ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Análisis y Comparación de Algoritmos con Filtro Colaborativo para Motor de Recomendación utilizando Mahout”

INFORME DE

MATERIA DE GRADUACIÓN

Previo la obtención de los Títulos de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentada por:

VERÓNICA DUARTE

ANTONNY GARCÍA

GUAYAQUIL – ECUADOR

2011

DEDICATORIA

A Dios por permitirme haber llegado hasta donde estoy, por guiar siempre mi camino y por darme la fortaleza que necesito para continuar.

A mis padres, Eugenia y Ramón, por ser los pilares fundamentales en mi vida, por enseñarme el valor del sacrificio, y por el apoyo que siempre me han brindado, por que sin ellos yo no habría llegado hasta aquí.

A mis amigos y profesores que han aportado con sus conocimientos, experiencias y las han compartido conmigo a lo largo de toda mi vida universitaria.

VERÓNICA DUARTE

A Dios por ser mi guía durante este periodo de aprendizaje, y ser el camino hacia la felicidad plena.

A mis padres Victoria y William, por entregarme su apoyo incondicional, y brindar consejos importantes para poder afrontar las adversidades que se han presentado en mi vida.

A mis amigos y profesores por su aportación intelectual y experiencia ganada, tanto en el campo de estudios como el del trabajo.

ANTONNY GARCÍA

AGRADECIMIENTO

Agradecemos a nuestras familias, compañeros y amigos que brindaron aportaciones y sugerencias en la elaboración de este trabajo, y de manera especial a la Ing. Vanessa Cedeño por compartir su experiencia y conocimiento para la culminación de este proyecto.

DECLARATORIA EXPRESA

"La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**"

(Reglamento de Graduación de la ESPOL)

_____

VERÓNICA DUARTE

_____

ANTONNY GARCÍA

TRIBUNAL DE SUSTENTACIÓN

Ing. Jorge Aragundi.

SUB-DECANO FIEC

Ing. Vanessa Cedeño

DIRECTOR DE PROYECTO

Ing. José Rodríguez.

MIEMBRO PRINCIPAL

RESUMEN

Los sistemas de recomendación han ganado popularidad hoy en día, convirtiéndose en una herramienta de gran importancia para sitios web de todo tipo que buscan ofrecer al usuario un contenido personalizado.

Estos sistemas se basan en algoritmos que son los encargados de generar las predicciones o recomendaciones que se mostrarán a través del SR. Existen varios enfoques que utilizan diferentes algoritmos para realizar esta labor. Definir cuál es más óptimo y eficiente, depende de la situación en la que se vaya a emplear y de las características propias del algoritmo.

En este trabajo se realiza un análisis y comparación de los dos principales algoritmos de recomendación basados en filtro colaborativo y que son usados en los SR: el algoritmo basado en usuario y el basadoen ítems proporcionados por el proyecto Mahout de Apache.

Se estudian los parámetros que pueden afectar los resultados obtenidos con los algoritmos y las pruebas que se realizaron con el fin de mostrar cómo se afectan si se cambia los valores de dichos parámetros. Y finalmente se muestra, en base a los resultados, que algoritmo es el mejor luego de haber seleccionado los mejores valores para los parámetros.

INDICE GENERAL

Resumen ..... **VIII**

indice general ..... **IX**

indice de figuras ..... **XI**

indice de tablas ..... **XII**

introducción ..... **XIII**

capítulo 1 ..... **15**

1. Antecedentes y justificación. **15**

1.1. ANTECEDENTES 15
1.2. DESCRIPCIÓN DEL PROBLEMA 16
1.3. OBJETIVOS 17
1.4. ALCANCE 19

capítulo 2 ..... **20**

2. Principios básicos de los sistemas de recomendación. **20**

2.1. SISTEMAS DE RECOMENDACIÓN 20
2.1.1. Sistemas de recomendación basado en filtro colaborativo 22
2.2. APACHE MAHOUT 25
2.3. ALGORITMO: *RECOMMENDER DOCUMENTATION* 26

capítulo3 ..... **28**

3. Análisis de la solución. **28**

PRIMERA ETAPA: FORMACIÓN DEL VECINDARIO. 29
SEGUNDA ETAPA: AGREGAR OPINIONES. 31
TERCERA ETAPA: RECOMENDACIÓN. 31

capítulo 4 ..... **34**

4. Diseño e implementación del análisis de los algoritmos basados en filtro colaborativo **34**

CONJUNTO DE DATOS 34
MÉTRICAS DE EVALUACIÓN. 36
PORCENTAJE DE PRUEBA Y TESTEO. 36
MÉTRICA DE SIMILARIDAD. 37

TAMAÑO DE LA VECINDAD. 37
capítulo 5 40
5. Pruebas y resultados. 40
5.1. EJECUCIÓN DE LAS PRUEBAS..... 40
5.2. ANÁLISIS DE LOS RESULTADOS..... 41
conclusiones 48
recomendaciones 50
anexos 51
ANEXO A..... 52
ANEXO B..... 53
ANEXO C 54
ANEXO D 55
bibliografía..... 56

INDICE DE FIGURAS

Figura 2-2–Definición de la arquitectura del algoritmo *RecommenderDocumentation*……**28**

Figura 3-1 – Etapas de un algoritmo Colaborativo basado en Memoria.……..**29**

Figura 3-2 – Ciclo iterativo que se producen con las etapas del algoritmo Colaborativo basado en Memoria……..……...**30**

Figura 4-1 – Algoritmo basado en usuario con filtro colaborativo, *Correlación de Pearson* para similaridad y método *NearestN* para definir vecindad…….**39**

Figura 4-2 – Algoritmo basado en usuario con filtro colaborativo, *Distancia Euclidiana* para similaridad y método NearestN para definir vecindad……...**39**

Figura 4-3 – Algoritmo basado en ítem con filtro colaborativo y *Correlación de Pearson* para definir la similaridad……...**40**

Figura 4-4 – Algoritmo basado en ítem con filtro colaborativo y *Distancia Euclidiana* para definir la similaridad……**40**

Figura 5-1 – Valor MAE obtenido de los distintos números de vecinos, para las métricas: coeficiente de correlación de Pearson y la distancia euclidiana……...**45**
.
Figura 5-2– Tiempo de ejecución, obtenido de los distintos números de vecinos, para las métricas: coeficiente de correlación de Pearson y la distancia euclidiana……....**46**

Figura 5-3– Valor MAE obtenido, de las métricas: coeficiente de correlación de Pearson y la distancia euclidiana, para los dos enfoques de sistema de recomendación basado en memoria…….**48**

Figura 5-4– Tiempo de ejecución obtenido, de las métricas: coeficiente de correlación de Pearson y la distancia euclidiana, para los dos enfoques de sistema de recomendación basado en memoria…….**48**

INDICE DE TABLAS

Tabla I – mediciones con respecto al mae, de los diferentes valores de vecinos..... **42**

Tabla II – mediciones con respecto al tiempo de ejecución, de los diferentes valores de vecinos. **42**

Tabla III – mediciones con respecto al mae, de los dos enfoques de sistemas de recomendación basado en memoria. **45**

Tabla IV – mediciones con respecto al tiempo de ejecución, de los dos enfoques de sistemas de recomendación basado en memoria. **47**

INTRODUCCIÓN

Los sistemas de Recomendación son herramientas que proporcionan ventajas de filtrado de información y que pueden facilitar a los usuarios la visualización cuando se trata de una gran cantidad de datos.

La idea es recomendar, a los usuarios, ítems o productos concretos en base a sus preferencias. De esta manera se pretende presentar información cada vez más personalizada.

Existen diversos tipos de SR, basados en diferentes algoritmos estadísticos y técnicas que varían en nivel de complejidad y se pueden ajustar a las necesidades propias de quienes los implementan.

Hasta ahora, la mayor aplicación que han tenido los SR, se ve reflejado en sitios web de comercio electrónico, donde se presenta al usuario por ejemplo el top ten de los ítems que tal vez le interesen o quizás la recomendación de algún producto especifico que podría gustarle porque a otros usuarios similares les ha gustado.

El contenido de este trabajo se distribuye de la siguiente manera: en el capítulo 1 se describe el problema, antecedentes, objetivos, justificación y alcance del presente trabajo. En el capítulo 2 se presentan tanto los

fundamentos teóricos de los sistemas de recomendación como los mecanismos para inferir dichas sugerencias. En el capítulo 3 se describe el análisis de la solución. En el capítulo 4 se explica la implementación utilizada para la obtención de los resultados. Las pruebas y resultados son mostrados en el capítulo 5. Finalmente, los anexos se encuentran en el capítulo 6.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN.

1.1. Antecedentes

Los sistemas de recomendación buscan ir un paso más allá, permitiendo a los usuarios obtener más información sobre temas, productos, grupos, etc. que tal vez les interese y para esto se basa en las preferencias y las opiniones de otros usuarios que deben ser recolectados y posteriormente procesados para analizar dichas preferencias.

"La idea que subyace tras ellos es encontrar usuarios con gustos similares a los de otro determinado y recomendar a este cosas que desconoce pero que gustan a aquellos con los que se tiene similitud"[1].

Por este motivo, herramientas como los motores de recomendación se han vuelto muy populares en la actualidad, porque facilitan a los sitios web el proponer temas a sus usuarios basándose en sus preferencias particulares.

Es el caso de Amazon, que es quizás el sitio de comercio electrónico más famoso en implementar las recomendaciones, basado en las compras y la actividad del sitio, recomienda libros y otros artículos que puedan ser de interés con el objetivo de incentivar la compra de una mayor cantidad de artículos [2].

En otros casos de redes sociales como Facebook y Twitter han usado los recomendadores para acercar usuarios que puedan compartir intereses comunes o que simplemente se conozcan.

También se puede mencionar los sitios de citas como Libimseti, incluso recomendar personas a otras personas. Y así una serie de sitios que actualmente los utilizan mucho los SR, adaptándolos para sus propios beneficios.

1.2. Descripción del Problema

En la actualidad se puede decir que el internet se ha convertido en una herramienta de mucha utilidad para todos. Mediante esta enorme red se encuentra datos casi sobre cualquier tema, lo cual la convierte en la

fuente más completa de información; sin embargo, esto podría ser un problema en algunos casos porque existe tanto contenido, tal vez mucho más de lo que las personas pueden abarcar de una manera efectiva. Puede tomar mucho tiempo encontrar los datos específicos, y los métodos de búsqueda tradicionales muchas veces no producen los resultados esperados [3].

Entonces, se tiene dos opciones, aprender a manejar tal cantidad de información lo cual puede tomar un tiempo considerablemente grande o utilizar técnicas que permitan filtrarla para que nuestro entorno, cuando se navega en la web, este conformado solo por aquella información que es relevante para nosotros.

La segunda opción resulta ser la más cómoda y con mejores beneficios. Para obtener estos resultados se debe utilizar los SR que sugieren información que puede interesar, librando al usuario así del molestoso exceso de información que hay en la web.

1.3. Objetivos

El tema de “Análisis y Comparación de Algoritmos de Filtro Colaborativo para Motor de Recomendación utilizando Mahout” fue

planteado para estudiar y explicar el funcionamiento de dos algoritmos que pueden ser aplicados para generar recomendaciones y que tienen enfoques distintos.

Para cumplir con este fin, se plantearon los siguientes objetivos:

- Analizar la estructura, funcionamiento y resultados arrojados por un algoritmo de recomendación basado en memoria con filtro colaborativo para almacenar y predecir la preferencia de una persona por un ítem particular que no conoce.

- Analizar la estructura, funcionamiento y resultados arrojados por un algoritmo de recomendación basado en modelo con filtro colaborativo para recomendar ítems basado en otros q son de agrado del usuario.

- Comparar ambos algoritmos para determinara sus diferencias, similitudes, ventajas y desventajas.

- Determinar mediante pruebas los mejores valores para obtener resultados óptimos con cada algoritmo.

- Comparar los resultados, generados por cada algoritmo en cuanto a métricas estadísticas.

1.4. Alcance

En este trabajo se realiza el análisis de los distintos factores que pueden afectar el resultado generado por ambos enfoques (basado en usuario y basado en ítem) del algoritmo con filtro colaborativo para motores de recomendación.

Se ejecutará el código evaluador de los algoritmos, alterando los valores de los parámetros y los tipos de métricas con el fin de elegir el código de cada algoritmo que tenga menor valor medio absoluto y el menor tiempo de ejecución. De esta manera se selecciona, de acuerdo a los resultados, el mejor.

CAPÍTULO 2

2. PRINCIPIOS BÁSICOS DE LOS SISTEMAS DE RECOMENDACIÓN.

2.1. Sistemas de recomendación

Los sistemas de recomendación aplican técnicas de descubrimiento de conocimiento para resolver el problema de producir recomendaciones personalizadas de información [4], respecto a productos, servicios, etc., que afectan el interés del usuario. Estos sistemas utilizan las preferencias del usuario para recomendarles ítems que aún no conocen.

Un ítem escualquier entidad empleada como objeto de recomendación, definida en el dominio del sistema, estos pueden ser libros, películas, documentos, incluso personas.

Entre los sistemas de recomendación más conocidos se encuentra Amazon [5], compañía estadounidense que brinda asesoría de forma personalizada en la compra de productos varios a través de la web; Facebook, la red social preferida [6] por más de 800 millones de personas en el mundo, tiene la capacidad de agrupar a un conjunto de personas con intereses comunes así como la función de búsqueda y sugerencias de amigos.

Recolección de datos

La recopilación de datos es una de las tareas fundamentales dentro del sistema de recomendación, para hacer aquello existen dos formas de recopilar información.

Explícita, donde el usuario expresa de forma clara y concisa su interés por uno o un grupo de ítems, por ejemplo:

- Pedir al usuario que califique en base a una escala proporcionada, algún tema particular.
- Solicita al usuario que pondere un conjunto de temas de una lista de temas preferidos.
- Solicitar al usuario de que seleccione de una lista de temas, los temas que él considera preferido

Implícita, donde el sistema recopila información, sin contar, con la intervención continúa del usuario, como por ejemplo.

- Guardar el historial de ítems visitados por el usuario.
- Analizar el número de visitas que recibe un ítem.
- Analizar el comportamiento del usuario en las redes sociales para así conocer sus gustos y preferencias.

2.1.1. Sistemas de recomendación basado en filtro colaborativo

Existen diversas técnicas de procesamiento de recomendaciones, empleadas por estos sistemas, entre los cuales cabe mencionar a los basado en contenido, utilidad, conocimiento y filtro colaborativo.

Los sistemas basados en filtro colaborativo tienen como idea básica proveer ítems recomendados en base a la opinión o preferencia que tienen otros usuarios, para eso mapea usuarios con intereses similares para luego crear recomendaciones basada en las relaciones creadas.

Enfoques del algoritmo de filtro colaborativo

Se presentan dos tipos de enfoque para el filtro colaborativo, basados en memoria y basados en modelo. Los algoritmos basados en memoria utilizan la base de datos que contiene una relación entre usuarios, ítems y valoración para generar las predicciones. Emplea métricas como correlaciones, coeficientes para determinar la similaridad que existe con el usuario, es decir un historial de valoraciones sobre los elementos, similar al del usuario actual y así poder inferir con un o una lista de ítems recomendados para el usuario activo.

Los algoritmos basados en modelo desarrollan primero un modelo de valoraciones del usuario, tratan el problema como uno de predicción estadística, calculando el valor esperado para cada ítem en función de las valoraciones anteriores.

Métricas de similaridad

Aquella medida que determina el grado de similaridad entre dos objetos o entidades es considerada como métrica de similaridad. Se pueden encontrar como correlaciones, coeficientes, distancia o matriz de similaridad. En un sistema de recomendación es empleado para determinar los usuarios e ítems similares entre sí.

Las métricas que se describen a continuación: el coeficiente de correlación de Pearson y la distancia euclidiana son las empleadas en las pruebas y análisis de algoritmos en este proyecto debido a que estas métricas toman en cuenta las valoraciones realizadas por los usuarios.

Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson mide la relación linealentre dos variables cuantitativas [7]. En este caso las variablescuantitativas son dos usuarios cualesquiera y los valores quetoman son las calificaciones implícitas a cada ítem del dataset.

Si *u* y *v* son dos usuarios, la similitud entre u y v se obtiene a través de la expresión:

$$sim(u,v) = \frac{\sum_{i=1}^{m}\left(r_{u,i} - \bar{r_u}\right)\left(r_{v,i} - \bar{r_v}\right)}{\sigma_u \sigma_v}$$

Donde $r_{u,i}$es la calificación que el usuario *u* asignó al ítem i, $r_{\bar{u}}$es la valoración media del usuario *u* y σ_u es la desviación estándar de las valoraciones del usuario *u* .

El rango del coeficiente es [-1,1], donde valores obtenidos en el rango (0,1] representa una correlación positiva, es decir si una de las variables incrementa en valor la otra también lo hará. Un coeficiente igual a cero significa que no existe una relación lineal entre ambas variables. Obtener valores en el rango [-1,0) representa una correlación inversa lo cual significa que mientras una aumenta la otra disminuye, o viceversa [8]. La medida actúa como un peso en la predicción de calificación de un usuario sobre un ítem desconocido por el mismo.

Distancia euclidiana

Esta métrica de similaridad mide la distancia entre dos usuarios *x* y *y*, Pensando en los ítems como las dimensiones y las

preferencias[9] como puntosa lo largo de esas dimensiones, la distancia se calcula utilizando todos los ítems (dimensiones) donde ambos usuarios han expresado su preferencia por ese tema. Esto es simplemente la raíz cuadrada de la suma de los cuadrados de las diferencias en la posición (de preferencia) a lo largo de cada dimensión.

La similaridad puede ser calculada como 1 / (1 + a distancia), por lo que los valores resultantes están en el rango (0,1].

Enfoques de los sistemas basados en memoria

Existen dos enfoques, uno de ellos, conocido como "*basado en el usuario"*, recomienda al usuario los ítems que aún no ha sido evaluados por él, pero que han sido evaluados por otros usuarios con gustos similares.

Otro de los enfoques, es aquel "*basado en el ítem"*, se basa en el hecho que a las personas les suele gustar cosas similares a las que ya les gusta, por lo tanto recomienda a un usuario los ítems similares a los que ha preferido antes.

2.2. Apache Mahout

Es un proyecto que permite construir bibliotecas escalables de aprendizaje automático. Construido sobre el potente paradigma

map/reduce del proyecto Apache Hadoop, Mahout permite resolver problemas como clustering, filtrado colaborativo y clasificación de terabytes de datos sobre miles de ordenadores.

2.3. Algoritmo: *RecommenderDocumentation*

Este motor de recomendación no está basado en Hadoop, anteriormente era un proyecto conocido como “Taste”, pero al ser incorporado en uno de los proyectos de Hadoop este adquiere mayor madurez,escalabilidad,flexibilidad, convirtiéndose en un algoritmo ideal para trabajar con recomendadores distribuidos.

Los algoritmos de recomendación deMahout no son sólo para la plataforma Java, sino que también puede funcionar como un servidor externo que expone la lógica de la recomendación de su aplicación a través de servicios web y HTTP.

Los paquetes que conforman la arquitectura del algoritmo son:

- DataModel (modelo de la base de datos)
- UserSimilarity (Similaridad entre usuarios)
- ItemSimilarity (Similaridad entre ítems)
- UserNeighborhood (vecindario de usuarios)
- Recommender (algoritmo de recomendación)

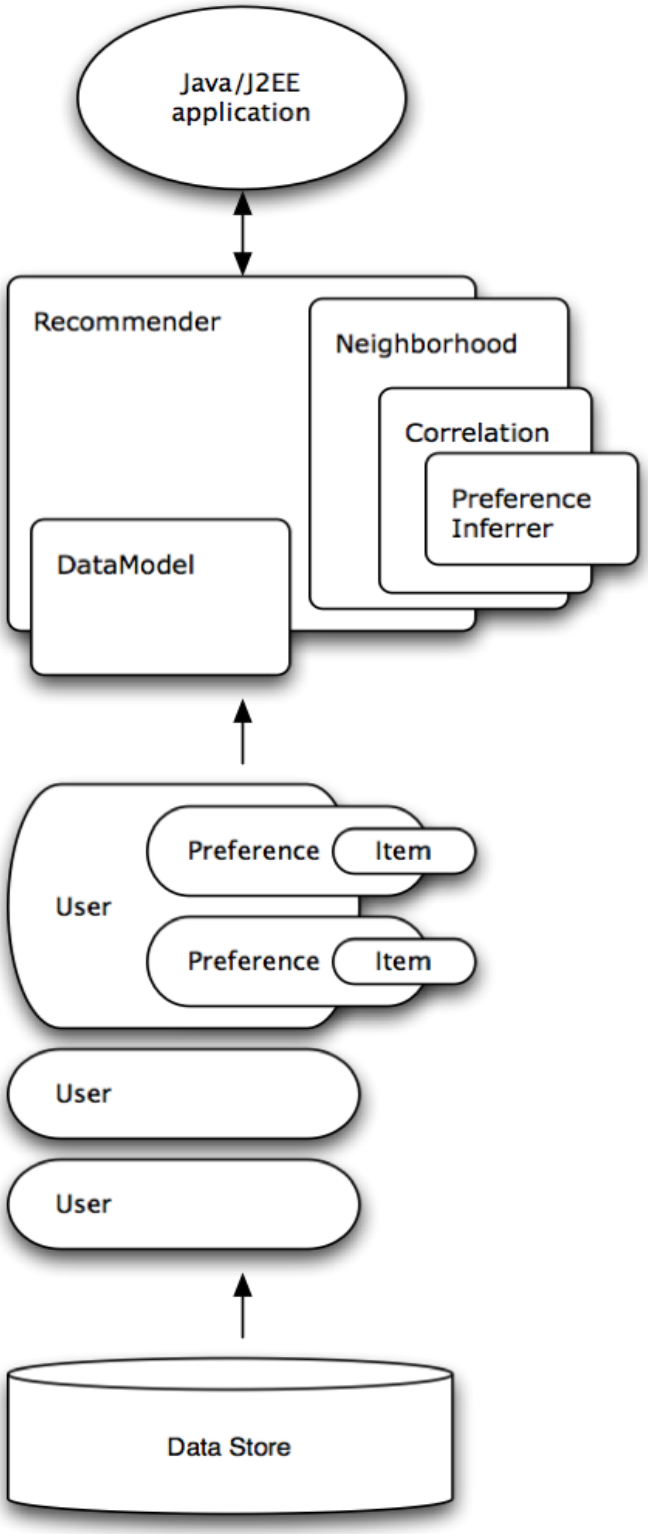


Figura 2-1–definición de la arquitectura del algoritmo *RecommenderDocumentation* [10],establecida por Mahout para explicar el modo de operación del mismo.

CAPÍTULO 3

3. ANÁLISIS DE LA SOLUCIÓN.

En este trabajo se ha realizado los análisis necesarios para poder encontrar el mejor algoritmo con Filtro Colaborativo basado en Memoria. Para esto se realizó una serie de pruebas y análisis que se mostrarán posteriormente.

Existen tres etapas en las que se basan los Sistemas de Recomendación con Filtro Colaborativo basado en Memoria.

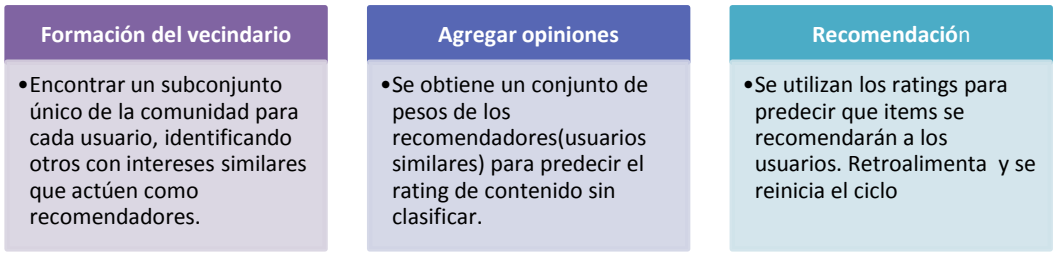


Figura 3.1 – Etapas de un algoritmo Colaborativo basado en Memoria.

Figura 3.2 – Ciclo iterativo que se producen con las etapas del algoritmo Colaborativo basado en Memoria.

Primera Etapa: Formación del vecindario.

En los enfoques colaborativos basados en usuario, durante la primera fase se selecciona a los usuarios cuyas preferencias son similares a las del usuario activo. Por el contrario, los enfoques basados en *ítem* extraen las preferencias del usuario que son más similares al producto objetivo.

Para el objetivo de este trabajo solo se implemento el uso de vecindario en los algoritmos de enfoque colaborativo basado en usuario para generar los resultados porque solo en estos casos se

necesita la información de otros usuarios para obtener las recomendaciones.

Para definir el vecindario lo ideal sería escoger todos los datos disponibles para seleccionar los mejores perfiles de usuarios y así lograr recomendaciones con menor margen de error. Sin embargo, esta opción no es la mejor si se cuenta con una abundante cantidad de datos y recursos limitados, en cuyo caso lo mejor es utilizar un muestreo aleatorio.

En cuanto al tamaño del vecindario, para definir el adecuado, se realizaron pruebas con respecto a los N vecinos más cercanos en un rango amplio para ir comprobando las variaciones de los resultados en cada caso. Es necesario tener en cuenta que si se elige un tamaño muy elevado se reduce la calidad de las recomendaciones obtenidas, mientras que si el tamaño es muy reducido se limitará la capacidad predictiva del sistema. La principal ventaja de esta técnica es que permite recomendaciones más precisas.

Una vez que se definió el tamaño de la vecindad, se procedió a establecer quiénes pueden ser considerados vecinos para generar las

recomendaciones en base a sus *ratings.* Esto es lo que se denomina Similitud.Las métricas para definir esta similitud son variadas.

Para evaluar este parámetro se utilizaron los métodos de Distancia Euclidiana y Correlación de Pearson.

Segunda Etapa: Agregar Opiniones.

En esta etapa se busca recolectar las calificaciones de otros usuarios con respecto a él o los productos que se van recomendar, cuando se refiere al enfoque colaborativo basado en usuario. En el caso del enfoque basado en *ítem,* se busca recolectar los ratings que el usuario (para el cual son las recomendaciones) ha dado a productos similares.

Para la ejecución de las pruebas en este trabajo no fue necesario recolectar dichas valoraciones. Se utilizó el conjunto de datos de GroupLens obtenido en internet.

Tercera etapa: Recomendación.

En esta etapa se genera la recomendación utilizando la estructura formada en las etapas anteriores. El algoritmo debe predecir el nivel de interés de un usuario en base a los datos que se entregados.

Para definir qué tan buenas son las recomendaciones generadas se utilizan las métricas de evaluación, que miden la calidad de los resultados, con el objetivo de analizar los puntos fuertes y débiles, visualizar las variables involucradas y ver cómo afectan los diferentes parámetros y sus variaciones.

Según Herlocker [11] una métrica para la exactitud de un sistema de recomendación mide, empíricamente, que tan cerca está el ordenamiento de ítems predichos para un usuario por el sistema con respecto al ordenamiento verdadero que el usuario haría, según su preferencia, de los mismos ítems.

A pesar de que el tema de los sistemas de recomendación es muy extenso, todavía no existen estándares necesarios para su evaluación. Es por eso que las métricas que se utilizan pueden variar dependiendo del caso, de la función del SR, del tipo de algoritmo, etc.

Esta diversidad de métricas conlleva a tres grandes problemas:

1. Si dos investigadores distintos, evalúan sus sistemas con distintas métricas, los resultados no son comparables.

2. Si la métrica usada no está estandarizada, se puede pensar que los investigadores han elegido la métrica más adecuada de cara a obtener los resultados deseados.
3. Sin una métrica estandarizada, cada investigador debe realizar un esfuerzo extra en identificar o desarrollar una métrica apropiada.

Teniendo en cuenta estas consideraciones, se procedió a elegir la métrica de evaluación denominada Error absoluto Medio (MAE). Esta es la más usada en cuanto a métricas estadísticas, ya que es fácil de entender y de representar al momento de querer mostrar la calidad de los resultados en forma gráfica. Mientras menor MAE se obtenga mejor será el resultado generado por el algoritmo de recomendación.

CAPÍTULO 4

4. DISEÑO E IMPLEMENTACIÓN DEL ANÁLISIS DE LOS ALGORITMOS BASADOS EN FILTRO COLABORATIVO.

En este capítulo se describe la metodología usada para el análisis de los algoritmos basados en filtro colaborativo. Posteriormente se detalla el plan de pruebas usado.

Conjunto de datos.

Se ha elegido el conjunto de datos MovieLens (http://www.grouplens.org/node/12), que fue recolectado por el proyecto de búsqueda de GroupLens de la Universidad de Minnesota y que ahora está disponible en el link mencionado. Se pueden descargar conjuntos de tres tamaños diferentes:

- 100.000 valoraciones.
- 1 millón de calificaciones.
- 10 millones de calificaciones.

En el caso de este trabajo se decidió trabajar con el conjunto de menor tamaño (100.000 valoraciones) porque es el idóneo para generar tiempos de respuestas más cortos.

Este conjunto de datos consta de tres archivos:

- **u.data**

  Consiste en 100.000 calificaciones de 943 usuarios para 1682 películas. Cada usuario calificó por lo menos 20 películas. Los datos están ordenados de forma aleatoria y los campos están clasificados de la siguiente manera:

  Id_usuario | id_item | calificación | fecha_hora.

- **u.user**

  Contiene la información de los usuarios clasificados y ordenados de la siguiente manera:

  Id_usuario | edad | género | ocupación | código_postal.

- **u.item**

  Contiene la información de los ítems(películas) clasificados y ordenados de la siguiente manera:

  Id_pelicula | titulo | fecha_lanzamiento | fecha_video_lanzamiento | IMDb URL | desconocido | acción | aventura | animación | para_niños|

comedia | Crimen | Documental | Drama | Fantasía | Cine negro | Horror | Musical | Misterio | Romance | Sci-Fi | Thriller | Guerra | Del Oeste.

Los últimos 19 campos son referentes al género de la película y los registros corresponden a valores booleanos. Una película puede pertenecer a uno o varios géneros.

Para realizar las pruebas solo se necesitará del archivo “u.data“que contiene las calificaciones.

Métricas de evaluación.

Entre las métricas de mencionadas en el capítulo 2 se utilizó MAE (error medio absoluto) debido a que es la medida más sencilla de utilizar y de interpretar. El método en el lenguaje de programación java, se denomina *AverageAbsoluteDifferenceRecommenderEvaluator()*.

Porcentaje de Prueba y Testeo.

Tal y como es recomendado en la página deGroupLensel libro *Mahout in Action*se decidió separar los 100.000 registros de calificaciones, que se encuentran en el archivo u.data de MovieLens, en un 80% para *training* y 20% para *test. Se usaron* los mismos porcentajes para ambos algoritmos.

Métrica de similaridad.

Se ha empleado dos métricas de similaridad: Correlación de Pearson, cuyo método correspondiente en java es *PearsonCorrelationSimilarity(model)*y Distancia Euclidiana con su método java*EuclideanDistanceSimilarity(model)* que se probaron con el conjunto de datos para seleccionar la mejor entre las 2. Los resultados se muestran en el siguiente capítulo.

Tamaño de la vecindad.

Este es un parámetro propio de los algoritmos basados en usuario (memoria) que tiene mucho impacto en los resultados obtenidos. Por lo tanto el valor que se defina para este es muy importante. Se ha elegido el algoritmo *NearestN* para definir la vecindad. El número de vecinos fue definido, de acuerdo a las pruebas que presentamos en el siguiente capítulo.

Con estos datos construimos la estructura del algoritmo basado en usuario y el algoritmo basado en ítem con filtro colaborativo, utilizando para ello el API de Mahout de la versión 4.0.

Se realizaron combinaciones con los diferentes parámetros y se muestran a continuación.

```
import  org.apache.mahout.cf.taste.recommender.*;
import  org.apache.mahout.cf.taste.similarity.*;
import  java.io.*;
import  java.util.*;

import  org.apache.mahout.cf.taste.eval.*;
import  org.apache.mahout.cf.taste.common.TasteException;
//import  org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
import  org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;


class Recomenderintro {
 public static void main(String[] args) throws Exception {

    DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
    /*-------------------------------------------------------------------------*/

    RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
        public Recommender buildRecommender(DataModel model) throws TasteException {
    /*algoritmos de similaridad basado en el usuario*/
        UserSimilarity similarity = new PearsonCorrelationSimilarity(model);// similarity es un valor entre 0 y 1
    /*-------------------------*/
    /* algoritmos de vecindad basado en el usuario */
        UserNeighborhood neighborhood = new NearestNUserNeighborhood(7, similarity, model);//N vecinos(aqui debe ir el mejor v
    /*-------------------------*/
        return new GenericUserBasedRecommender(model, neighborhood, similarity);//usuario
        }
    };
```

Figura 4.1 Algoritmo basado en usuario con filtro colaborativo, *Correlación de Pearson* para similaridad y método *NearestN* para definir vecindad.

```
import  org.apache.mahout.cf.taste.recommender.*;
import  org.apache.mahout.cf.taste.similarity.*;
import  java.io.*;
import  java.util.*;
import  org.apache.mahout.cf.taste.eval.*;
import  org.apache.mahout.cf.taste.common.TasteException;
//import  org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
import  org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;


class RecomenderintroUser {
 public static void main(String[] args) throws Exception {

    DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
    /*-------------------------------------------------------------------------*/

    RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
        public Recommender buildRecommender(DataModel model) throws TasteException {
        /*algoritmos de similaridad basado en el usuario*/
         UserSimilarity similarity = new EuclideanDistanceSimilarity(model);
         /*-------------------------*/
    /* algoritmos de vecindad basado en el usuario */
         UserNeighborhood neighborhood = new NearestNUserNeighborhood(7, similarity, model);//N vecinos(aqui debe ir el mejor v
    /*-------------------------*/
         //return new GenericUserBasedRecommender(model, neighborhood, similarity);//usuario
        }
    };
```

Figura 4.2 – Algoritmo basado en usuario con filtro colaborativo, *Distancia Euclidiana* para similaridad y método NearestN para definir vecindad.

```java
import  org.apache.mahout.cf.taste.impl.similarity.*;
import  org.apache.mahout.cf.taste.model.*;
import  org.apache.mahout.cf.taste.neighborhood.*;
import  org.apache.mahout.cf.taste.recommender.*;
import  org.apache.mahout.cf.taste.similarity.*;
import  java.io.*;
import  java.util.*;
import  org.apache.mahout.cf.taste.eval.*;
import  org.apache.mahout.cf.taste.common.TasteException;
//import  org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
import  org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;


class RecomenderintroItem {
 public static void main(String[] args) throws Exception {

    DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
    /*-------------------------------------------------------------------------------------*/

    RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
        public Recommender buildRecommender(DataModel model) throws TasteException {
    /*algoritmos de similaridad basado en el item*/
         ItemSimilarity similarity = new PearsonCorrelationSimilarity(model);// similarity es un valor entre 0 y 1
    /*-------------------------*/
         return new GenericItemBasedRecommender(model, similarity);//item
        }
    };
```

Figura 4.3 – Algoritmo basado en ítem con filtro colaborativo y *Correlación de Pearson* para definir la similaridad.

```java
import  org.apache.mahout.cf.taste.impl.similarity.*;
import  org.apache.mahout.cf.taste.model.*;
import  org.apache.mahout.cf.taste.neighborhood.*;
import  org.apache.mahout.cf.taste.recommender.*;
import  org.apache.mahout.cf.taste.similarity.*;
import  java.io.*;
import  java.util.*;
import  org.apache.mahout.cf.taste.eval.*;
import  org.apache.mahout.cf.taste.common.TasteException;
//import  org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
import  org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;


class RecomenderintroItem {
 public static void main(String[] args) throws Exception {

    DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
    /*-------------------------------------------------------------------------------------*/

    RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
        public Recommender buildRecommender(DataModel model) throws TasteException {
    /*algoritmos de similaridad basado en el item*/
         ItemSimilarity similarity = new EuclideanDistanceSimilarity(model);
    /*-------------------------*/
         return new GenericItemBasedRecommender(model, similarity);//item
        }
    };
```

Figura 4.4 – Algoritmo basado en ítem con filtro colaborativo y *Distancia Euclidiana* para definir la similaridad.

CAPÍTULO5

5. PRUEBAS Y RESULTADOS.

5.1. Ejecución de las pruebas

Las pruebas se realizaron en un equipo cuyas características son las siguientes:

- Procesador: Intel Core I3 2.3Ghz
- Disco duro: 10Gb
- Memoria RAM: 2048 Mb
- Sistema Operativo: Ubuntu 10.10 32 bits
- Mahout Versión: 0.4

La realización de esta etapa tiene dos propósitos. El primero, realizar una comparación entre los distintos valores de número de vecinos (múltiplos de 100),empleados para definir el tamaño de la vecindad respecto al usuario, para hacer aquello, los valores seleccionados se encuentran en un rango del 100 al 1000, con cada variación, se realiza pruebas con cada una de las métricas de similaridad definidas en el capítulo 2:el coeficiente de correlación de Pearson y la distancia euclidiana.

El segundo propósito para justificar la realización de pruebas, es efectuar una comparación entre los dos tipos de enfoque, para los sistemas de recomendación basada en memoria: usuario e ítem. Con el fin de llevar a cabo esta prueba se eligió como parámetro de entrada para el enfoque basado en usuario, los dos mejores valores de vecinos obtenidos en la prueba anterior, cada una representa a las métricas de similaridad mencionadas en la prueba anterior.

En ambas pruebas se consideró como métrica de evaluación, el puntaje obtenido en el MAE, siendo esta la que tiene mayor relevancia en las pruebas, esto se debe a que dicha métrica dice que tan bueno es, según los datos de entrada, los algoritmos de recomendación evaluados, otro indicador considerado es el tiempo de ejecución del algoritmo de recomendación.

Los resultados de cada prueba ejecutada se encuentran en los anexos.

5.2. Análisis de los resultados

En las tablas I, II se muestran las mediciones obtenidas, con respecto al indicadores de evaluación MAE y Tiempo de ejecución, para los distintos valores de vecinos, comparados tanto con la métrica de

similaridad coeficiente de correlación de Pearson como la distancia euclidiana.

Número de vecinos	MAE	
	PearsonCorrelation	EuclideanDistance
100	0,95235338	0,83654373
200	0,91921116	0,80228761
300	0,9001536	0,84204603
400	0,89697539	0,84416329
500	0,87755667	0,86021461
600	0,90938966	0,85154518
700	0,93324102	0,8433514
800	0,90042604	0,8279683
900	0,89312209	0,82955307
1000	0,92635875	0,80015912

Tabla I– mediciones con respecto al MAE, de los diferentes valores de vecinos.

Número de vecinos	Tiempo de ejecución	
	PearsonCorrelation	EuclideanDistance
100	14,29	22,91
200	16,89	26,18
300	18,67	33,93
400	16,61	25,18
500	25,17	34,39
600	18,44	26,12
700	17,22	18,55
800	19,52	27,74
900	21,48	13,43
1000	12,75	19,93

Tabla III–mediciones con respecto al Tiempo de ejecución, de los diferentes valores de vecinos.

En la figura 5-1, el eje vertical representa el valor MAE, mientras que el eje horizontal representa el número de vecinos. Como se observa en el gráfico, los valores de MAE oscilan entre 0.8 y 1, indicando que los valores de MAE obtenidos son muy bajos, es decir los resultados obtenidos no son buenos, esta tendencia se da a que el número de vecinos seleccionados no es lo suficiente grande para agrupar usuarios con características similares.

Los valores obtenidos de MAE, más bajos, resultan provenir de las pruebas efectuadas con la métrica de similaridad “Distancia Euclidiana”, donde los valores oscilan entre 0,80 y 0,86.

Los resultados de esta prueba indican que el error absoluto medio (MAE en inglés), predicen que el promedio de los errores absolutos, por cada tamaño de vecindad, donde el error absoluto se determina a partir de la diferencia entre la valoración predicha y la valoración real por un ítem, es menor y por consecuencia mejor, cuando se emplea la métrica de similaridad “Distancia Euclidiana” en vez de su contraparte, la métrica “coeficiente de correlación de Pearson”.

Los mejores valores MAE obtenidos para el coeficiente de correlación de Pearson y distancia Euclidiana son 0,8776 y 0,8002

respectivamente, en donde los números de vecinos fueron 500 y 1000 respectivamente.

En la figura 5-2, el eje vertical representa el tiempo de ejecución del algoritmo basado en usuario, mientras que el eje horizontal representa el número de vecinos. Con respecto a la prueba realizada, empleando como indicador de evaluación, el tiempo de ejecución, la métrica distancia Euclidiana, es la que tiene menor variación de tiempo, así como el menor de los tiempos.

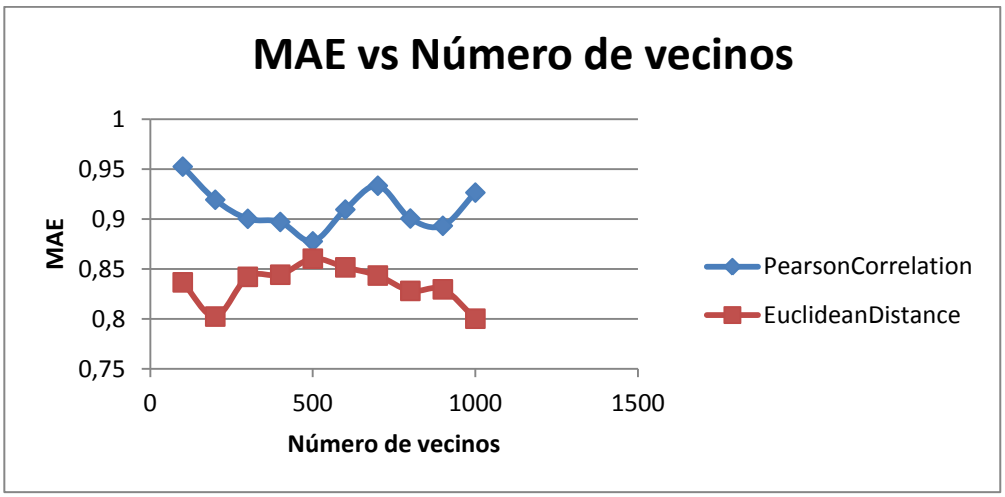


Figura 5-1– valor MAE obtenido de los distintos números de vecinos, para las métricas: coeficiente de correlación de Pearson y la distancia euclidiana.

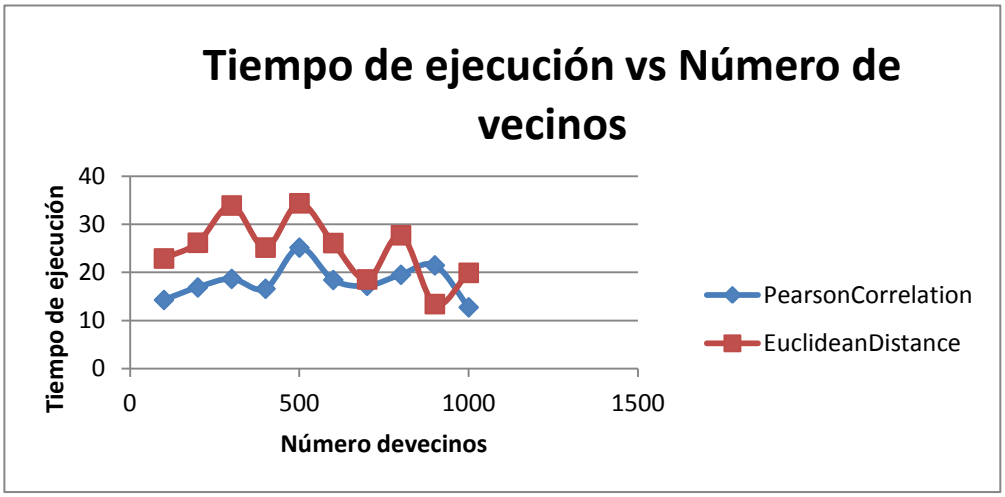


Figura 5-2– Tiempo de ejecución, obtenido de los distintos números de vecinos, para las métricas: coeficiente de correlación de Pearson y la distancia euclidiana.

En las tablas III, IV se muestran las mediciones obtenidas, con respecto alos indicadores de evaluación: MAE y Tiempo de ejecución respectivamente; Evaluando los dos enfoques de sistemas de recomendación basado en memoria: Basado en usuario y basado en ítem, se procede a comparar tanto, con la métrica de similaridad coeficiente de correlación de Pearson, como con la distancia euclidiana.

Métrica de similaridad	MAE	
	Basado en Usuario	Basado en Item
PearsonCorrelation	0,87755667	0,93203789
EuclideanDistance	0,80015912	0,83846361

Tabla IIIII – mediciones con respecto al MAE, de los dos enfoques de sistemas de recomendación basado en memoria.

En la figura 5-3, el eje vertical representa los distintos valores MAE, mientras que el eje horizontal se encuentra representado por los dos enfoques de los sistemas de recomendación basados en memoria. Como se aprecia en la figura los valores MAE obtenidos del enfoque basado en usuario, son menores con respecto al basado en ítem, con esto el algoritmo basado en usuario, resulta ser mejor, tanto, usando la métrica de similaridad de coeficiente de correlación de Pearson como la distancia euclidiana. Esto indica que resultó mejor obtener los ítems preferidos, de acuerdo, a la selección que hacen otros usuarios, que poseen características similares al usuario activo, frente a, los ítems seleccionados anteriormente por el usuario activo.

El tiempo de ejecución, de acuerdo a la figura 5-4, da como resultado al enfoque basado en ítem, como los tiempos de ejecución más corto, esto se debe a que no se debe realizar ninguna tarea adicional como por ejemplo normalizar las calificaciones de los ítems, para tener un mejor resultado en las pruebas.

métrica de similaridad	Tiempo de ejecución	
	Usuario	Item
PearsonCorrelation	12,75	7,06
EuclideanDistance	13,43	7,62

Tabla IVV – mediciones con respecto al Tiempo de ejecución, de los dos enfoques de sistemas de recomendación basado en memoria.

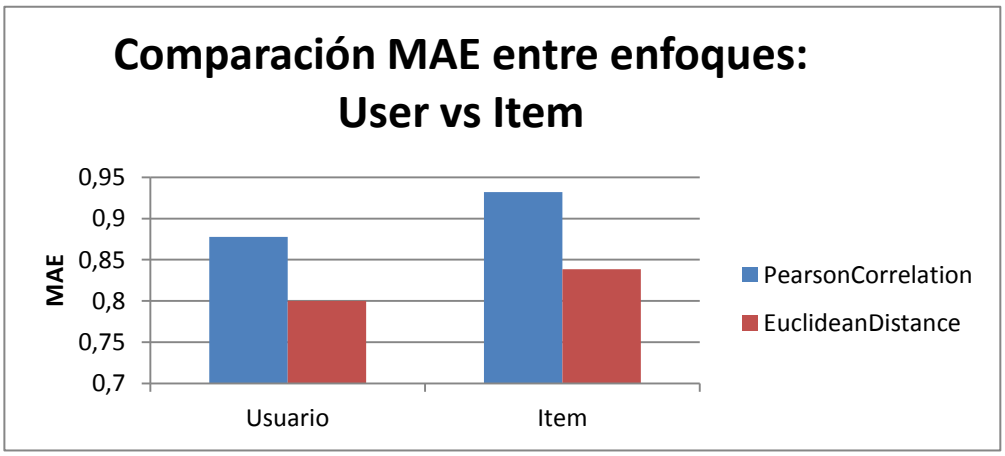


Figura 5-3–valor MAE obtenido, de las métricas: coeficiente de correlación de Pearson y la distancia euclidiana, para los dos enfoques de sistema de recomendación basado en memoria.

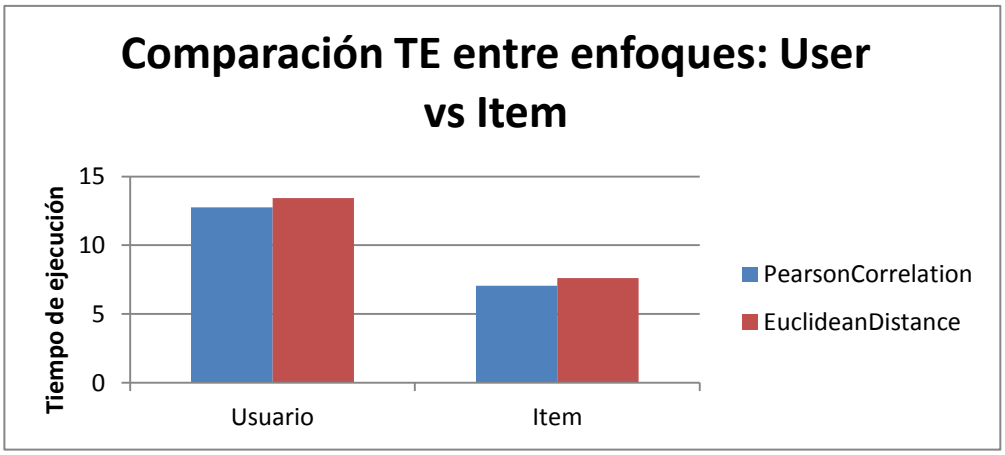


Figura 5-4–Tiempo de ejecución obtenido, de las métricas: coeficiente de correlación de Pearson y la distancia euclidiana, para los dos enfoques de sistema de recomendación basado en memoria.

CONCLUSIONES

Las conclusiones son:

1. Los sistemas de recomendación son herramientas de ayuda para la toma de decisiones que son muy utilizadas en la actualidad. Las más comunes son aquellas relacionadas con el ocio, donde se podrían implementar estos algoritmos para recomendar actividades, y el comercio electrónico para recomendar a los clientes nuevos productos. Otras aplicaciones también interesantes podría ser utilizarlos en sistemas tutores que ayuden a los estudiantes a decidir que materias tomar en un semestre. En fin, existe un amplio espectro de aplicaciones en las que se pueden emplear estos algoritmos de recomendación.

2. Al realizar las pruebas de ejecución de los algoritmos se observó que el tiempo de ejecución oscila entre los 14 y 34 segundos, un tiempo de respuesta que debería ser mucho menor pero al utilizar un conjunto de datos tan grande (100.000 calificaciones) es un tiempo justificable.

3. Se construyó, empleando los valores más óptimos, el algoritmo basado en Usuario con Filtro Colaborativo que utiliza como métrica de similaridad a la *Correlación de Pearson;* La técnica*NearestN*empleada

para definir la vecindad,empleóun tamaño de 1000. Se dividió el conjunto de datos en un 80% de *trainning* y un 20% de *test.* Y se evaluó utilizando la métrica estadística MAE (Error Medio Absoluto).

4. Se construyó,empleando los valores más óptimos, el algoritmo basado en ítem con Filtro Colaborativo, que utiliza como métrica de similaridad a la*Correlación de Pearson*. Se dividió el conjunto de datos en un 80% de *trainning* y un 20% de *test.* Y se evaluó utilizando la métrica estadística MAE (Error Medio Absoluto).

5. En base a, que ambos algoritmos fueron construidos con las mejores características en igualdad de condiciones, y luego de compararlos, se obtuvo que el algoritmo basado en Usuario con Filtro Colaborativo es mejor que el algoritmo basado en Ítem, ya que genera una tasa de MAE mucho menor y también el tiempo de ejecución es mínimo con respecto al de usuario.

RECOMENDACIONES

Las recomendaciones son:

1. Los elementos que conforman la estructura de un algoritmo, con Filtro Colaborativo para Motor de Recomendación, afectan los resultados que se obtienen de los mismos. Por lo tanto, elegir la forma de similaridad, el tamaño de la vecindad, el método para definir la vecindad, el tamaño del conjunto de datos, es un proceso que debe definirse de manera muy cautelosa con un análisis previo para obtener los mejores resultados.

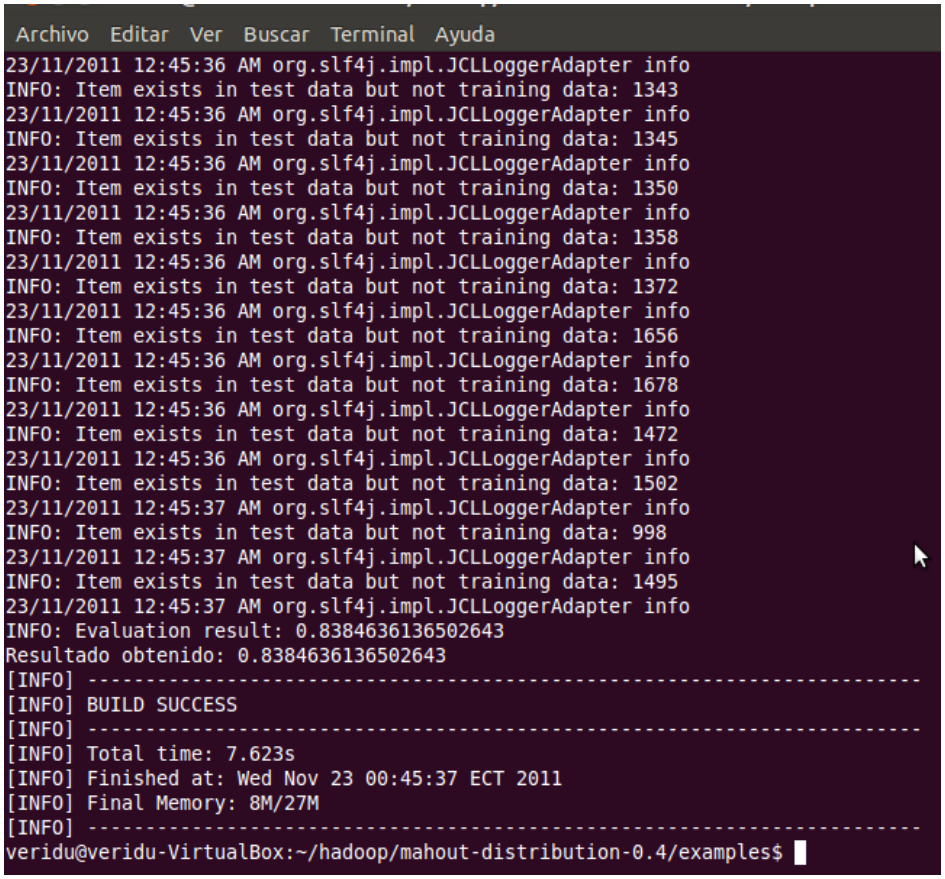
2. Las pruebas se realizaron en un solo ordenador debido a que los algoritmos que utilizamos no tienen características de ser distribuidos; sin embargo, estos algoritmos pueden ser modificados, al ser de código libre, podrían ser adaptados.

3. Como trabajo futuro se puede considerar este trabajo para implementar los algoritmos sobre la plataforma de Hadoop utilizando el paradigma Map/Reduce para procesamiento masivo y escalable de datos. De esta forma se lograría reducir los tiempos de ejecución y de respuesta, haciendo que el algoritmo sea aun mejor.

ANEXOS

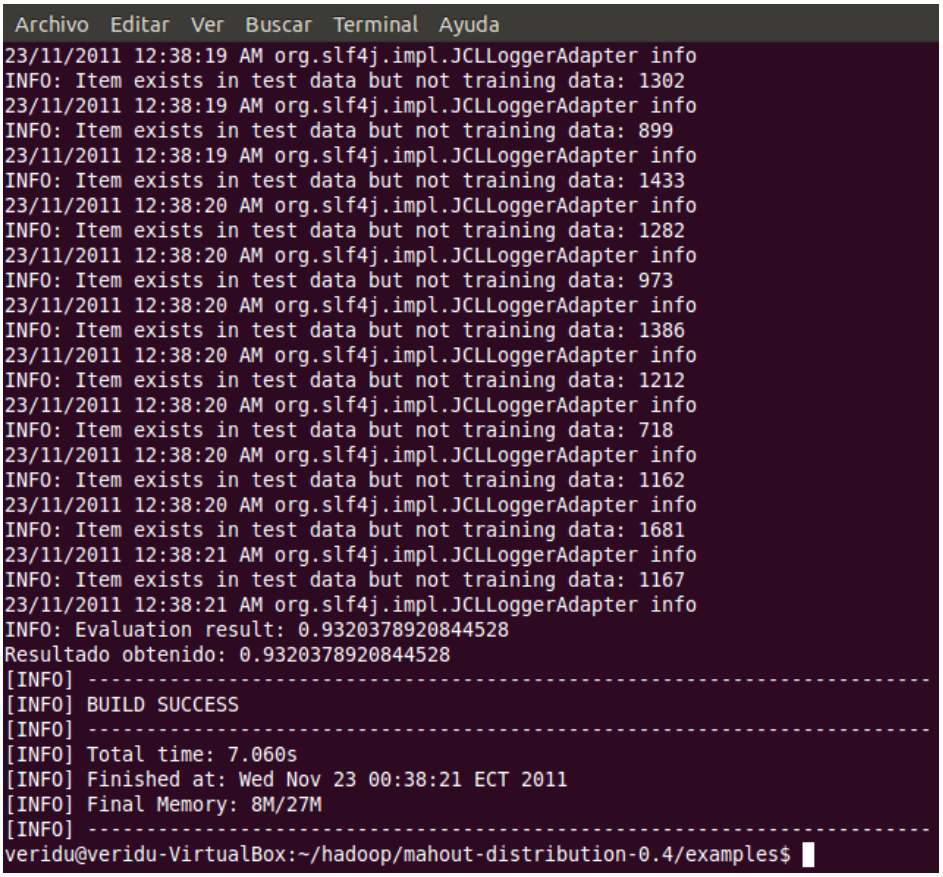
ANEXO A

Ejecución del algoritmo basado en Ítem utilizando *EuclidianDistance*como métrica de similaridad



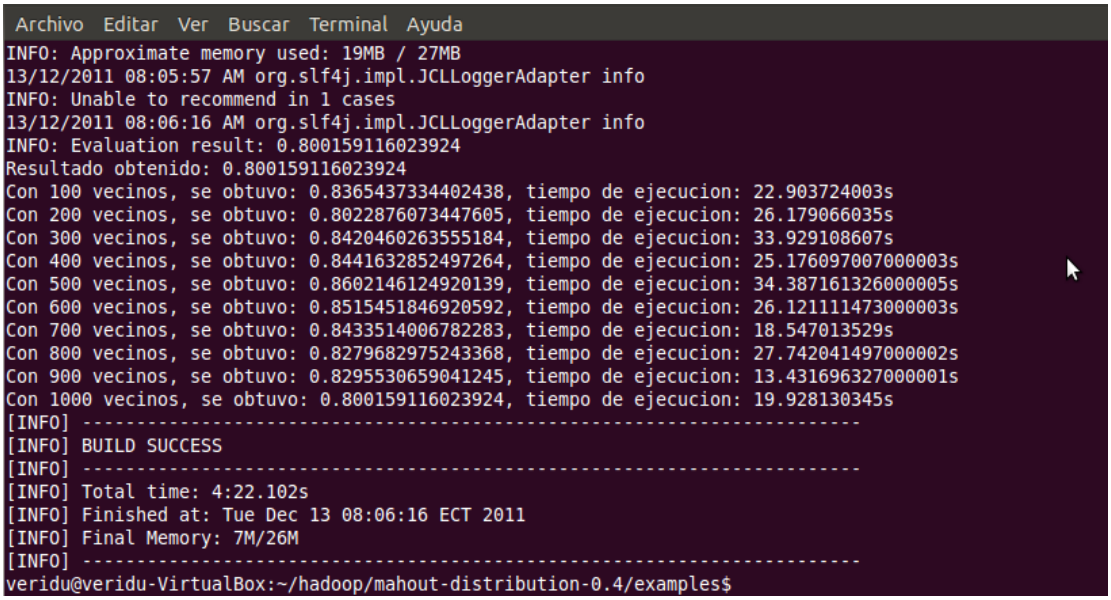
ANEXO B

Ejecución del algoritmo basado en Ítem utilizando *PearsonCorrelation* como métrica de similaridad.



ANEXO C

Ejecución del algoritmo basado en usuario utilizando *EuclidianDistance*como métrica de similaridad y*NearestN* como método de selección de vecindad con número de vecinos entre 100 y 1000.



ANEXO D

Ejecución del algoritmo basado en usuario utilizando *PearsonCorrelation* como métrica de similaridad y*NearestN* como método de selección de vecindad con número de vecinos entre 100 y 1000.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
INFO: Approximate memory used: 19MB / 27MB
13/12/2011 08:55:58 AM org.slf4j.impl.JCLLoggerAdapter info
INFO: Unable to recommend in 4 cases
13/12/2011 08:56:10 AM org.slf4j.impl.JCLLoggerAdapter info
INFO: Evaluation result: 0.9263587474897694
Resultado obtenido: 0.9263587474897694
Con 100 vecinos, se obtuvo: 0.9523533871174602, tiempo de ejecucion: 14.290664439s
Con 200 vecinos, se obtuvo: 0.9192111596597259, tiempo de ejecucion: 16.891109890000003s
Con 300 vecinos, se obtuvo: 0.900153602460709, tiempo de ejecucion: 18.674230398000002s
Con 400 vecinos, se obtuvo: 0.8969753899751459, tiempo de ejecucion: 16.608005011s
Con 500 vecinos, se obtuvo: 0.8775566634348887, tiempo de ejecucion: 25.171660260000003s
Con 600 vecinos, se obtuvo: 0.9093896558369543, tiempo de ejecucion: 18.43672683s
Con 700 vecinos, se obtuvo: 0.933241024984013, tiempo de ejecucion: 17.219328747000002s
Con 800 vecinos, se obtuvo: 0.9004260412903416, tiempo de ejecucion: 19.526224549000002s
Con 900 vecinos, se obtuvo: 0.8931220866561322, tiempo de ejecucion: 21.476513829s
Con 1000 vecinos, se obtuvo: 0.9263587474897694, tiempo de ejecucion: 12.74999867s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 3:14.664s
[INFO] Finished at: Tue Dec 13 08:56:10 ECT 2011
[INFO] Final Memory: 7M/26M
[INFO] ------------------------------------------------------------------------
veridu@veridu-VirtualBox:~/hadoop/mahout-distribution-0.4/examples$
```

BIBLIOGRAFÍA

[1]Galán S., Filtrado Colaborativo y Sistemas de Recomendación, http://www.it.uc3m.es/jvillena/irc/practicas/06-07/31.pdf, 2007.

[2]Owen S., Anil R., Dunning T., Friedman E., Mahout in Action, Manning Publications Co., 2009, 2-3.

[3]Velez O., SantoC., Sistemas Recomendadores: Un enfoque desde los algoritmos genético, artículo obtenido en *La ScientificElectronic Library Online*, 2006, 23.

[4]Sarwar B., Karypis G., Konstan J., Riedl John, Item-Based Collaborative Filtering Recommendation Algorithms, ACM, 2001, Páginas 1-2.

[5]Macías X., De la Rosa F., Generación de recomendaciones de ítems musicales basado en las valoraciones implícitas y las similitudes de los usuarios utilizando Hadoop para procesamientos masivos y escalables, ESPOL, 2009, Páginas 1-2.

[6]Facebook stadistic, http://www.facebook.com/press/info.php?statistics, 2011.

[7]Del Pino J., Salazar G., Análisis de métricas de similaridad usadas en un filtro colaborativo basado en el usuario para recomendar materias de pregrado, Repositorio Espol, 2011.

[8]Weisstein, Eric W., Correlation Coefficient, http://mathworld.wolfram.com/CorrelationCoefficient.html, MathWorld, Wolfram Research, fecha de consultanoviembre 2011.

[9]The Apache Software Fundation, https://builds.apache.org/job/Mahout-Quality/javadoc/index.html?org/apache/mahout/cf/taste/impl/similarity/EuclideanDistanceSimilarity.html, Mahout API, 2011.

[10]Drost I. y Owen S., RecommenderDocumentation, https://cwiki.apache.org/confluence/display/MAHOUT/Recommender+Documentation, 26 de agosto de 2011.

[11]HerlockerJ., KonstanJ., TerveenL., and Riedl J., Evaluating collaborative filtering recommender systems.