

Integrantes:

- Cortés Macías Gretel Penélope 317312184
- Velázquez Barrón Marilú Yatzael 318353492
- Peña Nuñez Axel Yael 318279754
- Escalante Castañeda Lenin Alberto 420003193

Actividades:

- Explica con tus propias palabras los algoritmos Apriori y Eclat.

Explicación de los Algoritmos Apriori y Eclat:**Apriori:**

Explicación: El algoritmo Apriori es utilizado para la minería de reglas de asociación en bases de datos transaccionales. Se basa en el principio Apriori, que establece que un conjunto de elementos es frecuente si y solo si todos sus subconjuntos también son frecuentes. El algoritmo opera en dos fases: encuentra itemsets frecuentes y luego genera reglas de asociación fuertes a partir de estos itemsets.

Proceso:

- Comienza con itemsets de tamaño 1 y los extiende para formar itemsets más grandes.
- Utiliza el soporte mínimo para filtrar itemsets infrecuentes.
- Genera reglas de asociación a partir de itemsets frecuentes, aplicando un umbral de confianza.

Eclat:

Explicación: El algoritmo Eclat es otra técnica para la minería de reglas de asociación. A diferencia de Apriori, Eclat trabaja con una representación vertical de la base de datos, donde cada registro tiene un item seguido de los identificadores de las transacciones en las que aparece. Utiliza operaciones de intersección para encontrar itemsets frecuentes de manera eficiente.

Proceso:

- Comienza con itemsets de tamaño 1 y los combina utilizando operaciones de intersección.
 - Utiliza el soporte mínimo para filtrar itemsets infrecuentes.
 - Genera reglas de asociación de manera similar a Apriori.
- Si el espacio de todas las reglas de asociación es exponencial, es decir, $O(m)$. Donde m es el número de elementos. En caso de ser cierto, ¿a qué se debe dicha complejidad?

Complejidad Exponencial:

La complejidad exponencial $O(m)$ en el espacio de todas las reglas de asociación se debe al crecimiento exponencial del número de posibles subconjuntos de elementos. Para un conjunto de m elementos, el número total de subconjuntos posibles es 2^m , lo que lleva a una complejidad exponencial. Este crecimiento se debe a que cada elemento puede estar presente o ausente en un subconjunto, generando todas las combinaciones posibles.

- ¿Qué implica que tengas $Lift(A \rightarrow X) = 0$?

$Lift(A \rightarrow X) = 0$, indica que el modelo no está proporcionando ninguna mejora en la predicción en comparación con hacer predicciones al azar. Es decir, la tasa de éxito del modelo para esa regla es igual a la tasa de éxito esperada al azar. En términos prácticos, esto sugiere que la regla específica no está siendo útil para hacer predicciones en el conjunto de datos dado.

TID	Lista de elementos
100	1, 2, 4
101	2, 4, 5
102	2, 3, 4
103	1
104	1, 2, 3
105	2, 3, 5
106	1, 3, 4
107	2, 3, 5
108	2, 3

- Si el soporte mínimo es de 2. ¿Cuál es el porcentaje?

Porcentaje de Soporte Mínimo de 2: Para calcular el porcentaje de soporte, necesitamos determinar cuántas transacciones contienen conjuntos de elementos frecuentes que cumplan con el soporte mínimo. En este caso, si el soporte mínimo es de 2, debemos contar cuántas transacciones tienen al menos 2 elementos en común.

De las transacciones proporcionadas, las siguientes cumplen con el criterio de soporte mínimo de 2: -

TID 100: 11, 12, 14

- TID 101: 12, 14, 15

- TID 102: 12, 13, 14

- TID 104: 11, 12, 13

- TID 105: 12, 13, 15

- TID 106: 11, 13, 14

- TID 107: 12, 13, 15

- TID 108: 12, 13

Por lo tanto, el porcentaje de soporte es (8 transacciones con conjuntos de elementos frecuentes) / (9 transacciones totales) * 100 = 88.88%.

- Si la confianza requerida es del 70%, debes encontrar los conjuntos de elementos frecuentes utilizando Apriori. Luego genera las reglas fuertes de asociación utilizando el soporte mínimo y la confianza mínima.

Generación de Reglas con Apriori (Soporte Mínimo: 2, Confianza Mínima: 70%):

1. Identificar los conjuntos de elementos frecuentes que cumplen con el soporte mínimo especificado de 2. Los conjuntos de elementos frecuentes encontrados fueron los siguientes:

- {12, 13}: 5 transacciones

- {12, 15}: 3 transacciones

- {12, 14}: 3 transacciones

- {11, 12}: 2 transacciones

- {11, 13}: 2 transacciones

- {11, 14}: 2 transacciones

- {13, 14}: 2 transacciones

- {13, 15}: 2 transacciones

- {12, 13, 15} : 2 transacciones

2. Generar todas las posibles reglas de asociación a partir de los conjuntos de elementos frecuentes obtenidos en el paso anterior.

3. Calculamos la confianza de cada regla y se seleccionaron aquellas que cumplieran con la confianza mínima requerida del 70%.

$\{13\} \rightarrow \{12\}$ Confianza: 83.33 %, Soporte: 55.55 %
 $\{14\} \rightarrow \{12\}$ Confianza: 75.00 %, Soporte: 33.33 %
 $\{12\} \rightarrow \{13\}$ Confianza: 71.42 %, Soporte: 55.55 %
 $\{12\} \rightarrow \{14\}$ Confianza: 42.85 %, Soporte: 33.33 %
 $\{12\} \rightarrow \{15\}$ Confianza: 42.86 %, Soporte: 33.33 %
 $\{13\} \rightarrow \{14\}$ Confianza: 33.33 %, Soporte: 22.22 %
 $\{13\} \rightarrow \{15\}$ Confianza: 33.33 %, Soporte: 22.22 %
 $\{13, 15\} \rightarrow \{12\}$ Confianza: 100.00 %, Soporte: 22.22 %
 $\{15\} \rightarrow \{12\}$ Confianza: 100.00 %, Soporte: 33.33 %

Las reglas fuertes de asociación obtenidas fueron las siguientes:

$\{14\} \rightarrow \{12\}$ confianza: 75.00 %, soporte: 33.33 %
 $\{15\} \rightarrow \{12\}$ confianza: 100.00 %, soporte: 33.33 %
 $\{13, 15\} \rightarrow 12$ confianza: 100.00%, soporte: 22.22%
 $\{12\} \rightarrow \{13\}$ confianza: 71.42%, soporte: 55.55%
 $\{13\} \rightarrow \{12\}$ confianza: 83.33%, soporte: 55.55%
 las demás nos dieron una confianza menor a 70%

Dado la siguiente tabla, donde se puede observar la información de algunas transacciones en formato horizontal. Utilizando Equivalence Class Transformation (ECLAT) realiza lo siguiente:

TID	Lista de elementos
T1	4, 5
T2	2, 3, 4, 5
T3	1, 3, 5
T4	3, 4
T5	1, 4
T6	4, 5
T7	2, 3, 5
T8	2, 5
T9	3, 4, 5

- Proporciona una tabla con 3 columnas donde identifiques los items (itemSet con $k = 1$) que aparecen en el conjunto de transacciones y calcula su soporte.

Vamos a realizar cada uno de los pasos solicitados utilizando Equivalence Class Transformation (ECLAT).

Considerando que el soporte mínimo es de 20%.

Paso 1: Identificación de los items (itemSet con $k=1$) y cálculo de su soporte.

TID	Item	Soporte
T3, T5	l1	22.22%
T2, T7, T8	l2	33.33%
T2, T3, T4, T7, T9	l3	55.55%
T1, T2, T4, T5, T6, T9	l4	66.66%
T1, T2, T3, T6, T7, T8, T9	l5	77.77%

- Proporciona todas las posibles intersecciones de la columna Transacciones $k = 1$ donde obtenemos los itemsets de longitud $k + 1$.

Paso 2: Posibles intersecciones de la columna Transacciones $K=1$ para obtener itemsets de longitud

$k+1$.

TID	ItemSet ($k=2$)	Soporte
T3	l1, l3	11%
T5	l1, l4	11%
T3	l1, l5	11%
T2, T7	l2, l3	22%
T2	l2, l4	11%
T2, T7, T8	l2, l5	33%
T2, T4, T9	l3, l4	33%
T2, T3, T7, T9	l3, l5	44%
T1, T2, T6, T9	l4, l5	44%

y para recordar que nuestro soporte mínimo es: 20%.

TID	ItemSet ($k=2$)	Soporte
T2, T7	l2, l3	22%
T2, T7, T8	l2, l5	33%
T2, T4, T9	l3, l4	33%
T2, T3, T7, T9	l3, l5	44%
T1, T2, T6, T9	l4, l5	44%

- Proporciona las intersecciones de la tabla anterior (itemSet $k = 2$) obteniendo los (itemset $k = 3$).
Hint: Piensa en el principio de downward closure te puede ahorrar trabajo.
Paso 3: Intersecciones de la tabla anterior (itemSet $k=2$) para obtener itemsets de longitud $k+1$ (itemset $k=3$).

TID	ItemSet ($k=3$)	Soporte
T2, T7	l2, l3, l4	22.22%
T2, T4, T9	l3, l4, l5	33.33%

- **¿En qué punto finaliza el algoritmo de ECLAT?**

El algoritmo de ECLAT finaliza cuando ya no puede generar más itemsets frecuentes o cuando se alcanza el umbral de soporte mínimo establecido. En cada iteración, se generan itemsets de mayor longitud a partir de la información recopilada en las iteraciones anteriores, y el proceso continúa hasta que ya no se pueden formar itemsets frecuentes.

- **El algoritmo permite una identificación de itemSets frecuentes, pero no produce reglas de asociación. ¿Qué propones para concluir nuestra tarea?**

Para concluir la tarea y obtener reglas de asociación a partir de los itemsets frecuentes identificados por ECLAT, se puede utilizar un paso adicional, como el que se realiza en el algoritmo Apriori. Este paso implica la generación de reglas a partir de los itemsets frecuentes y la aplicación de umbrales de confianza para seleccionar las reglas más fuertes.

- **Menciona las principales diferencias entre el algoritmo Apriori y ECLAT.**

Representación de Datos:

Apriori: Utiliza una representación horizontal de la base de datos, donde cada transacción es una fila con los elementos que contiene.

ECLAT: Trabaja con una representación vertical de la base de datos, donde cada elemento tiene una lista de transacciones en las que aparece.

Generación de Itemsets:

Apriori: Genera itemsets candidatos de una longitud mayor en cada iteración, explorando gradualmente conjuntos más grandes.

ECLAT: Utiliza la técnica de Equivalence Class Transformation para generar itemsets frecuentes a partir de los itemsets de longitud menor.

Complejidad Computacional:

Apriori: Puede requerir múltiples pasadas sobre la base de datos, lo que podría ser costoso computacionalmente.

ECLAT: Generalmente es más eficiente, ya que trabaja con la representación vertical y utiliza una estrategia de búsqueda basada en conjuntos de ítems.

Reglas de Asociación:

Apriori: Tiene un paso adicional para generar reglas de asociación a partir de los itemsets frecuentes.

ECLAT: Se centra en la identificación de itemsets frecuentes y puede necesitar un paso adicional para obtener reglas de asociación.