Universidad Nacional de Luján

Lic. en Sistemas de Información

Bases de Datos Masivas

Generación de recomendaciones a partir de reglas de asociación

Salvador Woinilowicz
Legajo: 121546

Resumen

Este trabajo es un primer acercamiento al desarrollo de sistemas de recomendación, utilizando como herramienta un algoritmo clásico de minería de datos llamado Apriori para generar reglas de asociación. Haciendo uso de estas reglas sobre un conjunto de datos masivos de consumo de títulos de series animadas para generar recomendaciones y evaluando su desempeño, se encontró que aplicando distintos enfoques a la hora de generar las reglas y de armar el ranking de selección de las mismas, podemos obtener un sistema de recomendación eficiente y muy sencillo de implementar.

1. Introducción

Los sistemas de recomendación se han vuelto muy populares en los últimos años y se utilizan en diversas aplicaciones web. Los sistemas de recomendación (SR) son herramientas de software que se utilizan para proporcionar sugerencias al usuario según su requerimiento. Las sugerencias se asocian con varios procesos de toma de decisiones, tales como qué artículos comprar, qué música escuchar, qué película mirar. Ítem es el término general usado para denotar qué recomienda el sistema a los usuarios. Un SR normalmente hace foco en un tipo de ítem específico, su diseño, su interfaz gráfica de usuario y el núcleo de la técnica de recomendación utilizada para generar todas las recomendaciones están personalizadas para proporcionar sugerencias efectivas para ese tipo específico de ítem [1]. En la actualidad, los SR tienen un nivel de eficiencia alto ya que pueden asociar elementos de nuestros perfiles de consumo como el historial de compras, selección de contenidos e inclusive nuestras horas de actividad, para realizar las recomendaciones [2].

Existen dos tipos de filtros o sistemas de recomendación:

1) Los filtros colaborativos: generalmente basan su lógica en las características del usuario. Es por ello que los datos disponibles del usuario son su eje; el sistema analiza las compras anteriores, preferencias, calificaciones que ha dado de otros productos, el importe medio de las compras, etc. y busca otros usuarios que se parecen a él y que han tomado decisiones similares. Los productos que han tenido éxito en estos casos, seguramente también le interesará al nuevo usuario [2].

2) Los filtros basados en contenido: el producto es la base de la predicción, en lugar del usuario. Utilizan las características del artículo (marca, precio, calificaciones, tamaño, categoría, etc.) para hacer las recomendaciones. En un ejemplo de streaming cinematográfico, los productos serían las películas y los datos disponibles son título, portada, sinopsis, año, país de origen, director, actores, género, duración, etc. Para enriquecer más al sistema, se valoran las calificaciones que el usuario ha hecho sobre las películas -calificaciones explícitas, como las puntuaciones con estrellas o favoritos, o implícitas, cuántas veces la ha visto o si la ha visto completa- así como las características propias del usuario. Estos datos, centrados en el producto y alineados con datos del usuario, son la materia prima del sistema de recomendación [2].

Las reglas de asociación tienen como objetivo encontrar relaciones dentro de un conjunto de transacciones, en concreto, ítems o atributos que tienden a ocurrir de forma conjunta [4].
El término transacción hace referencia a cada grupo de eventos que están asociados de alguna forma, por ejemplo, la cesta de la compra en un supermercado, las páginas web visitadas por un usuario o las películas o series vistas en una plataforma de *streaming*.
A cada uno de los eventos o elementos que forman parte de una transacción se le conoce como ítem y a un conjunto de ellos *itemset*. Una transacción puede estar formada por uno o varios ítems, en el caso de ser varios, cada posible subconjunto de ellos es un itemset distinto.
Una regla de asociación se define como una implicación del tipo "si X entonces Y" (X⇒Y), donde X e Y son *itemsets* o ítems individuales. El lado izquierdo de la regla recibe el nombre de antecedente o *left-hand-side (LHS*) y el lado derecho el nombre de consecuente o *right-hand-side (RHS)* [3][5].
En este trabajo se utilizaron reglas de asociación a través del algoritmo **Apriori** [7]. Este tiene dos etapas que son: identificar todos los itemsets que ocurren con una frecuencia por encima de un determinado límite (*itemsets* frecuentes) y convertir esos *itemsets* frecuentes en reglas de asociación.
Esta propuesta es una variante de SR basados en contenido donde las recomendaciones dependen de las elecciones de ítems anteriores de los usuarios. Los ítems, en el caso de estudio, serán títulos de series japonesas animadas (animes).

2. Selección de los datos

El dataset es producto de un scrapping hecho al sitio web www.myanimeList.net [5] que proporciona a sus usuarios un sistema basado en listas para organizar y calificar tanto anime como manga. Los datos contienen información sobre los títulos y su audiencia. Apunta a ser una muestra representativa de la comunidad "otaku" para un análisis de tendencias dentro de este grupo. El dataset contiene información sobre usuarios (género, localidad, fecha de nacimiento, etc), sobre títulos anime (fecha de aire, género, productora, etc) y las listas de cada usuarios. Un usuario puede agregar series o mangas a su lista, agregar etiquetas ("plan to watch", "completed", "watching", "dropped") y calificarlas en una escala del 1 al 10. Los datos están compuestos de 3 archivos de texto en formato *Comma Separated Values (CSV)* con las siguientes descripciones:

1. **AnimeList.csv** contiene una lista de anime, con título, sinónimos de título, género, estudio, licencia, productor, duración, calificación, puntaje, fecha de emisión, episodios, fuente (manga, novela ligera, etc.) entre otros.

2. **UserList.csv** contiene información sobre los usuarios que miran anime, a saber, nombre de usuario, fecha de registro (join_date), última fecha en línea, fecha de nacimiento, sexo, ubicación.

3. **UserAnimeList.csv** contiene listas de anime de todos los usuarios. Por cada registro, aquí hay nombre de usuario, ID de anime, puntaje, estado y timestamp cuando se actualizó este registro por última vez.

.3. Preprocesamiento y Transformación de los datos

El dataset de transacciones (UserAnimeList.csv) es el que interesa para el caso de estudio. Este posee en total:

- 32 millones de registros
- 108400 usuarios únicos entre esas transacciones
- 6668 series animadas

Para el objetivo de recomendar se seleccionaron las siguientes columnas :

Tabla 1. Columnas totales, descripción y selección para generar recomendaciones

Columnas iniciales	Descripción	Selección	Mapeo
Username	Nombre del usuario dentro de la plataforma.	SI	usuario_id
animeid	Id del título animé.	SI	anime_id
mywatchedepisodes	Episodios mirados por el usuario.	SI	my_watched_episodes
mystartdate	Fecha de comienzo de visualización del título.	NO	-
myfinishdate	Fecha de finalización de visualización del título.	NO	-
myscore	Calificación del título, provisto por el usuario.	SI	my_score
mystatus	Etiqueta de estado de visualización del título, provisto por el usuario.	SI	my_status
myrewatching	Cantidad de visualizaciones repetidas del título.	NO	-
myrewatchingep	Cantidad de episodios que fueron visualizados de manera repetida.	NO	-
mylastupdated	Fecha de actualización del registro.	NO	-
my_tags	Etiquetas o comentarios provistos por el usuario.	NO	-

Se realizaron operaciones de preprocesamiento con el objetivo de eliminar errores y reducir el volumen de datos. Estas tareas fueron: :

- Se reemplazó el campo username por el id correspondiente al mismo, el cuál obtenemos en *users.csv* (el dataset con información de los usuarios)
- Se reemplazaron los valores null por valores vacíos, ya que nos permite otra flexibilidad a la hora de hacer comparaciones sobre esos valores.
- Se hace un recorte en las transacciones que posean el campo *mystatus* mayor que 4. Ya que estos valores indican que el usuario dejó de mirar una serie o que planea verla. Para estos casos las transacciones no son relevantes.

Como resultado de las operaciones anteriores, el dataset de estudio queda con un poco más de 22 millones de transacciones de 32 millones iniciales. A estas transacciones se le aplica un último filtro donde quedarán solo las que posean títulos con un soporte mayor o igual al 0,02 (es decir, que tengan una cobertura del 2%). Quedando finalmente un total de aproximadamente 4,9 millones de transacciones para la experimentación (Tabla 3).

Tabla 2. Características principales del dataset final

Característica	Cantidad
Registros/Transacciones Totales	4852593
Títulos únicos en las transacciones	150
Usuarios únicos en las transacciones	107.322

Imputación de datos faltantes

A partir de un análisis de la distribución de calificaciones que los usuarios colocan sobre los títulos que aparecen en las transacciones, se observa que gran porcentaje de las mismas poseen calificaciones iguales a “0”. Por lo que se realiza imputación de las calificaciones en “0”, tanto para usuarios que no concluyeron un título como para los que vieron un título completo y no calificaron.

Media ponderada

La estrategia utilizada para imputar es la media ponderada que se calcula de la siguiente manera:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} = \frac{x_1 w_1 + x_2 w_2 + x_3 w_3 + \ldots + x_n w_n}{w_1 + w_2 + w_3 + \ldots + w_n}$$

donde X es una serie de datos numéricos no vacía:

$$X = \{x_1, x_2, x_3 \ldots, x_n\}$$

En nuestro caso serán las calificaciones que tienen las transacciones de un determinado título y donde W corresponde a los pesos, que en nuestro caso está determinado por el porcentaje de episodios vistos de un título en particular.

$$W = \{w_1, w_2, w_3, \ldots, w_n\}$$

En cada transacción donde no haya calificación, se imputa por la media ponderada del título en la transacción y se redondea para que dicho valor ponderado caiga en algunos de los valores del rango de calificación. Esta estrategia de imputación a diferencia de la media geométrica, afecta más a los títulos que fueron puntuados por usuarios de manera prematura, es decir, títulos fueron puntuados antes de que estos vieran la mayor parte de

los episodios. Aunque como mostramos a continuación en la figura 1, la mayor parte de los títulos fueron visualizados en su totalidad.

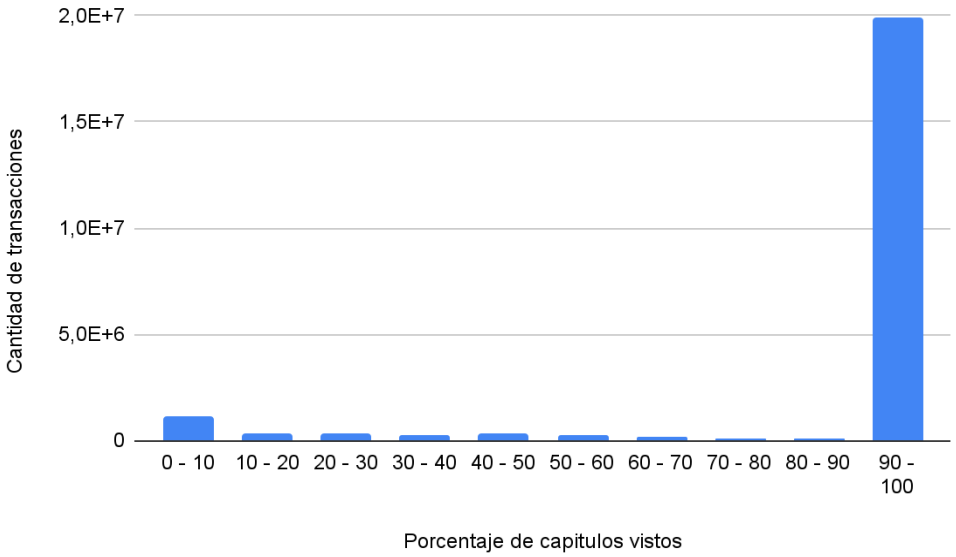


Figura 1. Cantidad de transacciones frente a Porcentaje de capítulos

Todas las imputaciones fueron realizadas antes del recorte de transacciones con soporte mayor a 0,02 mencionado anteriormente. A continuación vemos la distribución de las calificaciones antes y después de las imputaciones (figura 2 y 3).

Figura 2. Distribución de las calificaciones en las transacciones antes de la imputación.

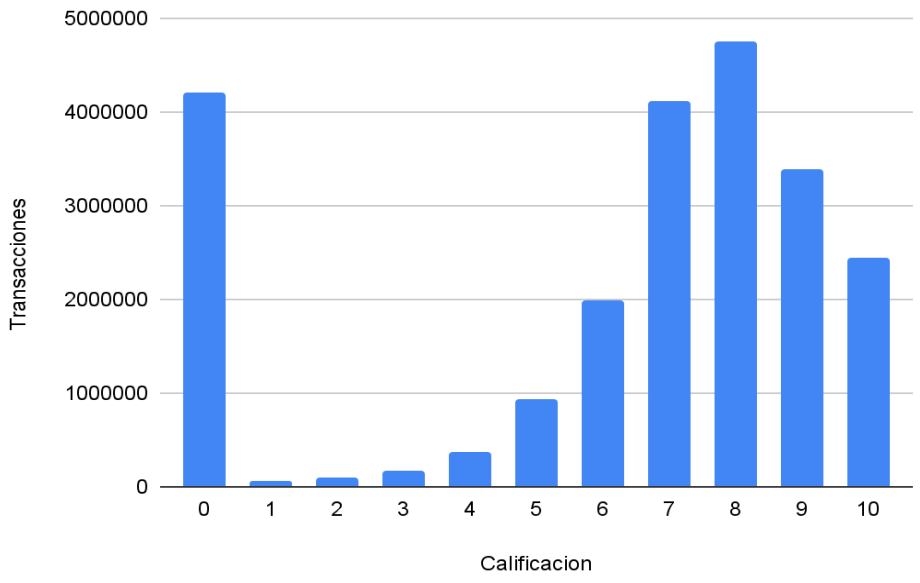
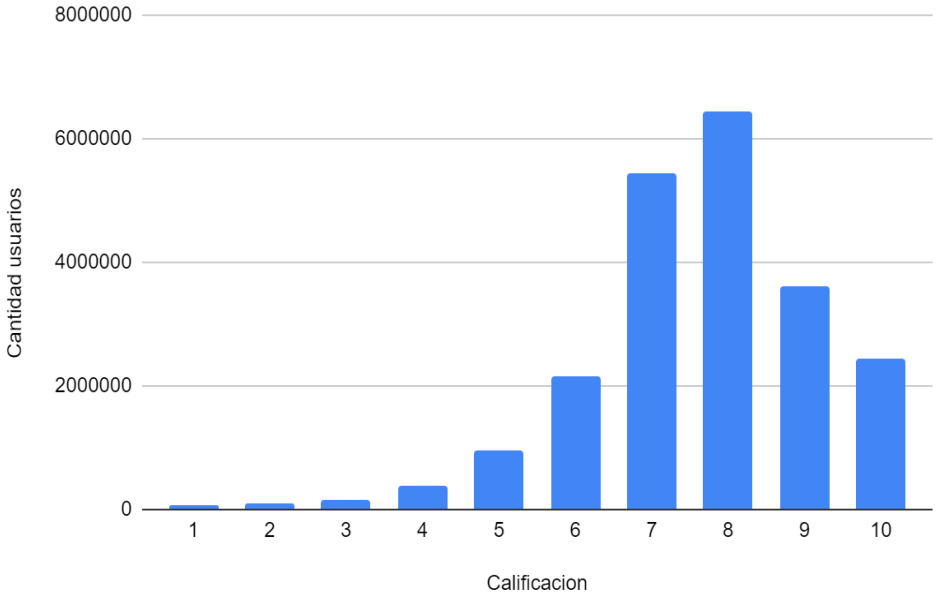


Figura 3. Distribución de las calificaciones en las transacciones después de la imputación.



4. Data Mining

El proceso de generación de reglas de asociación consiste en encontrar, para un conjunto de transacciones, todas las reglas que tengan un soporte y confianza superiores a unos valores mínimos a los que se denomina *min_sup* y *min_conf*, respectivamente. Este proceso puede descomponerse en 2 sub-tareas para facilitar su implementación.

1. Generación de los itemsets frecuentes cuyo objetivo es descubrir todos los *itemsets* que satisfacen el umbral *min_sup*. Estos itemsets descubiertos serán los denominados itemsets frecuentes.

2. Generación de Reglas cuyo objetivo es extraer, a partir de los itemsets frecuentes encontrados en el paso previo, todas las reglas que tengan un grado de confianza alto.

Se utilizó el algoritmo **Apriori.** Este es uno de los primeros algoritmos desarrollados para la búsqueda de reglas de asociación, siendo una herramienta de referencia para el minado de reglas que sigue siendo uno de los más empleados gracias a su sencillez a la hora de ser implementado.

Flujo de trabajo

El siguiente diagrama es un mapa del trabajo realizado y todas sus partes.
A continuación de este se encuentra una explicación de cada proceso.

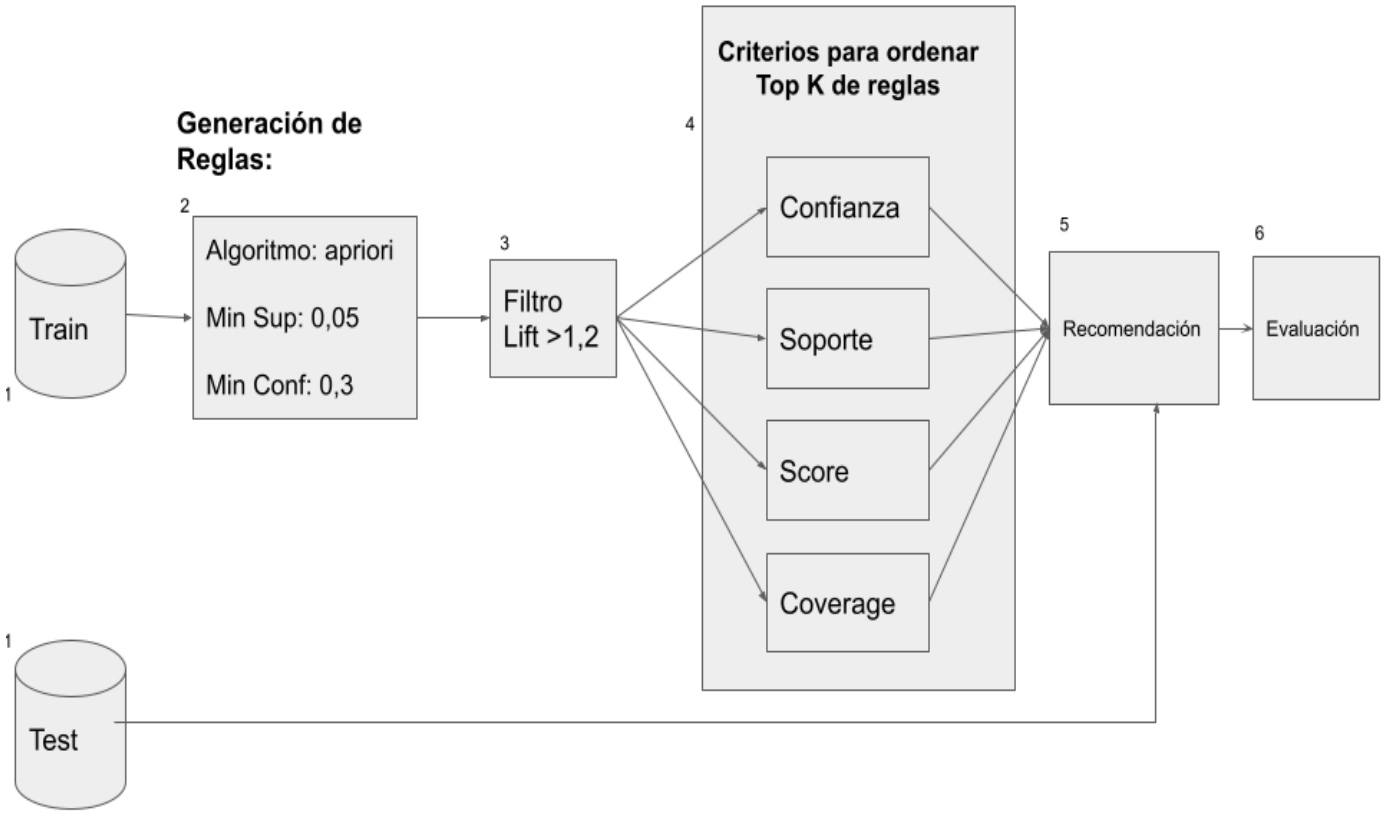


Diagrama 1. Flujo de trabajo y parámetros utilizados.

1. Conjunto de entrenamiento y testing

Se utilizó una estrategia de “hold out” separando en 70% de las transacciones para la generación de reglas y dejando un 30% para la validación (diagrama 2).

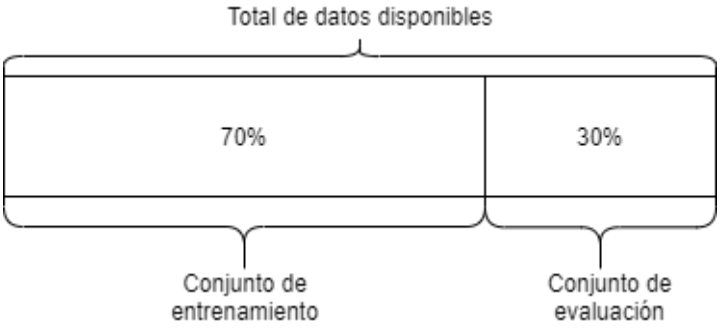


Diagrama 2. distribución de los conjuntos de entrenamiento y test

Conformación del “carrito de compras”

Se seleccionan del conjunto de entrenamiento 30 ítems/transacciones por usuario de manera aleatoria para conformar los “carritos de compra” de entrenamiento. El conjunto de pruebas contiene todas sus transacciones para validar los resultados de la recomendación. Estos son los datos de entrada para el algoritmo implementado, junto al top K de reglas generadas.

2. Generación de reglas

Se corre el algoritmo Apriori sobre los datos de entrenamiento aplicando los umbrales mínimo definidos de confianza y soporte. Para la confianza el umbral mínimo es de 30%, para el soporte el umbral mínimo será de 5%. Se seleccionan las reglas de tamaño 2 de 1 *itemset*, sugerido por [6] para datasets grandes.

3. Filtro Lift

El Lift es la medida que permite determinar si el antecedente de la regla está relacionado con el consecuente de la misma o si estas provienen de ítems que son estadísticamente independientes. La medida está definida como:

$$Lift = \frac{Confianza\ de\ la\ regla}{Proporción\ a\ priori\ del\ consecuente}$$

Lo que es lo mismo decir, que el Lift es la confianza de la regla sobre el soporte del consecuente de la misma. Cuando el Lift sea cercano a 1 implica que la ocurrencia de dos

títulos *A* y *B* son eventos independientes. Esto significa que el conocimiento de la ocurrencia de *A* no altera la probabilidad de la ocurrencia de *B*. Cuanto más se aleje el valor de Lift de 1, mayor evidencia de que la regla no se debe a un suceso aleatorio, es decir, mayor evidencia de que la regla representa un patrón real. Por lo tanto, las relaciones que interesan están donde el Lift sea bien distinto de 1. En nuestro caso, un valor de Lift alejado de 1 asegura que el contenido presente en el antecedente tiene una buena relación con el consecuente y no se trata de títulos que se asociaron de manera azarosa, sino que existe una relación.
Finalmente, este paso de aplicar Lift consiste en generar un filtro de reglas con valores mayores a 1,2. Este valor es empírico y garantiza que existe dicha relación entre los ítems, y que no existe independencia en estos. Este valor podría calibrarse en futuros trabajos.

4. Criterio para ordenar el Top K de reglas

Hay 4 escenarios para ordenar las reglas que obtuvimos en el paso anterior. También se eligen distintos valores de K (5, 10 y 20) para evaluar los resultados que serán mostrados más adelante. Los valores de K fueron seleccionados empíricamente ya que aseguran los mejores resultados, mientras que los K mayores a 20 aportan malos resultados.

Los 4 criterios para ordenar las reglas son:

- Ranking por confianza reglas
- Ranking por Soporte de las reglas
- Ranking por Score (calificación) del consecuente de las reglas
- Ranking por Coverage de antecedente de las reglas

5. Algoritmo de recomendación

Para realizar la recomendación hay 4 escenarios donde vamos a recibir un top k de reglas distintas ordenado por los criterios descritos anteriormente. También se recibe como parámetro de entrada el conjunto de transacciones de prueba descrito en el paso 1. El algoritmo busca los antecedentes de las K reglas que existen en las transacciones de un usuario determinado. Luego, busca si ese usuario tiene el consecuente de cada antecedente encontrado generando la recomendación.

6. Evaluación

Por último, se hace el cálculo de recall y precisión para cada recomendación. Verificando cuántos consecuentes de las reglas de asociación estuvieron dentro de las transacciones de cada usuario, es decir, si el usuario vió el título que el sistema recomienda. En base a estos resultados podemos ver la certeza del recomendador. Las métricas elegidas fueron: *precisión, recall* y *F-Score*. Se ejecutan las recomendaciones por cada usuario del conjunto de prueba y se calculan los valores acumulados.

5. Evaluación y análisis de resultados

A continuación se listan los resultados obtenidos. Se divide por enfoque a la hora de ordenar el Top K de reglas:

Criterio	Cant. reglas	Top K	Promedio precisión	Promedio recall	F-Score
Confianza	1085	5	0,1830	0,0328	0,0557
		10	0,2184	0,0659	0,1013
		20	0,1950	0,1264	0,1534
Score	1085	5	0,2869	0,0538	0,0906
		10	0,2598	0,1144	0,1588
		20	0,1911	0,1707	0,1803
Support	1085	5	0,3877	0,0701	0,1187
		10	0,2852	0,1361	0,1843
		20	0,1950	0,1264	0,1534
Coverage	1085	5	0,2875	0,0408	0,0715
		10	0,2033	0,0665	0,1002
		20	0,1948	0,1678	0,1803

Tabla 4. Comparación de los mejores criterios y parámetros elegidos.

En todos los casos los valores del soporte mínimo y confianza mínima son iguales a los que se ven en el diagrama 1. Los valores de K para armar los rankings de reglas fueron los N que aportaron valores más interesantes teniendo en cuenta como principal indicador la precisión y el recall de dichas recomendaciones.
En las próximas figuras podemos ver una comparación más detallada de cada criterio de ordenamiento de reglas y que valores de precisión y recall nos fueron entregando:

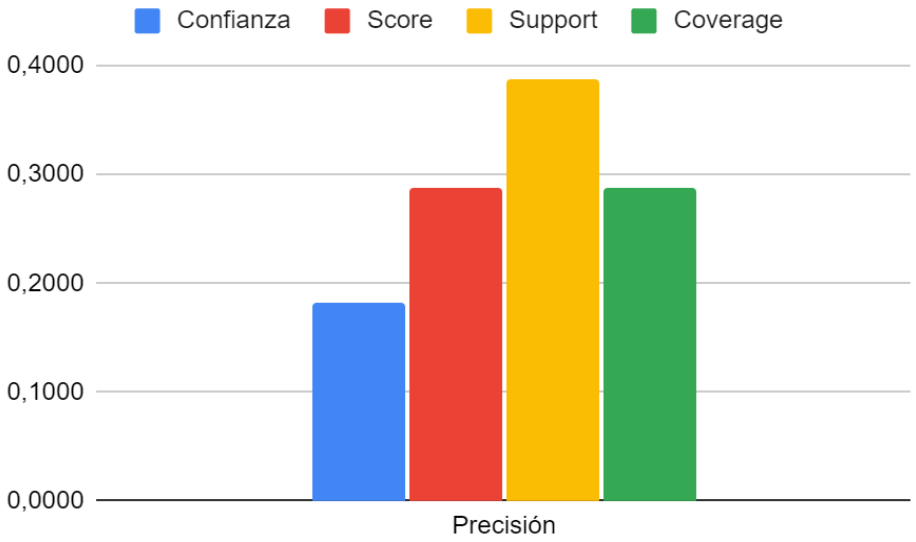


Figura 5. Valores de la precisión con un top K de reglas igual a 5

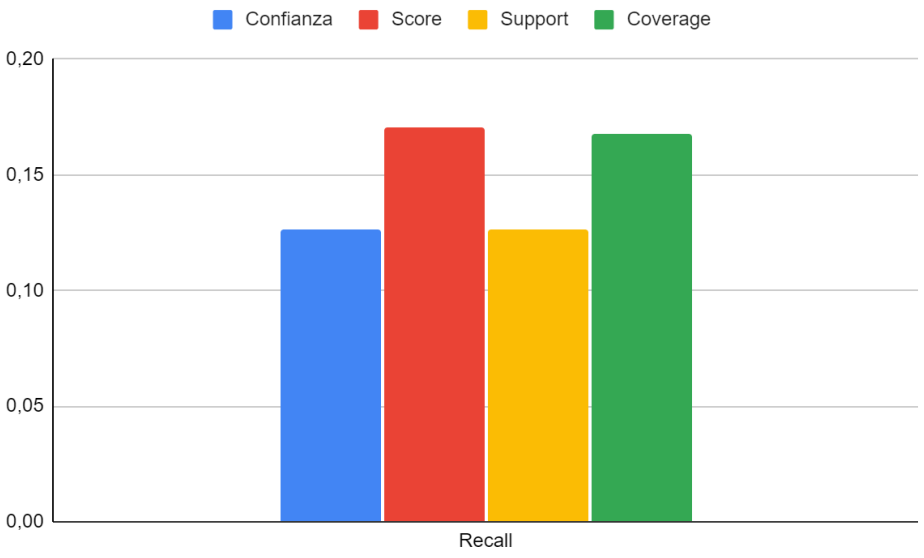


Figura 6. Valores de recall con un top K de reglas igual a 20

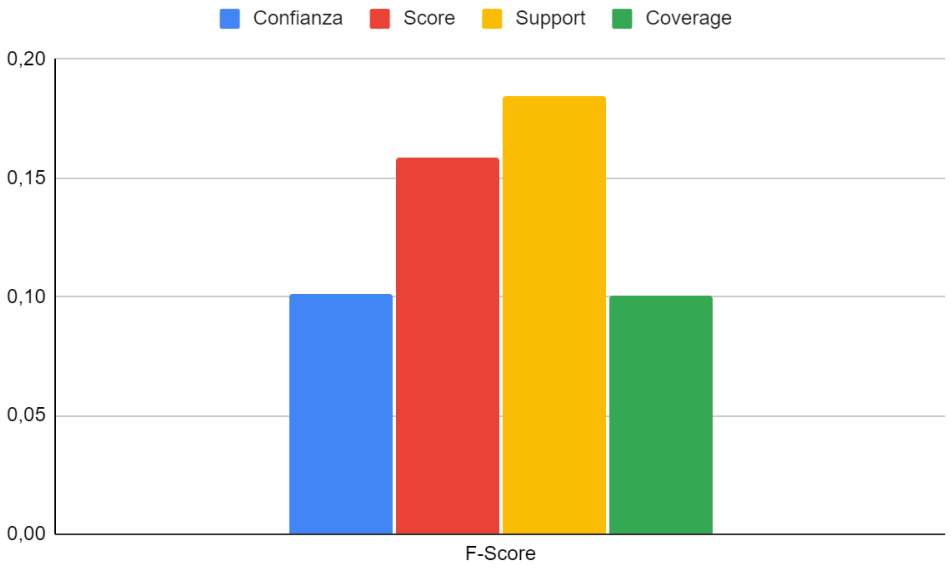


Figura 7. Valores del F-Score con un top K de reglas igual a 10

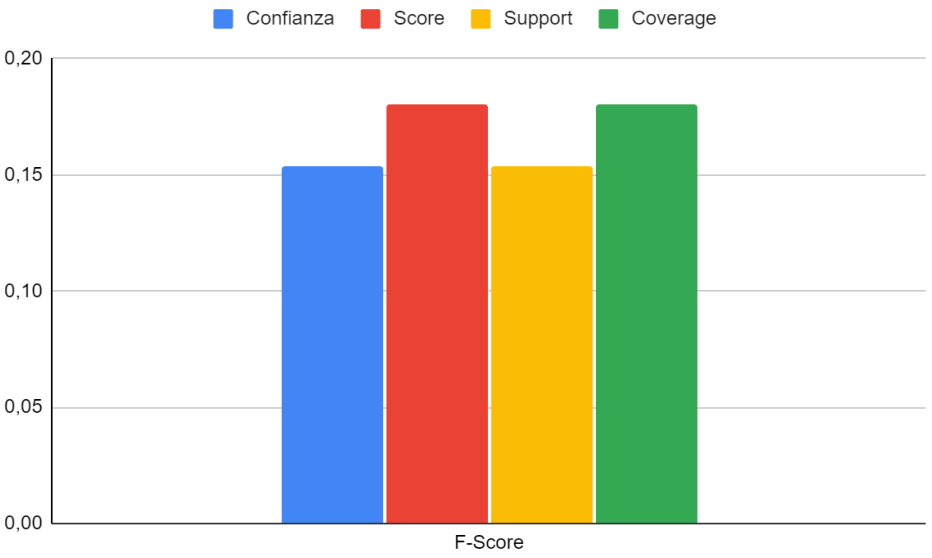


Figura 8. Valores del F-Score con un top K de reglas igual a 20

6. Conclusiones

Como conclusión general del trabajo un algoritmo de recomendación utilizando reglas de asociación ofrece resultados aceptables. Por esto, se puede afirmar que utilizar una técnica como reglas de asociación para generar modelos de recomendación resulta ser efectiva, ya que en el mejor escenario posible se puede logran recomendaciones que aciertan casi un 40% de las veces.
Si nos enfocamos en los criterios para generar los Top K de reglas, se podría afirmar que soporte, coverage y score son criterios parecidos en cuanto a resultados, pero varían un poco dependiendo el tamaño del K elegido. En cambio la confianza no parece ser buena como criterio para ordenar reglas y generar recomendaciones.
Otro punto importante es que durante la experimentación, hubo un cambio significativo en los resultados cuando se agregó el filtro por Lift. Esto redujo las reglas de asociación considerablemente dejando las más interesantes para recomendar, incrementando los valores de precisión generales en cada escenario.

7. Trabajos futuros

Al finalizar este trabajo surgen nuevas preguntas y nuevos experimentos que podrían realizarse. A continuación, se detalla una lista de posibles trabajos o ideas que pueden realizarse en el futuro:

- Crear un sistema de recomendación basado en filtros colaborativos, utilizando los datos de los usuarios (con el mismo dataset) y luego comparar su performance contra los resultados obtenidos en este trabajo.
- Utilizar otro criterio de orden para el *top K* de reglas. Tal vez alguno mixto que utilice, por ejemplo, el score y el soporte de las reglas.
- Utilizar otros criterios donde se tenga en cuenta la cantidad de títulos que un usuarios volvió a ver y la duración en capítulos de esos títulos como parte del *scoring.*

8. Referencias

[1] B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. International Journal of Computer Applications, 110(4), 31–36. doi:10.5120/19308-0760

[2] Uman, I. (2018). El efecto Netflix: cómo los sistemas de recomendación transforman las prácticas de consumo cultural y la industria de contenidos. *Cuadernos de Comunicólogos*, *27*.

[3] Joaquín Amat Rodrigo (Junio, 2018). Reglas de asociación y algoritmo Apriori con R.

[4] Lin, W. (2000). Association rule mining for collaborative recommender systems (Master's thesis, Worcester Polytechnic Institute.).

[5] Jiawei Han, Micheline Kamber, Jian Pei. 2012. Tercera edición. Data Mining: Concepts and Techniques. Cap. 2 y Cap. 3

[6] Osadchiy, T., Poliakov, I., Olivier, P., Rowland, M., & Foster, E. (2019). Recommender system based on pairwise association rules. Expert Systems with Applications, 115, 535-542.

[7] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast discovery of association rules. Advances in knowledge discovery and data mining, 12(1), 307-328.