Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD IZTAPALAPA

MAESTRÍA EN CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

SISTEMAS DE RECOMENDACIÓN

Idónea Comunicación de Resultados que para obtener el grado de:

MAESTRA EN CIENCIAS
(Ciencias y Tecnologías de la Información)

PRESENTA:
Lic. Adriana Almaraz Pérez

ASESOR
Dr. John Goddard Close

SINODALES:

Dr. Adán Díaz Hernández
M. en C. Fabiola Margarita Martínez Licona
Dr. John Goddard Close

25 de Septiembre del 2013

I. RESUMEN

Los Sistemas de Recomendación son de gran utilidad para tratar la sobrecarga de información de la web ya que consisten en herramientas de software y técnicas que proveen sugerencias al usuario respecto a un producto en específico que le pueda ser de su interés. Este proyecto está enfocado a Sistema de Recomendación de Películas, no obstante puede ser adaptado para algún otro tipo de producto como música, libros, restaurantes, etc.

Se realizó experimentación para comparar el comportamiento de las recomendaciones con:

Diferentes distancias: euclidiana, Manhattan y correlación de Pearson con el fin de ver cual daba menor valor en la raíz del error cuadrático de la media (RMSE) y el error medio absoluto (MAE) en las recomendaciones.

Diferentes técnicas de agrupamiento: affinity propagation, bisecting K-Means, K-Means, K-Medoids y X-Mean.

Diferente conjunto de datos: Se realiza experimentación tomando en cuenta opiniones de usuarios y opiniones de expertos para ver cuáles dan menor error RMSE y MAE.

La implementación fue utilizada en su mayoría en MATLAB (ver A.2.), mientras WEKA (ver A.1.) fue utilizada para X-Mean y K-Means así como la implementación de K-Medoids.

El mejor resultado en la experimentación se encontró en el conjunto de usuarios utilizando distancia euclidiana, bisecting K-Means y con un K = 37.

Dentro de las conclusiones más relevantes encontradas se puede mencionar las siguientes:

- No necesariamente una opinión experta es la mejor ya que en el caso de conjunto de expertos los errores salieron más altos, esto puede deberse a la poca información que ofrece cada experto.
- Un conjunto distribuido mas uniformemente en general da mejores resultados que aquellos que se distribuyen de una forma muy esparcida, esto lo podemos observar más adelante en los resultados de las técnicas que distribuyen mejor los datos como bisecting K-Menas, X-Means y affinity propagation contra los que los tienen muy dispersos como K-Means o K-Medoids.
- Un número de K óptimo es difícil de encontrar, aquí se propusieron algunos basándonos en algunas hipótesis, métodos que la obtienen, sin embargo, no se encontró una que pudiera ser siempre la apropiada.

II. ABSTRACT

Recommender Systems are useful for dealing with the large amount of information found on the web. They consist of software tools and techniques which provide suggestions to the user about a specific product that may be his interest. This project focuses on Recommender Systems for movies; however it can be adapted to any other type of Recommender Systems.

Experiments were carried out to compare the behavior of the recommendations with:

Different distances: euclidean, Manhattan and Pearson correlation in order to see which value of RMSE or MAE was lowest.

Different clustering techniques: affinity propagation, bisecting K-Means, K-Means, K-Medoids and X-Mean.

Different data sets: testing is performed taking user reviews and expert reviews to see which gave the lowest value of RMSE or MAE.

The implementations generally were performed using MATLAB (see A.2.), while WEKA (see A.1.) was used for X-Mean and K-Means as well as our implementation of.

The best result of the experiment was find in the users set with Euclidian distance, bisecting K-Means and K=37.

The most important conclusions can be found included the following:

- Not necessarily an expert opinion is the best because in the case of experts set out higher errors, this may be due to the limited information provided by each expert.

- A more evenly distributed set generally gives better results than those that are distributed in a very sparse, this can be seen later in the results of the techniques that distribute data better as bisecting K-Means, X-Means and affinity propagation against those that have highly dispersed as K-Means and K-Medoids.

- A number of optimal K is hard to find, here was proposed some based on certain assumptions that the obtained methods, however, there was no one that could always be appropriate.

III. AGRADECIMIENTOS

Al **Consejo Nacional de Ciencia y Tecnología** (**CONACyT**) y a la **Universidad Autónoma Metropolitana** (**UAM**) por haber otorgado el financiamiento para la realización de este proyecto de investigación.

A mi asesor el **Dr. John Goddard Close** por la confianza que depositó en mí y todo el apoyo brindado en la realización de este proyecto de investigación.

A mi hija **Azzlit Raziel Gómez Almaraz** quien siempre ha sido y será mi motor para seguir adelante.

A mis padres **Juan Jesús Almaraz Ayala** y **Ofelia Pérez Romero** por todo el apoyo que me han brindado.

A mis hermanos **María Luisa Almaraz Pérez** y **Juan Carlos Almaraz Pérez** por formar parte de mi vida y animarme a seguir adelante.

A **Diego Israel Lara Arvizu** por el apoyo incondicional que ha tenido para conmigo y animarme a realizar la maestría.

A los **maestros** y **compañeros** que me apoyaron, escucharon y compartieron sus conocimientos conmigo.

A todos ellos GRACIAS!

CONTENIDO

1. INTRODUCCIÓN

Con el paso del tiempo, la cantidad de información que podemos encontrar en la web es cada vez mayor, ya que ésta crece rápidamente. La información la podemos encontrar en diferentes tipos, por ejemplo en texto, imágenes, videos o audios y tomar decisiones sobre ésta, es cada vez más complejo. La información en alto volumen y variada provoca que los usuarios entren en conflicto para poder decidir la mejor opción a sus necesidades, dando lugar a malas decisiones. Para evitarlo y ayudar al usuario a tomar una mejor decisión existen los Sistemas de Recomendación (SR).

Los SR son de gran utilidad para tratar la sobrecarga de información de la web ya que consisten en herramientas de software y técnicas que proveen sugerencias al usuario respecto a un producto en específico que le pueda ser de su interés [1] como por ejemplo qué comprar, qué película ver, qué música escuchar, qué noticia o libro leer, etc. *ver* **Figura 1**.

Figura 1. *Algunos ejemplos de productos*

Se le pueden dar al usuario sugerencias no personalizadas como el top ten de películas, este tipo de sugerencias son las más fáciles de generar por lo cual no son dirigidas por un SR. Existen sugerencias personalizadas donde se toman en cuenta datos del usuario y su relación con los productos, en este tipo de sugerencias los SR son de gran utilidad ya que tomará los datos existentes para generar una recomendación que le

sea del interés del usuario. La precisión con la que se dará la recomendación dependerá de la técnica utilizada para el SR, está a su vez, dependerán de los tipos de datos con los que se cuenten para implementar el SR, lo anterior lo podemos ver en la **Figura 2**.

Figura 2. *Funcionamiento general de los SR*

Los SR surgieron como área independiente de investigación a mediados de la década de los 90's aumentando drásticamente el interés sobre estos en los últimos años, algunas de las razones por las cuales se desea explotar esta tecnología son:

- *Incrementar su número de ventas*. Probablemente es la razón más importante de los SR comerciales. Cuando el usuario está realizando una compra y le sugieren algo que le puede ser de utilidad es muy probable que el usuario también adquiera ese producto incrementando así las ventas.
- *Incrementar la satisfacción del usuario*. Es muy importante que el SR dé recomendaciones afines al usuario para que éste se vaya contento con la recomendación y la decisión que ha tomado con su ayuda, ya sea al comprar un producto, escuchar una canción o ver una película.
- *Incrementar la fidelidad del usuario*. Si el SR da una recomendación del interés del usuario y éste queda satisfecho seguramente regresará a utilizar ese sistema para futuras consultas.

- *Comprender mejor lo que el usuario quiere*. Un SR va "aprendiendo" a través de la actividad del usuario por lo cual se le pueden dar mejores recomendaciones ya que conoce un poco más los gustos del usuario.

Dentro de las aplicaciones más comunes en las cuales se utilizan Sistemas de Recomendación se pueden mencionar las siguientes:

- *Entretenimiento*. Aquí se encuentran aquellos productos que brindan entretenimiento al usuario como pueden ser películas, videos, música, etc.
- *Contenido*. Se encuentran noticias personalizadas, recomendaciones para documentos, páginas web, aplicaciones electrónicas de enseñanza, y filtros de correo electrónico.
- *Comercio electrónico (e-commerce).* Aquí encontramos recomendaciones para que los clientes compren libros, cámaras, PC's, televisiones, etc.
- *Servicios*. Aquí se dan recomendaciones para servicios de viaje, casas en renta, etc.

Existen distintas clasificaciones de los SR de acuerdo a las técnicas utilizadas para realizar recomendaciones, las cuales pueden ser *filtrado colaborativo, basado en contenido e hibrido,* existiendo también SR *demográficos, basados en conocimiento y basados en la comunidad.*

Dentro de los Sistemas de Recomendación se presentan retos a cubrir tales como escasez de datos, escalabilidad, sinónimos, oveja gris, vulnerabilidad de ataques, diversidad vs precisión, el valor del tiempo, evaluación de recomendaciones, interfaz de usuario [2] [3]. Estos retos se explicarán más adelante en el Capítulo 2.

1.1. Antecedentes

1.1.1. Clustering

En el proyecto se compararán diferentes técnicas de agrupamiento las cuales utilizan k-NN. La técnica estándar para realizar agrupamientos es K-Means, la cual tiene algunas variantes como X-Means, bisecting K-Means y K-Medoids.

En [4] y [5] se compara el tiempo computacional que tardan en realizar los agrupamientos tanto K-Means como K-Medoids. En ambos artículos se muestra que el mejor tiempo se obtiene con K-Means. En estos trabajos se menciona que K-Means es eficiente para bases de datos pequeñas y K-Medoids para bases de datos grandes.

En [6] se realiza experimentación y comparación con la base de datos iris utilizando K-Means y K-Medoids obteniendo que K-Medoids utiliza menos tiempo computacional que K-Means para realizar los agrupamientos. Esto se puede observar en la **Tabla 1.**

Algoritmo	Tiempo Computacional (segundos)
K-Means	13.9790
K-Medoids	13.9330

Tabla 1. *Compración de tiempos entre K-Means y K-Medoids* **[6]**

En [7] eligen utilizar bisecting K-Means para manejar grandes cantidades de datos que necesitan ser analizados como proceso de imágenes y agrupamiento de documentos ya que las técnicas de agrupamiento tradicionales como K-Means no logran procesar los datos de una forma rápida. Aquí se utiliza la técnica bisecting K-Means en un entorno off-line. Se realiza experimentación con dos bases de datos de movielens [8], las cuales son la base de datos de 100K (en [7] llamada MLP) y la base de datos de 1 M (en [7] llamada MLM). Se calcula el error MAE y se presentan los resultados en la **Tabla 2.**

N		20	40	60	80	100
			MLP			
CF	MAE	0.790	0.773	0.772	0.773	0.774
	T	149	150	150	150	151
BKM	MAE	0.831	0.793	0.779	0.758	0.758
	T	6	11	14	18	25
			MLM			
CF	MAE	0.766	0.754	0.749	0.747	0.747
	T	1516	1520	1523	1526	1531
BKM	MAE	0.819	0.781	0.762	0.749	0.742
	T	68	108	147	233	311

Tabla 2. *Resultados MAE con CF y BKM*

Donde MLP corresponde a la bd de 100 K y MLM a la bd de 1M **[7]**

La tabla muestra los resultados MAE obtenidos para MLP y MLM en el número de vecinos (N) mostrados en la parte de arriba. Se puede observar que utilizando filtrado colaborativo (CF) los mejores resultados MAE para MLP y MLM fueron 0.772 y 0.747 con 60 y 80 vecinos respectivamente. Para bisecting K-Means (BKM) fueron 0.758 y 0.742 con 80 y 100 vecinos respectivamente. En estos resultados la diferencia es muy poca, sin embargo, en el tiempo T si existe gran diferencia mostrándose mejores tiempos con BKM [7].

Una técnica que ha mostrado buenos resultados es affinity propagation la cual está basada en paso de mensaje entre los datos. En [9] se utiliza un subconjunto de 900 imágenes en escala a grises de Olivetti (una base de datos de caras) para realizar la comparación entre affinity propagation y K-centers. Se encuentra que affinity propagation da menor error que el mejor resultado de los 100 obtenidos con K-centers, esto se puede ver en la **Figura 3.**

Figura 3. *Comparación de precisión con Affinity Propagation.* **[9]**

En la **Figura 4** se muestra que el resultado obtenido en una corrida con affinity propagation es mejor que alguna de las 10,000 realizadas con K- centers.

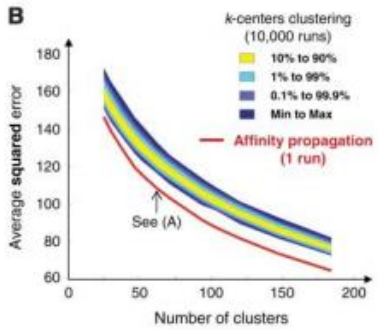


Figura 4. *Grafica comparativa de errores de Affinity Propagation vs K-centroies.* **[9]**

1.1.2. Sistemas de Recomendación

Se han realizado estudios comparativos de los diferentes sistemas de recomendación. Una de las comparaciones que se ha realizado se muestra en la **Tabla 3** [10]. Aquí se indican los datos requeridos para cada uno de los SR así como la desventaja de cada uno de ellos.

Sistema de Recomendación	Tipo de información	Tipo de conocimiento	Método de obtención de datos	Desventajas
Basado en Conocimiento	Demográficos, personal, atributos de usuarios	Reglas de decisión.	Aprendizaje maquinal	Perfil del usuario subjetivo y estático.
Basado en Contenido	Contenidos de páginas web	Descripción de productos en el perfil del usuario (Conjunto de atributos que identifiquen un producto), relación de producto-producto.	Modelado de documentos, filtrado de información, extracción de información.	Depende de la disponibilidad del contenido, pérdida de significados semánticos.
Colaborativo	Perfiles de usuarios (Lista de intereses de otros usuarios dentro de la comunidad)	Matriz de similitud (compartir características de las preferencias de otros usuarios en la comunidad).	Vecinos más cercanos (k-NN), basado en similitudes.	Problema de escases de datos, problema con la calificación de nuevos productos y nuevos usuarios, se invade la privacidad del usuario.

Demográfico	Datos demográficos de los usuarios como edad, género, fecha de nacimiento, educación, etc.	Pertenencia a alguna categoría.	Métodos de clasificación, intereses dentro de los grupos.	Depende de la disponibilidad de los datos demográficos, menos preciso ya que la calidad de los datos demográficos suele ser pobre.

Tabla 3. *Comparación de datos entre tipos de Sistemas de Recomendación [10]*

Los sistemas de recomendación colaborativos son los más utilizados en diferentes aplicaciones como recomendaciones de páginas, películas, música, etc. Este proyecto presenta el trabajo realizado enfocándonos a un sistema de recomendación colaborativo. Como se observa en la **Tabla 3**, estos sistemas de recomendación utilizan vecinos más cercanos (k-NN) y está basado en similitudes.

Se han realizado comparaciones de resultados dentro de sistemas de recomendación cuando en el filtrado colaborativo se utilizan opiniones de usuarios y cuando se utilizan opiniones de expertos. En [11] podemos encontrar la comparación de resultados de error de los experimentos realizados (ver **Figura 5**).

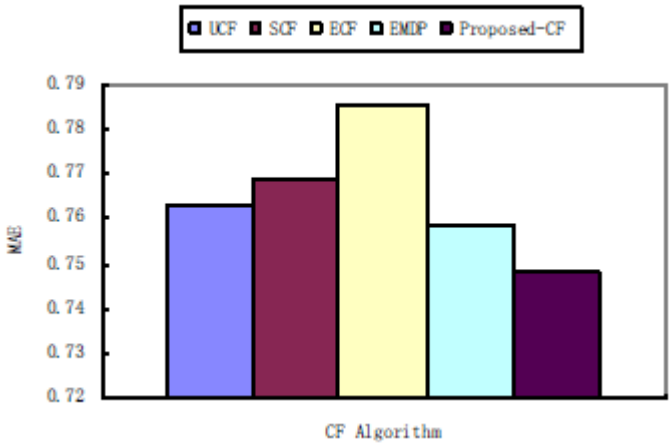


Figura 5. *Comportamiento de Diferentes Algoritmos de Filtrado Colaborativo.* **[11]**

En esta gráfica podemos observar los resultados MAE dados al utilizar el filtrado colaborativo con usuarios (UCF) y al utilizar expertos (ECF). Podemos ver que el mejor resultado se da al utilizar opiniones de usuarios.

1.2. Hipótesis

Para el desarrollo del presente proyecto se tomaron los siguientes puntos a considerar:

- Disminuir el tiempo de respuesta de Sistemas de Recomendación.
- Minimizar el error MAE y RMSE del Sistema de Recomendación.

Para tratar los puntos antes mencionados se realizaron las siguientes hipótesis.

Hipótesis 1. *El tiempo en realizar una recomendación en un conjunto de datos es menor si se realiza sobre un conjunto de datos agrupado previamente que sobre el conjunto de datos completo.* Para esta hipótesis se realizaron experimentos separando nuestro conjunto de datos en distintos números de agrupamientos y se compararon los tiempos que se lleva realizar la recomendación en cada uno de ellos.

Hipótesis 2. *El tiempo en realizar la recomendación dependerá de la técnica utilizada, siendo menor en un conjunto de datos particionado uniformemente, es decir, en afinity propagation y bisecting K-Means el tiempo de respuesta será menor.* Para ello, se realizaron experimentos con cinco diferentes técnicas de agrupamiento, las cuales son affinity propagation, K-Means, K-Medoids, bisecting K-Means y X-Means. Se observa la distribución de los datos en cada una de las técnicas de agrupamiento y se compara la distribución de los datos y el tiempo que se tarda en dar respuesta en cada uno de los casos.

Hipótesis 3. *El tiempo en realizar una recomendación dependerá de la métrica de distancia utilizada, siendo el menor tiempo utilizando correlación de Pearson.* Se experimenta con distancia euclidiana, distancia Manhattan y correlación de Pearson para posteriormente comparar las distancias y observar el comportamiento del tiempo en cada una de ellas.

Hipótesis 4. *Los errores MAE y RMSE serán menores si incluimos opiniones de expertos.* Se realiza experimentación de las recomendaciones dentro de un conjunto de usuarios y un conjunto de expertos para posteriormente comparar los resultados.

Hipótesis 5. *Los errores MAE y RMSE dependerán del algoritmo de clustering utilizado previamente, siendo menores en un conjunto de datos particionado uniformemente, es decir, se obtendrán menor error en Affinity Propagation y Bisecting K-Means.* Se observa la distribución de los datos en cada una de las técnicas de agrupamiento y se compara la distribución de los datos y los valores de error obtenidos.

Hipótesis 6. *Los errores MAE y RMSE dependerán de la métrica distancia utilizada dando menor error en Correlación de Pearson.* Se obtienen los resultados de error para cada una de las distancias utilizadas y posteriormente se realiza la comparación con base a ellas.

1.3. Estructura del Reporte

El reporte está estructurado de la siguiente manera:

Capítulo 1. Introducción acerca de los Sistemas de Recomendación. Se mencionarán las técnicas más utilizadas y los retos que se presentan dentro de los Sistemas de Recomendación.

Capítulo 2. Descripción de las técnicas más utilizadas dentro de los Sistemas de Recomendación y los retos que se presentan en estos. También se describen brevemente algunos ejemplos de Sistemas de Recomendación.

Capítulo 3. Se describen las bases de datos utilizadas como apoyo para crear la base de datos utilizada en este proyecto. Se encuentra la descripción de la base de datos utilizada para este proyecto.

Capítulo 4. Se describen los métodos de agrupamiento utilizados en el proyecto.

Capítulo 5. Se describen las métricas de evaluación utilizadas en el proyecto.

Capítulo 6. Se mencionan algunos retos importantes tratados en el desarrollo del proyecto.

Capítulo 7 y 8. Se muestran los resultados del proyecto, el trabajo futuro y las conclusiones.

2. SISTEMAS DE RECOMENDACIÓN

Los Sistemas de Recomendación se encargan de proveer recomendaciones al usuario, para ello existen diversas técnicas que se pueden utilizar dependiendo el contexto que se tenga. Las técnicas más utilizadas en los Sistemas de Recomendación son *Filtrado Colaborativo, Basado en Contenido, Demográfico, Basado en Conocimiento, Basado en la Comunidad e Híbrido.*

Dentro de los Sistemas de Recomendación también existen varios retos a afrontar ente ellos escasez de datos, escalabilidad, sinónimos, oveja gris, vulnerabilidad de ataques, diversidad vs precisión, el valor del tiempo, evaluación de recomendaciones e interfaz de usuario.

En este capítulo se describirán brevemente las técnicas y retos más comunes dentro de los Sistemas de Recomendación así como algunos ejemplos de éstos.

2.1. Técnicas de los Sistemas de Recomendación

Comúnmente para un Sistema de recomendación se utiliza una técnica híbrida la cual es una combinación de dos o más técnicas *(***Figura 6***)* dando como resultado recomendaciones más precisas, es decir, con menor error.

Figura 6. *SR Híbrido*

2.1.1. Filtrado Colaborativo

La técnica más utilizada es el Filtrado Colaborativo el cual toma en cuenta las calificaciones realizadas por diferentes usuarios hacia los productos calculando la similitud entre ellos para realizar la recomendación.

Para utilizar filtrado colaborativo es necesario contar con datos mínimos para realizar la recomendación. Estos datos son productos, usuarios y ratings de usuarios sobre productos (**Figura 7**).

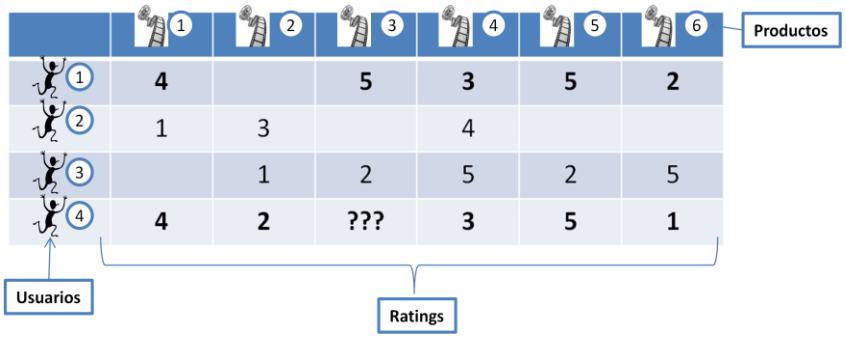
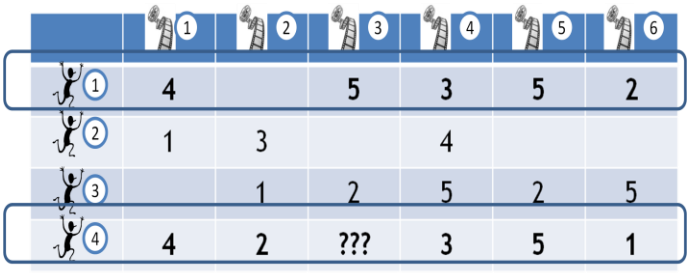


Figura 7. *Estructura de datos mínima para un SR Colaborativo*

El Filtrado Colaborativo puede ser basado en usuario y basado en producto [3].

El ***filtrado colaborativo basado en usuario*** (memoria), es el método más utilizado para realizar recomendaciones. Las recomendaciones las hace tomando en cuenta las calificaciones que el usuario dadas a cada producto, podemos observar esto de forma gráfica en la **Figura 8**. Para ello se deben seguir los siguientes pasos [12] [13] [14] [15]:

1. Analizar el historial de compras/consultas del usuario.
2. Calcular la similitud entre usuarios, es decir, ver qué usuario tiene mayor similitud con el usuario activo (el que solicita la recomendación). Para encontrar la similitud entre usuarios, las medidas más comunes son el Coeficiente de Correlación de Pearson, Similitud de Cosenos, Distancia Manhattan y Distancia Euclidiana.
3. Realizar la recomendación de productos.
4. Realizar la recomendación según el Top-N

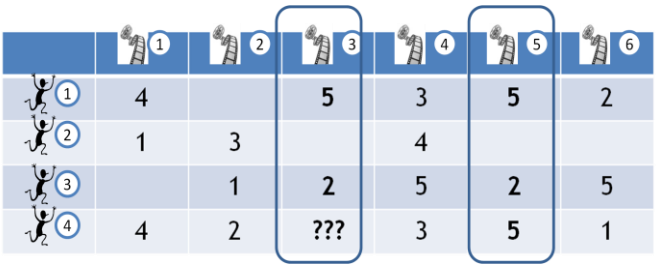


	1	2	3	4	5	6
1	4		5	3	5	2
2	1	3		4		
3		1	2	5	2	5
4	4	2	???	3	5	1

Figura 8. *Filtrado Colaborativo basado en usuario*

En la **Figura 8** se muestra la idea general de las recomendaciones basadas en usuarios. La tabla que se muestra es una tabla de ratings usuario-producto (película) donde se encuentran registradas las calificaciones que cada usuario ha dado a cada producto. Las calificaciones en el producto van de 1 a 5, siento 1 la peor calificación y 5 la mejor calificación. En este caso, el usuario 4 que solicita la recomendación. Se realiza el cálculo de la similitud hacia los usuarios 1, 2 y 3 para encontrar el más afín y poder realizar la recomendación en base al resultado.

El ***filtrado colaborativo basado en producto*** (modelo), se utilizan las calificaciones de los productos para entrenar un modelo y realizar las recomendaciones. En la **Figura 9** podemos observar que en vez de tomar en cuenta las calificaciones por usuarios, se toman en cuenta las calificaciones por producto. Las técnicas más comunes utilizadas en este tipo de filtrado son redes bayesianas, técnicas de clustering, y clasificadores.



	1	2	3	4	5	6
1	4		5	3	5	2
2	1	3		4		
3		1	2	5	2	5
4	4	2	???	3	5	1

Figura 9. *Filtrado Colaborativo basado en producto*

2.1.2. Basado en Contenido

Un SR basado en contenido toma en cuenta el contenido de los productos que el usuario ha seleccionado anteriormente para realizar la recomendación, es decir, busca aquellos productos similares que el usuario ha seleccionado.

Para realizar una recomendación basada en contenido es necesario contar con al menos usuario, historial del usuario, productos y características de los productos.

Figura 10. *Ejemplo de un SR Basado en Contenido*

En la **Figura 10** se muestra un ejemplo de SR basado en contenido. En la parte de lado izquierdo de la flecha se agrupan los artículos que ha comprado el usuario los cuales son unos googles, una gorra y una tabla de natación. En el lado derecho, podemos observar diversos objetos, si tomamos las características de los objetos antes comprados por el usuarios, el SR lo que mostrará será el traje de baño pues tiene características similares.

2.1.3. Demográficos

Los SR Demográficos realizan recomendaciones dependiendo del perfil demográfico del usuario como pueden ser edad, lenguaje, localidad, etc. Para estos SR es necesario contar con los productos, datos demográficos de los productos, usuarios y datos demográficos de los usuarios.

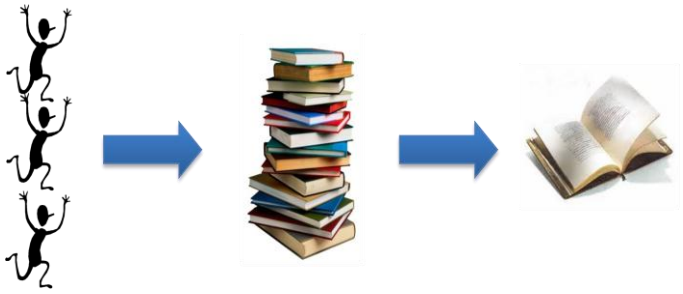


Figura 11. Ejemplo de un SR Demográfico

En la **Figura 11** se muestra un ejemplo sencillo de la forma de trabajar de un SR demográfico. La primera imagen representa al usuario y la segunda el conjunto de libros que se tiene para realizar la recomendación. Ambos (usuario y libros) tienen información demográfica. Por ejemplo si el usuario es un niño que habla inglés, se busca en los libros aquellos que sean infantiles en inglés y de ellos se tomarán algunos para realizará la recomendación.

2.1.4. Basado en Conocimiento

En un SR basado en conocimiento se recolecta la información de qué tanto ha satisfecho un producto al usuario estableciendo así una relación entre lo que necesita/le agrada a el usuario y la recomendación. La información mínima con la que se debe contar para este tipo de sistemas son: producto, características del producto, usuario y necesidades del usuario.

Figura 12. Ejemplo de un SR Basado en Conocimiento

En la **Figura 112** podemos ver un sencillo ejemplo del funcionamiento de estos tipos de sistemas. Si el usuario anteriormente ha comprado productos de 3 marcas diferentes y las ha calificado, se puede obtener el producto con mayores calificaciones positivas recomendando así un producto de dicha marca.

2.1.5. Basado en Comunidad

Si lo que deseamos es realizar una recomendación en base a los amigos del usuario lo ideal es utilizar una técnica basada en comunidad la cual obtendrá datos de los amigos del usuario para realizar la recomendación de tal forma que se aplique el dicho “Dime con quién andas y te diré quién eres”. Para estos SR es necesario contar como mínimo con usuarios, amigos de cada uno de los usuarios y productos calificados por los amigos del usuario.

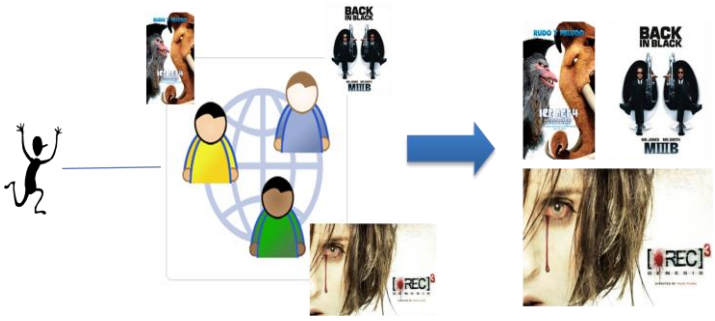


Figura 13. *Ejemplo de un SR Basado en Comunidad*

En la **Figura 13** podemos observar un ejemplo de este tipo de recomendación. Se tiene un usuario con 3 amigos y a cada uno de ellos le gusta una película diferente, con esto la recomendación que se le da al usuario con base en a las películas que le gustan a sus amigos.

2.2. Retos dentro de los Sistemas de Recomendación

Dentro de los SR hay varios retos que se pueden enfocar a temas de investigación/experimentación. A continuación se explicarán brevemente alguno de ellos.

2.2.1. Escasez de Datos

Cuando llega un usuario o producto nuevo al SR, estos no cuentan con información previa para poder obtener y realizar la recomendación, presentándose así el problema de *escasez de datos*. En este caso, la tarea de encontrar sus similares se vuelve más complicada ya que un nuevo producto no puede ser recomendado hasta que un usuario lo haya calificado y a nuevos usuarios no se les darán buenas recomendaciones por la falta de calificaciones en su historial de compras. Esto puede reducir la efectividad de los SR y por lo tanto generar malas predicciones.

2.2.2. Escalabilidad

La *escalabilidad* dentro de un SR se refiere a la forma en la cual crece la información dentro de éste. Cuando la información tanto de usuarios como de productos crece rápidamente decimos que se presenta la escalabilidad.

2.2.3. Sinónimos

En ocasiones encontramos *sinónimos* dentro de los identificadores de un producto y por tal motivo, algunos pueden no ser tomados en cuenta para la recomendación. Por ejemplo se pueden tener una película que dentro de su descripción tenga “Películas para niños” y otra muy similar que tenga “Película infantil” sin embargo si no se tienen considerados los sinónimos no se encontrarán dentro del mismo grupo aunque tenga características similares.

2.2.4. Oveja Gris

En muchas ocasiones los usuarios no ayudan a la realización de las recomendaciones ya que no están de acuerdo o en desacuerdo con algún grupo de personas, es decir, el perfil del usuario pertenece a diferentes grupos de usuarios y en muchas ocasiones grupos opuestos. Cuando esto sucede se dice que el usuario es una *oveja gris.* Este tipo

de usuarios no ayuda a dar buenas recomendaciones y es difícil determinar para ellos una recomendación adecuada. Este problema se presenta sobre todo en aquellos sistemas que usan filtrado colaborativo.

2.2.5. Vulnerabilidad de ataques

La *vulnerabilidad de ataques* se encuentra más dentro del comercio electrónico. Se presenta cuando se trata de promover o inhibir injustamente algunos productos. Esto se puede dar en diferentes casos, por ejemplo, cuando nadie puede proveer recomendaciones, cuando se dan demasiadas recomendaciones positivas para su propio material, y recomendaciones negativas para sus competidores.

2.2.6. Diversidad vs Precisión

Cuando la tarea es recomendar productos que sean apreciados para un usuario en particular, es más sencillo recomendar productos populares o con mayor calificación, sin embargo, esta recomendación no siempre es útil para el usuario ya que las opciones más populares son más fáciles de encontrar, incluso difíciles de evitar sin necesidad de utilizar un SR. Una lista de buenas recomendaciones debe contener productos que no sean fáciles de localizar para los usuarios y que le sean de utilidad tratando así el reto de *diversidad vs precisión.*

2.2.7. El valor del tiempo

Es importante que al realizar una recomendación, esta se dé en el menor tiempo posible, encontrando así el reto *el valor del tiempo*. Entre mayor sea la cantidad de datos que se tangan, mayor es la dificultad de tratar este reto.

2.2.8. Evaluación de las recomendaciones

La *evaluación de las recomendaciones* es un reto y a pesar de que se tienen diversas métricas de evaluación, el elegir la mejor según la situación y tarea dada es una pregunta

abierta. Comparar diferentes algoritmos de recomendación es problemático porque cada uno puede resolver diferentes tareas.

2.2.9. Interfaz de Usuario

La *interfaz de usuario* es muy importante para facilitar la aceptación de los usuarios y las recomendaciones hechas deben verse de forma clara y presentarse de una manera sencilla y fácil de navegar sobre ellas sin importar la cantidad de recomendaciones que se le darán al usuario.

2.3. Ejemplos de Sistemas de Recomendación

En la actualidad existen gran variedad de SR en su mayoría son sistemas híbridos basados en Filtrado Colaborativo. A continuación se mencionan y describen brevemente algunos ejemplos de estos que existen o han existido.

2.3.1. Tapestry

Tapestry fue un sistema experimental de correo diseñado para soportar filtrado basado en contenido y filtrado colaborativo [16] llamado también solamente “filtrado colaborativo” (término dado por Golberg), este surgió en 1992 y fue desarrollado por Xerox Palo Alto Research Center (Xerox PARC). Tapestry era más que un sistema de correo electrónico ya que permitía a los usuarios calificar los mensajes como buenos o malos, o bien, realizar anotaciones de texto asociados con esos mensajes, estas anotaciones podían ser compartidas entre usuarios y así era posible encontrar documentos basados en estos comentarios [15]. Al ser un experimento pionero, surgieron muchos problemas ya que solo funcionaba correctamente con pequeños grupos de personas y eran necesarias consultas de palabras específicas para obtener resultados lo que dificultaba en gran medida el propósito último del filtrado colaborativo. También tenía otras carencias como la falta de privacidad [17].

Tapestry al ser el primer sistema de recomendación y a pesar de todas las deficiencias que tuvo, fue importante para el crecimiento de los Sistemas de Recomendación, sobre todo de los colaborativos.

2.3.2. Netflix

En Octubre del 2006 *Netflix* lanzó un concurso para mejorar su sistema de recomendación en un 10% o más donde el premio sería 1 millón de dólares, se creía que era un trabajo de unas cuantas semanas, sin embargo fue hasta el 2009 cuando se dio a conocer al ganador de este premio, siento así AT&T quien tuvo la mayor mejora en el error cuadrático medio (RMSE) sobre el algoritmo interno de Netflix llamado Cinematch.

El reto de Netflix consistía en un sistema de recomendación de películas, para este concurso se proporcionó una base de datos de entrenamiento con 500,000 usuarios y calificaciones sobre 18,000 películas con lo que se tenían más de 100 millones de ratings [18] donde cada rating es dado por cuatro elementos: <user, movie, date of grade, grade>, el usuario y las películas son ID's enteros y los grados van de 1-5 estrellas. Los datos utilizados para el concurso de Netflix son [19]:

Training set (99,072,112 ratings)
Probe set (1,408,395 ratings)
Qualifying set (2,817,131 ratings), el cual consiste de de:
Test set (1,408,789 ratings), usado para determinar a los ganadores
Quiz set (1,408,342 ratings), usado para calcular las puntuaciones

Al principio de la competencia, lo que se utilizó comúnmente fue el filtrado colaborativo basado en producto utilizando vecinos más cercanos.

Los resultados obtenidos al final del concurso se muestran en la **Tabla 4**:

Rank	Team	Best RMSE score	Improvement (%)
1	BellKor's Pragmatic Chaos	0.8556	10.07%
2	Grand Prize Team	0.8571	9.91%
3	Opera Solutions and Vandelay United	0.8573	9.89%
4	Vandelay Industries!	0.8579	9.83%
5	Pragmatic Theory	0.8582	9.80%
6	BellKor in BigChaos	0.8590	9.71%
7	Dance	0.8605	9.55%
8	Opera Solutions	0.8611	9.49%
9	BellKor	0.8612	9.48%
10	BigChaos	0.8613	9.47%

Tabla 4. *Líders del premio Netflix en julio del 2009 [3]*

2.3.3. Amazon.com

En *Amazon.com* se utilizan los algoritmos de recomendación para personalizar la tienda en línea para cada cliente y puede tener un cambio radical para cada uno, es decir, no es lo mismo que se le muestra a un ingeniero que a una mamá primeriza, usa recomendaciones como una herramienta de marketing dirigido en muchas campañas de correo electrónico y en muchos sitios de páginas web incluyendo la demanda de su propia página.

El algoritmo utilizado en Amazon.com es llamado producto-to-producto, el cual trabaja con información implícita que va dando el usuario (cliente), es decir, toma en cuenta las compras que ha realizado para sugerirle nuevos productos sin necesitad de realizar encuestas o que califiquen un producto y produce recomendaciones en tiempo real y genera recomendaciones de alta calidad [20].

2.3.4. MovieLens

MovieLens es un Sistema de Recomendación de películas gratuito que utiliza el filtrado colaborativo para generar recomendación de películas, este servicio lo provee GroupLens Research el cual es parte del departamento de Ciencias de la Computación e

Ingeniería en la Universidad de Minnesota [8]. En este sistema el usuario puede calificar las películas que ha visto indicando que tanto es de su agrado, esta información la utiliza el sistema para generar una recomendación personalizadas de otras películas que pueden ser de interés para el usuario.

2.3.5. Last.fm

Last.fm es un sistema que se encarga de realizar recomendaciones personalizadas tomando en cuenta el tipo de música que el usuario ha escuchado. Este Sistema de Recomendación funciona a través de las lista de música que tienen los usuarios en su PC o ipod. Para ello se ofrece un programa llamado “scrobbler” que es el encargado de llenar automáticamente la lista de los usuarios. Con ello es posible realizar listas personalizadas de los temas que más se escuchan, recomendaciones de músicas y conciertos y encontrar personas afines musicalmente hablando. [21]

2.3.6. Jester

Jester es un Sistema de Recomendación de chistes que utiliza el filtrado colaborativo para realizar la recomendación basada en los ratings dados por el usuario previamente a los diferentes chistes. [22]

2.3.7. Book-crossing

Este es un ejemplo de SR demográfico y de contenido. Cada libro que le guste a un usuario es registrado y etiquetado con una identificación BookCrossing (BCID). Una vez registrado se comparte y con el BCID único se puede restrear y ver en que partes ha sido leído y quien lo ha leído ayudando así a encontrar personas afines. [23]

3. BASES DE DATOS

En este capítulo se mencionarán las bases de datos de apoyo utilizadas para la elaboración del reporte así como la descripción de la base de datos creada que incluye opiniones de expertos.

Las bases de datos de netflix, movielens, eachmovie con las que se contaron fueron obtenidas de Personalized Recommendation Algorithms Toolkit (PREA) [24]. Estas bases de datos originalmente están en un formato .arff con datos escasos como se muestra a continuación.

```
@RELATION movievote
@ATTRIBUTE UserId NUMERIC
@ATTRIBUTE 'Rate for Dinosaur Planet[1]'    NUMERIC
@ATTRIBUTE 'Rate for Isle of Man TT 2004 Review[2]' NUMERIC
…
@DATA
…
{0 9, 443 4, 1443 2003-06-02}
…
{0 14, 334 5, 1334 2005-11-04}
…
{0 16, 897 3, 1897 2005-12-17}
…
```

Los datos que se encuentran en el .arff representan lo siguiente:

@RELATION representa el nombre que se le da a la relación que se describirá más adelante, en este caso ***movievote.***

@ATTRIBUTE representa los atributos e instancias que se encuentran en el archivo y el tipo de dato que es. El primer dato corresponde al usuario (instancia) y a partir del segundo se encontrará el id de la película (atributo) y la calificación que el usuario le ha dado.

@DATA representa los datos que se tienen. Tomando como ejemplo {0 9, 443 4, 1443 2003-06-02}, se puede observar la correspondencia de cada uno en la *Figura 14*:

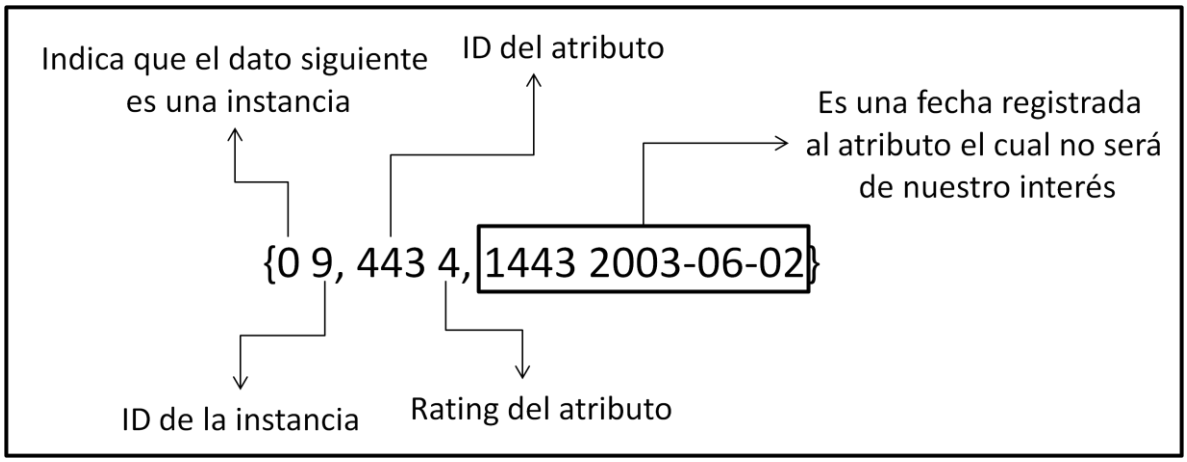


Figura 14. Descripción del @DATA

Las bases de datos para su manejo se transformaron en un archivo .arff sin datos escasos quedando de la siguiente manera:

```
@RELATION movievote
@ATTRIBUTE UserId NUMERIC
@ATTRIBUTE 'Rate for Dinosaur Planet[1]'    NUMERIC
@ATTRIBUTE 'Rate for Isle of Man TT 2004 Review[2]' NUMERIC
…
@DATA
…
9, 4, 0, 3, ...
…
14, 5, 0, 0, …
…
16, 0, 3, 5, …
…
```

Donde en la sección @DATA la primera columna representa la instancia y el resto representa el rating de cada atributo.

3.1. Netflix

En PREA nos proporcionan dos bases de datos de *netflix* donde se encuentras los votos de los usuarios hacia las películas, estas bases de datos tienen diferente número de instancias y atributos mostrados en la **Tabla 5**.

Base de Datos	Instancias	Atributos
netflix_3m1k	4427	1000
netflix_5m3k	8662	3000

Tabla 5. *Bases de datos Netflix*

3.2. Movielens

De igual forma nos proporcionan dos bases de datos de *movieLens* donde se encuentras los votos de los usuarios hacia las películas, estas bases de datos tienen diferente número de instancias y atributos mostrados en la **Tabla 6**.

Base de Datos	Instancias	Atributos
movieLens_1M	6040	3883
movieLens_100K	943	1682

Tabla 6. *Bases de Datos Movie Lens*

3.3. Eachmovie

Una base de datos también obtenida de PREA es *eachmovie* donde se encuentras los votos de los usuarios hacia las películas, esta base de datos 1648 instancias y 74424atributos (**Tabla 7***)*.

Base de Datos	Instancias	Atributos
eachmovie	1648	74424

Tabla 7. *Eachmovie*

3.4. BD utilizada en el proyecto

La base de datos que fue utilizada para la experimentación de este proyecto y a la cual corresponden los resultados mostrados se basó en su mayoría a la base de datos *hetrec2011-movielens-2k* la cual se proporcionó en el 2011 en el segundo *Taller Internacional sobre la Heterogeneidad de la información y la fusión en SR* [25], esta base de datos es una extensión a la base MovieLens el cual contiene ratings personales y etiquetas acerca de películas, esta base de datos tiene ligas hacia Internet Movie Database (IMDb) y RottenTomatoes (RT), cada película tiene identificadores IMDb y RT, títulos en inglés y español, URL de imágenes, géneros, directores, actores (ordenados por popularidad), ratings de usuarios y promedios de ratings de expertos (tomados de RT), ciudades y locaciones de filme.

Esta base de datos se completó con datos obtenidos directamente de la página de RT mediante API's proporcionados para desarrolladores en RT [26] , con estos API´s se incluyeron películas más recientes así como la calificación de expertos por cada película, cabe mencionar que se realizó un proceso de homologación para los ratings tanto de usuarios como de expertos a fin de que estas calificaciones quedaran en una escala de 1-100 en ambos casos. La base de datos obtenida al finalizar el concentrado y elegir los datos requeridos para los experimentos queda conformada como se muestra en la **Tabla 8**.

Base de Datos	Instancias	Atributos
Propia (usuarios)	2113	9375
Propia (expertos)	3911	9375

Tabla 8. *Base de datos Propia*

4. MÉTODOS DE AGRUPAMIENTO

El agrupamiento de datos (clustering), también conocido como análisis de grupos, análisis de segmentación, análisis de taxonomía o clasificación no supervisada [27], es el proceso de examinar una colección de puntos y agruparlos en *clusters* de acuerdo a las distancias entre ellos teniendo como objetivo que todos los puntos dentro de un mismo *cluster* tenga una distancia pequeña de cualquier otro [28].

Los métodos de clustering utilizados en este proyecto son K- Means, X-Mean, Bisecting K-Means, K-Medoids, y Affinity Propagation, los cuales se explicarán brevemente.

En las descripciones que se darán a continuación se utiliza el término similitud a aquella medida obtenida al aplicar alguna de las métricas de similitud o distancia descritas en los apéndices A.3.-A.6.

4.1. K-Means

El algoritmo *K-Means* es de los más utilizados, fue diseñado para agrupar datos numéricos en los que cada grupo tiene un centro llamado media o centroide. Puede agrupar un conjunto de datos en k categorías y el objetivo es minimizar la suma de los cuadrados de las distancias entre los datos y el centroide correspondiente. El algoritmo de K-Means se muestra en la **Tabla 9**.

Algoritmo K-Means
1. Inicializar los K centroides.
2. Formar los clusters iniciales asignando cada elemento al centroide más cercano, es decir, aquel con el que tenga una mayor similitud según la métrica utilizada.
3. Recalcular los nuevos centroides obteniendo la media de los puntos en cada agrupamiento.
4. Se repite el paso 2 y 3 hasta que los centroides no cambien.

Tabla 9. *Algoritmo K-Means*

K-Means no garantiza converger en un óptimo global, sin embargo, llegará a converger al menos en un mínimo local, el número de iteraciones y la eficiencia del resultado dependerá de los datos iniciales.

4.2. X-Means

El algoritmo *X-Means* es una variante de K-Means con una mejor estructura con el fin de resolver las principales deficiencias de K-Means tal como el número de K que tiene que ser proporcionado por el usuario y los resultados son mínimos locales [29]. X-Mean fue propuesto por Pelleg an More (2000) con el fin de encontrar "el verdadero" número de grupos para el conjunto de datos.

A diferencia de K-Means, este algoritmo no necesita como entrada el número de K, a X-Mean se le da un rango mínimo y máximo de K para encontrar el K óptimo para el conjunto de datos dado. El algoritmo lo podemos ver en la **Tabla 10** [27].

Algoritmo X-Means

1. Dar un rango para el valor de K [K_{min}, K_{max}]
2. Comenzar el algoritmo con K=K_{min}
3. Se añaden centroides nuevos al dividir algunos centroides en 2 según el criterio Schwarz el cual está definido como:

   $$BIC(M_j) = l_j(D) - \frac{p_j}{2}\log n$$

   Donde D es el conjunto de datos $D = \{X1, X2, ..., Xn\}$ que contiene n objetos. M_j son los modelos que corresponden a las soluciones con diferentes valores de K dado por $Mj = \{C_1, C_2, ..., Ck\}$. $l_j(D)$ es la probabilidad $P(Mj|D)$ de acuerdo al modelo j-ésimo y toma el punto con mayor probabilidad. p_j es el número de parámetros en M_j.
4. Se repite paso 3 hasta llegar a K = K_{max}
5. El conjunto de centroides con mejor resultado será la salida final

Tabla 10. *Algoritmo X-Mean*

4.3. Bisecting K-Means

Bisecting K-Means es una variante del algoritmo K-Means con la diferencia de ser este un algoritmo que divide de dos en dos el conjunto de datos hasta llegar al número de K deseado. Este algoritmo (**Tabla 11**) recibe el número de K y deja de partir el conjunto de datos cuando se ha llegado a éste.

Algoritmo Bisecting K-Means
1. Elegir el conjunto a dividir 2. Aplicar el algoritmo K-Means con K=2 para dividir el conjunto en 2 subconjuntos. 3. Repetir el paso 2 y tomar la partición que produce el agrupamiento con mayor similitud global, es decir, aquel conjunto que al calcular la suma de las similitudes entre elementos sea mayor. 4. Repetir el paso 1, 2 y 3 hasta que converja.

Tabla 11. *Algoritmo Bisecting K-Means*

Este algoritmo es más eficiente que K-Means y converge en un número finito de pasos.

4.4. K-Medoids

K-Medoids es similar a K-Means con la diferencia de que aquí siempre el "medoid" será un dato existente dentro del conjunto de datos. En la **Tabla 12** podemos ver el algoritmo.

Algoritmo K-Medoids
1. Inicializar los K ejemplares (medoids) 2. Formar los clusters iniciales asignando cada elemento al ejemplar más cercano, es decir, aquel con el que tenga mayor similitud. 3. Recalcular los nuevos ejemplares. 4. Se repite el paso 2 y 3 hasta que los ejemplares no cambien.

Tabla 12. *Algoritmo K-Medoids*

El algoritmo más común para obtener los nuevos ejemplares es el Partitioning Around Medoid (PAM) el cual se muestra en **Tabla 13**.

Algoritmo PAM
1. Intercambiar el medoid *M* y el dato *D* *2.* Calcular el costo total de la configuración. *a.* Realizar paso 1 y 2 para cada *M* y cada *D*, donde *D* no es medoid 3. Seleccionar la configuración con menor costo.

Tabla 13. *Algoritmo PAM*

Este algoritmo tiende a converger en menos iteraciones que K-Means.

4.5. Affinity Propagation

Affinity Propagation es un algoritmo de agrupamiento rápido y flexible elaborado por y Frey and Dueck en el 2007. Este algoritmo agrupa los datos de una forma similar a K-Means solo que toma ejemplares en vez de centroides. Frey y Dueck aplicaron affinity propagation en una gran base de datos de caras de humanos encontrando que este algoritmo da menor riesgo de error y da una solución más rápida.

En este algoritmo no es necesario dar un número específico de agrupamientos ya que encuentra el óptimo automáticamente.

Una limitación importante de affinity propagation es que se requiere gran espacio de memoria ya que requiere 4 matrices de nxn donde n es el número de datos a ser agrupados. [30]

El algoritmo affinity propagation es también conocido como el *algoritmo de paso de mensajes.* Aquí cada producto que será agrupado envía mensajes a todos los demás indicando la relación que tiene con cada uno de ellos. Cada uno de los receptores regresa un mensaje a cada uno de los remitentes indicando qué tan viable es asociarse con cada uno de ellos. Los remitentes toman esa información y ven qué productos son mejores candidatos para el agrupamiento. Este paso de mensajes se realiza hasta que se encuentra el mejor producto asociado a cada uno. La mejor asociación de cada producto es el ejemplar de cada uno de ellos y aquellos que tengan al mismo ejemplar corresponderán al mismo grupo.

Este algoritmo trabaja esencialmente con 3 matrices: matriz de similitud (s), matriz de disponibilidad (a) y matriz de responsabilidad (r) y el resultado se guarda en una matriz de criterio (c).

En la **Tabla 14** se muestra el algoritmo de affinity propagation [31].

Algoritmo Affinity Propagation

1. Realizar hasta que los mensajes no cambien:
   a. Cada dato envía a todos los demás un mensaje que indique que tan probable es que sea su ejemplar (responsabilidad).

$$r_{ij} = s_{ij} - max_{k \neq j}(s_{ik} + a_{ik})$$

   Donde r_{ij} es la responsabilidad que hay entre el punto i y el punto j, s_{ij} y s_{ik} es la similitud entre el punto i y el punto j o k según sea el caso y a_{ik} es la disponibilidad del dato i con el dato k.

   b. Cada dato regresa un mensaje indicando que tanto puede servir este como ejemplar del emisor (disponibilidad).

$$a_{ij} \begin{cases} \sum_{k \neq j} \max(0, r_{kj}) & i = j \\ \min\left[0, r_{jj} + \sum_{k \notin \{i,j\}} \max(0, r_{kj})\right] & i \neq j \end{cases}$$

   Donde a_{ij} es la disponibilidad que hay entre el punto i y el punto j, r_{kj} es la responsabilidad entre el punto k y el punto j.

***Tabla 14**. Algoritmo Affinity Propagation*

Una vez que los mensajes dejan de cambiar se dice que se ha llegado a la convergencia, sin embargo, este algoritmo no la garantiza así como tampoco garantiza siempre una óptima solución.

Si el algoritmo ha llegado a converger, el conjunto $K = \{k | a_{kk} + r_{kk} > 0\}$ se elige como el conjunto de ejemplares. Cada punto i que no es ejemplar es asignado al ejemplar con mayor similitud.

5. MÉTRICAS DE EVALUACIÓN

El objetivo final de cualquier métrica de evaluación en los SR es tener la capacidad de calcular el "buen comportamiento" del SR. Un "buen comportamiento" se presenta cuando al usuario se le guía a un producto de su interés/utilidad. El objetivo general de un SR se compone de dos tareas diferentes: generar sugerencias que sean aceptables para el usuario, y filtrar los productos interesantes/útiles. [32]

Los Sistemas de Recomendación pueden evaluarse tanto on-line como off-line. On-line se utiliza cuando se quieren hacer pruebas en un entorno real. Off-line cuando se hace sobre una simulación de usuarios reales. Este proyecto fue realizado off-line.

Una métrica de evaluación dará un valor a las recomendaciones dadas por el SR. Este valor puede ser en porcentaje, tiempo o alguna medida adimensional.

Para realizar la evaluación en el proyecto, se realizaron predicciones de los ratings que los usuarios puedan dar a las películas comparando estos resultados con los ratings reales obteniendo así una medida de error. Las métricas de error más utilizadas dentro de los Sistemas de Recomendación y comparadas en este proyecto son el error medio absoluto (MAE por sus siglas en inglés) y error cuadrático de la media (RMSE por sus siglas en inglés).

Sea $p(u,i)$ la predicción de un rating, $r(u,i)$ el rating real y n el número total de ratings sobre lo usuarios, describimos las métricas antes mencionadas a continuación.

5.1. Error Medio Absoluto (MAE)

Está dada por **(1)** [32] [12]:

$$MAE = \frac{\sum_{u,i} |r(u,i) - p(u,i)|}{n} \qquad \textbf{(1)}$$

Donde el producto *i* debe ser calificado por un usuario u para obtener *r(u,i)*. Entre más grande sea el número de observaciones disponibles *n*, mejor será la estimación, es decir, el valor MAE será menor.

Esta métrica de precisión es la más utilizada dentro de los SR. Mide la desviación absoluta media entre la predicción de un rating *p(u,i)* y los reales de cada usuario r*(u,i)*.

5.2. Raíz del Error Cuadrático de la Media (RMSE)

Está dada por **(2)** [3] [12]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(p(u,i) - r(u,i))^2} \qquad \textbf{(2)}$$

Esta métrica intensifica los errores por lo cual es comúnmente utilizada si se requiere una mayor precisión entre el rating real y el predicho. RMSE fue la métrica utilizada para decidir al ganador en el premio Netflix para la recomendación de películas obteniendo como ganador a aquel que diera como resultado el valor mínimo en esta métrica.

6. RETOS DENTRO DEL PROYECTO

Para un Sistema de Recomendación hay que tomar en cuenta diversas situaciones que pueden influir, los aspectos considerados en este proyecto son la forma de inicializar los centroides/medoides a la hora de realizar los agrupamientos, el número de agrupamientos adecuado para nuestro conjunto de datos, qué evaluación realizar para nuestras recomendaciones, cómo incluir información de expertos y manejar los datos faltantes dentro de nuestro conjunto de datos. En este capítulo se explica cómo fue tomado cada uno de estos puntos.

6.1. Clustering

6.1.1. ¿Cómo inicializar los centroides/medoides?

Una de las hipótesis planteadas en el proyecto es que "La estabilidad de los agrupamientos depende del tipo de inicialización que se elija"

Inicialización Aleatoria

Este tipo de inicialización es el más sencillo y el más utilizados para inicializar los centroides/medoides, consiste en tomar K instancias aleatoriamente sin que éstas se repitan para tomarlos como los centroides/medoides iniciales y posteriormente aplicar el algoritmo que se muestra en la **Tabla 15**.

Algoritmo Inicialización Aleatoria
1. De 1 hasta K a. Elegir un dato aleatoriamente b. Si ya se eligió anteriormente el dato, repetir *a*

Tabla 15. *Inicialización Aleatoria*

Inicialización Puntos Más Lejanos

Este consiste en tomar una instancia al azar como el primer centroide/medoide, en base a este se busca aquella instancia con la distancia euclidiana más lejana tomando esta como el segundo centroide/medoide, y así sucesivamente hasta completar los K centroides/medoides. Esto lo podemos ver en la Tabla 16.

Algoritmo Inicialización Puntos Más Lejanos
1. Inicializar los K ejemplares (medoids)
2. Formar los clusters iniciales asignando cada elemento al ejemplar más cercano, es decir, aquel con el que tenga mayor similitud.
3. Recalcular los nuevos ejemplares.
4. Se repite el paso 2 y 3 hasta que los ejemplares no cambien.

Tabla 16. *Inicialización Puntos Más Lejanos*

6.1.2. ¿Qué valor de K tomar?

El encontrar el valor de K es una tarea complicada y no se puede establecer una regla para ello, lo más comúnmente utilizado es realizar la experimentación varias veces con diferentes números de K y elegir aquel que nos haya dado mejores resultados.

Heurística

La “regla de oro” para determinar el número de clusters dentro de un conjunto de datos está dada por $k \approx \sqrt{N/2}$ donde n corresponde al número de instancias del conjunto de datos.

Propuesta utilizada

El objetivo de la propuestas es que los agrupamientos tengan más o menos el mismo número de datos, para ello se propone $k \approx \sqrt{N}$ con el fin de minimizar el número de operaciones para realizar una recomendación y minimizar así el tiempo de respuesta.

La meta es dividir nuestros N datos en K grupos tal que cada grupo tenga más o menos el mismo tamaño, es decir aproximadamente $^{N}/_{K}$ elementos. Para realizar una recomendación primero se tiene que ver a qué cluster pertenece, es decir, calcular la similitud hacia los K centroides y obtener el más cercano, una vez realizado esto hay que calcular la similitud a cada uno de los miembros del cluster, es decir, si tomamos en cuenta que nuestros agrupamientos tienen $^{N}/_{K}$ elementos, las similitudes que se calcularán en este paso son $^{N}/_{K}$, es decir, en total se calcularan $K + {}^{N}/_{K}$ similitudes.

Sea f(K) la función para calcular el número de similitudes que deben realizarse en un conjunto de datos dividido en K subconjuntos uniformemente, tenemos:

$$f(K) = K + {}^{N}/_{K}$$

Obteniendo el primer orden de la función tenemos:

$$f'(K) = 1 - {}^{N}/_{K^2} = 0$$

Despejando tenemos:

$$K^2 = N$$

Para obtener la propuesta de obtención de K:

$$K = \sqrt{N}$$

EM

El algoritmo EM (Expectation Maximization) obtiene un número de K basado en probabilidades. Comienza “adivinando” la probabilidad de que una instancia permanezca a una clase para posteriormente re-estimar los parámetros de la probabilidad hasta obtener el número de K.

La implementación de este algoritmo se encuentra en WEKA y lo utilizamos para obtener el número de K [33].

X-Mean

Es una variante de K-Means, la finalidad de X-Mean es distribuir los datos uniformemente entre el número de clusters. Para realizar la partición de los datos se basa en el criterio de información bayesiana (BIC) el cual nos propone un número de K.

La implementación de este algoritmo se encuentra en WEKA y lo utilizamos para obtener el número de K [33].

K-Affinity Propagation

El algoritmo de agrupamiento K-Affinity Propagation (KAP) genera un número de K basado en paso de mensajes. Fue propuesto por primera vez en [34]. El código de este algoritmo en MATLAB lo podemos encontrar en [35].

Silhouette y Davies – Bouldin Index

Son métodos comúnmente utilizados para obtener un número natural de clusters. Fueron utilizados en este proyecto únicamente para obtener el número de K que obtenemos de cada uno. Estos están implementados en MATLAB en un toolbox para estimar el número de K. El código lo podemos encontrar en [36].

6.1.3. ¿Cómo manejar las BD con mayor dimensión?

Se realizó un análisis para ver la posibilidad de separar a los usuarios dependiendo el género que hayan calificado en su historial, sin embargo, tomando en cuenta los géneros de Rotten Tomatoes obtenemos el siguiente porcentaje de usuarios en cada uno (**Tabla 17**).

GENERO	% DE USUARIOS
Action & Adventure	100
Adult	2
Animation	95
Anime & Manga	27
Art House & International	96
Classics	99
Comedy	100
Cult Movies	67
Documentary	89
Drama	100
Faith & Spirituality	32
Gay & Lesbian	30
Horror	95
Kids & Family	98
Musical & Performing Arts	91
Mystery & Suspense	100
Romance	100
Science Fiction & Fantasy	100
Special Interest	90
Sports & Fitness	50
Television	60
Western	73

Tabla 17. *Correspondencia obtenida de la bd género - %usuarios*

Con esto podemos observar que 17 de 22 géneros contienen a más del 50% de los usuarios por lo cual no es una buena opción separarlos mediante este criterio y por lo tanto no se realizó experimentación tomando los datos de esta manera.

6.2. Recomendación

6.2.1. ¿Cómo evaluar la recomendación?

Existen diferentes formas de evaluar una recomendación dependiendo que es lo que queremos obtener de esa evaluación. En el caso del presente proyecto, lo que se requiere es saber que tan bien el sistema realiza las predicciones para lo cual utilizamos métricas de exactitud tales como MAE y RMSE, esta última medida fue la base para anunciar al ganador en el concurso que realizó NETFLIX en el 2006.

6.2.2. ¿Cómo incluir la información de expertos?

Esta información se obtuvo de Rotten Tomatoes, la forma en la que se incluyeron los expertos fue mediante un conjunto de datos externo a los usuarios con el fin de evaluar y comparar por separado la recomendación que se da a un usuario entre un conjunto de usuarios y la que se da entre un conjunto de expertos para ver cual arroja menor valor en MAE y RMSE.

6.2.3. ¿Cómo tratar los datos faltantes?

La base de datos que manejamos tiene datos dispersos, es decir, los usuarios/expertos no califican todas las películas por lo cual se tienen muchos “datos faltantes”. Las calificaciones que se dan a las películas son entre 1 y 100 (1 peor película, 100 mejor película) por lo que sustituimos los datos faltantes con 0 el cual no es tomado en cuenta para realizar la predicción de calificación del usuario necesaria para obtener los errores MAE y RMSE.

7. RESULTADOS

Aquí se muestran los resultados obtenidos en la experimentación realizada a lo largo del proyecto. Dentro de estos resultados podemos encontrar los números de clusters (K) y cómo se distribuyen los datos en cada una de las técnicas utilizadas. Se muestra la comparación de los resultados entre técnicas, distancias y conjunto de datos utilizados dentro de la experimentación.

7.1. Obtención de K

Los números de K que se obtuvieron con los algoritmos e índices descritos anteriormente se muestran en la **Tabla 18**:

Criterio para obtener K	Número de k
Heuristica $k \approx \sqrt{N/2}$	33
Propuesta $k \approx \sqrt{N}$	46
EM	3
X-Mean	4
K-Affinity Propagation	2
Silhouette index	2
Davies-Bouldin index	37

Tabla 18. *Número de K utilizados*

Estos números de K son los utilizados en los diversos experimentos de este proyecto para comparar los resultados obtenidos y ver si se puede tomar algún criterio como el mejor para obtener un valor óptimo de K.

7.2. Distribución de datos

A continuación se muestra una tabla con el número de datos mínimos y máximos que se encuentra en los clusters, lo que nos da una idea si los datos se han distribuido uniformemente.

	Affinity Propagation		K-Means		K-Medoids		Bisecting K-Means		X-Means	
k	**Menor**	**Mayor**	**Menor**	**Mayor**	**Menor**	**Mayor**	**Menor**	**Mayor**	**Menor**	**Mayor**
2	1030	1083	598	1515	827	1086	605	1508	590	1523
3	481	938	199	1189	13	1275	605	786	204	1189
4	431	681	105	923	4	1900	371	736	149	800
33	43	98	1	453	1	857	20	95	1	467
37	40	109	1	462	1	856	23	82	1	485
46	27	82	1	432	1	777	18	73	1	484

Tabla 19. *Distribución de Datos*

En la **Tabla 19** podemos observar a grandes rasgos que affinity propagation y bisecting K-Means tienen mejor distribuidos los datos que cualquiera de las otras técnicas. La distribución de los datos para cada una de las técnicas consideradas se pueden observar gráficamente en el apéndice A.1. A.7.

7.3. Experimentación y Comparación entre Técnicas de Clustering

Para la experimentación en este y los siguientes puntos se tomaron 50 usuarios al azar, estos usuarios fueron los mismos para todos los experimentos a fin de estar comparando la misma información en todos los casos. Una vez obtenidos los resultados para estos 50 usuarios se obtuvieron los promedios que se encuentran en los resultados mostrados más adelante.

Las técnicas con las cuales se hizo la experimentación son: affinity propagation (AP), bisecting K-Means (BK-Means), K-Means, K-Medoids y X-Mean. En la siguiente tabla

podemos observar qué técnica obtiene mejores y peores resultados en cada una de las distancias evaluadas.

Los resultados dentro del conjunto de usuarios se muestran en la **Tabla 20.** Comparación entre Técnicas de Clustering (usuarios):

Usr-Usr	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
Euclidiana	B K-Means 13.28 (K=37)	AP 13.67 (K=46)	B K-Means 17.29 (K=37)	AP 17.69 (K=46)	K-Means 7.6 (K=2)	K-Medodis 14.2 (K=46)
Correlación de Pearson	K-Medoids 13.7 (K=33 y 37)	K-Means 14.2 (K=4)	X-Means 16.47 (K=33)	B K-Means 18.43 (K=37)	K-Medoids 22.8 (K=2)	K-Mea 26.68 (K=3)
Manhattan	K-Medoids 13.53 (K=3)	AP 13.84 (K=33)	K-Medoids 17.56 (K=3)	K-Means 17.85 (K=37)	K-Medoids 6.95 (K=2)	B K-Means 8.75 (K=4)

Tabla 20. *Comparación entre Técnicas de Clustering (usuarios)*

En el apéndice A.1. A.8. se encuentran las gráficas para ver de una forma más clara los resultados.

Los resultados dentro del conjunto de expertos son los que se muestran en la **Tabla 21**:

Usr-Exp	MAE		RMSE		TIEMPO (Seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
Euclidiana	AP 13.46 (K=46)	K-Means 14.60 (K=46)	AP 17.49 (K=4)	K-Means 19.28 (K=46)	AP 9.84 (K=2)	X-Means 7.76 (K=2)
Correlación de Pearson	K-Means 13.64 (K=46)	K-Medoids 29.99 (K=4)	K-Means 17.24 (K=46)	K-Medoids 56.67 (K=4)	K-Medoids 21.31 (K=4)	AP 34.49 (K=46)
Manhattan	AP 13.56 (K=33)	K-Means 14.84 (K=37)	AP 17.61 (K=33)	K-Means 22.36 (K=37)	B K-Means 6.95 (K=33)	AP 12.55 (K=33)

Tabla 21. *Comparación entre Técnicas de Clustering (expertos)*

En el apéndice A.1. A.8. se encuentran las gráficas para ver de una forma más clara los resultados.

7.4. Experimentación y Comparación entre Distancias

Las medidas de similitud que se implementaron en el proyecto son: Distancia Manhattan, Distancia Euclidiana y Correlación de Pearson. La siguiente tabla nos describe con que distancia se obtuvo mejores resultados en cada una de las técnicas utilizadas.

Los resultados dentro del conjunto de usuarios se muestran en **Tabla 22**:

Usr-Usr	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
AP	E 13.43 (K=3)	P 14.08 (K=4)	E 17.42 (K=3)	P 18.19 (K=4)	M 6.96 (K=2)	P 25.50 (K=4)
Bisecting K-Means	E 13.28 (K=37)	P 14.14 (K=3)	E 17.29 (K=37)	P 18.43 (K=37)	M 7.06 (K=2)	P 26.30 (K=37)
K-Means	E 13.41 (K=33)	P 14.20 (K=4)	E 17.42 (K=37)	P 18.23 (K=2)	M 7.15 (K=2)	P 26.68 (K=3)
K-Medoids	E 13.45 (K=4)	P 13.99 (K=2)	P 17.43 (K=33, 37)	P 18.11 (K=2)	M 6.95 (K=2)	P 25.91 (K=4)
X-Mean	P 12.62 (K=33)	P 14.16 (K=4)	P 16.47 (K=33)	P 18.30 (K=4)	M 7.72 (K=4)	P 26.53 (K=4)

Tabla 22. *Comparación entre Distancias (usuarios)*

Donde E corresponde a la Distancia Euclidiana, P a la Correlación de Pearson y M a la Distancia Manhattan.

En el apéndice A.1. A.9. se encuentran las gráficas para ver de una forma más clara los resultados.

Los resultados dentro del conjunto de expertos se muestran en la **Tabla 23**:

Usr-Exp	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
AP	E 13.46 (K=46)	P 14.11 (K=37)	E 17.49 (K=4)	P 18.20 (K=37)	M 9.52 (K=46)	P 34.49 (K=46)
Bisecting K-Means	E 13.75 (K=4)	P 14.16 (K=46)	E 17.79 (K=4)	P 18.25 (K=46)	M 6.95 (K=33)	P 27.37 (K=3)
K-Means	E 13.73 (K=33)	P 21.65 (K=37)	P 17.24 (K=46)	M 22.36 (K=37)	M 7.12 (K=2)	P 30.86 (K=37)
K-Medoids	E 13.81 (K=2, 3, 4)	P 29.99 (K=4)	E 17.83 (K=2, 3, 4)	P 56.67 (K=4)	M 7.29 (K=2)	P 23.82 (K=37)
X-Mean	E 13.73 (K=33)	P 14.71 (K=4)	E 17.78 (K=33)	P 19.59 (K=4)	M 7.79 (K=37)	P 28.19 (K=2)

Tabla 23. *Comparación entre Distancias (expertos)*

Donde E corresponde a la Distancia Euclidiana, P a la Correlación de Pearson y M a la Distancia Manhattan.

En el apéndice A.1. A.10. se encuentran las gráficas para ver de una forma más clara los resultados.

7.5. Experimentación y Comparación entre conjuntos de datos (usuarios y expertos)

Se realizó la recomendación para los 50 usuarios dentro el conjunto de usuarios y dentro de un conjunto de expertos, los resultados para cada una de las técnicas se muestran en las siguientes tablas.

Los resultados para la distancia euclidiana son los mostrados en **Tabla 24**:

DISTANCIA EUCLIDIANA						
	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
AP	Usuarios 13.43 (K=3)	Expertos 13.77 (K=33)	Usuarios 17.42 (K=3)	Expertos 17.83 (K=33)	Usuarios 13.59 (K=2)	Expertos 12.32 (K=46)
Bisecting K-Means	Usuarios 13.28 (K=37)	Expertos 13.79 (K=33)	Usuarios 17.29 (K=37)	Expertos 17.83 (K=33)	Expertos 7.45 (K=4)	Usuarios 9.21 (K=37)
K-Means	Usuarios 13.41 (K=33)	Expertos 14.60 (K=46)	Usuarios 17.42 (K=37)	Expertos 19.28 (K=46)	Expertos 7.37 (K=46)	Expertos 12.96 (K=37)
K-Medoids	Usuarios 13.45 (K=4)	Expertos 13.84 (K=33)	Usuarios 17.48 (K=46)	Expertos 17.88 (K=46)	Usuarios 8 (K=2)	Usuarios 8.75 (K=46)
X-Mean	Usuarios 13.37 (K=33)	Expertos 13.79 (K=46)	Usuarios 17.39 (K=33)	Expertos 17.86 (K=46)	Usuarios 7.70 (K=3)	Expertos 14.75 (K=2)

Tabla 24. *Comparación entre Conjunto de Datos (Distancia Euclidiana)*

En el apéndice A.1. A.11. se encuentran las gráficas para ver de una forma más clara los resultados.

Los resultados para la distancia Manhattan se muestran en la **Tabla 25**:

DISTANCIA MANHATTAN						
	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
AP	Expertos 13.56 (K=33)	Usuarios 13.84 (K=33)	Usuarios 17.61 (K=3)	Usuarios 17.83 (K=37)	Usuarios 6.96 (K=2)	Expertos 12.55 (K=33)
Bisecting K-Means	Usuarios 13.55 (K=46)	Expertos 13.89 (K=4)	Usuarios 17.58 (K=46)	Expertos 17.95 (K=4)	Expertos 6.95 (K=33)	Usuarios 8.75 (K=4)
K-Means	Usuarios 13.63 (K=3)	Expertos 14.84 (K=37)	Usuarios 17.64 (K=3)	Expertos 22.36 (K=37)	Expertos 7.12 (K=2)	Expertos 10.72 (K=46)
K-Medoids	Usuario 13.53 (K=3)	Experto 14 (K=46)	Usuario 17.56 (K=3)	Experto 18.10 (K=46)	Usuarios 6.95 (K=2)	Usuarios 8.68 (K=37)
X-Mean	Usuarios 13.65 (K=3)	Expertos 13.89 (K=4)	Usuarios 17.66 (K=3)	Expertos 17.99 (K=4)	Usuarios 7.72 (K=4)	Expertos 8.2 (K=3)

Tabla 25. *Comparación entre Conjunto de Datos (Distancia Manhattan)*

En el apéndice A.1. A.13. se encuentran las gráficas para ver de una forma más clara los resultados.

Los resultados para la correlación de Pearson los podemos observar en la **Tabla 26**:

CORRELACIÓN DE PEARSON						
	MAE		RMSE		TIEMPO (seg)	
	MENOR	MAYOR	MENOR	MAYOR	MENOR	MAYOR
AP	Usuarios 13.83 (K=33)	Expertos 14.11 (K=37)	Usuarios 17.83 (K=33)	Expertos 18.20 (K=37)	Usuarios 24.26 (K=37)	Expertos 34.49 (K=46)
Bisecting K-Means	Expertos 13.91 (K=37)	Expertos 14.16 (K=46)	Expertos 17.91 (K=3)	Usuarios 18.43 (K=37)	Usuarios 23.63 (K=2)	Expertos 27.37 (K=3)
K-Means	Expertos 13.64 (K=46)	Expertos 17.27 (K=37)	Expertos 17.24 (K=46)	Expertos 21.65 (K=37)	Expertos 23.06 (K=2)	Expertos 30.86 (K=37)
K-Medoids	Usuarios 13.70 (K=33, 37)	Expertos 29.99 (K=4)	Usuarios 17.43 (K=33, 37)	Expertos 56.67 (K=4)	Expertos 21.31 (K=4)	Usuarios 25.91 (K=4)
X-Mean	Usuarios 12.62 (K=33)	Expertos 14.71 (K=4)	Usuarios 16.47 (K=33)	Expertos 19.59 (K=4)	Expertos 24.73 (K=46)	Expertos 28.19 (K=2)

Tabla 26. *Comparación entre Conjunto de Datos (Correlación de Pearson)*

En el apéndice A.1. A.13. se encuentran las gráficas para ver de una forma más clara los resultados.

8. CONCLUSIONES Y TRABAJO FUTURO

En el desarrollo del proyecto se realizaron varios experimentos con el fin de disminuir el tiempo de respuesta de un SR así como minimizar el error MAE y RMSE. Para ello se plantearon algunas hipótesis y aquí se presentará el resultado de cada una de ellas.

Hipótesis 1. *El tiempo en realizar una recomendación en un conjunto de datos es menor si se realiza sobre un conjunto de datos agrupado previamente que sobre el conjunto de datos completo.*

Se realizaron experimentos con diferentes particiones de un conjunto esperando obtener menor tiempo en aquellas particiones con un número mayor de clusters que en aquellas con un número menor de clusters. Los resultados obtenidos en las diferentes técnicas y distancias nos muestran que esto va a depender de la técnica y distancia utilizadas por lo cual no podemos tomar como verdad la hipótesis planteada.

Hipótesis 2. *El tiempo en realizar la recomendación dependerá de la técnica utilizada, siendo menor en un conjunto de datos particionado uniformemente, es decir, en affinity Propagation y Bisecting K-Means el tiempo de respuesta será menor.*

Las técnicas affinitty propagation y bisecting K-Means distribuyen más uniformemente los datos en los agrupamientos esperando con ello que en estas técnicas el tiempo de respuesta sea menor. Se realizó la experimentación con diferentes técnicas y distancias para ver el comportamiento en cada una de las combinaciones posibles. Los resultados obtenidos nos muestran que no necesariamente el tiempo será menor en los conjuntos distribuidos más uniformemente. En muchos casos el mejor resultado se encontró en las técnicas peor distribuidas (K-Menas, K-Medoids) por lo cual no podemos tomar esta hipótesis como cierta.

Hipótesis 3. *El tiempo en realizar una recomendación dependerá de la métrica de distancia utilizada, siendo el menor tiempo utilizando correlación de Pearson.*

En la literatura consultada se presenta a la correlación de Pearson como una buena métrica para obtener la distancias entre datos por lo cual se espera que los tiempos en este proyecto sean los mejores con esta distancia. Los resultados obtenidos nos indican que para nuestros datos la distancia de correlación de Pearson es la que más se tarda en realizar la recomendación en todos los casos. Los mejores resultados en razón del tiempo que se obtuvieron fueron con distancia Manhattan. Estos resultados son comprensibles por los tipos de datos y las operaciones que realiza cada una de las métricas sobre ellos. La correlación de Pearson hace más operaciones sobre un conjunto de datos que la distancia Manhattan.

Hipótesis 4. *Los errores MAE y RMSE serán menores si incluimos opiniones de expertos.*

Al incluir opiniones de expertos se espera menor error MAE y RMSE ya que se espera que sus calificaciones sean más adecuadas y las predicciones realizadas sean más cercanas a la realidad del usuario. Los resultados obtenidos y comparados tanto con usuarios y expertos nos indica que este resultado depende de la distancia y técnica utilizada. Con distancia Manhattan y distancia euclidiana el menor error MAE y RMSE obtenido es en el conjunto de usuarios. Estos resultados se deben a la escaza y limitada información que nos proporcionan las opiniones de expertos.

Hipótesis 5. *Los errores MAE y RMSE dependerán del algoritmo de clustering utilizado previamente, siendo menores en un conjunto de datos particionado uniformemente, es decir, se obtendrán menor error en affinity propagation y bisecting K-Means.*

A pesar de que estas técnicas son las que distribuyen mejor los datos, no en todos los casos se obtuvo el mejor resultado. Sin embargo tomando en cuenta la distancia euclidiana que nos proporcionó menor error MAE y RMSE, el mejor resultado se obtuvo al

utilizar bisectig K-Means. Por lo cual en este caso la hipótesis planteada se cumple para bisecting K-Means.

Hipótesis 6. *Los errores MAE y RMSE dependerán de la métrica distancia utilizada dando menor error utilizando Correlación de Pearson.*

En general, al utilizar correlación de Pearson los errores MAE y RMSE están por arriba de los errores obtenidos con distancia Manhattan y distancia euclidiana.

Para la base de datos utilizada la correlación de Pearson no fue una buena opción para reducir el error MAE y RMSE ya que daba valores muy por arriba de distancia euclidiana y distancia Manhattan. Los errores más pequeños se obtuvieron con distancia euclidiana. En cuanto a tiempo, el mejor tiempo se obtuvo con distancia Manhattan sin embargo la diferencia con distancia euclidiana resultó muy pequeña por lo cual nos quedamos con la distancia euclidiana para compara los grupos de datos y técnicas utilizadas.

El realizar una recomendación dentro del conjunto de expertos no da buenas predicciones. Tomando en cuenta la distancia euclidiana, los menores errores MAE y RMSE se muestran al realizar la recomendación dentro del conjunto de usuarios.

La técnica que dio mejores resultados fue bisecting K-Means con K= 37.

La mejor combinación de distancia, técnica, conjunto de datos y número de agrupamientos que dieron menor tiempo y error MAE y RMSE son: Bisecting K-Menas con K=37, distancia euclidiana y en el conjunto de usuarios.

Este trabajo puede ser ampliado tanto como queramos. Sería interesante ver el comportamiento de los resultados descritos anteriormente sustituyendo los datos faltantes con un promedio o media de los ratings dados y ver si estos darán mejores predicciones.

La métrica de similitud de cosenos en este caso no pudo ser utilizada por la cantidad de datos faltantes que se tenían. Al manejar de diferente manera los datos faltantes se podrían comprar los resultados se obtienen con esta métrica.

En este proyecto se ha realizado la experimentación realizando recomendaciones en un conjunto de datos de usuarios y en un conjunto de datos de expertos. Sería interesante ver el comportamiento de los resultados si combinamos estos conjuntos y sobre este realizamos la experimentación.

Pasa mejorar el tiempo de respuesta, sería conveniente realizar la experimentación en paralelo para observar los resultados y ver si esto minimiza considerablemente el tiempo.

En este proyecto la experimentación se realizó off-line. Sería interesante ver si se comporta de la misma forma al realizar las recomendaciones on-line.

A. APÉNDICES

A.1. WEKA

Es un Software de minería de datos en java que contiene una colección de algoritmos para tareas de minería de datos. Los algoritmos pueden ser aplicados directamente sobre la base de datos o bien desde el código java. Weka contiene herramientas para pre-procesamiento, clasificación, regresión, agrupamientos, reglas de asociación y visualización. Weka permite desarrollar nuevos sistemas de aprendizaje máquinal.

Weka es un software de código libre publicado bajo la licencia GNU [37].

A.2. MATLAB

MATLAB (matrix laboratory) es un lenguaje de alto nivel y un entorno interactivo para calculo numérico, visualización y programación. Utilizando matlab se peuden analizar los datos, desarrollar algoritmos y crear modelos y aplicaciones. El lenguaje, las herramientas y funciones incorporadas en matlab permiten explorar múltiples enfoques y llegar a una solución más rápida que con hojas de cáculo o lenguajes de programación tradicionales como C/C++ o java.

MATLAB se puede utilizar para diversas aplicaciones como procesamiento de señales y comunicaciones, procesamiento de imágenes y videos, sistemas de control , prueba y medición, finanzas computacional y biología computacional [38].

A.3. Coeficiente de Correlación de Pearson

Los coeficientes de correlación nos permiten identificar si un usuario está relacionado con los productos y que tan relacionados se encuentran.

El coeficiente de Correlación de Pearson mide la relación lineal que se presenta entre dos puntos aleatorios.

Sean $\overline{x}, \overline{y}$ las medias aritméticas de X e y, el Coeficiente de Corelación de Pearson se define como:

$$P(x,y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}} \qquad (3)$$

Este coeficiente nos permite también identificar si es una relación directa o inversa.

A.4. Similitud de Cosenos

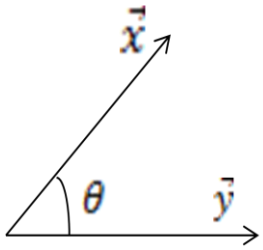
Aquí los elementos se consideran como vectores. El coseno del ángulo entre los vectores representa la similitud entre dos productos.

Sean $\vec{x}, \vec{y}$ los vectores, se definen las normas como:

$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \cdots + x_n^2}$, $\|\vec{y}\| = \sqrt{y_1^2 + y_2^2 + y_3^2 + \cdots + y_n^2}$ respectivamente.

Y la similitud de cosenos está dada por:

$$sim(x,y) = \cos\theta = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|} \qquad (4)$$



El ángulo θ más pequeño representa mayor similitud entre los productos.

A.5. Distancia Manhattan

Calcula la distancia que se debe recorrer para llegar de un punto a otro. La distancia Manhattan entre dos elementos es la suma de las diferencias de sus componentes correspondientes, se define como:

$$M(x, y) = \sum_i |x_i - y_i| \qquad (5)$$

Donde n es el número de variables, x_i, y_i son los valores de la i-ésima variable de los puntos *x* e *y* respectivamente.

A.6. Distancia Euclidiana

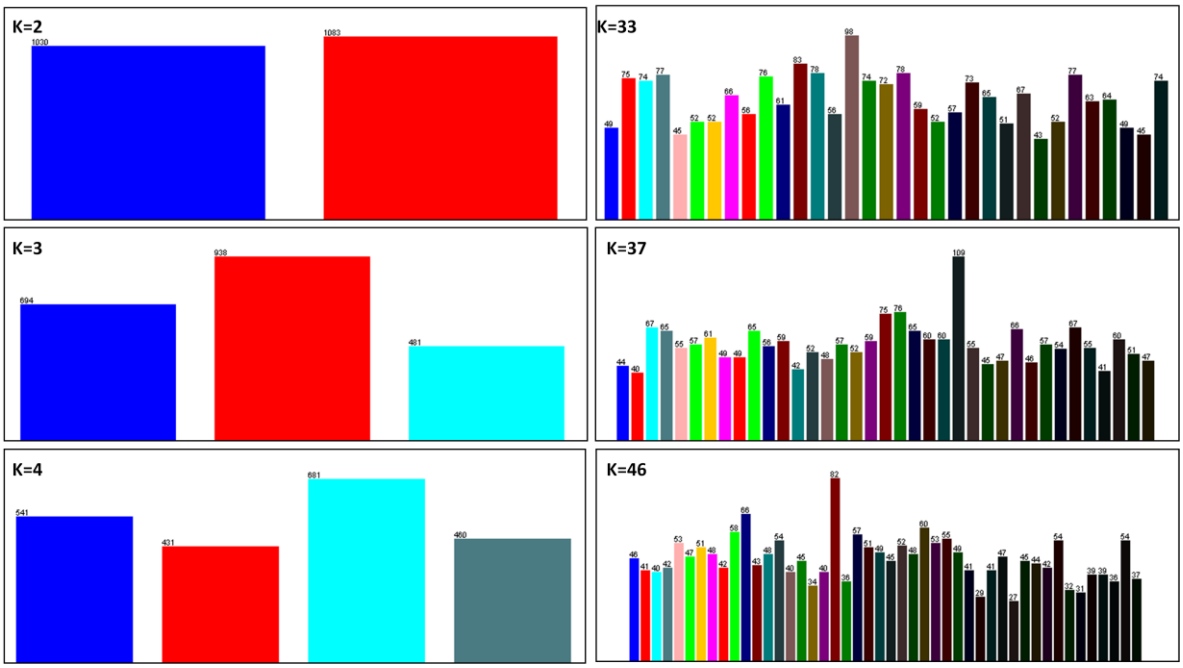
La medida de distancia más familiar es la distancia euclidiana entre el punto X y el punto Y y está dada por:

$$E(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \qquad (6)$$

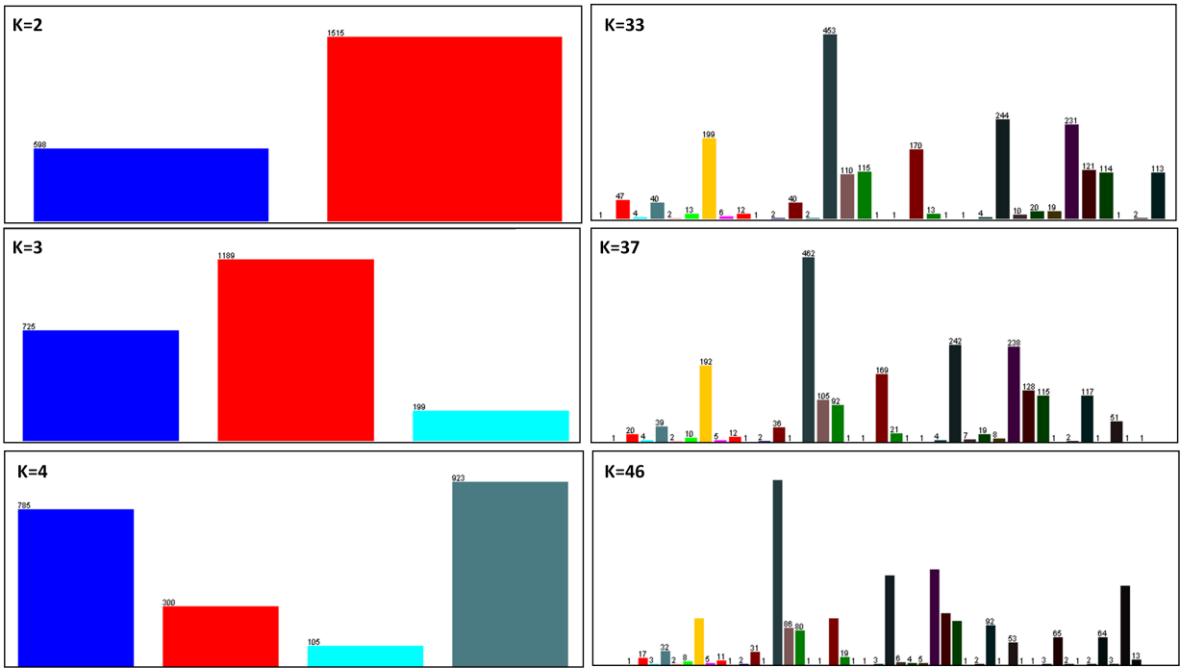
Lo cual implica el cálculo de la raíz cuadrada de la suma de los cuadrados de las diferencias entre los valores correspondientes.

A.7. Distribución de datos en las técnicas de clustering

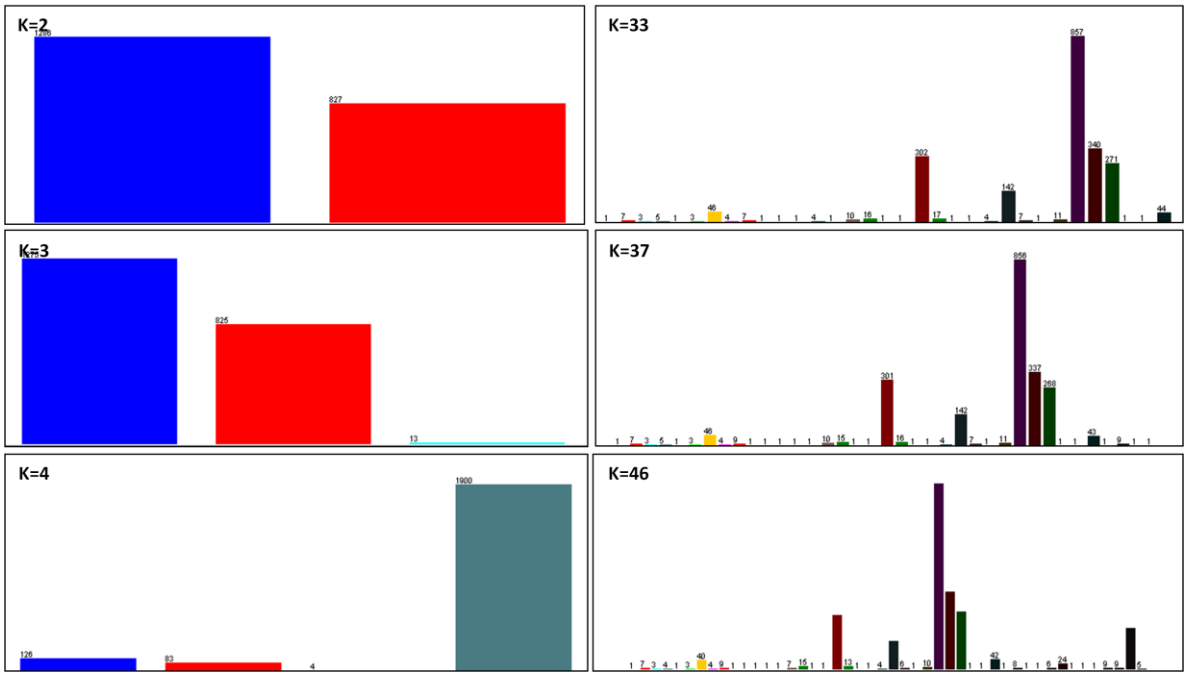
Affinity Propagation



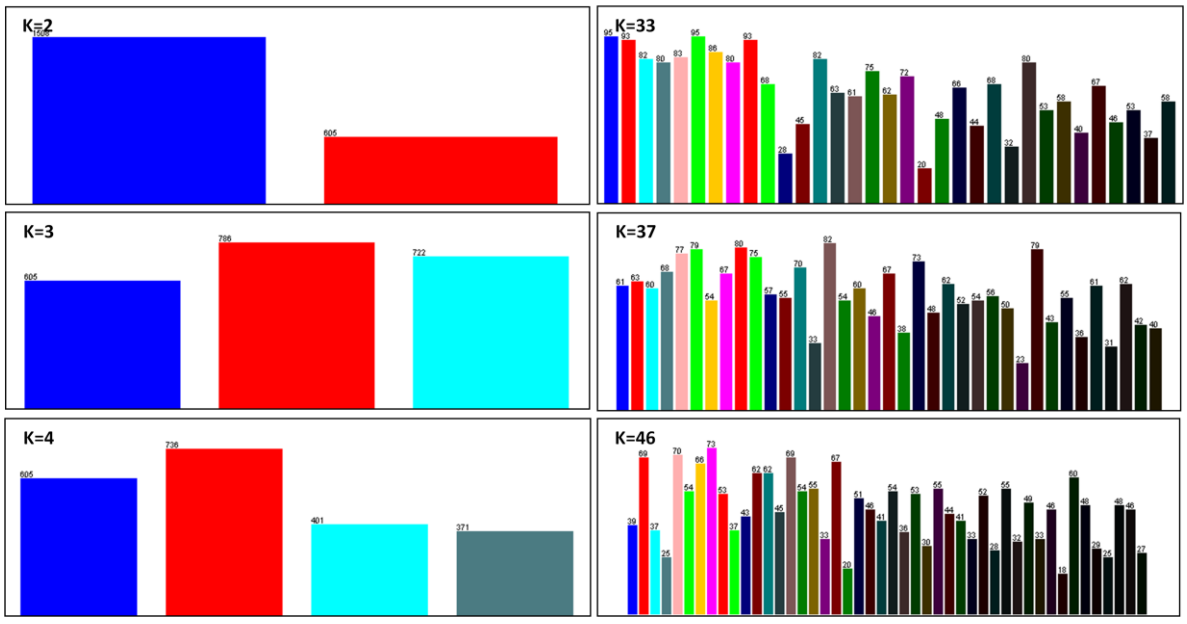
K-Means



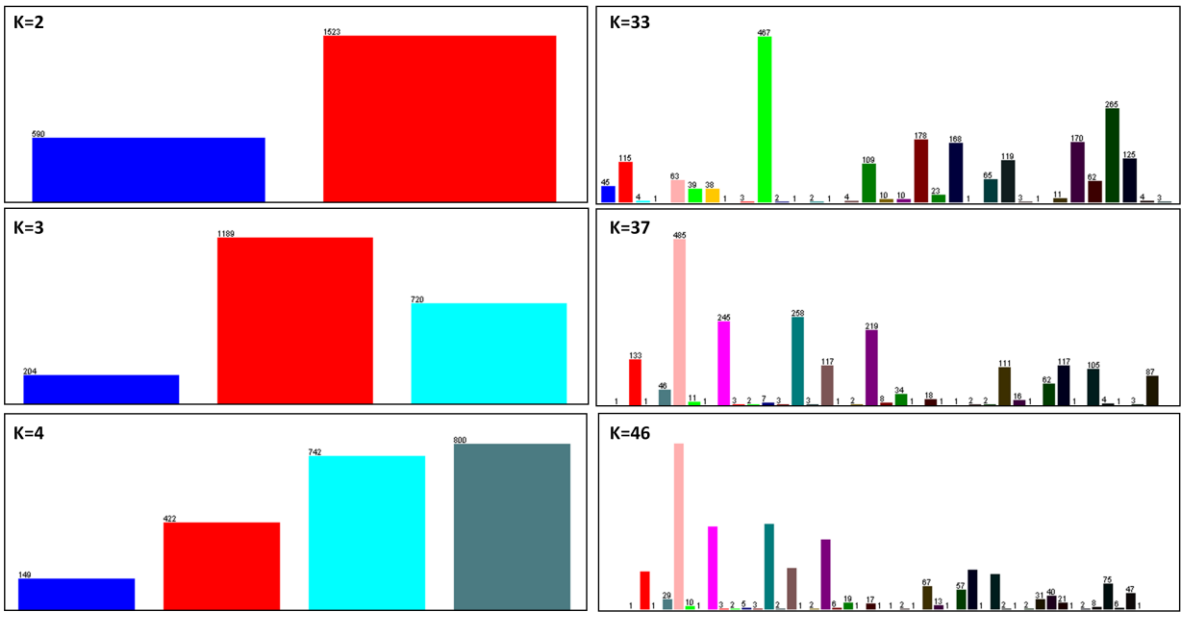
K-Medoids



Bisecting K-Means



X-Mean

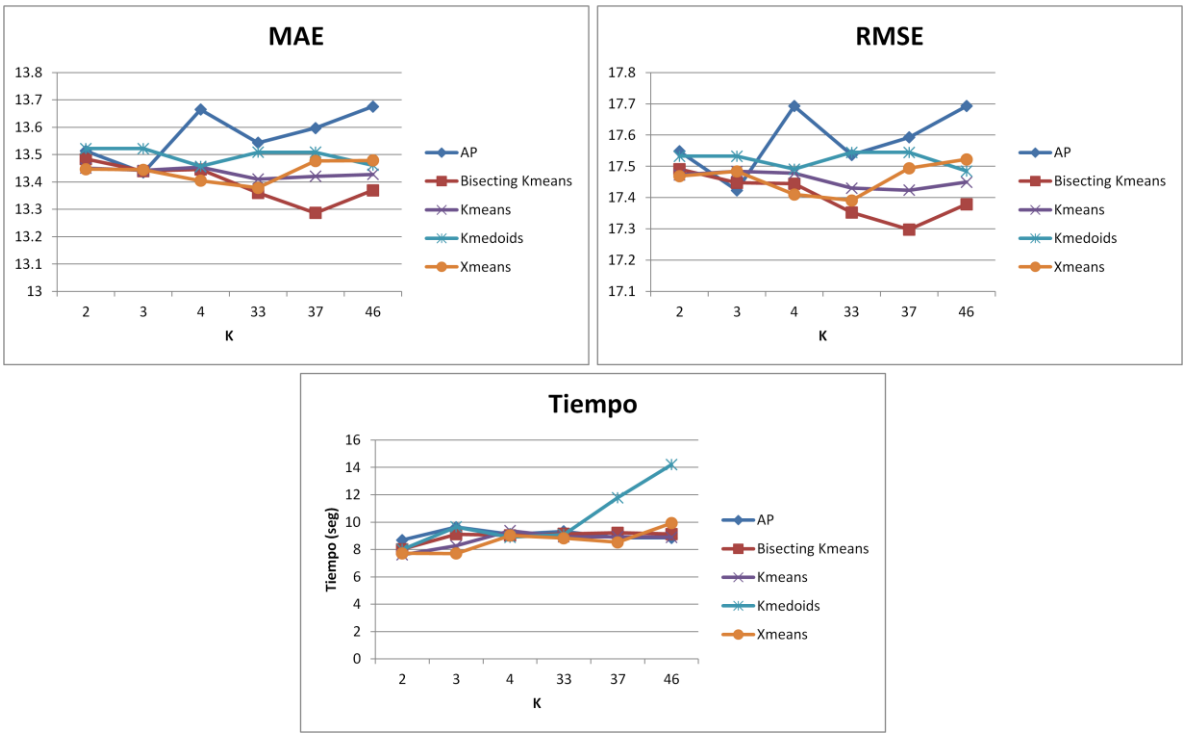


Se puede observar claramente que los datos se distribuyen más uniformemente cuando se utiliza Affinity Propagation o Bisecting K-Means para realizar el clustering. Sin embargo se ve una muy mala distribución en K-Means, K-Medoids y X-Means.

A.8. Resultados: Comparación Técnicas de Agrupamiento (Usuarios)

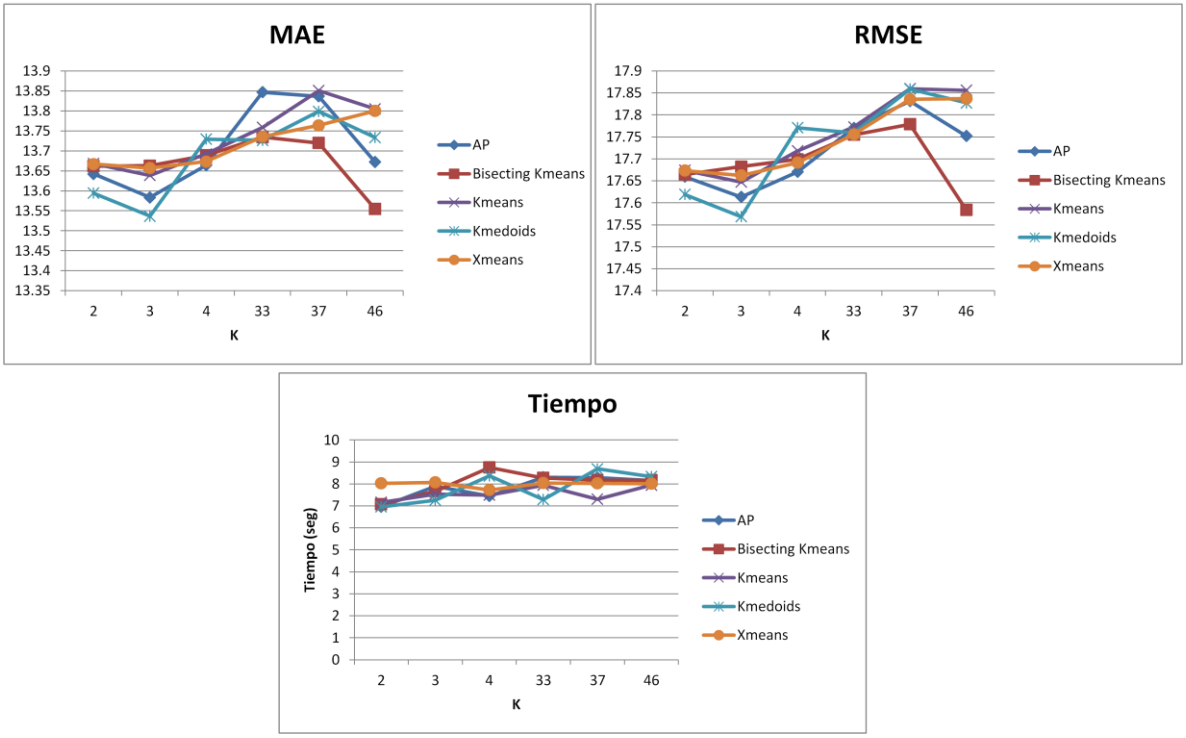
Aquí encontraremos los resultados obtenidos en la comparación de las diferentes técnicas de agrupación utilizadas en el proyecto así como las diferentes distancias. Los resultados presentados en este apéndice corresponden a las recomendaciones realizadas entre usuarios.

DISTANCIA EUCLIDIANA



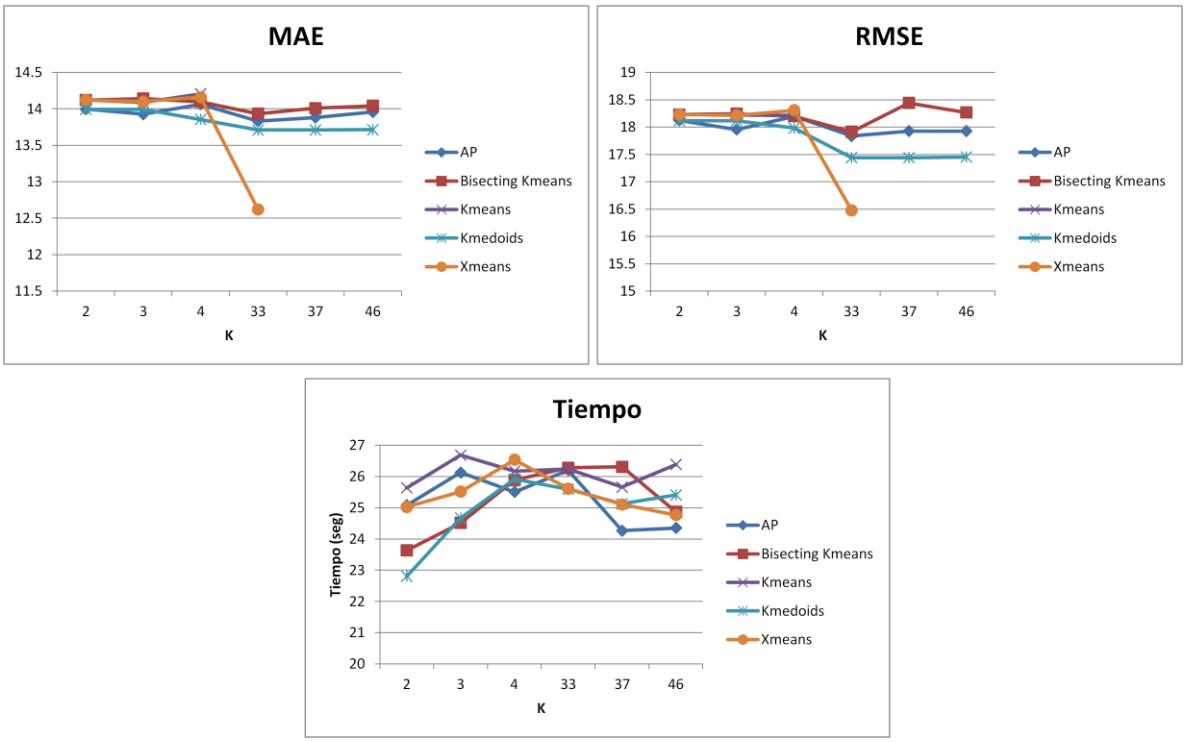
En la distancia euclidiana podemos observar que el mejor resultado cuando K=37 con bisecting K-Means, mientras que el peor resultado es con affinity propagation en K= 4 y K=46. Con respecto al tiempo podeos observar que más o menos todas las técnicas se comportan de la misma manera. Se eleva considerablemente el tiempo con K-Medoids en K=37 y K=46 siendo estos los peores tiempos, los mejores se pueden observar con todas las técnicas en K=2.

DISTANCIA MANHATTAN



Con la distancia Manhattan los resultados obtenidos son muy variados siendo los mejores resultados K-Medoids en K=3 y bisecting K-Means con K=46. Si deseamos comparar las técnicas con respecto al tiempo podemos ver que son casi los mismos tiempos sin embargo en K=2 se presenta un mejor tiempo para affinity propgation, bisecting K-Means, K-Means y K-Medoids. Este tiempo aumenta aproximadamente un segundo en K=46.

CORRELACIÓN DE PEARSON



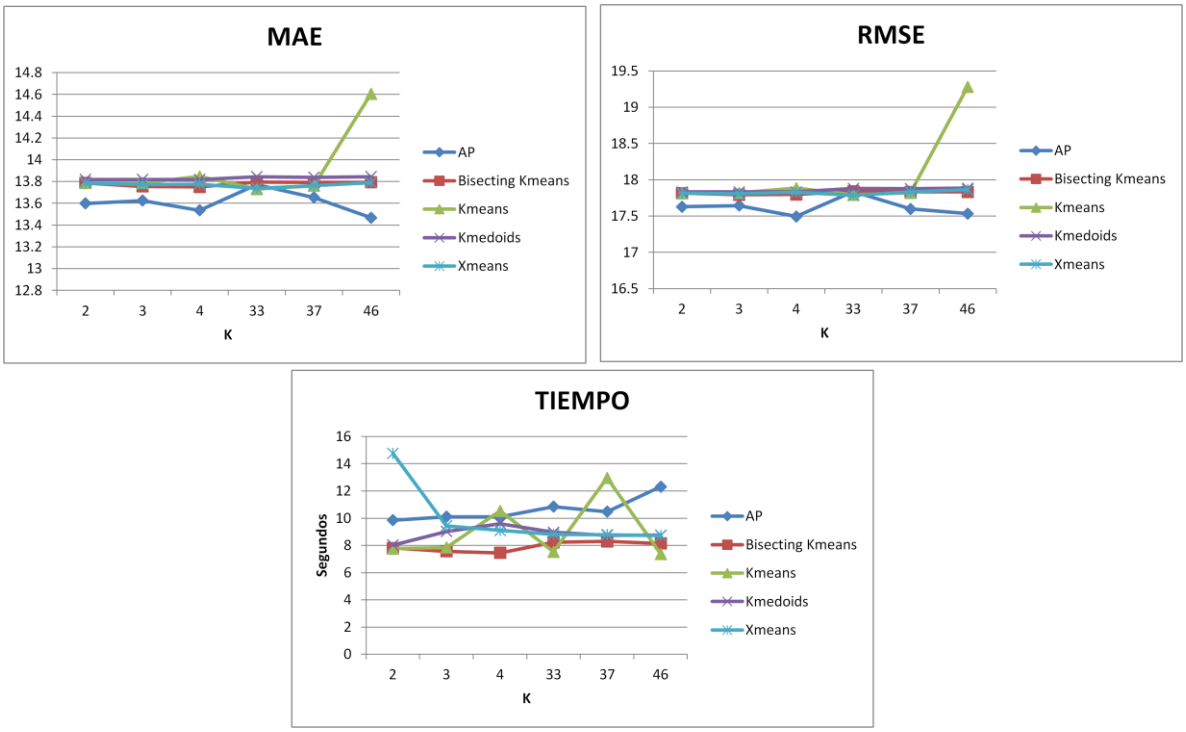
Utilizando coeficiente de correlación de Pearson observamos que el mejor resultado se da con X-Means en K= 33 que hasta ahora sería el mejor resultado para el caso de realizar las comparaciones en un grupo de usuarios. El tiempo menor se presenta con K-Medoids en K=2.

Comparando las 3 distancias en el conjunto de usuarios, podemos observar que el mayor tiempo para cualquier caso se da con la correlación de Pearson pues éste se encuentra arriba de los 20 seg mientras que las otras dos distancias se encuentran alrededor de los 8 seg.

A.9. Resultados: Comparación Técnicas de Agrupamiento (Expertos)

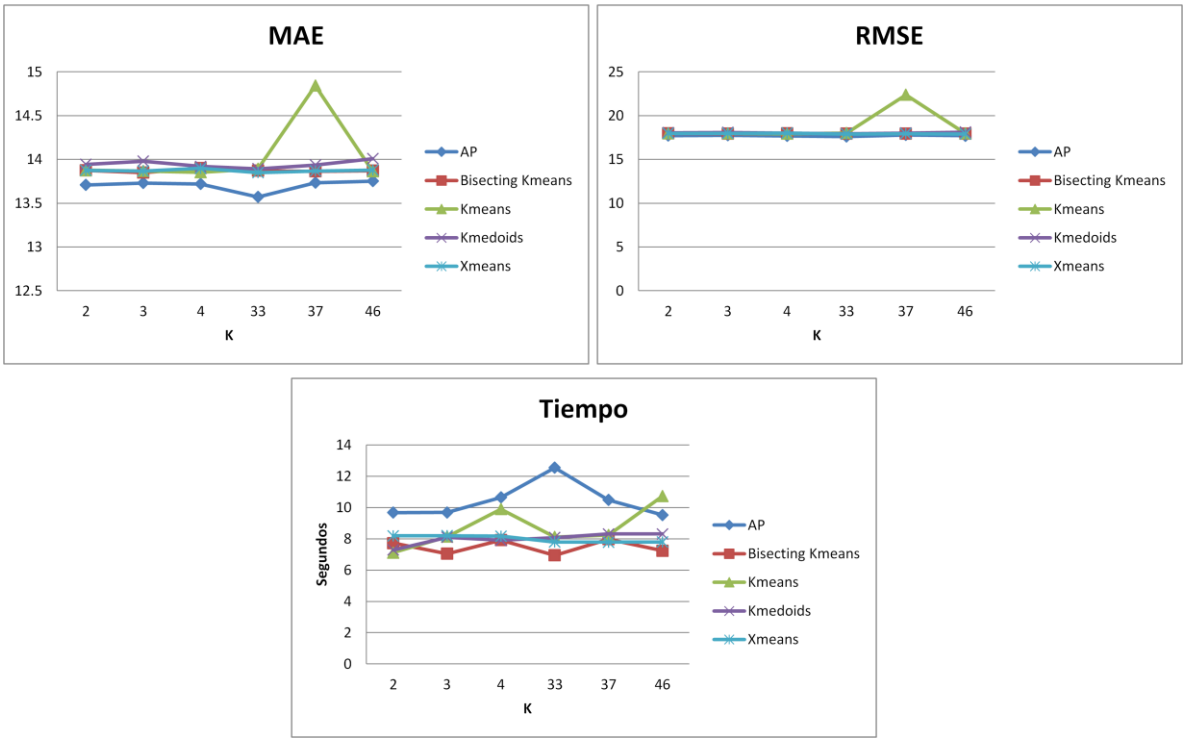
En esta parte se presentan los resultados obtenidos al realizar la recomendación dentro de un grupo de expertos.

DISTANCIA EUCLIDIANA



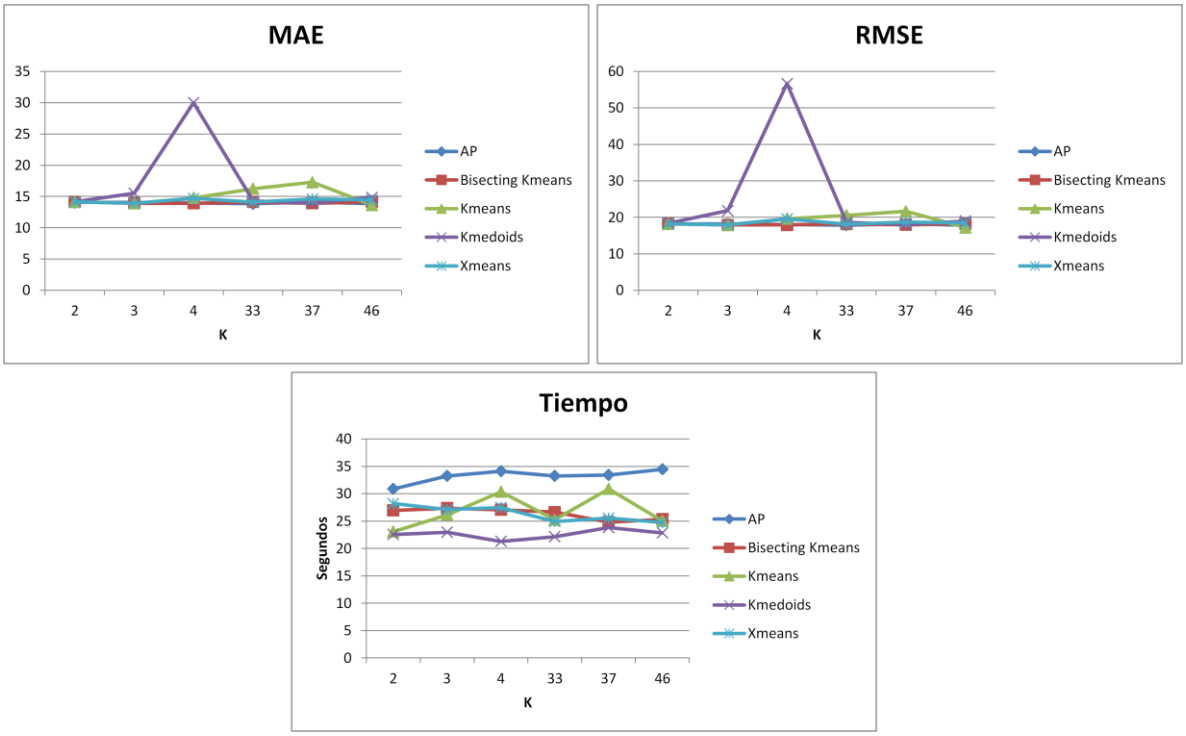
En el caso de utilizar distancia euclidiana para realizar la recomendación dentro de un grupo de expertos podemos observar que el mejor resultado se presenta con Affinity propagation en K=46 y el peor caso se encuentra con el mismo número de agrupamientos (K=46) pero con el algoritmo K-Means. En cuestión del tiempo, podemos observar que el menor tiempo es con Bisecting K-Means en K=4 y el peor tiempo con X-Mean con K=2. El mejor tiempo en la mayoría de los casos es presentado con el algoritmo bisecting K-Means.

DISTANCIA MANHATTAN



Con la distancia Manhattan en el caso de las recomendaciones tomando en cuenta expertos, el mejor resultado es con affinity propagation en K=33 mientras que el peor es con K-Means en K=37. Podemos observar que el RMSE sse mantiene prácticamente igual en todos los casos excepto en K-Means con K=37 que es el que dejaremos como el peor caso. Los peores tiempos en este caso se presentan con Affinity Propagation y los mejores con Bisecting K-Means. El mejor tiempo de todos es con Bisecting K-Means en K=33.

CORRELACION DE PEARSON

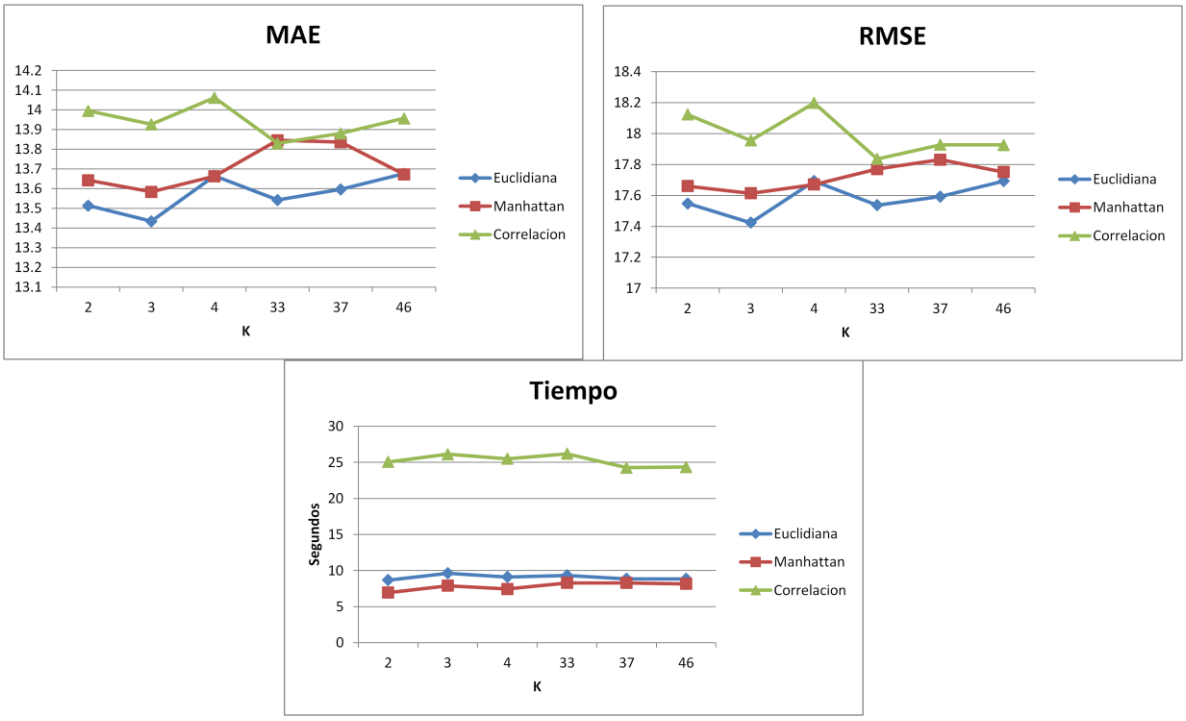


Con la correlación de Pearson se observa que MAE y RMSE se comportan más o menos de la misma manera siendo notablemente K-Medoids con K=4 el peor caso. Los mejores tiempos registrados con Correlación de Pearson son con K-Medoids y los peores con affinity propagation. Sin embargo, sucede lo mismo que en el caso de las recomendaciones realizadas entre usuarios, el tiempo con correlación de Pearson es mucho mayor al obtenido en Distancia Manhattan o Distancia Euclidiana.

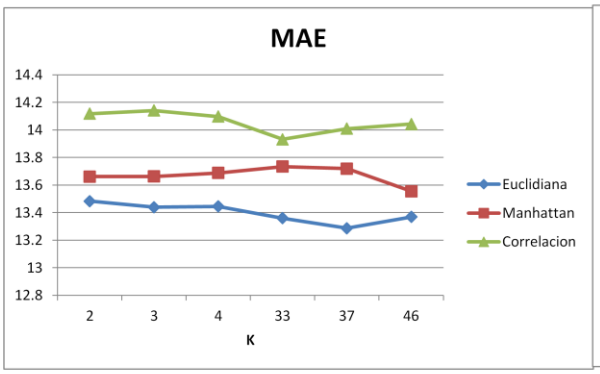
A.10. Resultados: Comparación Distancias (Usuarios)

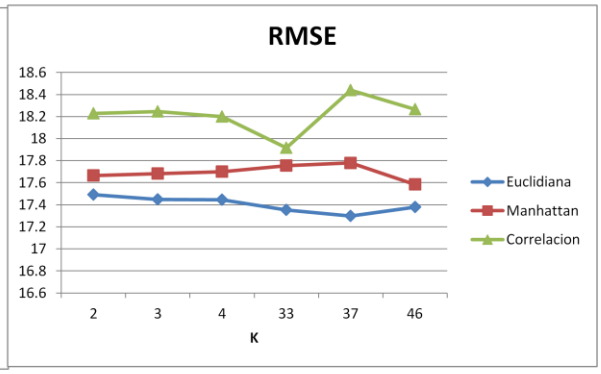
Aquí veremos las comparaciones entre las diferentes distancias utilizadas en el proyecto y ver cuál da mejores resultados en cuanto a errores y tiempos en cada una de las técnicas utilizadas.

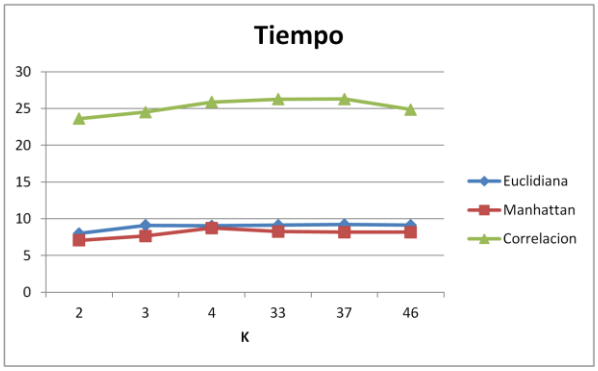
AFFINITY PROPAGATION



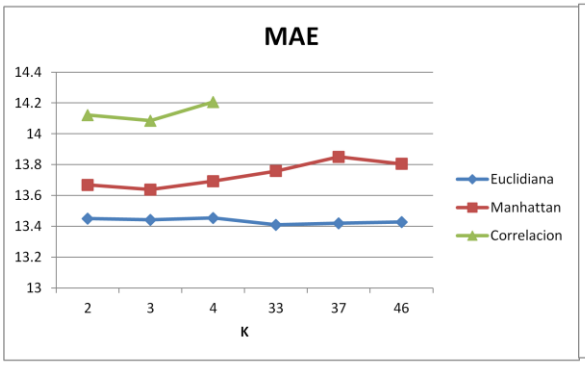
BISECTING KMEANS

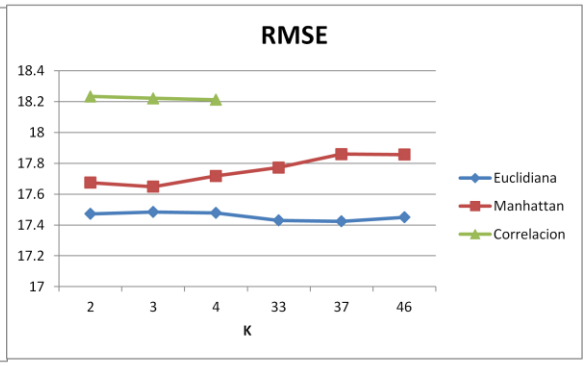


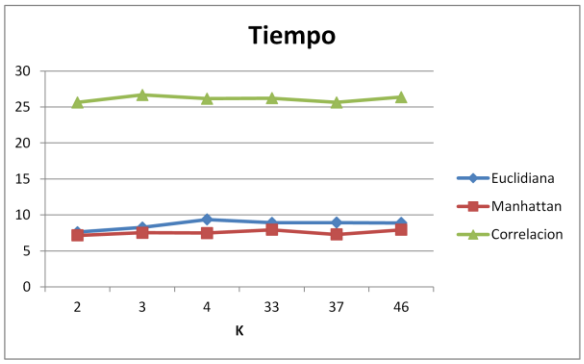




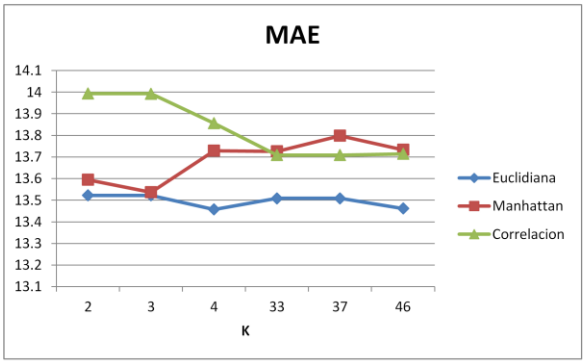
K-MEANS

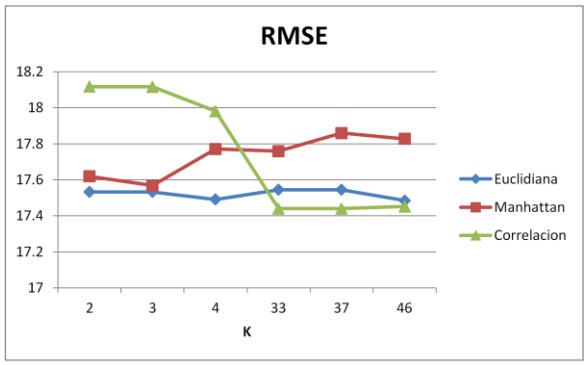


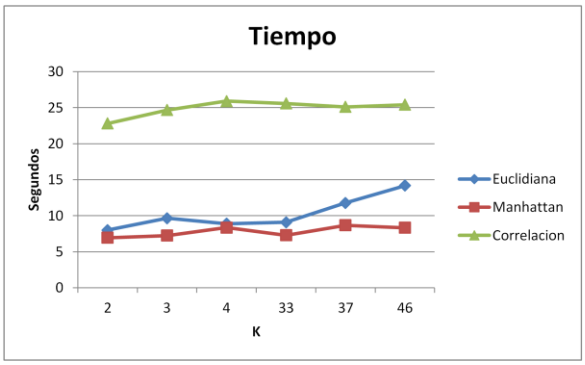




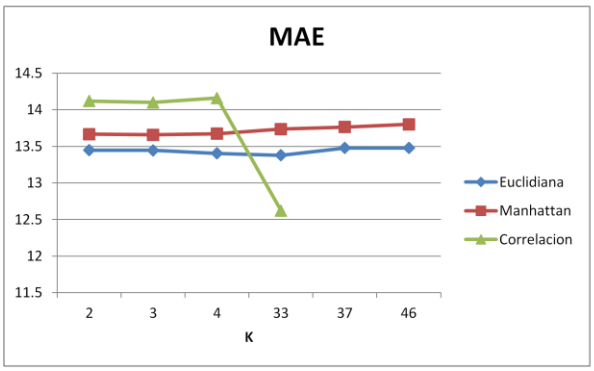
K-MEDOIDS

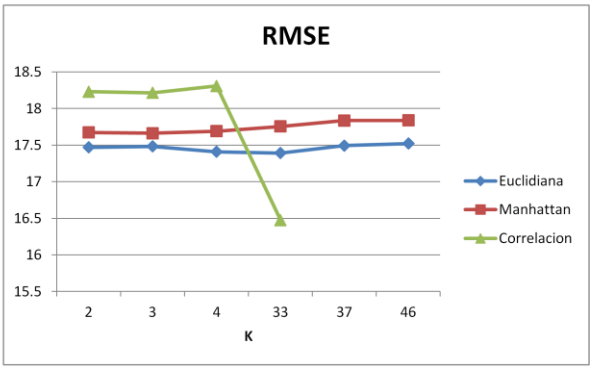


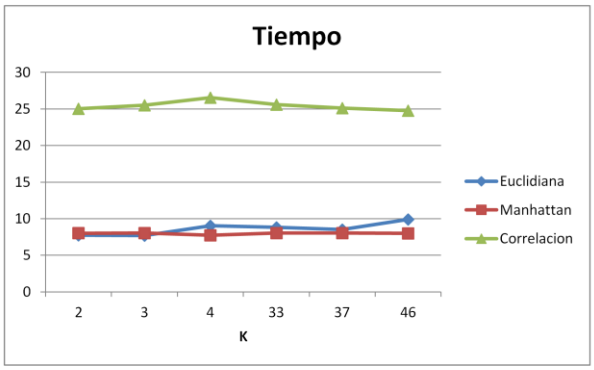




X-MEANS





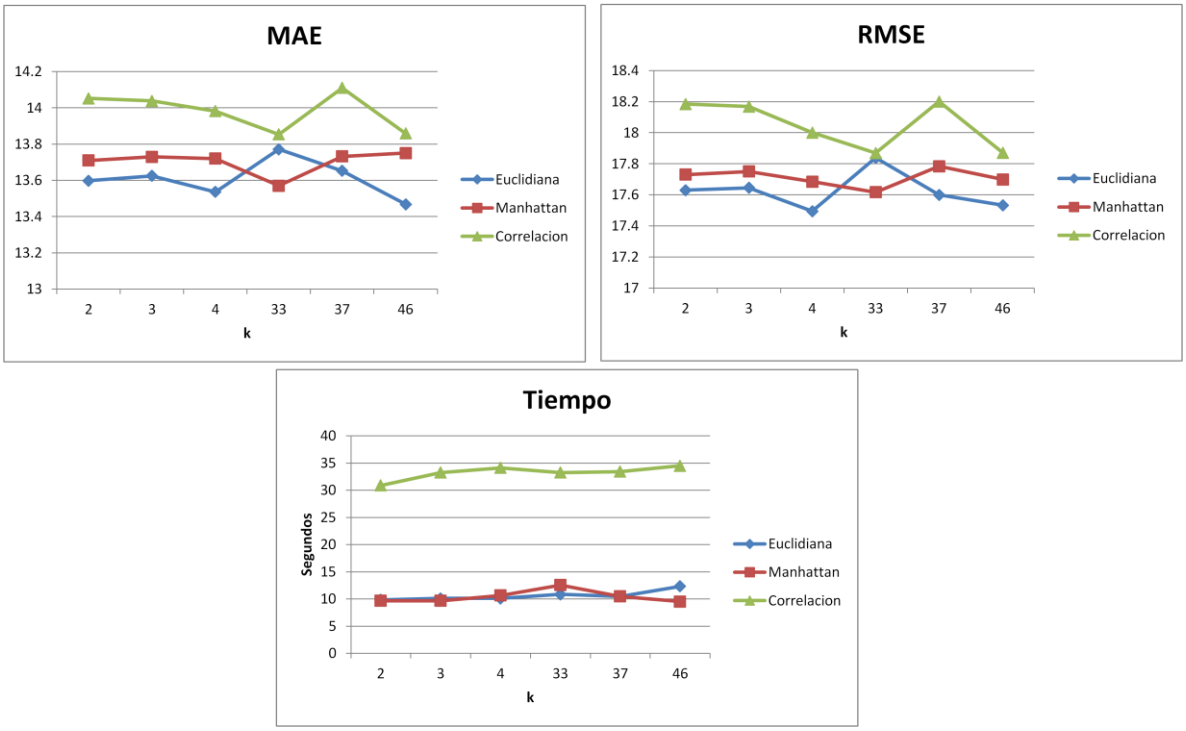


En estos resultados se puede observar que en general los mejores resultados se dan al utilizar la distancia Euclidiana salvo en la correlación de Pearson cuando K=33 donde los errores MAE y RMSE descienden considerablemente. Observando el tiempo en las gráficas anteriores, el mejor tiempo se obtiene cuando se utiliza distancia Manhattan y el peor tiempo en todos los cosas es con Correlación de Pearson.

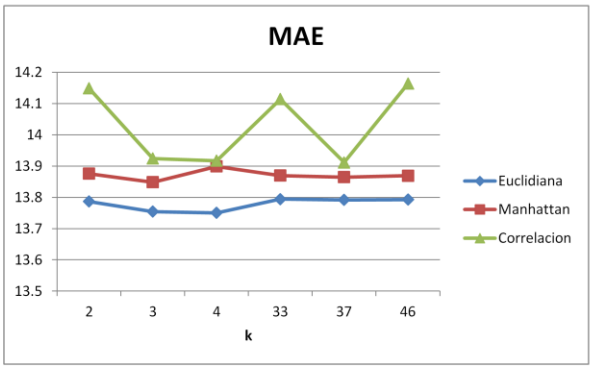
A.11. Resultados: Comparación de Distancias (Expertos)

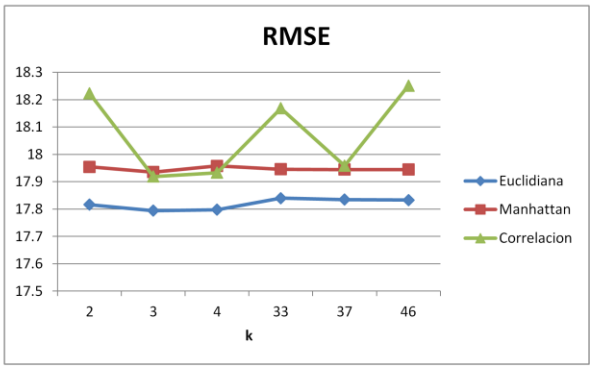
Aquí se presentan los resultados obtenidos al realizar la recomendación en el grupo de expertos. Podremos observar las comparaciones entre las distancias en cada una de las técnicas.

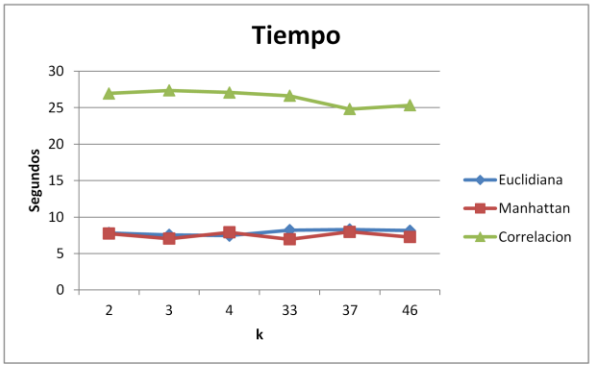
AFFINITY PROPAGATION



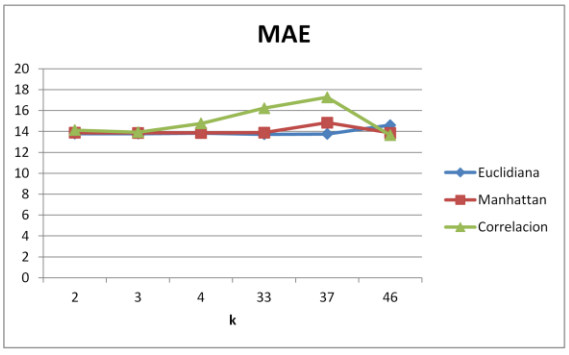
BISECTING KMEANS

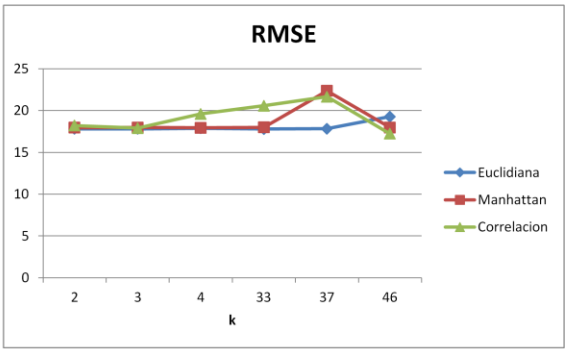


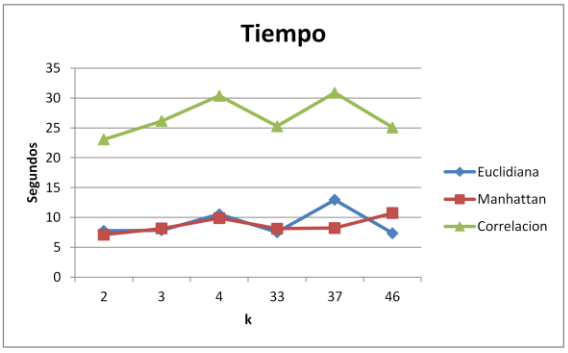




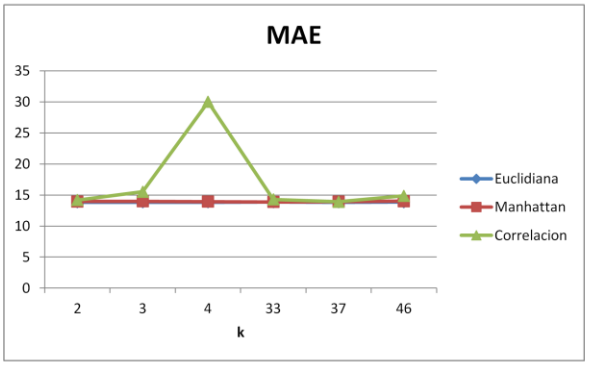
K-MEANS

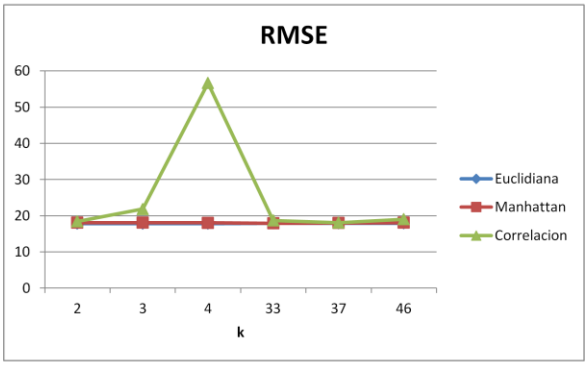


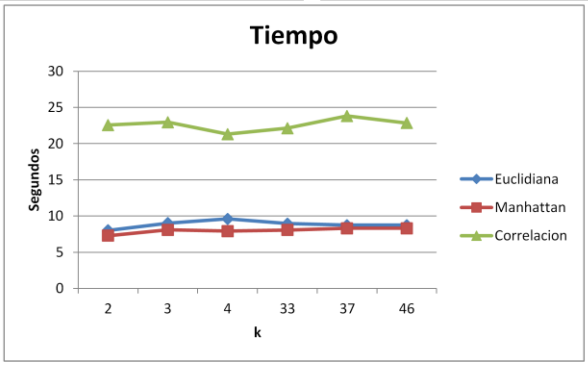




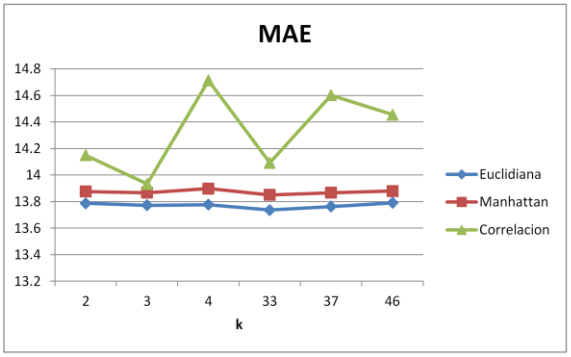
K-MEDOIDS

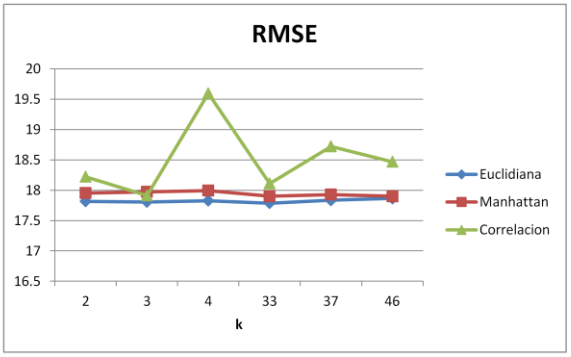


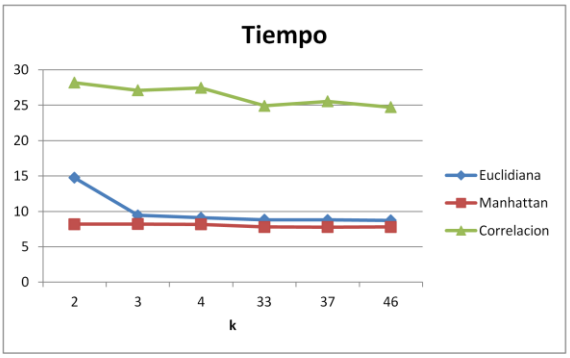




X-MEANS





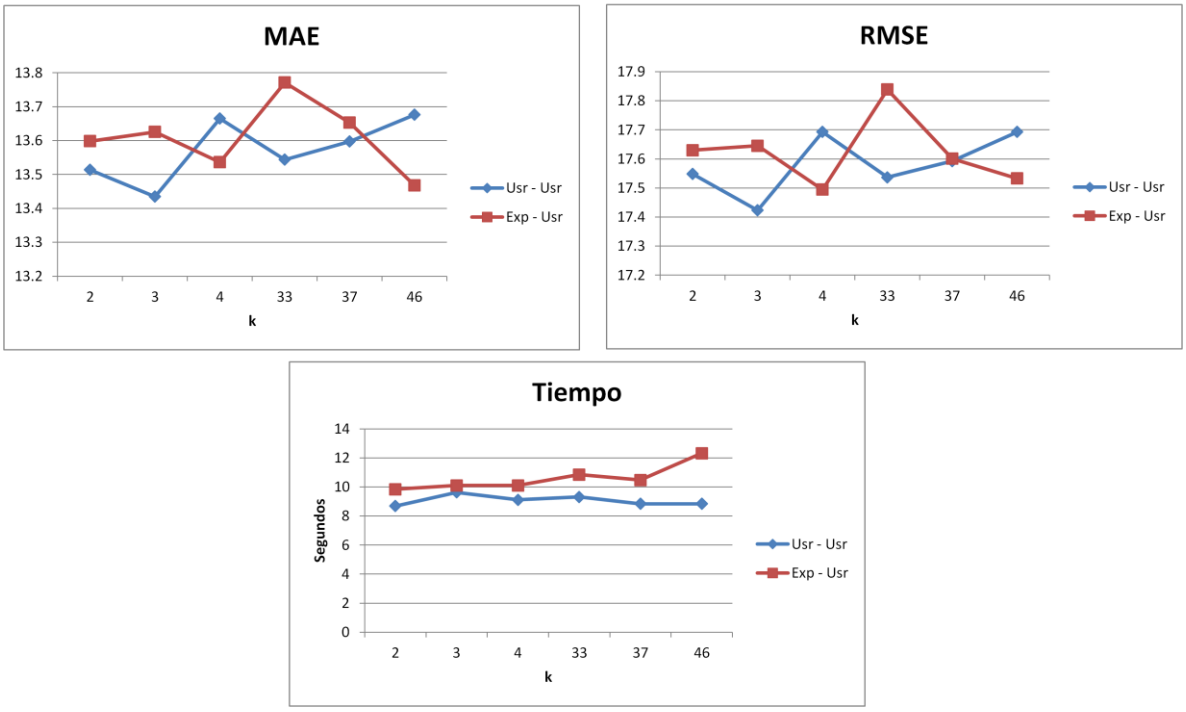


En este caso, en cada una de las técnicas se observa que el mejor tiempo (de igual forma que al realizar comparaciones entre usuarios) se da con la distancia Manhattan y el peor tiempo con la correlación de Pearson. En cuanto a los resultados de error, se observan los peores resultados con la correlación de Pearson y los mejores con la distancia Euclidiana, no quedando muy atrás con la distancia Manhattan en algunos casos como por ejemplo en K-Medoids que prácticamente ambas obtienen los mismos resultados.

A.12. Resultados: Comparación Conjunto de Datos (Distancia Euclidiana)

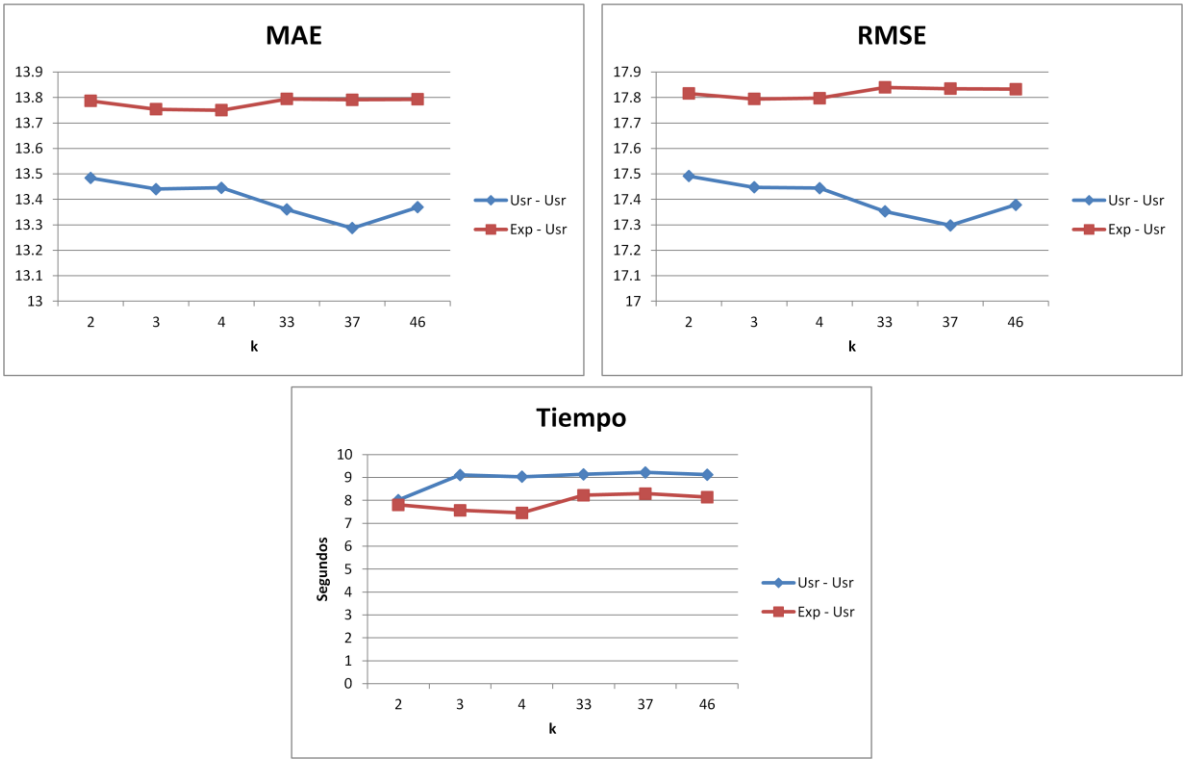
Para los resultados en los cuales compararemos los conjuntos de datos (usuarios y expertos) es importante mencionar que el número de usuarios que se tomó en cuenta fue de 2113 mientras que los expertos son 3911 lo que es un 85% más que los usuarios por lo que es importante considerarlo sobre todo para los resultados mostrados en cuestión de tiempos ya que puede haber poca diferencia en tiempos pero hay gran diferencia en cantidad de datos procesados.

AFFINITY PROPAGATION



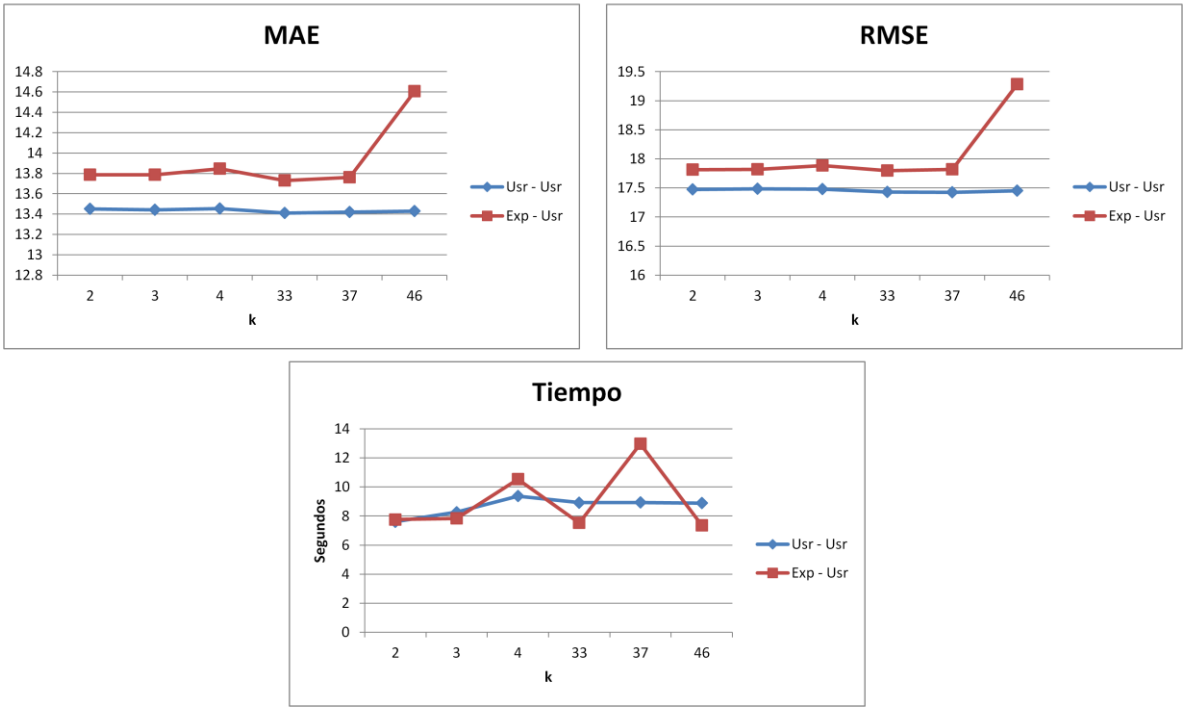
En el algoritmo de affinity propagation podemos observar que el menor error MAE y RMSE se obtiene usando el conjunto de datos de los expertos cuando K=2. En cuestión de tiempo se ve claramente que en todos los casos en el conjunto de usuarios se realiza en menor tiempo y la diferencia con el conjunto de expertos no es mucha, sin embargo la cantidad de datos que se tiene en el conjunto de expertos es 85% más grande.

BISECTING K-MEANS



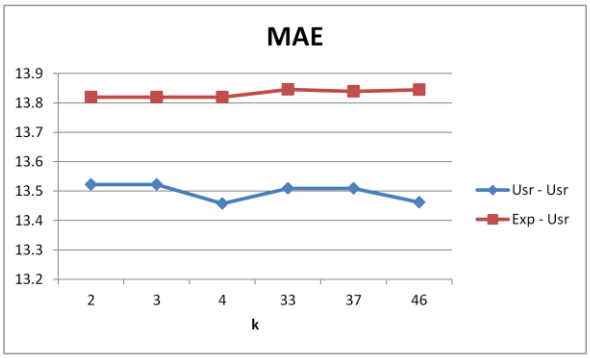
En bisecting K-Means se ve claramente que en todos los casos, al realizar una recomendación en el conjunto de usuarios los errores MAE y RMSE son menores que al utilizar opiniones de expertos. El tiempo en realizar la recomendación es menor utilizando expertos. Comparando con affinity propagation, bisecting K-Means da mejores resultados en ambos casos (usuarios y expertos) y es más rápido sobre todo con los expertos.

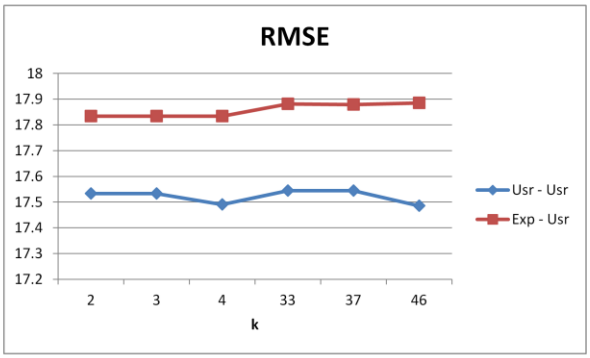
K-MEANS

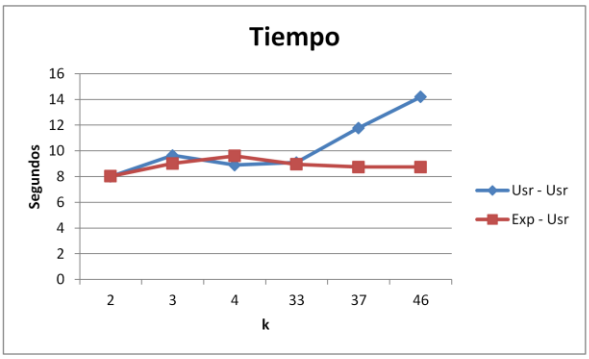


En K-Means podemos observar de igual manera que los resultados al realizar recomendaciones con usuarios es mejor que con expertos. Aquí podemos observar que el peor tiempo hasta ahora es en el conjunto de expertos con K=37 y el algoritmo K-Means. Hasta este momento el que ha dado mejores resultados es bisecting K-Means con K=37 tomando en cuenta a los usuarios.

K-MEDOIDS

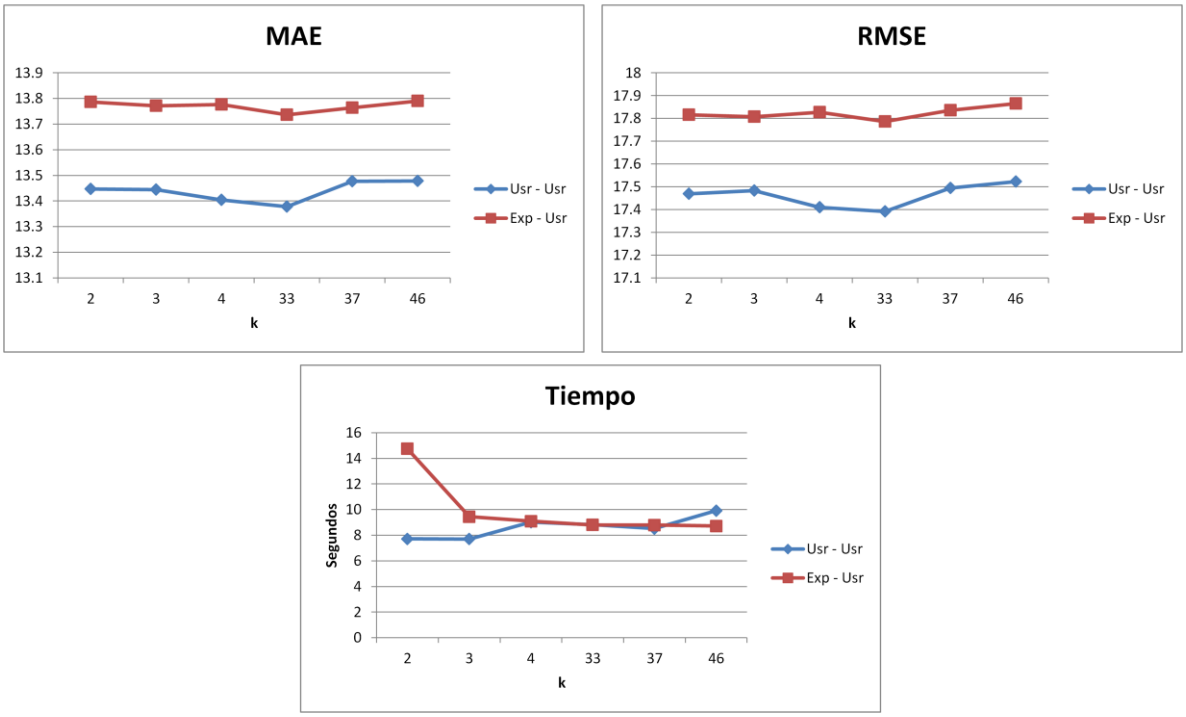






De igual manera en K-Medoids los mejores resultados se obtienen dentro del conjunto de expertos. No obstante, los mejores resultados siguen siendo con bisecting K-Means y K=37 tomando en cuenta los usuarios. Se puede observar que en general el tiempo que se tarda en realizar una recomendación con los expertos es menor que con los usuarios. Como se mencionó anteriormente, el conjunto de datos de expertos es 85% mayor que el de usuarios por lo cual podemos decir que K-Medoids realiza menos tiempo en la recomendación en el conjunto de expertos que de usuarios.

X-MEAN

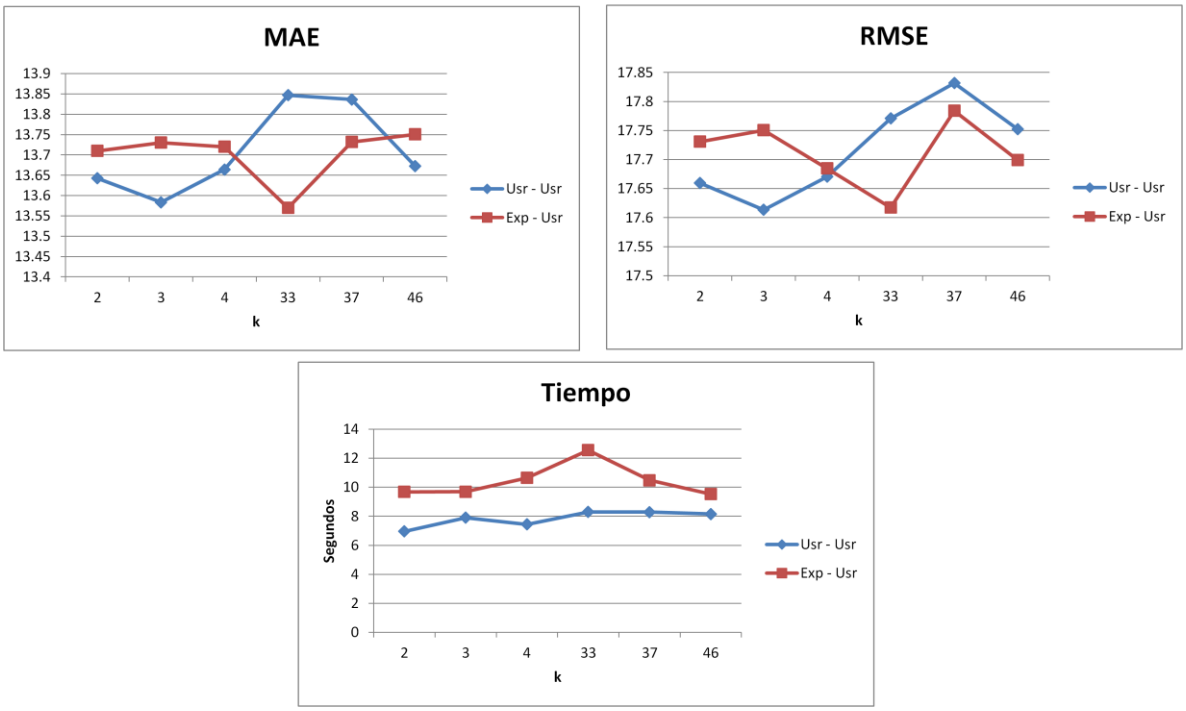


En X-Mean los mejores resultados igualmente se presentan con los expertos, siendo en K=33 el mejor de ellos. Considerando la cantidad de instancias en expertos y usuarios, podemos decir que es más rápido en el conjunto de datos de expertos.

Comparando todos los resultados con la distancia euclidiana podemos observar que el mejor resultado se dio en Bisecting K-Means con K=37 tomando en cuenta los usuarios. Considerando que el número de expertos es 85% mayor al número de usuarios, podemos decir que todos los algoritmos dan una recomendación más rápida tomando en cuenta opiniones de expertos, lo cual no quiere decir que den mejores resultados.

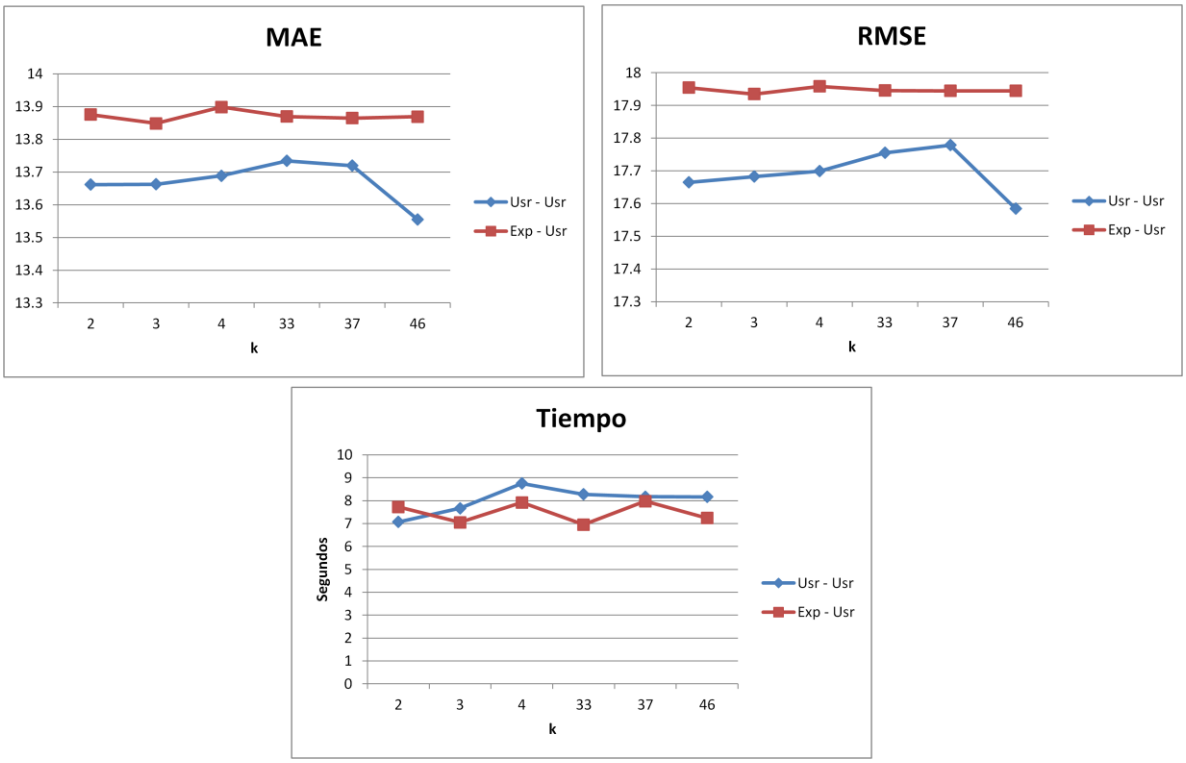
A.13. Resultados: Comparación Conjunto de Datos (Distancia Manhattan)

AFFINITY PROPAGATION



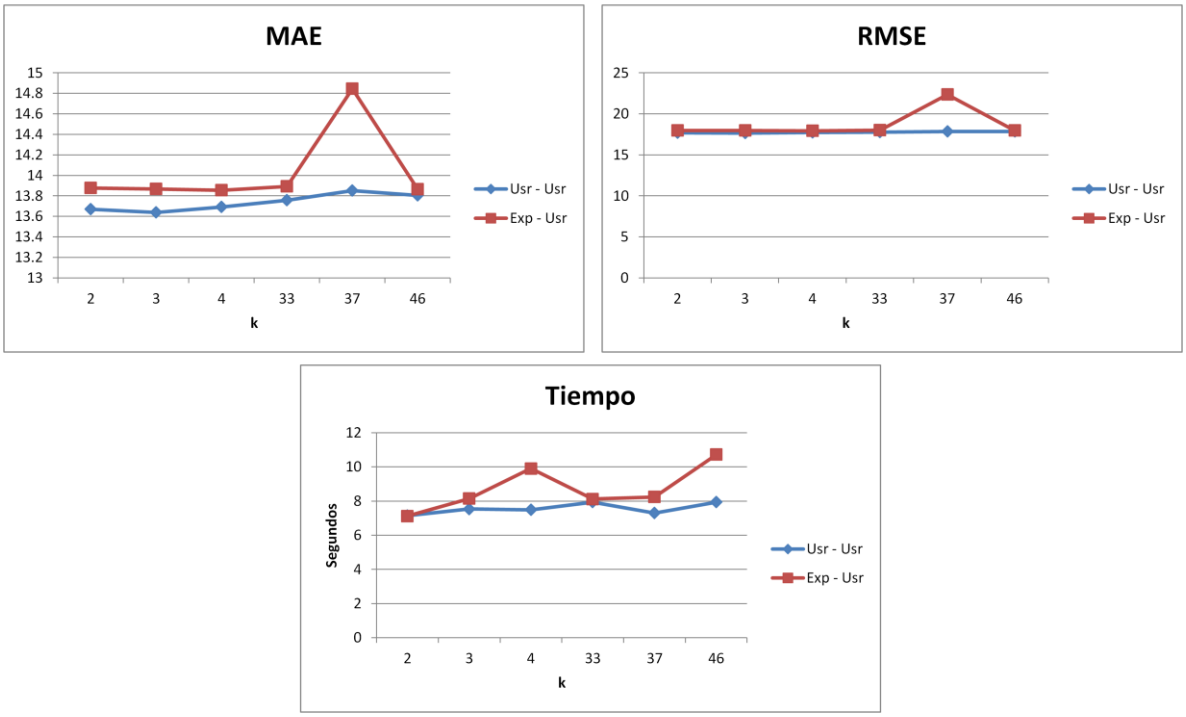
Para la distancia Manhattan observamos que el mejor resultado se obtiene con K=33 y en el conjunto de opiniones de expertos. Se tarda menos tiempo en realizar una recomendación dentro de los usuarios pero recordemos que tiene 85% menos usuarios que expertos.

BISECTING K-MEANS



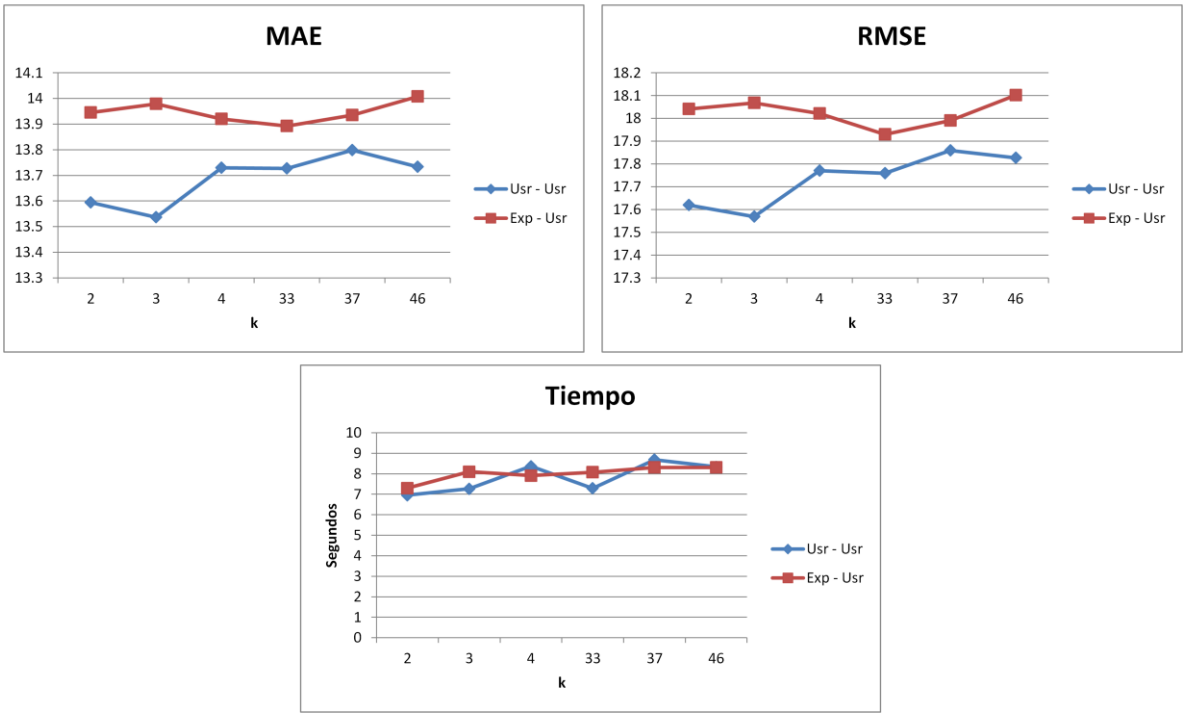
Bisecting K-Means con esta distancia igualmente da mejores resultados tomando en cuenta a usuarios. El mejor resultado obtenido es con K=46. Este algoritmo muestra una mejora en el tiempo.

K-MEANS



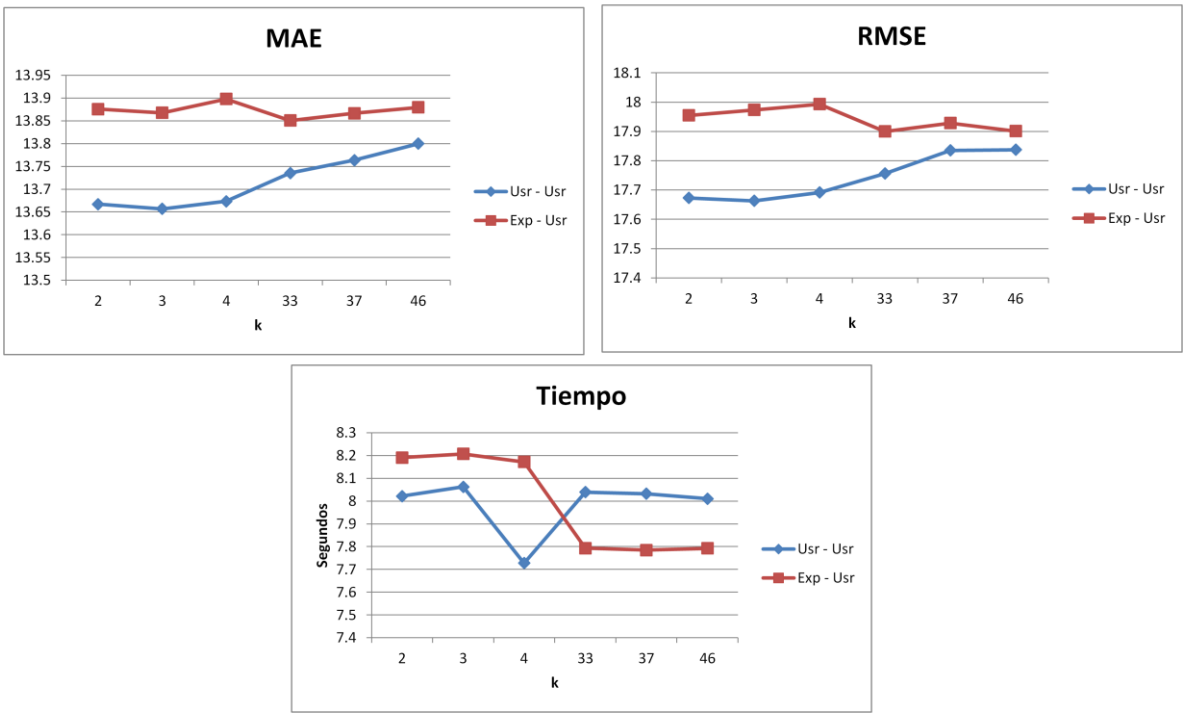
En K-Means podemos ver que el mejor resultado se obtiene con usuarios en K=3 que se puede observar en el error MAE. El RMSE es prácticamente el mismo valor para todos los casos a excepción de expertos con K=37.

K-MEDOIDS



K-Medoids obtiene mejores resultados si realizamos la comparación entre usuarios. En K=3 se da el mejor resultado.

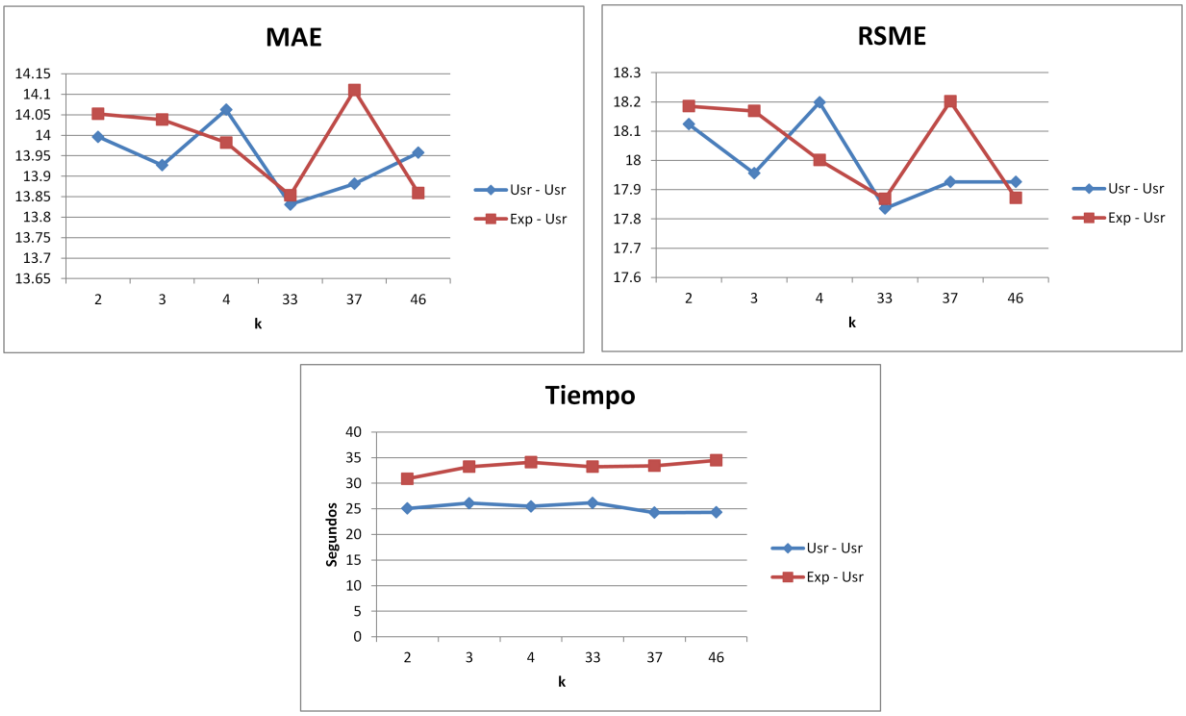
X-MEANS



Se observa que en X-Mean también los mejores resultados se dan dentro del grupo de usuarios. El tiempo menor lo observamos en el grupo de usuarios con K=4, sin embargo no es mucha la diferencia con los expertos en K=33,37 y 46. Recordando que tenemos más expertos que usuarios, el mejor resultado sería para estos casos.

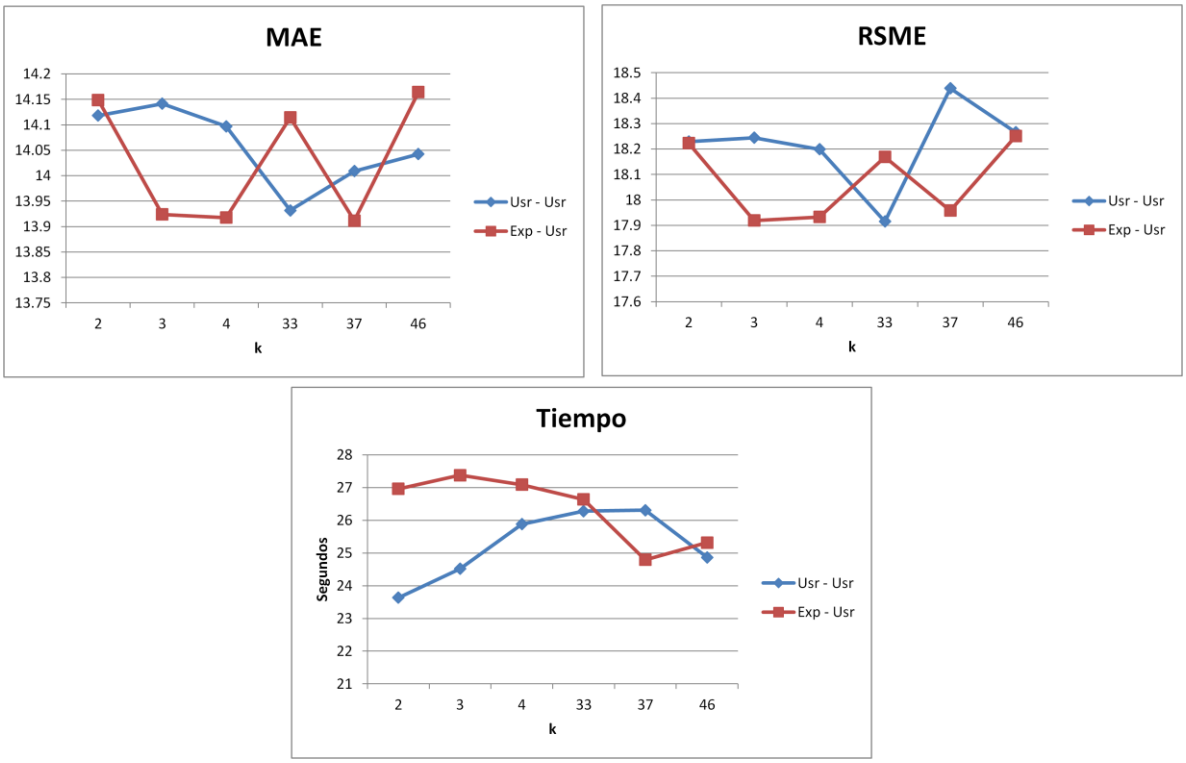
A.14. Resultados: Comparación Conjunto de Datos (Correlación de Pearson)

AFFINITY PROPAGATION



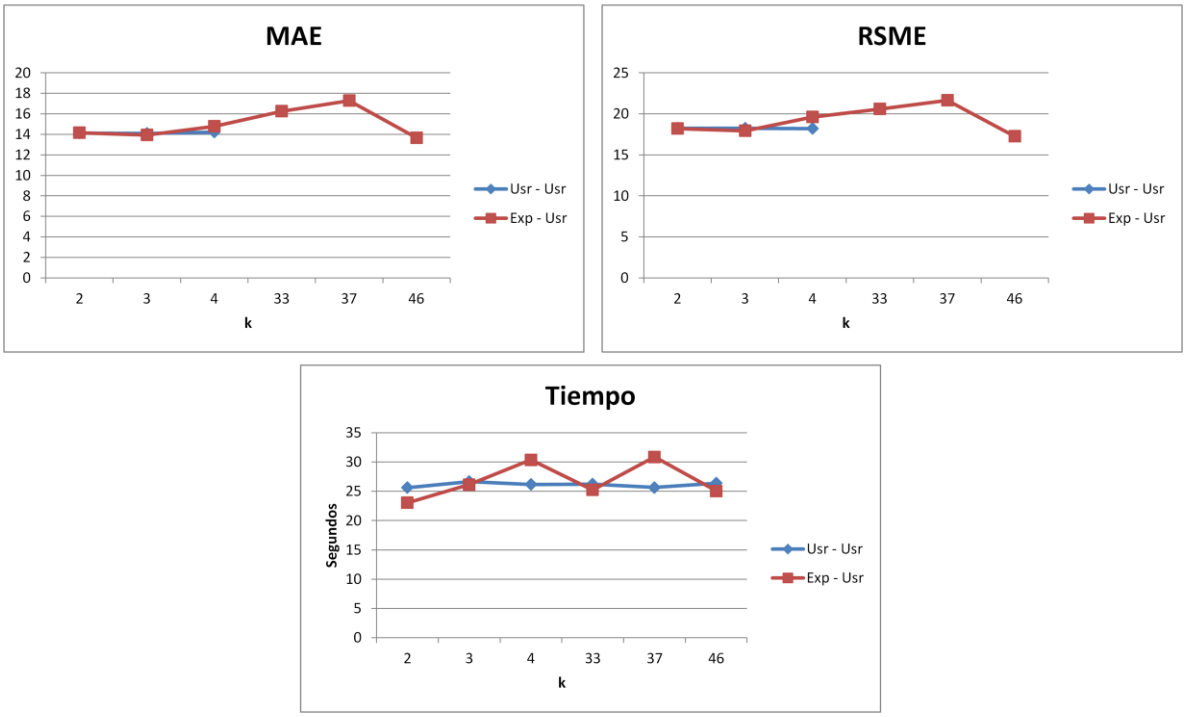
En el caso de la correlación de Pearson observamos el mejor resultado en K = 33 tanto para usuarios como para expertos. El tiempo en realizar una recomendación es menor en el grupo de usuarios.

BISECTING K-MEANS



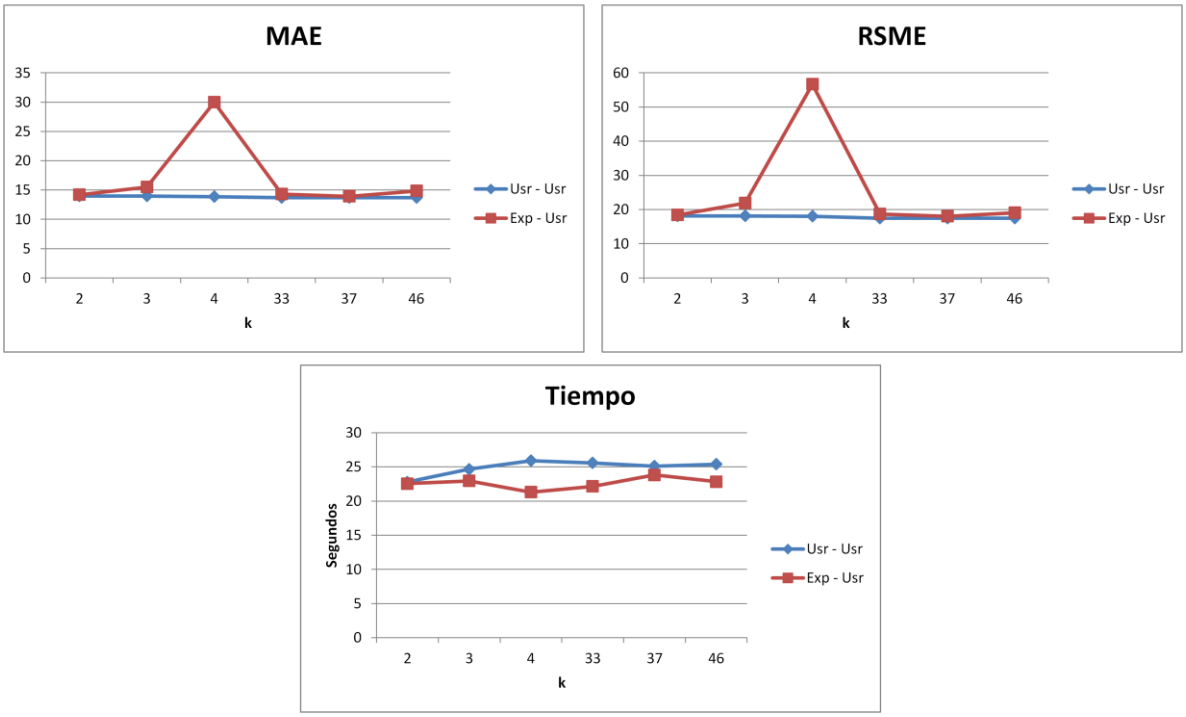
En el caso de bisecting K-Means con la correlación de Pearson los mejores resultados se dan con los expertos en K = 3, 4 y 37. El mejor tiempo en el caso de expertos se presenta en K=37.

K-MEANS



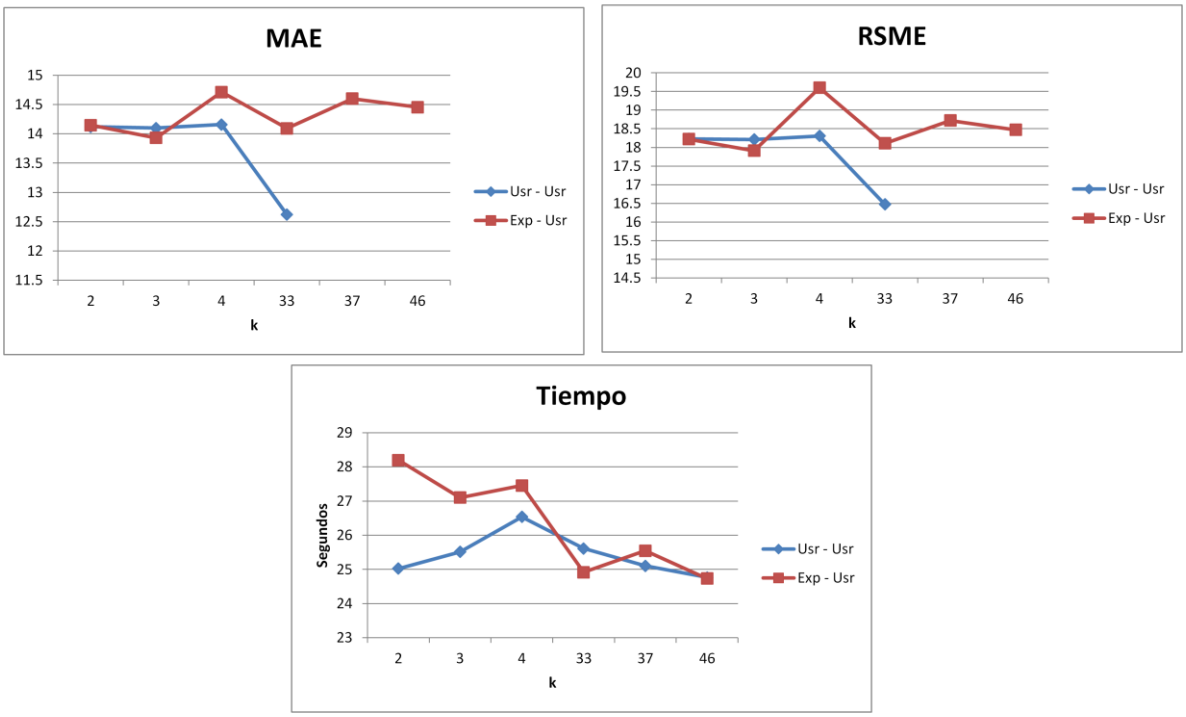
Con K-Means el mejor resultado se obtiene con expertos en K=46. Tomando en cuenta que se tienen más expertos que usuarios, podemos decir que la recomendación con los expertos, K-Means y la correlación de Pearson es más rápida para los expertos.

K-MEDOIDS



K-Medoids muestra un muy mal resultado en expertos con K=4 y la distancia de correlación de Pearson, en el resto de los casos tanto de usuarios como expertos los errores RMSE y MAE se comportan de manera similar. El tiempo es menor cuando se hace una recomendación tomando en cuenta a los expertos.

X-MEANS



El mejor resultado para este algoritmo podemos observar que es en K=33 tomando en cuenta los usuarios. El menor tiempo lo encontramos en K=46 ya sea con expertos o usuarios.

Realizando las observaciones correspondientes a las gráficas de los resultados obtenidos, podemos ver que el mejor resultado se obtiene con el algoritmo bisecting K-Means, la distancia Euclidiana y K=37.

IV. LISTA DE FIGURAS

V. LISTA DE TABLAS

VI. REFERENCIAS

[1] Francesco Ricci, Lior Rokach, Bracha Shapira. «Introduction to Recommender Systems.» En *Recommender Systems Handbook*, de Lior Rokach, Bracha Shapira, Paul B. Kantor Francesco Ricci, 1-29. Springer, 2011.

[2] Linyuan Lü, Matús Medo, Chi Ho Yeung, Yi-Cheng Zhanga, Zi-Ke Zhanga, Tao Zhou. «Recommender Systems.» *Physics Reports.* 2012.

[3] Xiaoyuan Su, Taghi M. Khoshgoftaar. «A Survey of Collaborative Filtering Techniques.» *Advances in Artificial Intelligence.* Vol. 2009. nº 4. 2009.

[4] T. Velmurugan, T. Santhanam. «Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points.» *Journal of Computer Science.* Vol. 6. nº 3. 2010. 363-368.

[5] Velmurugan, Dr. T. «Efficiency of K-Means and K-Medoids Algorithms for Clustering Arbitrary Data Points.» *Int.J.Computer Technology & Applications.* Vol. 3. nº 5. 2012. 1758-1764.

[6] Subhash K. Shinde, Uday V. Kulkarni. «Hybrid Personalized Recommender System Using Fast K-Medoids Clustering Algorithm.» *JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY* 2, nº 3 (2011).

[7] Alper Bilge, HuseyinPolat. «A scalable privacy-preserving recommendation scheme via bisecting K-Means clustering.» *Information ProcessingandManagement.* Vol. 49. nº 4. 2013. 912-927.

[8] MovieLens. «http://movielens.umn.edu/login.»

[9] Brendan J. Frey, Delbert Dueck. «Clustering by Passing Messages Between Data Points.» *Science Express.* Vol. 315. nº 5814. 2007. 972-976.

[10] Bahram Amini, Roliana Ibrahim, Mohd Shahizan Othman. «Discovering the Impact of Knowledge in Recommnder Systems: a Comparative Study.» *International Journal of Computer Science & Engineering Survey (IJCSES)* 2, nº 3 (2011).

[11] Long Yun, Yan Yang, Jing Wang, Ge Zhu. «Improving Rating Estimation in Recommender Using Demographic Data and Expert Opinions.» *IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS).* 2011. 120-123.

[12] Lathia, Neal. «Computing Recommendations with Collaborative Filtering.» *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling.* 2008.

[13] Satoshi Niwa, Takuo Doi, Shinichi Honiden. «Web Page Recommender System based on Folksonomy Mining for ITNG '06 Submissions.» *Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06).* 2006.

[14] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, Nuria Oliver. «The Wisdom of the Few A Collaborative Filtering Approach Based on Expert Opinions from the Web.» *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.* 2009. 532-539.

[15] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, William Nzoukou. «A Survey Paper on Recommender Systems.» *CoRR.* 2010.

[16] David Goldberg, David Nichols, Brian M. Oki and Douglas Terry. «Using collaborative filtering to weave an information Tapestry.» *Communications of the ACM - Special issue on information filtering* 35, nº 12 (1992): 61-70.

[17] Rodríguez, Antonio Pedro Albín. «Sistema de recomendación colaborativo basado en algoritmos de filtrado mejorados.» *Universidad de Jaén.* 2009.

[18] Robert M. Bell, Yehuda Koren, and Chris Volinsky. «All Together Now: A Perspective on the NETFLIX PRIZE.» *CHANCE.* Vol. 23. nº 1. 2010.

[19] Netflix_Prize. *http://en.wikipedia.org/wiki/Netflix_Prize.*

[20] Greg Linden, Brent Smith, and Jeremy York. «Amazon.com Recommendations Item-to-Item Collaborative Filtering.» *IEEE Internet Computing.* Vol. 7. nº 1. 2003. 76-80.

[21] Lastfm. *http://www.lastfm.es/.*

[22] Jester. *http://shadow.ieor.berkeley.edu/.*

[23] BookCrossing. *http://www.bookcrossing.com/.*

[24] PREA. «http://prea.gatech.edu/download.html#dataset.»

[25] HETREC2011. *http://ir.ii.uam.es/hetrec2011/.*

[26] RottenTomatoes. *http://developer.rottentomatoes.com/iodocs.*

[27] Guojun Gan, Chaoqun Ma, Jianhong Wu. *Data Clustering Theory, Algorithms, and Applications.* Editado por U.S. (12 de julio de 2007) Society for Industrial & Applied Mathematics. 2007.

[28] Jeffrey D. Ullman, Jure Leskovec, Anand Rajaraman. *Mining of Massive Datasets.* 2012.

[29] Dan Pelleg, Andrew Moore. «X-means: Extending K-Means with Efficient Estimation of the Number of Clusters.» En *Proceedings of the Seventeenth International Conference on Machine Learning*, de Morgan Kaufmann, 727-734. San Francisco, 2000.

[30] Thavikulwat, Precha. «Affinity Propagation: A Clustering Algorithm for Computer-Assisted Business Simulations and Experimential Exercises.» *Developments in Business Simulation and Experiential Learning* 35 (2008).

[31] Givoni, Inmar-Ella. «Beyond Affinity Propagation: Message Passing Algorithms for Clustering.» *University of Toronto.* 2012.

[32] Félix Hernández del Olmo, Elena Gaudioso. «Evaluation of recommender systems: A new approach.» *Expert Systems with Applications: An International Journal* 35, nº 3 (2008): 790-804.

[33] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. «The WEKA Data Mining Software: An Update.» *ACM SIGKDD Explorations Newsletter* 11, nº 1 (2009): 10-18.

[34] Xiangliang Zhang, Wei Wang, Kjetil Nørvag, Michele Sebag. «K-AP: Generating Specified K Clusters by Efficient Affinity Propagation.» *10th IEEE International Conference on Data Mining (ICDM' 2010)*, 2010: 1187-1192.

[35] K-AP. *https://www.lri.fr/~xlzhang/software.htm*.

[36] MATLAB, ToolBox. *http://www.mathworks.com/matlabcentral/fileexchange/13916-simple-tool-for-estimating-the-number-of-clusters/all_files*.

[37] Weka. «http://www.cs.waikato.ac.nz/ml/weka/.»

[38] MATLAB. «http://www.mathworks.com/products/matlab/index.html.»

[39] Sergio M. Savaresi, Daniel L. Boley. «A comparative Analysis on the bsecting K-Means and the PDDP clustering algorithms.» *Intelligent Data Analysis* 8, nº 4 (2003): 345-362.

Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

UNIDAD IZTAPALAPA

SISTEMAS DE RECOMENDACIÓN

Tesis que presenta:

Adriana Almaraz Pérez

Para obtener el grado de:

Maestra en Ciencias

(Ciencias y Tecnologías de la Información)

Asesor: Dr. John Goddard Close

Jurado Calificador:

Presidente: Dr. Adán Díaz Hernández

Secretario: M. en C. Fabiola Margarita Martínez Licona

Vocal: Dr. John Goddard Close

México, D.F. a 25 de Septiembre del 2013