Item*QUALITY*
h. Q17*PopularQUALITY* > Q17Item*QUALITY*
i. Q17*PopularQUALITY* = Q17Item*QUALITY*
j. Q17*SVDQUALITY* < Q17Item*QUALITY*
k. Q17*SVDQUALITY* > Q17Item*QUALITY*
l. Q17*SVDQUALITY* = Q17Item*QUALITY*
m. Q17User*QUALITY* < Q17Item*QUALITY*
n. Q17User*QUALITY* > Q17Item*QUALITY*
o. Q17User*QUALITY* = Q17Item*QUALITY*
p. Q17*PersmeanQUALITY* < Q17*LuceneQUALITY*
q. Q17*PersmeanQUALITY* > Q17*LuceneQUALITY*
r. Q17*PersmeanQUALITY* = Q17*LuceneQUALITY*
s. Q17*PopularQUALITY* < Q17*LuceneQUALITY*
t. Q17*PopularQUALITY* > Q17*LuceneQUALITY*
u. Q17*PopularQUALITY* = Q17*LuceneQUALITY*
v. Q17*SVDQUALITY* < Q17*LuceneQUALITY*
w. Q17*SVDQUALITY* > Q17*LuceneQUALITY*
x. Q17*SVDQUALITY* = Q17*LuceneQUALITY*
y. Q17User*QUALITY* < Q17*LuceneQUALITY*
z. Q17User*QUALITY* > Q17*LuceneQUALITY*
aa. Q17User*QUALITY* = Q17*LuceneQUALITY*
ab. Q17*PopularQUALITY* < Q17*PersmeanQUALITY*
ac. Q17*PopularQUALITY* > Q17*PersmeanQUALITY*
ad. Q17*PopularQUALITY* = Q17*PersmeanQUALITY*
ae. Q17*SVDQUALITY* < Q17*PersmeanQUALITY*
af. Q17*SVDQUALITY* > Q17*PersmeanQUALITY*
ag. Q17*SVDQUALITY* = Q17*PersmeanQUALITY*
ah. Q17User*QUALITY* < Q17*PersmeanQUALITY*
ai. Q17User*QUALITY* > Q17*PersmeanQUALITY*
aj. Q17User*QUALITY* = Q17*PersmeanQUALITY*
ak. Q17*SVDQUALITY* < Q17*PopularQUALITY*
al. Q17*SVDQUALITY* > Q17*PopularQUALITY*
am. Q17*SVDQUALITY* = Q17*PopularQUALITY*
an. Q17User*QUALITY* < Q17*PopularQUALITY*
ao. Q17User*QUALITY* > Q17*PopularQUALITY*
ap. Q17User*QUALITY* = Q17*PopularQUALITY*
aq. Q17User*QUALITY* < Q17*SVDQUALITY*
ar. Q17User*QUALITY* > Q17*SVDQUALITY*
as. Q17User*QUALITY* = Q17*SVDQUALITY*

8.2 GROUPS

8.2.1 Q13 - Do you think that the recommender is recommending interesting content you hadn't previously consider?

Ranks

		N	Mean Rank	Sum of Ranks
Q13*LuceneEffectiveness* - Q13Item*Effectiveness*	Negative Ranks	6[a]	3,50	21,00
	Positive Ranks	0[b]	,00	,00
	Ties	4[c]		
	Total	10		
Q13*PersmeanEffectiveness* - Q13Item*Effectiveness*	Negative Ranks	8[d]	4,50	36,00
	Positive Ranks	0[e]	,00	,00
	Ties	2[f]		
	Total	10		
Q13*PopularEffectiveness* - Q13Item*Effectiveness*	Negative Ranks	4[g]	4,00	16,00
	Positive Ranks	2[h]	2,50	5,00
	Ties	4[i]		
	Total	10		
Q13*SVDEffectiveness* - Q13Item*Effectiveness*	Negative Ranks	4[j]	3,13	12,50
	Positive Ranks	1[k]	2,50	2,50
	Ties	5[l]		
	Total	10		
Q13User*Effectiveness* - Q13Item*Effectiveness*	Negative Ranks	3[m]	2,67	8,00
	Positive Ranks	1[n]	2,00	2,00
	Ties	6[o]		
	Total	10		
Q13*PersmeanEffectiveness* - Q13*LuceneEffectiveness*	Negative Ranks	3[p]	3,00	9,00
	Positive Ranks	2[q]	3,00	6,00
	Ties	5[r]		
	Total	10		
Q13*PopularEffectiveness* - Q13*LuceneEffectiveness*	Negative Ranks	2[s]	1,50	3,00
	Positive Ranks	3[t]	4,00	12,00
	Ties	5[u]		
	Total	10		
Q13*SVDEffectiveness* - Q13*LuceneEffectiveness*	Negative Ranks	2[v]	5,00	10,00
	Positive Ranks	6[w]	4,33	26,00
	Ties	2[x]		
	Total	10		
Q13User*Effectiveness* - Q13*LuceneEffectiveness*	Negative Ranks	1[y]	2,00	2,00
	Positive Ranks	4[z]	3,25	13,00
	Ties	5[aa]		
	Total	10		
Q13*PopularEffectiveness* - Q13*PersmeanEffectiveness*	Negative Ranks	0[ab]	,00	,00
	Positive Ranks	4[ac]	2,50	10,00
	Ties	6[ad]		
	Total	10		
Q13*SVDEffectiveness* - Q13*PersmeanEffectiveness*	Negative Ranks	1[ae]	3,00	3,00
	Positive Ranks	5[af]	3,60	18,00
	Ties	4[ag]		
	Total	10		
Q13User*Effectiveness* - Q13*PersmeanEffectiveness*	Negative Ranks	1[ah]	2,00	2,00
	Positive Ranks	5[ai]	3,80	19,00
	Ties	4[aj]		
	Total	10		
Q13*SVDEffectiveness* - Q13*PopularEffectiveness*	Negative Ranks	3[ak]	3,67	11,00
	Positive Ranks	3[al]	3,33	10,00
	Ties	4[am]		
	Total	10		
	Negative Ranks	2[an]	3,50	7,00

Q13User*Effectiveness* - Q13*PopularEffectiveness*	Positive Ranks	3[ao]	2,67	8,00
	Ties	5[ap]		
	Total	10		
Q13User*Effectiveness* - Q13*SVDEffectiveness*	Negative Ranks	3[aq]	3,00	9,00
	Positive Ranks	3[ar]	4,00	12,00
	Ties	4[as]		
	Total	10		

a. Q13*LuceneEffectiveness* < Q13Item*Effectiveness*
b. Q13*LuceneEffectiveness* > Q13Item*Effectiveness*
c. Q13*LuceneEffectiveness* = Q13Item*Effectiveness*
d. Q13*PersmeanEffectiveness* < Q13Item*Effectiveness*
e. Q13*PersmeanEffectiveness* > Q13Item*Effectiveness*
f. Q13*PersmeanEffectiveness* = Q13Item*Effectiveness*
g. Q13*PopularEffectiveness* < Q13Item*Effectiveness*
h. Q13*PopularEffectiveness* > Q13Item*Effectiveness*
i. Q13*PopularEffectiveness* = Q13Item*Effectiveness*
j. Q13*SVDEffectiveness* < Q13Item*Effectiveness*
k. Q13*SVDEffectiveness* > Q13Item*Effectiveness*
l. Q13*SVDEffectiveness* = Q13Item*Effectiveness*
m. Q13User*Effectiveness* < Q13Item*Effectiveness*
n. Q13User*Effectiveness* > Q13Item*Effectiveness*
o. Q13User*Effectiveness* = Q13Item*Effectiveness*
p. Q13*PersmeanEffectiveness* < Q13*LuceneEffectiveness*
q. Q13*PersmeanEffectiveness* > Q13*LuceneEffectiveness*
r. Q13*PersmeanEffectiveness* = Q13*LuceneEffectiveness*
s. Q13*PopularEffectiveness* < Q13*LuceneEffectiveness*
t. Q13*PopularEffectiveness* > Q13*LuceneEffectiveness*
u. Q13*PopularEffectiveness* = Q13*LuceneEffectiveness*
v. Q13*SVDEffectiveness* < Q13*LuceneEffectiveness*
w. Q13*SVDEffectiveness* > Q13*LuceneEffectiveness*
x. Q13*SVDEffectiveness* = Q13*LuceneEffectiveness*
y. Q13User*Effectiveness* < Q13*LuceneEffectiveness*
z. Q13User*Effectiveness* > Q13*LuceneEffectiveness*
aa. Q13User*Effectiveness* = Q13*LuceneEffectiveness*
ab. Q13*PopularEffectiveness* < Q13*PersmeanEffectiveness*
ac. Q13*PopularEffectiveness* > Q13*PersmeanEffectiveness*
ad. Q13*PopularEffectiveness* = Q13*PersmeanEffectiveness*
ae. Q13*SVDEffectiveness* < Q13*PersmeanEffectiveness*
af. Q13*SVDEffectiveness* > Q13*PersmeanEffectiveness*
ag. Q13*SVDEffectiveness* = Q13*PersmeanEffectiveness*
ah. Q13User*Effectiveness* < Q13*PersmeanEffectiveness*
ai. Q13User*Effectiveness* > Q13*PersmeanEffectiveness*
aj. Q13User*Effectiveness* = Q13*PersmeanEffectiveness*
ak. Q13*SVDEffectiveness* < Q13*PopularEffectiveness*
al. Q13*SVDEffectiveness* > Q13*PopularEffectiveness*
am. Q13*SVDEffectiveness* = Q13*PopularEffectiveness*
an. Q13User*Effectiveness* < Q13*PopularEffectiveness*
ao. Q13User*Effectiveness* > Q13*PopularEffectiveness*
ap. Q13User*Effectiveness* = Q13*PopularEffectiveness*
aq. Q13User*Effectiveness* < Q13*SVDEffectiveness*
ar. Q13User*Effectiveness* > Q13*SVDEffectiveness*
as. Q13User*Effectiveness* = Q13*SVDEffectiveness*

8.2.2 Q16 - How much do you think that the recommended movies are relevant?

Ranks

		N	Mean Rank	Sum of Ranks
Q16*LuceneQuality* - Q16Item*Quality*	Negative Ranks	10[a]	5,50	55,00
	Positive Ranks	0[b]	,00	,00
	Ties	0[c]		
	Total	10		
Q16*PersmeanQuality* - Q16Item*Quality*	Negative Ranks	10[d]	5,50	55,00
	Positive Ranks	0[e]	,00	,00
	Ties	0[f]		
	Total	10		
Q16*PopularQuality* - Q16Item*Quality*	Negative Ranks	6[g]	4,42	26,50
	Positive Ranks	2[h]	4,75	9,50
	Ties	2[i]		
	Total	10		
Q16*SVDQuality* - Q16Item*Quality*	Negative Ranks	6[j]	3,50	21,00
	Positive Ranks	0[k]	,00	,00
	Ties	4[l]		
	Total	10		
Q16User*Quality* - Q16Item*Quality*	Negative Ranks	6[m]	4,33	26,00
	Positive Ranks	1[n]	2,00	2,00
	Ties	3[o]		
	Total	10		
Q16*PersmeanQuality* - Q16*LuceneQuality*	Negative Ranks	4[p]	3,50	14,00
	Positive Ranks	2[q]	3,50	7,00
	Ties	4[r]		
	Total	10		
Q16*PopularQuality* - Q16*LuceneQuality*	Negative Ranks	1[s]	2,50	2,50
	Positive Ranks	6[t]	4,25	25,50
	Ties	3[u]		
	Total	10		
Q16*SVDQuality* - Q16*LuceneQuality*	Negative Ranks	2[v]	4,50	9,00
	Positive Ranks	8[w]	5,75	46,00
	Ties	0[x]		
	Total	10		
Q16User*Quality* - Q16*LuceneQuality*	Negative Ranks	1[y]	3,00	3,00
	Positive Ranks	6[z]	4,17	25,00
	Ties	3[aa]		
	Total	10		
Q16*PopularQuality* - Q16*PersmeanQuality*	Negative Ranks	0[ab]	,00	,00
	Positive Ranks	6[ac]	3,50	21,00
	Ties	4[ad]		
	Total	10		
Q16*SVDQuality* - Q16*PersmeanQuality*	Negative Ranks	0[ae]	,00	,00
	Positive Ranks	7[af]	4,00	28,00
	Ties	3[ag]		
	Total	10		
Q16User*Quality* - Q16*PersmeanQuality*	Negative Ranks	1[ah]	1,50	1,50
	Positive Ranks	6[ai]	4,42	26,50
	Ties	3[aj]		
	Total	10		
Q16*SVDQuality* - Q16*PopularQuality*	Negative Ranks	5[ak]	5,20	26,00
	Positive Ranks	4[al]	4,75	19,00
	Ties	1[am]		
	Total	10		
Q16User*Quality* - Q16*PopularQuality*	Negative Ranks	6[an]	4,25	25,50
	Positive Ranks	3[ao]	6,50	19,50
	Ties	1[ap]		
	Total	10		
Q16User*Quality* - Q16*SVDQuality*	Negative Ranks	4[aq]	3,38	13,50
	Positive Ranks	3[ar]	4,83	14,50

	Ties	3[as]
	Total	10

a. Q16*LuceneQuality* < Q16Item*Quality*
b. Q16*LuceneQuality* > Q16Item*Quality*
c. Q16*LuceneQuality* = Q16Item*Quality*
d. Q16*PersmeanQuality* < Q16Item*Quality*
e. Q16*PersmeanQuality* > Q16Item*Quality*
f. Q16*PersmeanQuality* = Q16Item*Quality*
g. Q16*PopularQuality* < Q16Item*Quality*
h. Q16*PopularQuality* > Q16Item*Quality*
i. Q16*PopularQuality* = Q16Item*Quality*
j. Q16*SVDQuality* < Q16Item*Quality*
k. Q16*SVDQuality* > Q16Item*Quality*
l. Q16*SVDQuality* = Q16Item*Quality*
m. Q16User*Quality* < Q16Item*Quality*
n. Q16User*Quality* > Q16Item*Quality*
o. Q16User*Quality* = Q16Item*Quality*
p. Q16*PersmeanQuality* < Q16*LuceneQuality*
q. Q16*PersmeanQuality* > Q16*LuceneQuality*
r. Q16*PersmeanQuality* = Q16*LuceneQuality*
s. Q16*PopularQuality* < Q16*LuceneQuality*
t. Q16*PopularQuality* > Q16*LuceneQuality*
u. Q16*PopularQuality* = Q16*LuceneQuality*
v. Q16*SVDQuality* < Q16*LuceneQuality*
w. Q16*SVDQuality* > Q16*LuceneQuality*
x. Q16*SVDQuality* = Q16*LuceneQuality*
y. Q16User*Quality* < Q16*LuceneQuality*
z. Q16User*Quality* > Q16*LuceneQuality*
aa. Q16User*Quality* = Q16*LuceneQuality*
ab. Q16*PopularQuality* < Q16*PersmeanQuality*
ac. Q16*PopularQuality* > Q16*PersmeanQuality*
ad. Q16*PopularQuality* = Q16*PersmeanQuality*
ae. Q16*SVDQuality* < Q16*PersmeanQuality*
af. Q16*SVDQuality* > Q16*PersmeanQuality*
ag. Q16*SVDQuality* = Q16*PersmeanQuality*
ah. Q16User*Quality* < Q16*PersmeanQuality*
ai. Q16User*Quality* > Q16*PersmeanQuality*
aj. Q16User*Quality* = Q16*PersmeanQuality*
ak. Q16*SVDQuality* < Q16*PopularQuality*
al. Q16*SVDQuality* > Q16*PopularQuality*
am. Q16*SVDQuality* = Q16*PopularQuality*
an. Q16User*Quality* < Q16*PopularQuality*
ao. Q16User*Quality* > Q16*PopularQuality*
ap. Q16User*Quality* = Q16*PopularQuality*
aq. Q16User*Quality* < Q16*SVDQuality*
ar. Q16User*Quality* > Q16*SVDQuality*
as. Q16User*Quality* = Q16*SVDQuality*

8.2.3 Q17 - Do you think that the recommended movies are not well-chosen?

Ranks

		N	Mean Rank	Sum of Ranks
Q17*LuceneQuality* - Q17Item*Quality*	Negative Ranks	8[a]	5,25	42,00
	Positive Ranks	1[b]	3,00	3,00
	Ties	1[c]		
	Total	10		
Q17*PersmeanQuality* - Q17Item*Quality*	Negative Ranks	9[d]	5,00	45,00
	Positive Ranks	0[e]	,00	,00
	Ties	1[f]		
	Total	10		
Q17*PopularQuality* - Q17Item*Quality*	Negative Ranks	5[g]	3,60	18,00
	Positive Ranks	1[h]	3,00	3,00
	Ties	4[i]		
	Total	10		
Q17*SVDQuality* - Q17Item*Quality*	Negative Ranks	7[j]	4,00	28,00
	Positive Ranks	0[k]	,00	,00
	Ties	3[l]		
	Total	10		
Q17User*Quality* - Q17Item*Quality*	Negative Ranks	6[m]	4,17	25,00
	Positive Ranks	1[n]	3,00	3,00
	Ties	3[o]		
	Total	10		
Q17*PersmeanQuality* - Q17*LuceneQuality*	Negative Ranks	3[p]	3,00	9,00
	Positive Ranks	2[q]	3,00	6,00
	Ties	5[r]		
	Total	10		
Q17*PopularQuality* - Q17*LuceneQuality*	Negative Ranks	2[s]	3,00	6,00
	Positive Ranks	5[t]	4,40	22,00
	Ties	3[u]		
	Total	10		
Q17*SVDQuality* - Q17*LuceneQuality*	Negative Ranks	2[v]	3,50	7,00
	Positive Ranks	5[w]	4,20	21,00
	Ties	3[x]		
	Total	10		
Q17User*Quality* - Q17*LuceneQuality*	Negative Ranks	1[y]	4,50	4,50
	Positive Ranks	5[z]	3,30	16,50
	Ties	4[aa]		
	Total	10		
Q17*PopularQuality* - Q17*PersmeanQuality*	Negative Ranks	0[ab]	,00	,00
	Positive Ranks	5[ac]	3,00	15,00
	Ties	5[ad]		
	Total	10		
Q17*SVDQuality* - Q17*PersmeanQuality*	Negative Ranks	1[ae]	4,00	4,00
	Positive Ranks	7[af]	4,57	32,00
	Ties	2[ag]		
	Total	10		
Q17User*Quality* - Q17*PersmeanQuality*	Negative Ranks	1[ah]	2,50	2,50
	Positive Ranks	6[ai]	4,25	25,50
	Ties	3[aj]		
	Total	10		
Q17*SVDQuality* - Q17*PopularQuality*	Negative Ranks	4[ak]	4,00	16,00
	Positive Ranks	3[al]	4,00	12,00
	Ties	3[am]		
	Total	10		
Q17User*Quality* - Q17*PopularQuality*	Negative Ranks	4[an]	3,50	14,00
	Positive Ranks	3[ao]	4,67	14,00
	Ties	3[ap]		
	Total	10		
Q17User*Quality* - Q17*SVDQuality*	Negative Ranks	2[aq]	3,25	6,50
	Positive Ranks	3[ar]	2,83	8,50
	Ties	5[as]		

Total	10

a. Q17*LuceneQuality* < Q17Item*Quality*
b. Q17*LuceneQuality* > Q17Item*Quality*
c. Q17*LuceneQuality* = Q17Item*Quality*
d. Q17*PersmeanQuality* < Q17Item*Quality*
e. Q17*PersmeanQuality* > Q17Item*Quality*
f. Q17*PersmeanQuality* = Q17Item*Quality*
g. Q17*PopularQuality* < Q17Item*Quality*
h. Q17*PopularQuality* > Q17Item*Quality*
i. Q17*PopularQuality* = Q17Item*Quality*
j. Q17*SVDQuality* < Q17Item*Quality*
k. Q17*SVDQuality* > Q17Item*Quality*
l. Q17*SVDQuality* = Q17Item*Quality*
m. Q17User*Quality* < Q17Item*Quality*
n. Q17User*Quality* > Q17Item*Quality*
o. Q17User*Quality* = Q17Item*Quality*
p. Q17*PersmeanQuality* < Q17*LuceneQuality*
q. Q17*PersmeanQuality* > Q17*LuceneQuality*
r. Q17*PersmeanQuality* = Q17*LuceneQuality*
s. Q17*PopularQuality* < Q17*LuceneQuality*
t. Q17*PopularQuality* > Q17*LuceneQuality*
u. Q17*PopularQuality* = Q17*LuceneQuality*
v. Q17*SVDQuality* < Q17*LuceneQuality*
w. Q17*SVDQuality* > Q17*LuceneQuality*
x. Q17*SVDQuality* = Q17*LuceneQuality*
y. Q17User*Quality* < Q17*LuceneQuality*
z. Q17User*Quality* > Q17*LuceneQuality*
aa. Q17User*Quality* = Q17*LuceneQuality*
ab. Q17*PopularQuality* < Q17*PersmeanQuality*
ac. Q17*PopularQuality* > Q17*PersmeanQuality*
ad. Q17*PopularQuality* = Q17*PersmeanQuality*
ae. Q17*SVDQuality* < Q17*PersmeanQuality*
af. Q17*SVDQuality* > Q17*PersmeanQuality*
ag. Q17*SVDQuality* = Q17*PersmeanQuality*
ah. Q17User*Quality* < Q17*PersmeanQuality*
ai. Q17User*Quality* > Q17*PersmeanQuality*
aj. Q17User*Quality* = Q17*PersmeanQuality*
ak. Q17*SVDQuality* < Q17*PopularQuality*
al. Q17*SVDQuality* > Q17*PopularQuality*
am. Q17*SVDQuality* = Q17*PopularQuality*
an. Q17User*Quality* < Q17*PopularQuality*
ao. Q17User*Quality* > Q17*PopularQuality*
ap. Q17User*Quality* = Q17*PopularQuality*
aq. Q17User*Quality* < Q17*SVDQuality*
ar. Q17User*Quality* > Q17*SVDQuality*
as. Q17User*Quality* = Q17*SVDQuality*