Universidad Juárez Autónoma de Tabasco

Tesis de Doctorado

Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana

Que presenta

Freddy de la Cruz Ruiz

Para obtener el grado de

Doctor en Ciencias de la Computación

Directora

Dra. Juana Canul Reich

Cuerpo Académico

Inteligencia Artificial

Línea de Generación y Aplicación del Conocimiento

Ciencia de Datos y Optimización

Cunduacán, Tabasco, México Marzo 2023

Universidad Juárez Autónoma de Tabasco

Tesis de Doctorado

Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana

Que presenta

Freddy de la Cruz Ruiz

Para obtener el grado de

Doctor en Ciencias de la Computación

Directora

Dra. Juana Canul Reich

Jurado: **Dr. Erick de la Cruz Hernández**
Dr. Rafael Rivera López
Dr. Oscar Alberto Chávez Bosquez
Dr. José Hernández Torruco
Dra. Betania Hernández Ocaña
Dra. Cristina López Ramírez

Cuerpo Académico

Inteligencia Artificial

Línea de Generación y Aplicación del Conocimiento

Ciencia de Datos y Optimización

Cunduacán, Tabasco, México Marzo 2023

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

F5: Liberación de dirección de tesis

Cunduacán, Tabasco., a 01 de febrero de 2023.

MTE. Oscar Alberto González González
Director de la División Académica de Ciencias y Tecnologías de la Información
Presente

Por medio de la presente me permito comunicarle que después de haber concluido la dirección de la Tesis: "Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana.", elaborada por el ***C. Freddy de la Cruz Ruiz***, del Doctorado en Ciencias de la Computación, considero que puede continuar con los trámites para la obtención del grado.

Sin otro particular, aprovecho la ocasión para enviarle un cordial saludo.

A t e n t a m e n t e

Dra. Juana Canul Reich

RECIBIDO
- 1 FEB. 2023
D.A.C.Y.T.I.
DIRECCIÓN

c.c.p. Dr. Eddy Arquímedes Gracia Alcocer. Encargada del despacho de la Coordinación de Posgrado.
Director de Tesis.
Estudiante

01/feb/2023

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

F6: Respuesta de jurado

Cunduacán, Tabasco, a 08 de marzo de 2023.

MTE. Óscar Alberto González
Director de la División Académica de Ciencias y Tecnologías de la Información
Presente

En atención a los oficios girados por usted, en los que se nos designa como parte del jurado para efectuar la revisión de la tesis titulada ***"Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana"***, realizada por el ***C. Freddy de la Cruz Ruiz***, estudiante del Doctorado en Ciencia de la Computación, nos permitimos informarle que, en virtud de que ha atendido las observaciones realizadas, otorgamos nuestra aprobación para que continúe los trámites para la obtención del grado.

Sin otro particular, aprovechamos la ocasión para enviarle un cordial saludo.

Atentamente integrantes del jurado

Dra. Juana Canul Reich

Dra. Betania Hernández Ocaña

Dr. Erick de la Cruz Hernández

Dr. Rafael Rivera López

Dr. Oscar Alberto Chávez Bósquez

Dra. Cristina López Ramírez

Dr. José Hernández Torruco

c.c.p. Dr. Eddy Arquímedes Garcia Alcocer. Encargada del despacho de la Coordinación de Posgrado.
Estudiante.

RECIBIDO
13 MAR. 2023
D.A.C.Y.T.I.
DIRECCIÓN

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

DIVISIÓN ACADÉMICA DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

Cunduacán, Tabasco, a 13 de marzo de 2023.

Asunto: Cesión de derechos

MTE. Óscar Alberto González González
Director de la División Académica de Ciencias y Tecnologías de la Información
Presente

Los que suscriben la presente, declaramos que el trabajo de tesis titulado, "***Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana.***" es de nuestra autoría intelectual y por lo tanto cedemos los derechos de comunicación pública, reproducción, distribución, difusión en general y puesta a disposición electrónica de la citada tesis doctoral, de forma gratuita y no exclusiva, a la Universidad Juárez Autónoma de Tabasco, a la cual relevamos de cualquier sanción y asumimos responder a cualquier reclamo de derechos de autor ante las autoridades competentes.

Sin otro particular, aprovechamos la ocasión para enviarle un cordial saludo.

Atentamente

Dra. Juana Canul Reich

M. en C. B.B. Freddy de la Cruz Ruiz

c.c.p. Dr. Eddy Arquímedes Garcia Alcocer. Encargada del despacho de la Coordinación de Posgrado.
Estudiante.

Cunduacán, Tabasco a 15 de marzo de 2023

Oficio No. 0444/DACYTI/CP/2023

Asunto: Autorización de impresión de Tesis

C. Freddy de la Cruz Ruiz
Matricula: 191H18005

En virtud de que cumple satisfactoriamente los requisitos establecidos en el Reglamento General de Estudios de Posgrado vigente en la Universidad, informo a Usted que se autoriza la impresión del trabajo recepcional **"Derivación de reglas de asociación para identificar bacterias coexistentes detonantes de la Vaginosis Bacteriana"**, para presentar examen y obtener el Grado de Doctor en Ciencias de la Computación.

Sin otro particular, aprovecho la ocasión para enviarle un afectuoso saludo.

Atentamente

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

MTE. Óscar Alberto González González
Director

DIVISIÓN ACADÉMICA DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN

C.c.p. Dr. Eddy Arquímedes García Alcocer. - Encargado del Despacho de la Coordinación de Posgrado DACYTI
Archivo.
Consecutivo.
MTE´OAGG/EAGA

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda. C.P. 86690.
Cunduacán, Tabasco. México.
Tel: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870
E-mail: direccion.dacyti@ujat.mx

Agradecimientos

Primero que nada a Dios por haberme dado la capacidad intelectual para realizar estos estudios de posgrado.

A los profesores de DACyTI por la oportunidad de aceptarme en su programa de doctorado aunque soy del área de ciencias de la salud.

A mi amigo Edgar Herrera Ramon Morgado por ayudarme a resolver los problemas técnicos con el SO GNU/Linux distro Open SuSe. Es notable su amplio conocimiento con este SO a pesar de que no tiene una licenciatura en informática. Sin su ayuda no hubiera podido instalar la función fim4r diseñada para windows y solo probada para Linux Mint.

Del mismo modo a mi amigo el profesor Reinerio Zapata Salazar de DAMC por la disposición a ayudarme siempre a resolver los problemas de dependencias y compilación de binarios de los paquetes de R usados en esta investigación.

A mi amigo y compañero de doctorado Henry que me apoyo en innumerables ocasiones cuando tenia alguna dificultad.

A mi familia que soportó mis ausencias por estar atendiendo este programa de estudio.

Publicaciones

de la Cruz Ruiz, F., & Canul-Reich, J. (2022). Reglas de asociación para el estudio de la vaginosis bacteriana.

de la Cruz-Ruiz, F., Canul-Reichb, J., de la Cruz-Hernándezc, E., & Rivera-Lopezd, R. (2022). Analysis of Bacterial Association Patterns that trigger Bacterial Vaginosis. *International Journal of Combinatorial Optimization Problems & Informatics*, *13*(4).

de la Cruz-Ruiz, F., Canul-Reich, J., de la Cruz-Hernándezc, E., & Rivera-Lopezd, R. (2023). Impact of data balancing a multiclass dataset before the creation of association rules to study bacterial vaginosis. Intelligent Medicine.

de la Cruz-Ruiz, F., Canul-Reich, J. Apriori, Eclat, and FP-Growth Algorithms to Study Bacterial Baginosis. In the 5th International Conference on Communication and Computational Technologies held during January 28-29 2023 in India.

Índice general

Índice de figuras

Índice de figuras

Índice de tablas

Resumen

La vaginosis bacteriana es una condición clínica causada por un desequilibrio en la comunidad de ***Lactobacillus*** protectores de esta mucosa. Se manifiesta en forma de síndrome polimicrobiano en el cual la producción de peróxido de hidrógeno H_2O_2 y ácido láctico deja de producirse.
Esta condición clínica se caracteriza por leucorrea adherente y homogénea, irritación vaginal y olor vaginal a pescado.
Debido a que las técnicas de diagnóstico clásicas son subjetivas en esta investigación se aborda el estudio de la vaginosis con las técnicas de aprendizaje automático. Dentro de estas técnicas, las reglas de asociación modelan las relaciones entre las bacterias que desencadenan la condición clínica.
Los experimentos consistieron en investigar los porcentajes de soporte y confianza para que el algoritmo Apriori cree las reglas de asociación. Para realizar la validación estadística se investigaron 8 métricas de calidad. El conjunto de reglas reportado por estas métricas fue validado por el experto en Biología para determinar que los patrones tengan significancia biológica.
Durante el diagnóstico clínico de las enfermedades es común observar un mayor número de casos negativos con respecto a los casos positivos para dicha enfermedad. Debido al sesgo en los datos hacia el diagnóstico negativo de la enfermedad, el conjunto de datos en estudio se balanceó para investigar si la creación de reglas mejora en su rendimiento y los patrones reportados por los algoritmos continúan siendo biológicamente significativos. Los resultados mostraron que los patrones creados con el conjunto de datos balanceados son mejores con respecto a los valores reportados por las métricas de calidad y clínicamente, en comparación con los patrones creados con los datos no balanceados.
También se crearon las reglas con los algoritmos: Eclat y FP-Growth. Los algoritmos con mayor rendimiento fueron Eclat y FP-Growth. El algoritmo Eclat con respecto al número de reglas presentadas y el FP-Growth con

respecto a la calidad de las reglas. Los resultados obtenidos con las reglas reportadas fueron estadísticamente significativas y biológicamente acorde con lo que se reporta en la clínica.

Capítulo 1

Generalidades

En esta sección se introduce al problema que la vaginosis bacteriana representa para la paciente. Las técnicas de diagnóstico y principalmente se introduce el algoritmo de reglas de asociación como técnica del aprendizaje automático para modelar las bacterias que interactúan para desarrollar vaginosis bacteriana. Del mismo modo se realiza el planteamiento del problema que representa la vaginosis y se delimitan los alcance de esta investigación.

1.1. Introducción

La vaginosis bacteriana (VB) es una condición clínica causada por un desequilibrio en la comunidad de ***Lactobacillus***, entre ellos ***L. jenseni***, ***L. gasseri***, ***L. crispatus*** y ***L. inners*** [1]. Esta condisión clínica se manifiesta en forma de síndrome polimicrobiano en el cual la producción de peróxido de hidrogeno H_2O_2 y ácido láctico se ve alterada por el reemplazo de una gran variedad de bacterias anaerobias y mycoplasmas [2]. Esta condición clínica se caracteriza por leucorrea adherente y homogénea, irritación vaginal y olor vaginal a pescado que es consecuencia de las diaminas tales como putresina, cadaverina y trimetilamina [3, 5]; elevado pH vaginal (>4.5) y la presencia de células clave (células del epitelio escamoso con bacterias adheridas).
Las bacterias asociadas a vaginosis bacteriana están presentes tanto en el estado de salud como de enfermedad. En el estado de salud estas bacterias están en una densidad de crecimiento muy bajo y esta caracteristica es regulada por ***Lactobacillus crispatus*** que esta en una densidad de crecimiento alto y por lo tanto protege a esta mucosa. ***Lactobacillus crispatus*** deja de

proteger a la mucosa vaginal y por lo tanto las bacterias asociadas a vaginosis bacteriana crecen en una densidad de crecimiento alto. De igual manera, otros lactobacilos, tales como ***L. inners*** tienen en una densidad de crecimiento alto y esta bacteria se asocia a flora vaginal alterada.
El diagnóstico de la vaginosis bacteriana suele ser complicado puesto que son múltiples especies las que están asociadas a la condición clínica. No obstante, se diagnostica usando los criterios de Amsel o la escala de Nugent. Sin embargo, estas pruebas tienen baja especificidad debido a la subjetividad de los criterios de evaluación [6]. El diagnóstico a partir de pruebas moleculares como la qPCR tiempo real está ganando terreno frente a las pruebas ya mencionadas [7, 8]. Esto se basa en la detección y evaluación cuantitativa de secuencias de ADN específicas en las que los cebadores especifícos para cada especie o género flanquean la región de interés que suele ser el gen del rRNA 16 S [10, 11].
Por otra parte, se han implementando las técnicas de aprendizaje automático (Machine Learning) para explorar las interacciones entre los microorganismos y su entorno en el hospedero con el objetivo de comprender el papel del microbioma en la salud y enfermedad. En cuanto a la vaginosis bacteriana los métodos del aprendizaje automático están demostrando su capacidad para modelar las complejas relaciones entre las comunidades microbianas detonantes de esta infección polimicrobiana [12, 13, 15, 16].
Las reglas de asociación, una de las técnicas del aprendizaje automático, también se están usando para resolver diferentes problemas de las áreas biológicas y de la salud [17, 18]. Una regla de asociación es una implicación que tiene la siguiente estructura: si X entonces Y donde a la X se le denomina antecedente (LHS) y a la Y se le denomina consecuente (RHS) [19].
Para crear las reglas de asociación los algoritmos Apriori, Eclat y FP-Growth buscan los conjuntos de elementos frecuentes en la primera etapa y en la segunda etapa crean las reglas de asociación [36]. Estos algoritmos necesitan dos métricas basicas como directrices durante la creación de reglas: el soporte [30, 22] y la confianza [30, 22]. Estos algoritmos reportan los conjuntos de reglas de asociación, y por lo tanto es imporante seleccionar las reglas que son de interés. Para realizar esa tarea se utilizan por una parte métricas de calidad [22, 23, 24, 25, 26, 27] y por otra, funciones que proporciona el paquete `Arules` [19, 25, 28]. Como resultado se tienen los modelos en forma de reglas de asociación significativos desde el punto de vista computacional. La última tarea del proceso de creación de reglas de asociación es determinar la significancia biológica. Esta tarea la realiza el experto en biología con la in-

tención de determinar si los modelos en forma de regla representan patrones acordes con lo que se observa en la clínica.

1.2. Planteamiento del problema

1.2.1. Definición del problema

La vaginosis bacteriana se caracteriza por la disbiosis de la flora normal protectora de la mucosa vaginal. La comprensión y diagnóstico preciso de esta condición clínica es complicada debido a la presencia de factores entre los que se encuentran la condición asintomática, ambigüedad en el diagnóstico, factores de riesgo y la gran cantidad de bacterias asociadas.
Si no se atiende adecuadamente esta condición puede facilitar la adquisición de infecciones de transmisión sexual. También las bacterias que están causando la vaginosis pueden producir enfermedad inflamatoria pélvica (EIP), y salpingitis. En pacientes embarazadas la VB está asociada a un mayor riesgo de aborto, parto pretérmino, bajo peso al nacer, entre otras [5].
Se ha observado que las bacterias asociadas a vaginosis están presentes en el estado de salud lo que propicia que se reporten falsos positivos o falsos negativos cuando se hace el diagnóstico. Para abordar esta problemática es imprescindible contar con criterios que discriminen la densidad del crecimiento bacteriano asociado a vaginosis del crecimiento asociado al estado de salud y poder reportar verdaderos positivos o verdaderos negativos, respectivamente.

1.2.2. Delimitación de la investigación

En este estudio se propone aplicar reglas de asociación sobre un conjunto de datos que tiene registros sobre vaginosis bacteriana de una población de pacientes femeninos sexualmente activas con un intervalo de edad de 18 a 50 años que acudieron al laboratorio de investigación en enfermedades infecciosas y metabólicas de la Universidad Juárez Autónoma de Tabasco a su inspección ginecológica de rutina anual. Esto con la intención de investigar si las asociaciones entre variables que se identifiquen por este método muestran significancia biológica con el diagnóstico positivo de VB.
Dado que las reglas de asociación pueden modelar las relaciones implícitas entre las variables al mostrar las asociaciones que existen entre estas, en este

estudio se propone obtener un modelo en forma de reglas de asociación que describa las relaciones bacterianas que contribuyen en el desarrollo de VB.

1.3. Pregunta de investigación e hipótesis

¿Tienen significado biológico las reglas de asociación identificadas en el conjunto de datos bajo estudio?
¿Qué porcentaje de soporte y confianza en las reglas de asociación con significancia biológica son aceptables en la construcción de un diagnóstico positivo de VB?
¿Las reglas aceptadas por su significancia biológica asociadas al diagnóstico de VB positivo, cumplen con los criterios de las métricas de calidad?

Hi: Las reglas de asociación permiten encontrar relaciones frecuentes entre bacterias que al coexistir detonan la vaginosis bacteriana con una confianza de al menos 80 %.

1.4. Objetivo general

Crear un modelo con reglas de asociación que permita identificar la coexistencia de bacterias en un diagnóstico positivo de vaginosis bacteriana.

1.5. Objetivos específicos

1 Generar reglas de asociación con base en el algoritmo Apriori.

2 Generar reglas de asociación con base en el algoritmo Eclat.

3 Generar reglas de asociación con base en el algoritmo FP-Growth.

4 Calcular las métricas Soporte, Confianza, Lift, Convicción y Factor de Poder de la Regla (RPF) para las reglas generadas.

5 Identificar las reglas de asociación que representen un significado biológico.

6 Evaluar el grado de asociación de las reglas con el diagnóstico positivo de vaginosis bacteriana con métodos estadísticos.

7 Validar los resultados de los algoritmos Apriori, Eclat, y FP-Growth por un experto.

1.6. Justificación

La vaginosis bacteriana es una condición clínica muy común en mujeres sexualmente activas. Su etiología es polimicrobiana, lo que dificulta su diagnóstico. Debido a que las técnicas clásicas de diagnóstico tienen baja especificidad debido a la subjetividad de los criterios de evaluación se requieren técnicas de diagnóstico con mayor precisión y exactitud.
El modelo de reglas de asociación representa eficientemente la asociación que se desarrolla entre las bacterias Gram-negativas (Gram-) que interactúan entre sí para desencadenar vaginosis bacteriana. Conocer las bacterias que estos algoritmos ubican en el antecedente (LHS) de la regla de asociación es muy importante ya que esto guía objetivamente a los profesionales de la salud para hacer frente al problema que representa la vaginosis bacteriana.
Por un lado que el profesional de la salud conozca con un alto nivel de precisión diagnóstica las bacterias Gram- implicadas en el desarrollo de la vaginosis, lo que permite tomar mejores decisiones en el tratamiento de los pacientes.
Por otro evitar las recurrencias por mal tratamiento de la infección y las infecciones asociadas a transmisión sexual.

1.7. Organización del documento de tesis

Este documento está organizado de la siguiente manera:

En el Capítulo 2 se describe la fundamentación teórica que sirve como base para la descripción y comprensión del problema a resolver.

En el Capítulo 3 se expone el estado del arte que permite ubicar esta propuesta en el contexto de las Ciencias de la Computación, específicamente en el área de la Inteligencia Artificial.

En el Capítulo 4 se describe el modelo de reglas de asociación y el trabajo experimental realizado.

En el Capítulo 5 se describen las pruebas realizadas y los resultados obtenidos.

Finalmente, el Capítulo 6 plasma las conclusiones, contribuciones, resultados esperados con esta investigación y los posibles trabajos futuros.

Capítulo 2

Fundamentos

En este capítulo se argumenta la fundamentación teórica que sirve de base para el desarrollo de esta investigación. Se desarrolla cada concepto de lo más básico a lo mas complejo, tratando de abarcar en lo posible la esencia de cada concepto.

2.1. Vaginosis bacteriana

La vaginosis bacteriana es una disbiosis (perdida de la homeóstasis) de la flora normal residente en la mucosa vaginal. En la mucosa vaginal viven dos grupos de bacterias tanto en el estado de salud como en el estado de enfermedad.
Por una parte están las especies de ***Lactobacillus*** que protegen la mucosa vaginal y por otra las especies de bacterias Gram- que en el estado de salud viven en la mucosa vaginal en un estado de densidad muy bajo debido a los ***Lactobacillus*** productores de ácido láctico, pero cuando se da la disbiosis estos crecen en una mayor densidad para desarrollar la vaginosis bacteriana [3] (Figura 2.1).

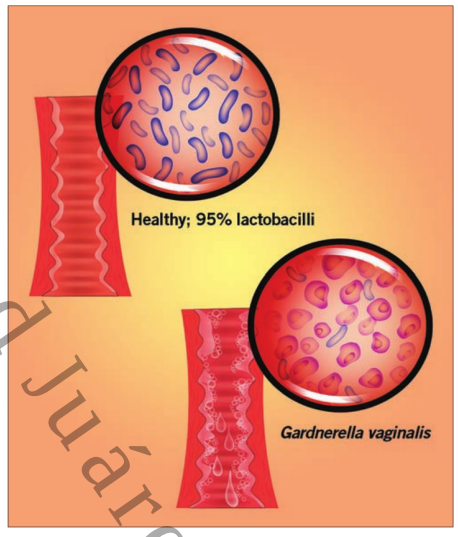

Figura 2.1. Etiología de la vaginosis bacteriana [3].

Los lactobacilos productores de ácido láctico mantienen el pH vaginal entre 4 y 4.5 en condiciones de salud o condiciones homeostáticas normales. En otras palabras se mantiene un ambiente interno estable y relativamente constante debido a que el pH vaginal mantiene un crecimiento limitado de las bacterias asociadas a vaginosis.
Cuando se pierde la homeostasis o en otras palabras se pierde ese ambiente estable y constante por el crecimiento en menor densidad de las especies de ***Lactobacillus*** se produce menos ácido láctico por lo que la mucosa vaginal se hace alcalina y bacterias como ***Gardnerella vaginalis*** producen succinato necesario para la proliferación de bacterias asociadas a vaginosis, las cuales sintetizan aminopeptidasas que liberan aminoácidos los cuales son descarboxilados para producir diaminas como putresina, cadaverina, trimetilamina y poliamina. Se sospecha que estas poliaminas son las responsables del olor a pescado en esta infección [4].

Para diagnosticar esta condición clínica se utilizan [6]:

1. Criterios de Amsel: solo toma en cuenta las características clínicas que presenta la paciente como células clave, el olor, entre otros.

2. Escala de Nugent: toma en cuenta para el diagnóstico los morfotipos bacterianos a través de tinción de Gram.

Debido a que las tecnicas anteriores dependen de la interpretación subjetiva de los criterios de evaluación y la vaginosis es muy recurrente se necesitan para el diagnóstico técnicas con mayor alcance o en otras palabras técnicas con alto nivel de precisión diagnóstica.

2.2. PCR en tiempo real

"La reacción en cadena de la polimerasa (PCR) es una técnica de los laboratorios de biología molecular que permiten la producción (amplificación) rápida de millones a miles de millones de un segmento específico de ADN (ácido desoxirribonucleico).
La PCR implica el uso de fragmentos cortos de ADN sintético, denominados cebadores, para seleccionar un segmento del genoma que se estudiará o analizará en múltiples sesiones de síntesis de ADN"[14].
La PCR tiene muchas variantes para estudiar diferentes características de los ácidos nucleícos. Entre dichas variantes se puede mencionar a PCR anidada, PCR de extensión solapada, PCR in situ, PCR múltiple, PCR con transcripción inversa (RT-PCR), PCR en tiempo real o PCR cuantitativo (qPCR).
La PCR en tiempo real es una técnica que combina la amplificación de ADN y la detección de este ADN en una misma mezcla de reacción al correlacionar el producto de PCR de cada uno de los ciclos con la señal de intensidad de un fluoróforo (molécula que al ser excitada por un fotón se torna fluorescente) [9].
La PCR a diferencia de las técnicas de diagnóstico mencionadas en la Sección 2.1 es muy eficiente para realizar el diagnóstico de VB [8, 10, 11]. Para realizar el diagnóstico primero se toma un exudado vaginal para obtener una muestra de las bacterias presentes en la mucosa vaginal. A partir de estas bacterias se extrae el ADN. Se determina la concentración de ADN y se verifica que el ADN no esté fragmentado.
En un tubo Eppendorf (pequeño contenedor cilíndrico de plástico, con un fondo cónico y típicamente una tapa unida al cuerpo del tubo para evitar su desprendimiento) se mezcla el ADN bacteriano con los desoxinucleótidos (dNTPs), ADN polimerasa (enzima que a partir de los cebadores duplica el ADN que sirve como plantilla durante la síntesis), cebadores específicos para cada bacteria y el cofactor para la ADN polimerasa $MgCl_2$.
El tubo Eppendorf con la mezcla se coloca en los pozos del termociclador y en este se programa la reacción de PCR, (ver Figura 2.2) [9].

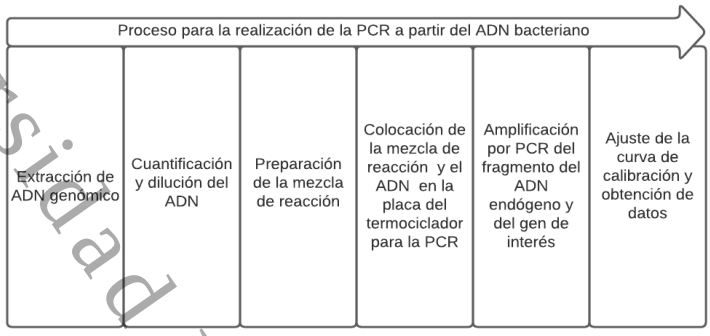

Figura 2.2. Técnica para el diagnóstico con la PCR tiempo real. Editado a partir de [9].

Cada ciclo de PCR consta de un paso de desnaturalización, alineamiento y elongación, (ver Figura 2.3).

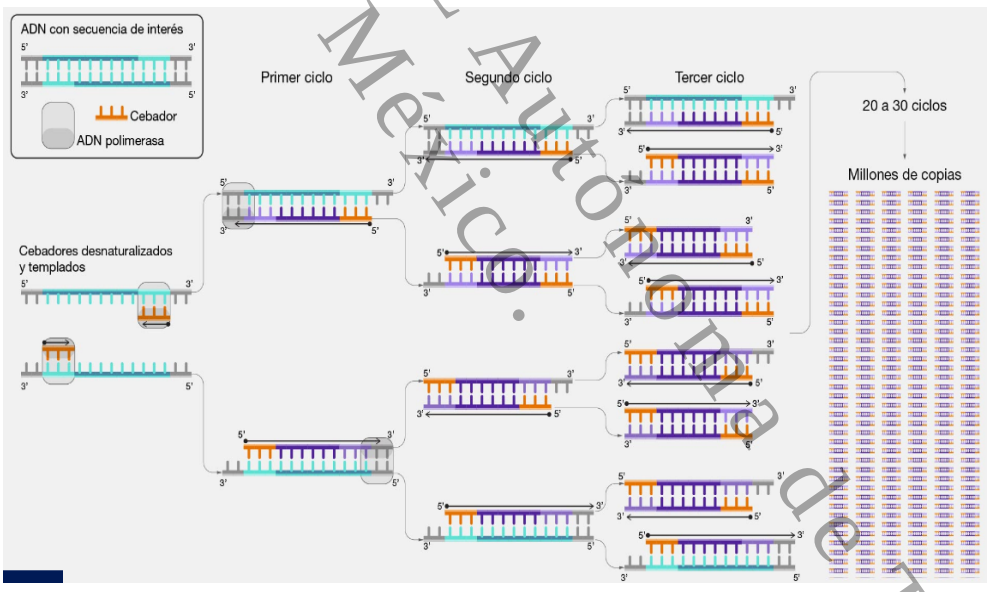

Figura 2.3. Ciclos de PCR [14].

El valor Cq (Ciclo de cuantificación) va a ser el ciclo en el que detecte el ADN de una bacteria. Este valor determina la densidad de crecimiento bacteriano.
Kuster et al. [8] evaluaron un ensayo de PCR multiplex semicuantitativo para el diagnóstico de vaginosis bacteriana (VB). Estos autores usan el ciclo

de cuantificación (Cq) del termociclador para determinar la densidad del crecimiento bacteriano y lo muestran en una serie de gráficas. En las gráficas cada panel representa los valores Cq obtenidos por qPCR multiplex para cinco especies bacterianas analizadas.
Los círculos representan la escala de Nugent y los valores Cq de los pacientes categorizados. La escala de Nugent 0-3 (sin vaginosis), 4-6 (intermedio) y 7-10 (vaginosis bacteriana) se presentan en el eje X y los valores Cq en el eje Y como se puede ver en la (Figura 2.4).

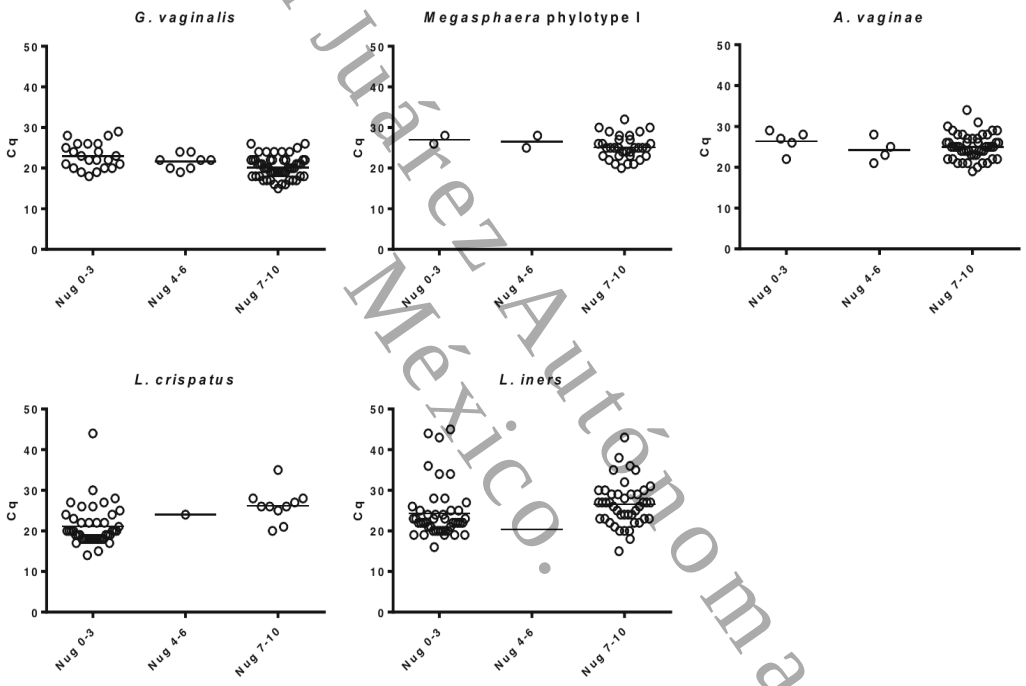

Figura 2.4. Valor Cq para determinar la densidad de crecimiento bacteriano [8].

Como se acaba de describir la PCR es muy sensible, específica, y eficiente para detectar el ADN de las bacterias asociadas a vaginosis bacteriana. Determinar que bacterias están interactuando entre sí para desarrollar vaginosis bacteriana es muy complejo ya que no se conoce con exactitud las bacterias involucradas en este síndrome polimicrobiano. Sin embargo, la minería de datos es la técnica adecuada para determinar que bacterias participan para desarrollar vaginosis bacteriana.

2.3. Minería de datos

2.3.1. Proceso KDD (Knowledge Discovery in Databases)

El proceso de descubrimiento del conocimiento en conjuntos de datos consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice [29]. En términos generales consiste en los siguientes pasos:

1. Etapa de selección.- se reconocerán e identificarán los datos relevantes y prioritarios del conjunto de datos y se definirán las metas desde el punto de vista de los objetivos del proyecto.

2. Etapa de pre-procesamiento/limpieza.- se analizará la calidad de los datos, se limpiarán las ambigüedades, ruido y seleccionarán estrategias para el manejo de datos desconocidos, nulos, duplicados.

3. Etapa de transformación/reducción.- se buscarán características presentes en el conjunto de ítems o transacciones, las cuales pueden tener atributos de diferentes tipos, por lo tanto no es necesario hacer una conversión a un tipo de datos específicos.

4. Etapa de minería de datos.- se seleccionarán las tecnicas de minería de datos apropiadas, para encontrar relaciones de interés dentro del conjunto de elementos en el conjunto de datos.

En la etapa de minería de datos del proceso KDD según los objetivos de la investigación se selecciona el o los algoritmos para realizar el estudio, (ver Figura 2.5). Para el caso de esta investigación se seleccionó reglas de asociación para determinar que bacterias están asociadas entre si para desencadenar vaginosis bacteriana.

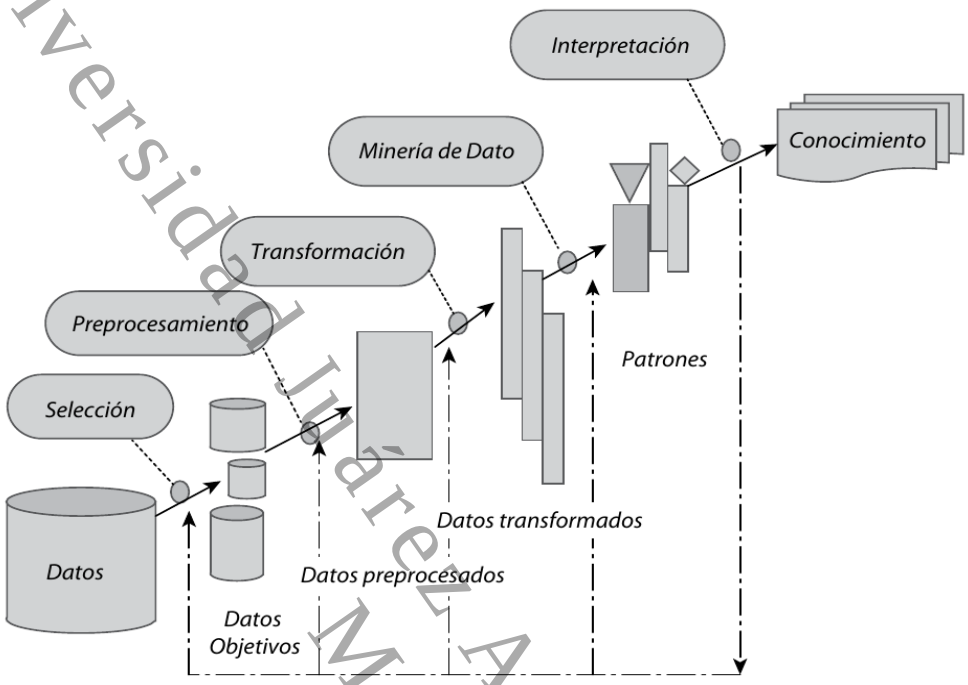

Figura 2.5. Etapas del proceso KDD [29].

2.3.2. Definición de reglas de asociación

El problema de la extracción de reglas de asociación se define como: Sea $I = \{i_1, i_2, ..., i_n\}$ un conjunto de atributos binarios n llamado elemento (items). Sea $D = \{t_1, t_2, ..., t_m\}$ un conjunto de transacciones llamada base de datos. Cada transacción en D tiene un único ID de transacción y contiene un subconjunto de elementos en I. Una regla es definida como una implicación de la forma: X $\Rightarrow$ Y, donde X, Y, $\subseteq$ I y X $\cap$ Y $= \emptyset$[19].
Una regla de asociación esta constituida de dos conjuntos de elementos unidos por una implicación (si $\Rightarrow$ entonces). El conjunto a la izquierda de la flecha se denomina antecedente (LHS) y el conjunto a la derecha de la flecha se denomina consecuente (RHS). El conjunto de elementos presentes en el antecedente en esta investigación serán las bacterias que interaccionan entre sí para desencadenar el elemento en el consecuente que será el caso positivo de vaginosis bacteriana, (ver Figura 2.6).

[1]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒ {VaginosisPos}

Figura 2.6. Modelo en forma de regla de asociación

Los algoritmos utilizados para crear el modelo de reglas de asociación son: Apriori, Eclat y FP-Growth. Estos algoritmos se describen en las subsecciones siguientes.

2.3.3. Algoritmo Apriori

El algoritmo *Apriori* incluido en el paquete `ARules` versión 1.6-8 permite la extracción de conjuntos de elementos frecuentes, conjuntos de elementos frecuentes maximales, conjuntos de elementos frecuentes cercanos y reglas de asociación.
La función *apriori(tr, parameter = list(supp = 0.07, conf = 0.9, minlen = 2, target = rules"), appearance = list(rhs = "VaginosisPos"))* recibe los siguientes parámetros: *data* que en esta investigación se nombró como tr, son un objeto de la clase transacción, *parameter* es un objeto de la clase APparameter. Este extrae reglas con un soporte mínimo de 0.1, una confianza mínima de 0.8, un máximo de 10 elementos y un tiempo máximo para la verificación de subconjuntos de 5 segundos, *appearence* es un objeto de la clase APappearance [38]. Con este argumento se puede restringir la apariencia del consecuente.
Este algoritmo primero busca todos los conjuntos de elementos frecuentes y posteriormente crea las reglas. Para la búsqueda de conjuntos de elementos frecuentes, el algoritmo Apriori se basa en una búsqueda de arriba hacia abajo y la búsqueda en amplitud en el espacio de búsqueda (con generación de candidatos), (ver Figura 2.7) y determina el valor de soporte contando directamente sus ocurrencias en la base de datos [39, 40, 41].

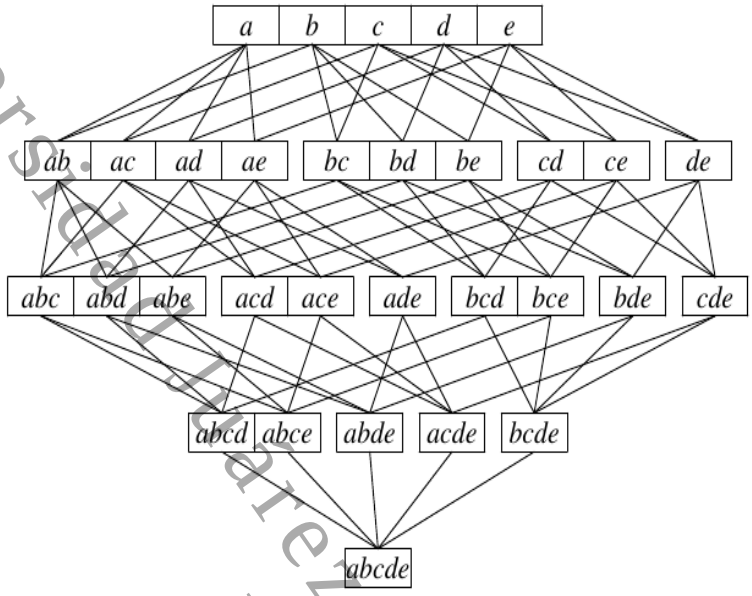

Figura 2.7. Diagrama de Hasse para subconjuntos de cinco elementos (se omite el conjunto vacío) [40].

Para estructurar la búsqueda se organiza la red de subconjuntos como un árbol de prefijos que para cinco elementos se muestra en la (Figura 2.8). En este árbol, esos conjuntos de elementos se combinan en un nodo que tiene el mismo prefijo en un orden arbitrario pero fijo de los elementos (en el ejemplo de cinco elementos, este orden es simplemente a, b, c, d, e). Con este diagrama, se construyen los conjuntos de elementos contenidos en un nodo del árbol, (ver Figura 2.8).

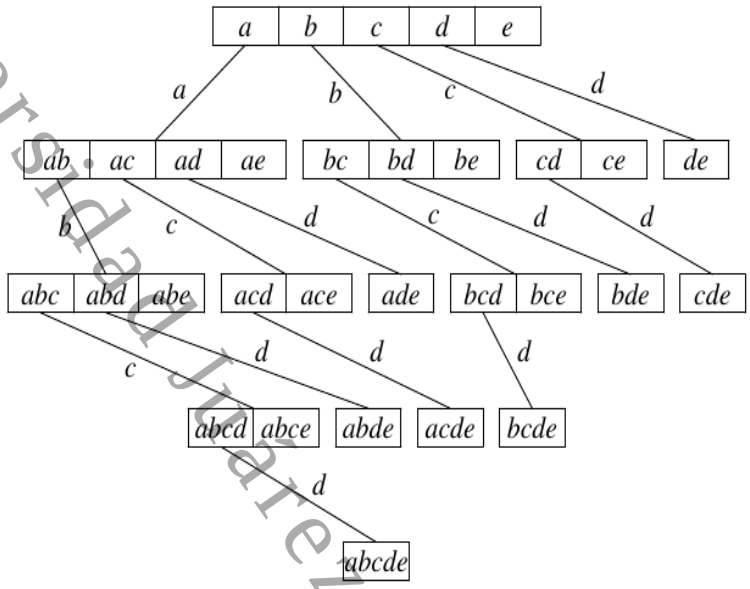

Figura 2.8. Árbol de prefijos para cinco elementos (se omite el conjunto vacío) [40].

Por lo tanto, los conjuntos de elementos frecuentes se encuentran naturalmente mediante una búsqueda de arriba hacia abajo en el lattice de búsqueda. El algoritmo Apriori realiza varias pasadas en la base de datos ya que, para determinar el valor del soporte, debe contar sus ocurrencias directamente en la base de datos, (ver Figura 2.9).
En la primera pasada el algoritmo simplemente cuenta las ocurrencias de elementos para determinar los conjuntos de elementos grandes de dimensión 1. Una pasada posterior digamos pasada k, consiste de 2 fases. Primero los conjuntos de elementos grandes L_{k-1} encontrados en la pasada *k-1* son usados para generar los conjuntos de elementos candidatos C_k usando la función *apriori-gen()*, (ver línea 3 de la Figura 2.9).

1. L_1 = {large 1-itemsets};
2. **for**(k=2; $L_{k-1} \neq 0$; k++) **do begin**
3. C_k = apriori-gen(Lk_1); % Nuevos candidatos
4. **forall** transacciones t ∈ D **do begin**
5. C_t = subset(C_k, t); % Candidatos contenidos en t
6. **forall** candidatos c ∈ C_t **do**
7. c.count++;
8. **end**
9. L_k {c ∈ C_k | c.count ≥ minsup}
10. **end**
11. Answer = $U_k L_k$;

Figura 2.9. Algoritmo Apriori [39].

La función *apriori-gen()* toma como argumento L_{k-1}, el conjunto de todos los conjunto de elementos grandes (k-1). Devuelve un superconjunto del conjunto de todos los conjuntos de elementos k grandes.
Esta función primero, en el paso de unión, se une L_{k-1} con L_{k-1}:

---

insert into C_k
select p.item$_1$, p.item$_2$, ..., p.item$_{k-1}$, q.item$_{k-1}$
from L_{k-1} p, L_{k-1} q
where p.item$_1$ = q.item$_1$, ..., p.item$_{k-2}$ = q.item$_{k-2}$, p.item$_{k-1}$ < q.item$_{k-1}$;

A continuación en el paso de poda se elimina todos los itemsets c ∈ C_k tal que algunos (k-1)-subset de c no esten en L_{k-1}:

---

1. **forall** itemsets c ∈ C_k **do**
2. **forall** (k-1)-subsets s of c **do**
3. **if** (s ∉ L_{k-1}) **then**

4. **delete** c of C_k;

Para una explicación más detallada del proceso que realiza el algoritmo para buscar los elementos frecuentes en el lattice de búsqueda consultar la referencia [41].
A continuación, se escanea la base de datos y se contabiliza el soporte de los candidatos en *Ck*. Se determinan de manera eficiente los candidatos en *Ck* que están contenidos en una transacción dada *t*. Para este proposito se usa la función *subset()*, (ver línea 5 de la Figura 2.9).
En la función *subset()* los conjuntos de elementos candidatos *Ck* se almacenan en un árbol hash. Un nodo del árbol hash contiene una lista de conjuntos de elementos (un nodo hoja) o una tabla hash (un nodo interior). En un nodo interior, cada cubo de la tabla hash apunta a otro nodo. Se define que la raíz del árbol hash está a profundidad *1*. Un nodo interior en profundidad *d* apunta a nodos en la profundidad *d + 1*.
Los conjuntos de elementos se almacenan en las hojas. Cuando se agrega un conjunto de elementos *c*, se comienza desde la raíz y se baja por el árbol hasta llegar a una hoja. En un nodo interior a la profundidad *d*, se decide qué rama seguir aplicando una función hash al elemento *d-ésimo* del conjunto de elementos. Todos los nodos se crean inicialmente como nodos hoja. Cuando el número de conjuntos de elementos en un nodo hoja supera un umbral especificado, el nodo hoja se convierte en un nodo interior.
Comenzando desde el nodo raíz, la función subset encuentra todos los candidatos contenidos en una transacción *t* de la siguiente manera. Si se está en una hoja, se encuentra cuáles de los conjuntos de elementos de la hoja están contenidos en *t* y se agrega referencias a ellos en el conjunto de respuestas. Si se esta en un nodo interior y se ha alcanzado mediante hash del elemento *i*, se aplica hash a cada elemento que viene después de *i* en *t* y se aplica recursivamente este procedimiento al nodo en el cubo correspondiente. Para el nodo raíz, se hace hash en cada elemento en *t* [39]. Para ilustrar el proceso descrito consultar el ejemplo aportado por el autor en la referencia [39].
Este algoritmo debe realizar todos los recorridos necesarios para buscar todos los conjuntos de elementos frecuentes. El rendimiento de este algoritmo mejora porque descarta todos los conjuntos de elementos infrecuentes y, por lo tanto, evita cálculos innecesarios [33, 34].

2.3.4. Algoritmo Eclat

El algoritmo Eclat (*Equivalence Class Clustering and bottom up Lattice Traversal*) extrae conjuntos de elementos frecuentes mediante operaciones de intersección simple para el agrupamiento de clases de equivalencia junto con la búsqueda en el lattice de arriba hacia abajo, (ver Figura 2.7). Al igual que el algoritmo Apriori usa el árbol de prefijos para estructurar la búsqueda de los conjuntos de elementos frecuentes, (ver Figura 2.8).
Este algoritmo requiere los siguientes parámetros: *data*, objeto de la clase transacción; *parameter*, objeto de la clase ECparameter o lista de nombres; *control*, objeto de la clase ECcontrol o lista de nombres para controles algorítmicos y el prámetro ... agrega argumentos adicionales por conveniencia a la lista de parámetros [40, 41].
Al igual que el algoritmo FP-Growth explicado en la siguiente sección, el algoritmo Eclat utiliza la búsqueda recursiva en profundidad en el espacio de búsqueda (con generación de candidatos) y determina el valor de soporte mediante la intersección de conjuntos, (ver Figura 2.10) [42].

1: **INPUT:** A file $\mathcal{D}$ consisting of baskets of items, a support threshold σ, and an item prefix I, such that $I \subseteq \mathcal{J}$.
2: **OUTPUT:** A list of itemsets $\mathcal{F}[I](\mathcal{D}, \sigma)$ for the specified prefix.
3: **METHOD:**
4: $\mathcal{F}[i] \leftarrow \{\}$
5: **for** all $i \in \mathcal{J}$ occurring in $\mathcal{D}$ **do**
6: $\mathcal{F}[I] := \mathcal{F}[I] \cup \{I \cup \{i\}\}$
7: # Create $\mathcal{D}_i$
8: $\mathcal{D}_i \leftarrow \{\}$
9: **for** all $j \in \mathcal{J}$ occurring in $\mathcal{D}$ such that $j > i$ **do**
10: $C \leftarrow \text{cover}(\{i\}) \cap \text{cover}(\{j\})$
11: **if** $|C| \geq \sigma$ **then**
12: $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{j, C\}$
13: # Depth-first recursion
14: Compute $\mathcal{F}[I \cup i](\mathcal{D}_i, \sigma)$
15: $\mathcal{F}[I] := \mathcal{F}[I] \cup \mathcal{F}[I \cup i]$

Figura 2.10. Algoritmo Eclat [35].

La llamada inicial a Eclat usa un valor I de {}, lo que significa que no se requiere un prefijo específico. Esta llamada inicial encontraría todos los conjuntos de elementos frecuentes de un solo elemento. Existen varios méto-

dos diferentes para almacenar los valores de soporte en el algoritmo recursivo Eclat. El enfoque más común es usar una estructura llamada trie. Un gráfico trie siempre contiene un nodo raíz vacío. A medida que se encuentran conjuntos de elementos, se agregan al trie insertando un nodo para cada elemento que forma el conjunto de elementos.
El elemento más a la izquierda corresponde a un elemento secundario del nodo raíz. El segundo elemento corresponde a un hijo del primer elemento de este conjunto frecuente. Ningún padre tendría más de un hijo con el mismo nombre de elemento; sin embargo, el nombre de un elemento puede aparecer en varias ubicaciones en el trie. El trie se genera para que el algoritmo pueda encontrar rápidamente el soporte de un conjunto de elementos atravesando el trie a medida que los elementos del conjunto se leen de izquierda a derecha. El nodo que contiene el elemento más a la derecha contiene el soporte para ese conjunto de elementos.
A medida que el algoritmo procesa la base de datos, se recorre el trie en busca de cada conjunto de elementos descubierto. Se crean nodos, si es necesario, para completar el trie para contener todos los conjuntos de elementos. Si los nodos ya existen, el nodo para el elemento más a la derecha en el conjunto de elementos tiene su soporte aumentado. Los nuevos nodos comienzan con un soporte de 1. Esto permite que Eclat use menos memoria que Apriori, porque las ramas centrales del trie permiten que los subconjuntos muy utilizados se almacenen solo una vez [35].
El algoritmo Eclat, a diferencia del algoritmo Apriori, solo pasa por la base de datos una vez. Debido a la agrupación que realiza este algoritmo para buscar los conjuntos de elementos frecuentes, necesita más tiempo de ejecución. Este algoritmo es computacionalmente más eficiente que Apriori a pesar del paso adicional de agrupamiento que realiza [33, 34].
Para generar reglas a partir de los conjuntos de elementos encontrados la función *ruleInduction()* proporciona el método para inducir todas las reglas de asociación que puede generar el conjunto de elementos dado a partir de un conjunto de datos de transacciones.
Esta función requiere los siguientes parámetros: x, el conjunto de los conjuntos de elementos a partir de los cuales se inducirán las reglas; el parámetro ..., más argumentos; transactions, las transacciones utilizadas para extraer los conjuntos de elementos. Puede omitirse por el método ´´ptree", si x contiene un (conjunto completo) conjunto de elementos frecuentes junto con sus recuentos de soporte; confidence, un valor numérico entre 0 y 1 que proporciona el umbral mínimo de confianza para las reglas; method, ´´ptree", ´´apriori";

reduce, eliminar elementos no utilizados para acelerar el proceso de conteo; y verbose, informe de progreso [43].

2.3.5. Algoritmo FP-Growth

Para ejecutar el algoritmo FP-Growth (crecimiento de patrones frecuentes) en R, debe hacerse a través de la implementación *fim4r*. Esta funcion interconecta los algoritmos implementados en `fim4r`. Los algoritmos incluyen: Apriori, Eclat, FP-Growth, Carpenter, IsTa, RElim y SaM. Esta función requiere los siguientes parámetros: *transactions*, objeto de tipo transacción; *method*, algoritmo a ser usado:

1. ´´apriori", ´´eclat", ´´fpgrowth"puede minar conjuntos de elementos frecuentes y reglas.
2. ´´relim", ´´sam"puede extraer conjuntos de elementos.
3. ´´carpenter", ´´ista"solo puede minar conjuntos de elementos cercanos.

target, el tipo de objetivo: ´´frequent", ´´closed", ´´maximal", ´´generators", ´´rules"; report, no se puede utilizar a través de la interfaz; *appear*, especifica la apariencia de los elementos en las reglas (solo para apriori, eclat, fpgrowth y las reglas blanco") especifica una lista con dos vectores (etiquetas de elementos y modificadores de apariencia) de la misma longitud. Los modificadores de apariencia son:

1. ´´-"(puede no aparecer).
2. ´´a"(solo en el antecedente de la regla /LHS).
3. ´´c"(solo en el consecuente de la regla /RHS).
4. ´´x"(puede aparecer en cualquier lugar).

El parámetro ... añade más argumentos y se pasan a *fim4r.x()* en el paquete *fim4r* (x es el método especificado). El soporte mínimo y la confianza mínima se pueden establecer como parámetros *supp* y *conf* (el rango es [0,100][0,100] por ciento) [41].
El algoritmo FP-Growth utiliza la búsqueda en profundidad en el espacio de búsqueda (sin generación de candidatos). Esto se hace usando un trie para almacenar las canastas reales, en lugar de almacenar candidatos como lo

hacen Apriori y Eclat. Apriori es en gran medida un algoritmo horizontal, primero en amplitud. Del mismo modo, Eclat es en gran medida un algoritmo vertical, primero en profundidad.
La estructura trie de FP-Growth proporciona una vista vertical de los datos. Sin embargo, FP-Growth también agrega una tabla de encabezado para cada artículo individual que tiene soporte por encima del nivel de soporte umbral. Esta tabla de encabezado contiene una lista vinculada a través del trie para conectar todos los nodos del mismo tipo. La tabla de encabezado le da a FP-Growth una vista horizontal de los datos, además de la vista vertical proporcionada por el trie [35]. Determina el valor de soporte contando directamente sus ocurrencias en la base de datos, (ver Figura 2.11) [44].

INPUT: A file $\mathcal{D}$ consisting of baskets of items, a support threshold σ, and an item prefix I, such that $I \subseteq \mathcal{J}$.
OUTPUT: A list of itemsets $\mathcal{F}[I](\mathcal{D}, \sigma)$ for the specified prefix.
$\mathcal{F}[i] \leftarrow \{\}$
for all $i \in \mathcal{J}$ occurring in $\mathcal{D}$ **do**
    $\mathcal{F}[I] \leftarrow \mathcal{F}[I] \cup \{I \cup \{i\}$
Create $\mathcal{D}_i$
    $\mathcal{D}_i \leftarrow \{\}$
    $H \leftarrow \{\}$
    for all $j \in \mathcal{J}$ occurring in $\mathcal{D}$ such that $j > i$ **do**
        if support$(I \cup \{i, j\}) \geq \sigma$ **then**
            $H \leftarrow H \cup \{j\}$
    for all $(tid, X) \in \mathcal{D}$ with $I \in X$ **do**
        $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(tid, X \cap H)\}$
Depth-first recursion
    Compute $\mathcal{F}[I \cup \{i\}](\mathcal{D}_i, \sigma)$
    $\mathcal{F}[I] \leftarrow \mathcal{F}[I] \cup \mathcal{F}[I \cup \{i\}]$

Figura 2.11. Algoritmo FP-Growth [35].

Debido a que el algoritmo FP-Growth no genera candidatos, requiere pocas inserciones en la base de datos. Este algoritmo utiliza un árbol de prefijos para representar las bases de datos de transacciones, por lo que no requiere un alto costo computacional. Este algoritmo es superior al Apriori debido a la búsqueda recursiva de prefijos que realiza. El FP-Tree de las bases de datos que crea este algoritmo es muy flexible ya que permite adaptarlo a los recursos de cálculo disponibles [33, 34].

La Sección 2.4 describe las métricas básicas utilizadas por los algoritmos ya descritos para crear las reglas de asociación y la Sección 2.5 las métricas de calidad usadas para seleccionar las reglas de interés.

2.4. Métricas básicas

Para que los algoritmos de reglas de asociación creen las reglas usan dos métricas básicas que son el soporte y la confianza, las cuales se definen a continuación:

2.4.1. Soporte

El soporte se define sobre conjuntos de elementos y da la proporción de transacciones que contienen a X. Se utiliza como una medida de significancia (importancia) de un conjunto de elementos. Dado que utiliza el recuento de transacciones, a menudo se denomina restricción de frecuencia. Un conjunto de elementos con soporte mayor que el umbral de soporte mínimo establecido, $supp(X) > \sigma$, se denomina conjunto de elementos frecuente o grande [22, 30].

$$supp(X) = \frac{|t \in D; X \subseteq t|}{|D|} = \frac{c_X}{|D|} = P(X) \tag{2.1}$$

donde X = conjunto de elementos frecuentes, c_X = representa el número de transacciones que contienen todos los elementos en x, D = conjunto de datos que contiene cada transacción (t) y P = es la probabilidad de que ocurra un determinado conjunto de elementos.

La desventaja del soporte es el problema de los elementos raros. Los elementos que ocurren con muy poca frecuencia en el conjunto de datos se eliminan, aunque aún producirían reglas interesantes y potencialmente valiosas. El problema de los elementos raros es importante para los datos de transacciones que generalmente tienen una distribución muy desigual de soporte para los elementos individuales (lo típico es una distribución de ley de potencia donde se usan pocos elementos todo el tiempo y la mayoría de los elementos se usan rara vez).

Rango [0, 1].

El valor del soporte dependerá del conjunto de datos bajo estudio, por ejemplo en conjuntos de datos desbalanceados el soporte se establece con valores bajos si es la clase minoritaria.

2.4.2. Confianza

La confianza se define como la proporción de transacciones que contienen Y en el conjunto de transacciones que contienen X. Esta proporción es una estimación de la probabilidad de ver el consecuente de la regla bajo la condición de que las transacciones también contengan el antecedente.
La confianza es directa y da diferentes valores para las reglas X ⇒ Y e Y ⇒ X. Las reglas de asociación tienen que satisfacer una restricción de confianza mínima, conf(X ⇒ Y)≥ γ [22, 30].

$$Conf(X \Rightarrow Y) = \frac{supp(X \Rightarrow Y)}{supp(X)} = \frac{supp(X \cup Y)}{supp(X)} = \frac{c_{XY}}{c_X} = \frac{P(X \cap) Y}{P(X)} = P(Y \mid X) \quad (2.2)$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), c_{XY}, c_X = es el evento en que una transacción contiene elementos X e Y y P = estimación de la probabilidad condicional de Y dado X.
La confianza no es cerrada hacia abajo y fue desarrollada junto con el soporte de Agrawal et al. (el llamado marco soporte-confianza). El soporte se usa primero para encontrar conjuntos de elementos frecuentes (significativos) que explotan su propiedad de cierre hacia abajo para podar el espacio de búsqueda. Luego, la confianza se usa en un segundo paso para producir reglas a partir de los conjuntos de elementos frecuentes que exceden un mínimo, umbral de confianza.
Rango [0, 1].
Dado que la confianza refleja la fuerza de la regla, se recomienda establecerla en un valor cercano a 1.

2.5. Métricas de calidad

Los algoritmos de reglas de asociación tienen el problema que crean demasiadas reglas que en la mayoría de los casos son redundantes o no significativas. El medio para seleccionar reglas significativas son las métricas de calidad y las funciones proporcionadas por el paquete Arules. Las métricas de calidad se describen en esta sección y las funciones proporcionadas por el paquete ARules se describen en la Sección 2.6.

2.5.1. Hiperconfianza

El nivel de confianza para la observación de conteos demasiado altos/bajos para las reglas X ⇒ Y usa el modelo hipergeométrico. Dado que los conteos se extraen de una distribución hipergeométrica (representada por la variable aleatoria C_{XY} con parámetros conocidos dados por los conteos n_X y n_Y, podemos calcular un intervalo de confianza para los conteos observados n_{XY} a partir de la distribución [22, 25]. La hiperconfianza reporta el nivel de confianza como:

$$hyper - conf(X \Rightarrow Y) = 1 - P[C_{XY} \geq c_{XY} \mid c_X, c_Y] \tag{2.3}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), C_{XY} = una variable aleatoria representando una distribución hipergeométrica y c_X y c_Y = representan el conteo de cada elemento.
Rango [0, 1]
Un nivel de confianza de, p. ej., > 0,95 indica que solo hay un 5 % de probabilidad de que el recuento alto de la regla se haya producido aleatoriamente. La hiperconfianza es equivalente a la estadística utilizada para calcular el p-valor en la prueba exacta de Fisher. Cada regla representa una prueba estadística y puede ser necesaria la corrección para comparaciones múltiples.

2.5.2. Hiperlift

Adaptación de la métrica lift donde en lugar de dividir por el conteo esperado bajo independencia ($E[C_{XY}] = n_X/n \times n_Y/n$) se utiliza un cuantil más alto de la distribución de conteo hipergeométrico. Esto es más sólido para conteos bajos y da como resultado menos falsos positivos cuando se usa hiper-lift para el filtrado de reglas [22, 25]. Hyper-lift se define como:

$$hyper - lift_\delta(X \Rightarrow Y) = \frac{c_{XY}}{Q_\delta[C_{XY}]} \tag{2.4}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), c_{XY} = es el número de transacciones que contienen X e Y y $Q_\delta[C_{XY}]$ es el cuantil de la distribución hipergeométrica con parámetros c_X y c_Y dado por δ (típicamente el cuantil 99 o 95 %).
Rango [0, ∞] (1 indica independencia).
Debido al ajuste de esta métrica, reporta valores cercanos a 1, lo que representa valores aceptables independientemente de las características del conjunto de datos.

2.5.3. Lift

Lift fue originalmente llamado interés por Brin et al. Más tarde, lift, el nombre de una medida equivalente popular en la publicidad y el modelado predictivo, se hizo más común. Lift mide cuántas veces más X e Y ocurren juntos de lo esperado si fueran estadísticamente independientes [22, 24]. El lift se define como:

$$\begin{aligned} lift(X \Rightarrow Y) = lift(Y \Rightarrow X) = \frac{conf(X \Rightarrow Y)}{supp(Y)} = \frac{P(Y \mid X)}{P(Y)} \\ = \frac{P(X \cap Y)}{P(X)P(Y)} = n\frac{n_{XY}}{n_X n_Y} \end{aligned} \tag{2.5}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), Supp (X) = conjuntos de elementos frecuentes, Supp (Y) = conjuntos de elementos frecuentes, $P(X \cap Y)$ = probabilidad de ocurrencia de transacciones que contengan X e Y, P(X) = probabilidad de ocurrencia de transacciones que contienen X y P(Y) = probabilidad de ocurrencia de transacciones que contienen Y.
Rango [0, ∞](1 representa la independencia)
Un valor de lift de 1 indica independencia entre X e Y. Los conjuntos de elementos raros con recuentos bajos (baja probabilidad), que por casualidad ocurren pocas veces (o solo una vez) pueden producir valores de lift enormes. Si el lift es > 1, eso nos permite saber el grado en que esas dos ocurrencias dependen una de la otra y hace que esas reglas sean potencialmente útiles para predecir el consecuente en conjuntos de datos.
Si el lif es < 1, eso nos permite saber que los elementos se sustituyen entre sí. Esto significa que la presencia de un elemento tiene un efecto negativo sobre la presencia de los otros elementos y viceversa.

2.5.4. Convicción

La convicción es una medida que evalúa el grado en que el término antecedente influye en la ocurrencia del término consecuente de una regla de asociación [22, 24]. La convicción se define como:

$$conviction(X \Rightarrow Y) = \frac{1 - supp(Y)}{1 - conf(X \Rightarrow Y)} = \frac{P(X)P(\bar{Y})}{P(X \cap \bar{Y})} \tag{2.6}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), $\overline{Y}$ = E ¬ Y es el evento que Y no aparezca en una transacción y P(X) = probabilidad de ocurrencia de X.
Rango [0, ∞] (1 indica independencia; las reglas que siempre se cumplen tienden a ∞)
Un valor alto de convicción significa que el consecuente es altamente dependiente del antecedente. Por ejemplo, en el caso de un valor de confianza perfecta, el denominador se convierte en 0 (debido a 1 - 1), por lo que el valor de la convicción se define como inf. Similar al lift, si los elementos son

independientes, la convicción es 1.

2.5.5. Coseno

El coseno es la media geométrica entre el factor de interés (I) y la métrica soporte, que es una medida de similitud ampliamente utilizada para los modelos de espacio vectorial. Se utiliza para medir la similitud entre LHS y RHS de una regla [22, 26]. Se define como:

$$cosine(X \Rightarrow Y) = \frac{supp(X \cup Y)}{\sqrt{supp(X)supp(Y)}} = \frac{P(X \cap Y)}{\sqrt{P(X)P(Y)}} = \sqrt{P(X \mid Y)P(Y \mid X)} \tag{2.7}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), Supp (X ∪ Y) = calcula el soporte del conjunto de elementos combinado, $\sqrt{supp(X)supp(Y)}$ = raíz cuadrada de la multiplicación del soporte del antecedente con el soporte del consecuente, P(X ∩ Y) = probabilidad de ocurrencia de transacciones que contengan X e Y, $\sqrt{P(X)P(Y)}$ = raíz cuadrada de la multiplicación de las probabilidades de ocurrencia del antecedente por la del consecuente y $\sqrt{P(X \mid Y)P(Y \mid X)}$ = cuadrado raíz de la multiplicación de las probabilidades condicionales de X dado Y e Y dado X.
Los valores válidos se encuentran en el rango [0, 1], donde un valor de 0,0 a 0,5 significa que no hay correlación, y de 0,51 a 1 significa que existe correlación.

2.5.6. Índice Gini

El índice Gini o la impureza Gini mide el grado o la probabilidad de que una variable en particular se clasifique incorrectamente cuando se elige al azar [22, 26]. Se define de la siguiente manera:

$$gini(X \Rightarrow Y) = P(X)[P(Y \mid X)^2 + P(\bar{Y} \mid X)^2] + P(\bar{X})[P(Y \mid \bar{X})^2 + P(\bar{Y} \mid \bar{X})^2] - P(Y)^2 - P(\bar{Y})^2 \tag{2.8}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente). Esta métrica se define en términos de las probabilidades estimadas a partir de una tabla de contingencia de 2 * 2.
P(X) = probabilidad de ocurrencia de X, $P(Y \mid X)^2$ = probabilidad de ocurrencia de Y dado X, $P(\overline{Y}|X)^2$ = probabilidad de que el evento Y no aparecerá en una transacción dada X, $P(\overline{X})$= probabilidad de ocurrencia de que el evento X no aparezca en la transacción, $P(Y|\overline{X})^2$ = probabilidad de ocurrencia de que el evento Y aparezca en la transacción dado que incluso X no aparecerá en la transacción, $P(\overline{Y} \mid \overline{X})^2$ = probabilidad de que no ocurra Y dado que incluso X no aparecerá en la transacción y $P(\overline{Y})$ = probabilidad de que no ocurra de y.
Rango [0, 1]
0 significa que la regla no proporciona ninguna información para el conjunto de datos.

2.5.7. Prueba exacta de Fisher

La prueba exacta de Fisher (prueba de significancia para identificar si las reglas representan patrones reales) calcula el p-valor de una tabla de contingencia de 2 * 2. Devuelve el p-valor asociado con la probabilidad de observar la regla solo por azar [25, 22]. Esta métrica se define de la siguiente manera:

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!} \tag{2.9}$$

ó

$$p - value = P(C_{XY} \geq n_{XY}) \tag{2.10}$$

donde a, b, c y d = elementos a, b, c y d que son las frecuencias exactas en una tabla de contingencia de 2 * 2. (a + b)! = factorial de la suma de a + b, n! = es el factorial del número de elementos de la tabla de contingencia, y a! = factoriales de a y así sucesivamente para cada elemento.
Rango [0, 1] (Escala del p-valor)

Entre más cerca esté el p-valor de cero, menor será la probabilidad de observar la regla solo por azar.

2.5.8. Factor de poder de la regla

RPF se centra en la importancia (pesa la confianza de una regla por su soporte) de la asociación entre el antecedente y el consecuente de las reglas. RPF funciona bien incluso cuando falla la confianza [22, 27]. Se define de la siguiente manera:

$$rpf(X \Rightarrow Y) = supp(X \cup Y) * conf(X \cup Y) \tag{2.11}$$

donde X = conjunto de elementos del lado izquierdo (antecedente), Y = conjunto de elementos del lado derecho (consecuente), $supp(X \cup Y)$ = calcula el soporte del conjunto de elementos combinado, y conf $(X \cup Y)$ = confianza de la regla.
Rango [0, 1]
RPF es más informativo sobre la importancia de las reglas. Cuando aumenta la asociación entre antecedente y consecuente, aumenta la importancia de la regla.

2.6. Funciones Arules

En esta sección se describen las funciones que aporta el paquete ARules para filtrar las reglas que son de relevancia por su significancia computacional y estadística.

2.6.1. Función *is.redundant()*

Cada métrica de calidad junto con la función *is.redundant()* filtran las reglas redundantes. Una regla es redundante si existe una regla más general con la misma o mayor confianza. Es decir, una regla más específica es redundante si es igual o incluso menos predictiva que una regla más general

[28]. Esta función recibe como argumento el conjunto de reglas y una métrica de calidad, y a partir de estos dos argumentos determina un subconjunto de regla no redundante.

2.6.2. Función *is.significant()*

La función *is.significant()* evalúa la significancia estadística de las reglas creadas. Esta función utiliza el método de Fisher con $\alpha = 0.01$ y el ajuste de Bonferroni [25]. Cada regla representa una prueba estadística y puede ser necesaria la corrección para comparaciones múltiples. Las opciones que la función acepta para hacer estas multiples comparaciones son: Ninguno, Bonferroni, Holm, FDR.
En esta investigación se usó el ajuste de Bonferroni ya que es el que está predeterminado en la función. Esta función recibe como argumentos el conjunto de reglas, la base de datos transaccional y los métodos estadísticos y devuelve un subconjunto de reglas estadísticamente significativas.

2.6.3. Función *is.maximal()*

La función *is.maximal()* reporta solo reglas maximales [19]. Un conjunto de elementos frecuentes es maximal si ningún otro conjunto de elementos frecuentes es su superconjunto. Una regla de asociación se define como maximal si se generó con un conjunto de elementos maximales. Esta función reporta un subconjunto en el que determina qué regla es maximal y cuál no.

2.7. Algoritmos de balanceo

En conjuntos de datos desbalanceados la clase minoritaria tiene menos instancias en comparación con la clase mayoritaria. Esto se ve en el diagnóstico médico donde la clase minoritaria determina el diagnóstico positivo. La consecuencia del desbalanceo es que los algoritmos pueden estar sesgados

hacia la clase mayoritaria. Por lo tanto para solucionar este problema se balancea el conjunto de datos con los algoritmos: SMOTE, ROSE y ADASYN. Las secciones siguientes describen a estos algoritmos.

2.7.1. Algoritmo SMOTE

El algoritmo SMOTE (*Synthetic Minority Oversampling TEchnique*) incluido en el paquete `smotefamily` versión 1.3.1 sobremuestrea la clase minoritaria creando casos sintéticos. El algoritmo recibe cuatro parámetros: X es un marco de datos o matriz de un conjunto de datos con atributos numéricos, *target* es un vector de clase objetivo con atributos correspondiente a un conjunto de datos X, K es el número de vecinos más cercanos durante el proceso de muestreo y *dup size* es el número o vector que representa los tiempos deseados de instancias minoritarias sintéticas sobre el número original de instancias mayoritarias [45].

La idea clave de SMOTE es introducir ejemplos sintéticos en lugar de aplicar una simple réplica de las instancias de la clase minoritaria. Estos nuevos datos se crean por interpolación entre varias instancias de las clases minoritarias que se encuentran dentro de un vecindario definido. El algoritmo se basa en los valores de las características y su relación.

Al comienzo del balanceo, el algoritmo selecciona una instancia x_i de la clase minoritaria y en función de una métrica de distancia, se eligen varios vecinos más cercanos de la misma clase (puntos x_{i1} a x_{i4}) del conjunto de entrenamiento, (ver Figura 2.12). Finalmente, se realiza una interpolación aleatoria para obtener nuevas instancias r_1 a r_4. El valor k igual a 3, 5, 7, etc. determina la base para interpolar ejemplos de sobremuestreo sintético en el espacio de características del vecindario a partir de instancias minoritarias.

Para ejecutar el algoritmo SMOTE primero se configura la cantidad total de sobremuestreo N, que puede configurarse para obtener una distribución de clases aproximada de 1:1 o descubrirse a través de un proceso de envoltura. Luego, se lleva a cabo un proceso iterativo, compuesto por varios pasos. Una instancia de clase minoritaria se selecciona al azar del conjunto de entrenamiento. A continuación, se obtienen sus K vecinos más cercanos (5 por defecto).

Finalmente, N de estas K instancias se eligen aleatoriamente para calcular las nuevas instancias por interpolación. Para ello se toma la diferencia entre el vector de características (muestra) considerado y cada uno de los vecinos seleccionados.

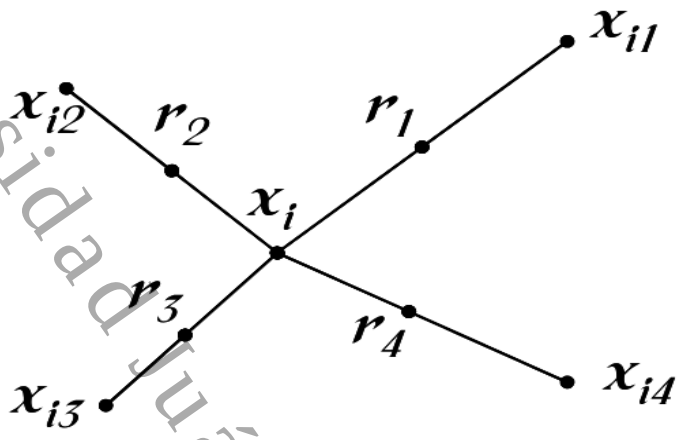

Figura 2.12. Creación de los puntos de datos sintéticos por el algoritmo SMOTE [45].

Esta diferencia se multiplica por un número aleatorio entre 0 y 1, y luego se suma al vector de características anterior. Esto provoca la selección de un punto aleatorio a lo largo del segmento de línea entre las características, (ver Figura 2.13) [46].

SMOTE algorithm

1: function SMOTE(T, N, k)
Input: T ; N ; k minority class examples, Amount of oversampling, nearest neighbors
Output: (N /100) * T synthetic minority class samples
Variables: Sample[][]: array for original minority class samples;
newindex: keeps a count of number of synthetic samples generated, initialized to 0;
Synthetic[][]: array for synthetic samples
2: if N <100 then
3: Randomize the T minority class samples
4: T = (N /100)*T
5: N = 100
6: end if
7: N = (int)N/100 . The amount of SMOTE is assumed to be in integral multiples of 100.
8: for i = 1 to T do
9: Compute k nearest neighbors for i, and save the indices in the nnarray
10: POPULATE(N, i, nnarray)
11: end for
12: end function

Figura 2.13. Algoritmo de balanceo SMOTE [45].

2.7.2. Algoritmo Rose

El algoritmo ROSE (*Random Over-Sampling Examples*) incluido en el paquete ROSE versión 0.0-4 crea datos sintéticos para aumentar el número de la clase minoritaria y proporciona la función *ovun.sample()* que crea muestras balanceadas utilizando ejemplos de minorías de sobremuestreo aleatorio, ejemplos de mayoría de submuestreo o combinación de sobremuestreo y submuestreo.
Tiene los siguientes argumentos para realizar el balanceo: *formula* es un objeto de clase fórmula (o uno que puede ser forzado a esa clase), *data* es un marco de datos opcional, list or environment (u objeto coercible a un marco de datos por *as.data.frame()*) en el que interpretar preferentemente ´´fórmula". Si no se especifica, las variables se toman de entorno (fórmula), *method* es uno entre c(´´over", ´´under", ´´both") para realizar ejemplos de sobremuestreo minoritarios, ejemplos de submuestreo mayoritario o combinación de sobre y submuestreo, respectivamente, p probabilidad de volver a muestrear la clase minoritaria y *seed* es un valor único, interpretado como un número entero, recomendado para especificar semillas y mantener el rastro de la muestra (reproducibilidad) [47].
ROSE utiliza *bootstrapping* suavizado para extraer muestras artificiales del vecindario de espacio de características alrededor de la clase minoritaria. Maneja datos continuos y categóricos mediante la generación de ejemplos sintéticos a partir de una estimación de densidad condicional de las dos clases [48].

Definición del algoritmo ROSE

Sea un conjunto de entrenamiento Tn, con N muestras $\{x_i, y_i\}$, i = 1, ..., N, la etiqueta de la clase $y_i \in C = \{y_0, y_i\}$. x_i son atributos de un vector aleatorio x definido en R^d, f(x) es la función de densidad de probabilidad. Sea Nj el número de ejemplos pertenecientes a la clase y_j.

El procedimiento ROSE se describe a continuación:

1. Selecciona $y^* = yj$ con probabilidad π_j.
2. Selecciona $\{x_i, y_i\} \in$ T R, tal que $y_i = y^*$, con probabilidad $\frac{1}{N_j}$.
3. Muestra x^* de K_{Hj} (., x_i) una distribución de probabilidad centrada en

x_i y matriz de covarianza H_j [48].

2.7.3. Algoritmo ADASYN

El algoritmo ADASYN (*Adaptive Syntetic Sampling Approach for Imbalanced Learning*) incluido en el paquete `smotefamily` versión 1.3.1 crea instancias positivas sintéticas. La función *ADAS()* toma tres argumentos: X es un marco de datos o matriz de conjunto de datos con atributos numéricos, *target* es un vector de la clase objetivo con atributo correspondiente a un conjunto de datos X y K es el número de vecinos más cercanos durante el proceso de muestreo [49].

La función *ADAS()* crea muestras sintéticas inversamente proporcionales a la densidad de los ejemplos en la clase minoritaria. La idea clave del algoritmo ADASYN es utilizar una distribución de densidad $\hat{r}_i$ como criterio para decidir automáticamente la cantidad de muestras sintéticas que deben generarse para cada ejemplo de datos minoritarios. Físicamente, $\hat{r}_i$ es una medida de la distribución de pesos para diferentes ejemplos de clases minoritarias según su nivel de dificultad en el aprendizaje.

ADASYN se basa en la idea de generar de forma adaptativa muestras de datos minoritarios según sus distribuciones. Se generan más datos sintéticos para muestras de clases minoritarias que son más difíciles de aprender en comparación con aquellas muestras minoritarias que son más fáciles de aprender [46].

El conjunto de datos resultante posterior a ADASYN no solo proporcionará una representación balanceada de la distribución de datos (de acuerdo con el nivel de balanceo deseado definido por el coeficiente β), sino que también obligará al algoritmo de aprendizaje a centrarse en esos ejemplos difíciles de aprender, (ver Figura 2.14).

Input

(1) Training dataset Dtr with m samples $\{x_i, y_i\}$, i = 1, ..., m, where x_i is an instance in the n dimensional feature space X and $y_i \in Y = 1, -1$ is the class identity label associated with x_i . Define ms and ml as the number of minority class examples and the number of majority class examples, respectively. Therefore, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure

(1) Calculate the degree of class imbalance:

$d = \frac{m_s}{m_l}$ (1)

where d $\in$ (0, 1].

(2) If d $< d_{th}$ then (d_{th} is a preset threshold for the maximum tolerated degree of class imbalance ratio):

(a) Calculate the number of synthetic data examples that need to be generated for the minority class:

$G = (m_l - m_s) * \beta$ (2)

Where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced data set is created after the generalization process.

(b) For each example $x_i \in$ minorityclass, find K nearest neighbors based on the Euclidean distance in n dimensional space, and calculate the ratio r_i defined as:

$r_i = \frac{\delta_i}{K}, i = 1, ..., m_s$ (3)

where δ_i is the number of examples in the K nearest neighbors of x_i that belong to the majority class, therefore $r_i \in [0, 1]$;

(c) Normalize r_i according to $\hat{r}_i = ri / \sum_{i=1}^{m_s} r_i$, so that $\hat{r}_i$ is a density distribution ($\sum_i \hat{r}_i = 1$)

(d) Calculate the number of synthetic data examples that need to be generated for each minority example x_i:

$g_i = \hat{r}_i * G$ (4)

where G is the total number of synthetic data examples that need to be generated for the minority class as defined in Equation (2).

(e) For each minority class data example x_i, generate g_i synthetic data examples according to the following steps:

Do the Loop from 1 to g_i:

(i) Randomly choose one minority data example, x_{zi}, from the K nearest neighbors for data x_i.

(ii) Generate the synthetic data example:

$s_i = x_i + (x_{zi} - x_i) \times \lambda$ (5)

where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$.

End Loop

Figura 2.14. Algoritmo de balanceo ADASYN [49].

2.8. Algoritmo Random Forest

Para que los algoritmos SMOTE y ADASYN creen las reglas de asociación necesitan el argumento K. El valor K alude al número de vecinos más cercanos para que a partir de estos se haga un interpolación de los datos durante el proceso de balanceo. El algoritmo Random Forest se utilizó para realizar la tarea de determinar el valor K adecuado para los algoritmos de balanceo.

Un bosque aleatorio se define de la siguiente manera: es un clasificador que consiste en una colección de clasificadores estructurados en árbol {h(x, ..., θ_k), k = 1, ...} donde {θ_k} son vectores aleatorios independientes, distribuidos de forma idéntica y cada árbol arroja un voto unitario para la clase más popular de la entrada x [50].

Un bosque aleatorio es un algoritmo de aprendizaje automático supervisado de proposito general que se construye a partir de algoritmos de árboles de decisión. `Random Forest` consiste en una gran cantidad de árboles de decisión individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio devuelve una predicción de clase y la clase con más votos se convierte en la predicción de nuestro modelo, (ver Figura 2.15). Lo que ha contribuido en gran medida a la popularidad de los bosques es el hecho de que se pueden aplicar a una amplia gama de problemas de predicción y tienen pocos parámetros para ajustar [50]. El método es generalmente reconocido por su precisión y su capacidad para manejar tamaños de muestra pequeños. También se puede utilizar en modo no supervisado para evaluar proximidades entre puntos de datos [51].

La función *randomForest()* acepta una gran cantidad de argumentos, los argumentos utilizados para esta investigación fueron los siguientes: *formula* es un marco de datos o una matriz de predictores, o una fórmula que describe el modelo a ajustar, *data* es un marco de datos opcional que contiene las variables del modelo. Por defecto, las variables se toman del entorno desde el que se llama a *randomForest()*, *proximity* calcula la medida de proximidad entre las filas [52].

Inicio
1. Seleccionar aleatoriamente "k" a partir de las características "f" totales.
2. Donde k <f
3. Para "k", calcular el nodo "d" utilizando el mejor punto de división
a. Dividir "d" en $d_1, d_2..., d_n$.
4. Repetir 1:3 hasta obtener "d_n".
5. Repetir pasos 1 hasta 4 para obtener "n" árboles y construir el bosque aleatorio B.
6. Predicción
6.1 Precondición: Conjunto de entrenamiento S = (x1,y1),..., (xn,yn), características f y n árboles en el Bosque B.
a. Función RanfomForest (S,F)
b. H ←Ø
c. Para i ∈ 1, ..., hacer B
S ←(i) una muestra de arranque de S
hi ←randomizedtreelearn (S(i)F)
H ←H ∪ hi
Fin Para
d. Retornar H
e. Fin Función
6.2 Función RandomForestClassifier
Hacer clasificación para n estimaciones de S(x_n, y_n) donde
Predicción es:
$D = f(x)$, específicamente $D = x_i, y_i^n = 1$
Se entrena $h_k(x)$ donde:
Cada clasificador $h_k(x) = h(x \mid \theta_k)$ es un predictor de n para $y = \pm 1$ asociado con cada entrada x
Fin función
Fin predicción

Figura 2.15. Pseudocódigo del algoritmo Random forest [53].

2.9. Lenguajes de programación

Esta sección describe a los lenguajes de programación R y AWK. El lenguaje de programación R fue usado para ejecutar a los algoritmos de reglas de asociación y crear las reglas. El lenguaje de programación AWK se usó en la segunda y tercera etapa de experimentos para filtrar las reglas de interés. El lenguaje de programación AWK detecta patrones en conjunto de datos a partir de usar expresiones regulares y reporta las coincidencias halladas en el conjunto de datos.

2.9.1. R

R es un entorno de programación simple y efectivo que admite condicionales, ciclos, funciones recursivas y posibilidad de entrada y salida [54]. Este lenguaje es orientado a objetos, por lo tanto todo es guardado en memoria como un objeto. Las facilidades de programación permiten la implementación de nuevos procedimientos así como el uso de funciones incluídas en paquetes que extienden la potencia de este lenguaje de programación.
Este lenguaje de programación se caracteriza principalmente por:

1. La robustez del lenguaje.
2. La constante actualización mantenida por la comunidad y la amplia literatura disponible.
3. Amplias facilidades de manipulación de bases de datos.
4. La obtención de informes con un formato predeterminado a través de Rmarkdown. También es notable la facilidad con la que se puede ejecutar LaTeXdesde R y hacer la edición del documento más fácil en cuanto a la programación.
5. Las facilidades gráficas.
6. Facilidades para la documentación de todo el proceso de manipulación de los datos y procesamiento estadístico en la investigación reproducible.

La versión de R utilizada en esta investigación fue: R version 4.2.1 (2022-06-23) – Funny-Looking Kid Copyright (C) 2022 The R Foundation for Statistical Computing Platform: x86_64-suse-linux-gnu (64-bit).

2.9.2. AWK

AWK es un lenguaje de programación cuya operación básica es buscar patrones en un conjunto de datos y realizar acciones específicas en las líneas o campos que contienen instancias de esos patrones [55]. En tareas relacionadas con el manejo de grandes volúmenes de datos, este lenguaje de programación resulta muy potente a la hora de buscar patrones específicos. Su precisión se deriva del uso de expresiones regulares que tienen un gran alcance para encontrar patrones en grandes conjuntos de datos.
Cuando se ejecuta awk con una expresión regular, este lenguaje busca en el archivo que contiene a los datos línea por línea el patrón que describe la expresión regular. Encontrado el patrón el lenguaje realiza una acción. En el caso específico de esta investigación, imprime las líneas que coinciden con el patrón buscado. Las reglas que crean Apriori, Eclat y FP-growth tienen la siguiente forma:

[1]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒ {VaginosisPos}
[2]{AtopobiumNeg, GardnerellaNeg, inersHighGrowthDensity} ⇒ {VaginosisNeg}
[3]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒ {UreaplasUreaNeg}

Las tres reglas anteriores ya están validadas estadísticamente, sin embargo, ese conjunto contiene reglas que no son de interés. En ese conjunto, sólo nos interesa la regla número 1. ¿Cómo extrae awk solo la regla número 1?

awk '$4 ~ /VaginosisPos/ {print $0}' file.txt

El programa anterior solo imprime las líneas que coinciden con la expresión regular.
La versión usada de este lenguaje en esta investigación fue: GNU AWK 4.2.1, API: 2.0 Copyright (C) 1989, 1991-2018 Free Software Foundation.

Capítulo 3

Revisión de literatura relacionada

Esta sección describe el uso del aprendizaje automático para estudiar a la vaginosis bacteriana. Se explora la factibilidad de estudiar a la vaginosis bacteriana con estos algoritmos. También se explora que tan factible es el uso del algoritmo de reglas de asociación en el ámbito del diagnóstico de enfermedades.

3.1. Estudio de la vaginosis bacteriana con aprendizaje automático

Para estudiar la vaginosis bacteriana, Baker et al. [12] analizaron un conjunto de datos con 1601 instancias y 418 atributos. Las instancias se dividen en tres subcategorías: series temporales, datos clínicos y médicos. Las series de tiempo cuentan el tiempo de estudio. Los datos clínicos tratan un cuestionario en el que se investigan los factores de riesgo de vaginosis y los criterios de Amsel, y los datos médicos identifican taxonómicamente las especies de bacterias asociadas a la vaginosis en base a la secuencia de rRNA de 16 Svedberg.

Ellos utilizaron 5 algoritmos de selección de atributos y a estos les agregaron 6 métodos de búsqueda. Una vez formados los subgrupos de atributos más significativos, eligieron los algoritmos de clasificación: A1 (Bagging), A2 (RBFNetwork), A3 (J48), A4 (NaïveBayes), A5 (AdaBoost.M1), A6 (RandomForest), A7 (LogitBoost), A8 (Kstar (K *)) y A9 (FT).

La validación cruzada se repitió 10 veces. Todos los algoritmos se ejecutaron en el entorno de la herramienta Weka. Los autores utilizaron una combina-

ción de cinco algoritmos de selección de atributos, seis métodos de búsqueda y tres algoritmos de clasificación (utilizados para los métodos de wrapper) ensamblados para crear 20 conjuntos distintos de selección de atributos. Basado en el tiempo de ejecución (0: 00: 02), reducción del número de atributos (14) y sensibilidad del 92 %; los autores concluyen que el algoritmo FS16 A9 es el mejor algoritmo para investigar el problema de la vaginosis bacteriana. También demuestran la viabilidad de estudiar la vaginosis bacteriana con técnicas de aprendizaje automático.

Continuando con la misma línea de investigación, Baker et al. [13] utilizando únicamente los datos clínicos y médicos del conjunto de datos (Ravel et al., 2011), analizaron la capacidad de predecir la clase en función de atributos clínicos (criterios de Amsel) o atributos médicos (regiones OTU basadas en la secuencia 16 S rRNA).

Para lograr sus objetivos, utilizaron el siguiente algoritmo de selección de atributos: WrapperSubsetEval que utiliza un clasificador: oneR, Bagging, NaïveBayes, para determinar el subconjunto de atributos. La validación cruzada se aplicó para aproximar la precisión del esquema de aprendizaje. Utilizaron 4 métodos de búsqueda.

Los algoritmos de clasificación utilizados fueron los siguientes: Bagging, RandomForest, NaiveBayes, RBFNetwork. Las métricas utilizadas fueron las siguientes: exactitud (AC), precisión (PR), sensibilidad (RC) y medida F (FM). Utilizaron una combinación de cinco algoritmos de selección de atributos, seis métodos de búsqueda y tres algoritmos de clasificación ensamblados para crear 20 conjuntos distintos de selección de atributos.

Además, seleccionaron nueve algoritmos de clasificación para sus experimentos. Teniendo en cuenta la precisión del 95,7527 %, el tiempo de ejecución (0:00:01), la reducción de atributos y la sensibilidad del 0,847 %, los autores concluyen que el algoritmo WNLR y el conjuntos de datos médico son los mejores para comprender el desarrollo de la vaginosis bacteriana.

Para clasificar las comunidades microbianas en las categorías BV+ y BV-, Beck y Foster [16] utilizaron tres técnicas de aprendizaje automático que incluyen programación genética (GP), bosques aleatorios (RF) y regresión logística (LR). El interés de los autores en el modelo de clasificación es analizar la precisión de la clasificación ya que eso determinará qué tan bien se clasifican las muestras en las categorías mencionadas.

Las técnicas mencionadas anteriormente se aplicaron al conjunto de datos publicado por Ravel et al. en 2011, y consta de 396 pacientes de los cuales 97 eran VB+ según la definición de la escala de Nugent. Estos autores clasi-

ficaron las comunidades microbianas amplificando y secuenciando la región variable V1 - V2 del gen 16S rRNA. También utilizaron el conjunto de datos de Srinivasan et al., que consta de 220 pacientes, 97 de los cuales eran BV+ según los criterios de Amsel. Estos autores clasificaron las comunidades microbianas amplificando y secuenciando la región variable V3 - V4 del gen 16S rRNA.
Los autores encontraron que RF y LR clasifican la clase de vaginosis con una precisión entre el 90 % y el 95 % y principalmente cuando se trata del conjunto de datos en el que se realizó el diagnóstico de VB+ con la escala de Nugent. También son más rápidos en términos de tiempo de ejecución en comparación con GP. Los autores argumentan que este estudio demuestra la viabilidad de utilizar modelos de clasificación para identificar importantes comunidades microbianas relacionadas con la VB.

3.2. Reglas de asociación en el estudio de enfermedades

Las reglas de asociación también se han utilizado para estudiar enfermedades como la enfermedad de Chagas y virus de la inmunodeficiencia humana (VIH). Marchan et al. [17] utilizaron reglas de asociación para investigar los factores de riesgo de transmisión de Chagas causado por ***Trypanosoma cruzi***. Utilizaron el proceso de minería de datos estándar de la industria cruzada CRISP-DM y la biblioteca Arules del paquete estadístico R con su función Apriori.
Esta técnica de aprendizaje automático crea reglas basadas en la restricción que ejercen las métricas de soporte y confianza, y de esta manera revela patrones de conocimiento ocultos en las bases de datos transaccionales.
El conjunto de datos que analizaron está compuesto de los datos de 293 familias según características epidemiológicas. Realizaron un diagnóstico serológico aleatorio en 88 individuos y determinaron la presencia de seropositividad o ausencia de seronegatividad de anticuerpos antitripanosoma IgM, IgA e IgG totales.
Aplicando la función Apriori del paquete Arules es posible predecir y asociar en un 93 % y 100 % múltiples factores de riesgo para una serología positiva y negativa, respectivamente. Solo 2 factores fueron determinados por el método

de Chi-cuadrado convencional.
Los autores concluyen que las reglas de asociación pueden mostrar relaciones ocultas entre algunos elementos de las variables. El uso de reglas de asociación puede mejorar la obtención de conocimiento oculto frente al uso clásico de técnicas de selección de variables.
Para estudiar pacientes con VIH/SIDA, Fernandez et al. [18], también utilizaron reglas de asociación. Realizaron la extracción de reglas de asociación utilizando el algoritmo Apriori y el proceso KDD. Analizaron una base de datos de registros clínicos y administrativos de pacientes infectados con VIH desde julio de 1980 hasta marzo de 2006.
La base de datos tiene un tamaño de 155 MB y consta de 111 tablas que contienen información sobre 6277 pacientes. Al preprocesar las tablas mencionadas se obtuvo una tabla con 6277 transacciones con 17 elementos por transacción. Utilizaron la herramienta de software libre ARView desarrollada específicamente para extraer reglas de asociación, programada en Java.
En el primer análisis, con la configuración por defecto (confianza mínima del 80 %, cobertura entre el 10 % y el 100 %, número de ítems entre 1 y 5) y sin imponer ninguna restricción, se obtuvieron 14.203 reglas de asociación. Para reducir el número de reglas, variaron los parámetros e incorporaron restricciones.
Los autores argumentan que este estudio es una aproximación al problema de la extracción de asociaciones entre variables. Informan cómo el uso de técnicas de minería de datos puede conducir a la extracción de patrones, confirmando en algunos casos el conocimiento que se tiene sobre las enfermedades y abriendo posibles vías de investigación biomédica.

Capítulo 4

Proceso para crear el modelo de reglas de asociación

La intención de estudiar y crear el modelo de reglas de asociación es para determinar que tan factible es este modelo para describir la interacción bacteriana que desarrolla la vaginosis bacteriana. Analizar si lo que describe el modelo esta en concordancia con lo que se observa en la clínica que presenta la paciente con vaginosis bacteriana.
Por lo tanto esta sección describe el proceso realizado para crear este modelo. Se realiza la selección de los datos, el preprocesamiento sobre el conjunto de datos. Se dan detalles del proceso llevado acabo para determinar los porcentajes de soporte y confianza apropiados para crear reglas de asociación. Se describe el uso de las métricas de calidad y las funciones proporcionadas por el paquete ARules para seleccionar las reglas con significancia estadística y biológica.

4.1. Descripción de los datos

El conjunto de datos estudiado en esta investigación fue proporcionado por Sanchez-Garcia et al. [10]. Este estudio se realizó entre agosto de 2016 y octubre de 2018 en Tabasco, un estado en la región sureste de México. La población objeto de estudio estuvo conformada por mujeres sexualmente activas de 18 a 50 años que se sometieron a su revisión ginecológica de rutina anual en el Laboratorio de Investigación en Enfermedades Metabólicas e Infecciosas de la Universidad Juárez Autónoma de Tabasco.

El conjunto de datos se compone de 201 observaciones y 58 variables. Todas las variables en el conjunto de datos son numéricas excepto la variable ID y Citología que son categóricas. Existen tres clases en el conjunto de datos: la clase para casos de vaginosis positiva (51), la clase para casos de vaginosis negativa (134) y la clase para casos de vaginosis indeterminada (16).
El conjunto de datos tiene tres enfoques. El enfoque cuantitativo el cual registra el valor numérico del Cq, en este se determina la densidad del crecimiento bacteriano. El enfoque cualitativo el cual registra la presencia o ausencia bacteriana. Y el enfoque mixto el cual es una mezcla de ambos enfoques.

4.2. Preprocesamiento del conjunto de datos

Se observa con mucha frecuencia que los datos biomédicos siempre están sesgados hacia una característica particular. En el caso de esta investigación donde se evalúa el caso positivo o negativo para vaginosis bacteriana el sesgo es hacia los casos negativos sobre vaginosis bacteriana. También es común observar que durante la colección de datos se pierdan algunos datos y estos se registren como datos faltantes o datos atípicos.
La tarea de preprocesamiento de datos consiste en depurar todas estas características en el conjunto de datos que de algún modo pueden generar ruido aleatorio cuando se inicia la etapa de minería de datos.
Para la minería de reglas de asociación primero todas las variables en el conjunto de datos bajo estudio se transformaron de variables numéricas a variables categóricas, ya que es el tipo de variable adecuada para extraer reglas de asociación. Por ejemplo, originalmente nuestro conjunto de datos viene con dos variables crispatus, una cuantitativa y otra cualitativa. Ambas variables se transformaron de la siguiente manera:

1. Variable cualitativa: nombre original Lactobacilluscrispatus<20 con valores de 1 para la presencia y 2 para la ausencia. Se cambió el nombre de la variable a crispatus con valores categóricos como crispatusPresent en lugar de 1 y crispatusAbsent en lugar de 2.

2. Variable cuantitativa: nombre original crispatusCq con valores continuos a partir de 0,0. Se dividió en tres categorías según el valor cuantitativo de la variable. Para el valor Cq igual a 0.0 se utilizó la constante denominada como indetectable, para el valor Cq <= 25 se utilizó la

constante denominada como alta densidad de crecimiento y para el valor Cq >25 se utilizó la constante denominada como baja densidad de crecimiento.

El ejemplo anterior descrito para la variable crispatus se aplicó para cada variable en el conjunto de datos. Las variables en el conjunto de datos como virus del papiloma humano (VPH), Clamidia, Gonorrea, que no están asociadas directamente con el desarrollo de vaginosis bacteriana se descartaron y solo aquellas variables que probablemente si están relacionadas directamente con la vaginosis bacteriana se mantuvieron y se muestran en la Tabla 4.1.

Tabla 4.1. Variables preprocesadas contenidas en el conjunto de datos sobre vaginosis.

Variable	Descripción
AGE30	Edad dividida en <= 30 y > 30
Megasphaera	*Magasphaera tipo 1.* Clasificado como positivo o negativo durante el diagnóstico.
Atopobium	*Atopobium vaginae.* Clasificado como positivo o negativo durante el diagnóstico.
Gardnerella	*Gardnerella vaginalis.* Clasificado como positivo o negativo durante el diagnóstico.
VBPCR	Diagnóstico de vaginosis por PCR (Polymerase Chain Reaction). Clasificado como vaginosis positiva, negativa o indeterminada.
MH	*Mycoplasma hominis.* Clasificado como positivo o negativo durante el diagnóstico.
MG	*Mycoplasma genitalium.* Clasificado como positivo o negativo durante el diagnóstico.
UP	*Ureaplasma parvum.* Clasificado como positivo o negativo durante el diagnóstico.
UU	*Ureaplasma urealyticum.* Clasificado como positivo o negativo durante el diagnóstico.
CrsipatusCqRange, GasseriCqRange, JenseniiCqRange, and InersCqRange	Cq (Ciclo de cuantificación en el termociclador) valor para el cual se detecta la densidad de crecimiento. Clasificado como densidad de crecimiento indetectable, baja y alta.

Esta investigación estudia el conjunto de datos original (descrito en la Sección 4.1). Este conjunto de datos se caractariza porque presenta un marcado desbalanceo en sus clases. Este desbalanceo es consecuencia de que en la clínica es muy común observar que un mayor número de pacientes presenten un diagnóstico negativo para una infección en comparación con el número de pacientes que presentan un diagnóstico positivo.
También se estudia un subconjunto del conjunto de datos original. En este subconjunto se descartó la clase indeterminada ya que nuestro objetivo

fue determinar que bacterias están interactuando entre sí para desarrollar la vaginosis bacteriana y la clase indeterminada no aporta mucho a nuestro objetivo.
El subconjunto resultante solo contiene dos clases (positiva 51 casos y negativa 134 casos). Al igual que el conjunto de datos original también este conjunto de datos esta desbalanceado. Para realizar la tarea de balanceo y evitar que los algoritmos usados para crear reglas de asociación sesguen su análisis hacia la clase mayoritaria este subconjunto de datos se sometió a balanceo. En la siguiente sección se describe el proceso de balanceo.

4.3. Proceso de balanceo

Para el balanceo la función *SMOTE(X = casoPosNeg[, -9], target = casoPosNeg$VBPCR, K = 9, dup_size = 0)* usa los parámetros establecidos. El parámetro X representa el conjunto de datos, target la clase a balancear, K es el número de vecinos más cercanos y *dup_size* indica cuántas veces la función SMOTE hará un bucle en la instancia original. La función *ADAS(X = casoPosNeg[, -9], target = casoPosNeg$VBPCR, K = 9)* usa los parámetros establecidos en la función. X representa el conjunto de datos, *target* la clase a balancear y K el número de vecinos más cercanos. Aquí en este parámetro se evaluaron los distintos valores de K para determinar el valor apropiado para los algoritmos de balanceo.
El valor K determina la base para interpolar ejemplos de sobremuestreo sintético en el espacio de características del vecindario a partir de las instancias minoritarias. Este parámetro K no esta presente en el algoritmo ROSE, este algoritmo balancea con la función *ovun.sample(VBPCR ~., data = casoPosNeg, method = over, p = 0.5, seed = 1)*. El parámetro VBPCR representa la clase a balancear, data el conjunto de datos, *method* indica que se haga un sobremuestreo, el valor p representa la probabilidad de remuestrear la clase minoritaria y *seed* es interpretado como un número entero, para especificar las semillas y mantener un seguimiento de la muestra.
Para determinar cuál es el valor apropiado de K para el algoritmo SMOTE y ADASYN se exploró el siguiente intervalo: $K = i$ donde $i = 3$, $i = 5$, $i = 7$, $i = 9$ e $i = 11$. Se utilizó la función *randomForest(VBPCR~., data=prueba, proximity=TRUE)* con los parámetros establecidos para esta función para realizar esta tarea con un conjunto de entrenamiento del 66 % y un conjunto de prueba del 34 %. Este algoritmo evalúa proximidades entre puntos de da-

tos y a partir de esa evaluación determina el valor K apropiado o con mejor rendimiento.
El algoritmo SMOTE balanceó al conjunto de datos con los siguientes parámetros: X = casoPosNeg[, -9], *target* = casoPosNeg$VBPCR, $K = 9$, *dup_size* = 0.
El algoritmo ADASYN balanceó el conjunto de datos con los siguientes parámetros: X = casoPosNeg[, -9], *target* = casoPosNeg$VBPCR, $K = 9$. Para determinar el valor K para el algoritmo ADASYN se usó el mismo procedimiento que SMOTE.
El algoritmo ROSE balanceó al conjunto de datos con los siguientes parámetros: VBPCR ~., *data* = casoPosNeg, *method* = over, $p = 0.5$, *seed* = 1.
Con respecto al algoritmo ROSE el valor p fue de 0.5, el valor predeterminado en la función. Estos algoritmos realizaron la tarea de balanceo sobremuestreando la clase minoritaria con 51 casos con respecto a la clase mayoritaria con 134 casos.

4.4. Determinación del umbral de soporte

Para crear las reglas de asociación, primero los algoritmos deben encontrar los conjuntos de elementos frecuentes. Estos conjuntos de elementos frecuentes son encontrados según el umbral de soporte asignado al algoritmo. Para determinar el umbral de soporte adecuado se exploraron porcentajes que van en el intervalo de 2 % hasta 50 %. Por cada porcentaje se realizó una corrida del algoritmo. A partir de los conjuntos de elementos frecuenctes se crearon las reglas. En esta etapa el porcentaje para la confianza se fijó en 80 % ya que lo que se esta investigando es el porcentaje para el soporte.
Estos porcentajes permitieron a los algoritmos crear los conjuntos de elementos frecuentes (aquellos que están con una frecuencia mayor o igual al umbral establecido).
El objetivo es encontrar conjuntos de elementos frecuentes que no sean redundantes o, en otras palabras, que sean específicos. Las métricas de calidad permiten evaluar esa redundancia o especificidad. También las funciones del paquete ARules permiten evaluar la redundancia de las reglas creadas a partir de conjuntos de elementos frecuentes redundantes.
En esta búsqueda experimental se determinó que el soporte de 7 % es el adecuado para el conjunto de datos original (desbalanceado) y el soporte de 14 % para el subconjunto de datos balanceado.

Es imprescindible explorar diferentes porcentajes de soporte para determinar el soporte adecuado para crear las reglas de asociación. Este porcentaje depende de las características del conjunto de datos, particularmente si hay desbalanceo.

4.5. Determinación del umbral de confianza

Explorado los diferentes porcentajes de soporte y determinado que el porcentaje de 7 % para el conjunto de datos original (desbalanceado) es el mejor, se exploraron para la confianza los porcentajes de 80 %, 85 %, 90 % y 95 %. En esta etapa el porcentaje para el soporte se fijó en 7 % ya que lo que se está investigando es el porcentaje para la confianza.
A partir de estos porcentajes se crearon las reglas y se exploraron hasta encontrar el porcentaje adecuado. Las reglas creadas como consecuencia de cada porcentaje de confianza investigado fueron analizadas según el valor reportado por cada métrica de calidad.
Como resultado de esta exploración experimental se determinó el umbral de confianza en 90 % para cada uno de los conjuntos de datos estudiados en esta investigación. Este porcentaje en la confianza permitió a los algoritmos crear las reglas de asociación (aquellas en las que la probabilidad de ocurrencia es mayor o igual al umbral establecido).

4.6. Creación y selección de reglas de asociación

Para cada conjunto de datos (original desbalanceado y subconjunto balanceado) con los soportes y confianza previamente investigados se corrieron los algoritmos de reglas de asociación Apriori, Eclat y FP-growth. La función *apriori(tr, parameter = list(supp = 0.05, conf = 0.9, minlen = 2, target = rules"), appearance = list(rhs = "VaginosisPos"))* con los parámetros establecidos crea las reglas de asociación. El parámetro *tr* es el conjunto de datos de tipo transacción. En el parámetro *supp* se exploraron todos los porcentajes de soporte. El parámetro *minlen* se refiere al número de elementos en el antecedente de la regla. El parámetro *target* le dice a la función que cree reglas de asociación y el parámetro *appearence* hace que la función solo reporte reglas que en su consecuente tengan el elemento VaginosisPos.

La función *eclat(tr, parameter = list(supp = 0.07, maxlen = 5)* con sus parámetros extrae los conjuntos de elementos frecuentes. El parámetro *supp* establece el porcentaje de soporte para extraer los conjuntos de elementos frecuentes y el parámetro *maxlen* la longitud máxima de elementos frecuentes.

Para extraer las reglas de asociación a partir de los conjuntos de elementos frecuentes creados por el algoritmo ECLAT se usó la función *ruleInduction(itemsets, confidence = .9)* para generar reglas a partir de los conjuntos de elementos encontrados.

El parámetro *itemsets* representa los conjuntos de elementos frecuentes y *confianza* representa a la métrica con la que la función se guía para crear las reglas. Por lo tanto el soporte se usa primero para encontrar conjuntos de elementos frecuentes (significativos) con el algoritmo ECLAT. Luego, la confianza se usa en un segundo paso para crear reglas a partir de los conjuntos de elementos frecuentes que exceden un mínimo umbral de confianza con la función *ruleInduction()*.

Para ejecutar el algoritmo FP-Growth en el lenguaje de programación R a través del paquete ARules se hace a través de la función *fim4r(tr, method = fpgrowth, target = rules, supp = .7, conf = .90)*. Para que cree las reglas necesita los parámetros ya establecidos en la función. El parámetro *tr* representa el conjunto de datos tipo transacción, *method* representa al algoritmo que en este caso es el algoritmo FP-Growth, el parámetro *target* le pide al algoritmo que cree reglas de asociación, *supp* representa el umbral de soporte para buscar los conjuntos de elementos frecuentes y *conf* representa el umbral de confianza para que el algoritmo cree las reglas, (ver figura 4.1).

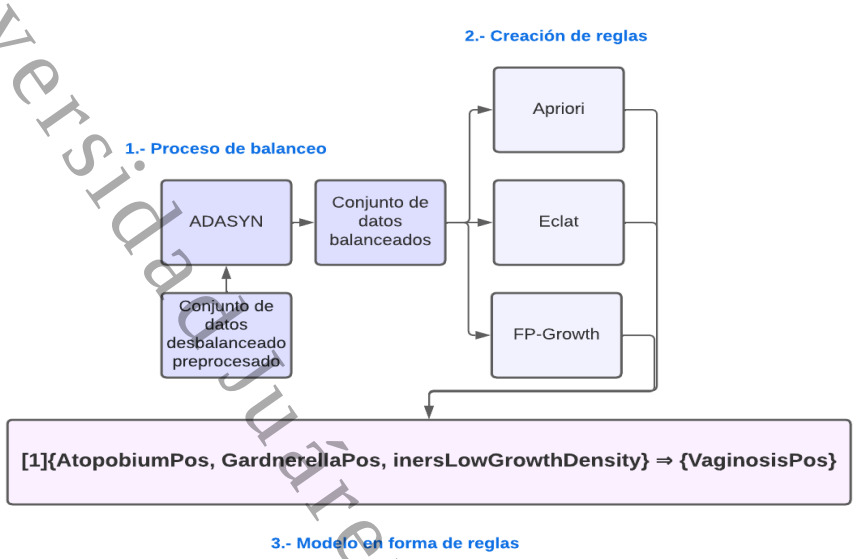

Figura 4.1. Algoritmos para creación de regls de asociación.

El problema con estos algoritmos es que crean demasiadas reglas en la fase de explosión combinatoria y principalmente si el porcentaje de soporte es bajo. Para resolver este problema por una parte se usan en esta investigación las métricas de calidad hiper-confianza, lift, hyper-lift, convicción, coseno, factor de poder de la regla, gini, y prueba exacta de Fisher. La función *interestMeasure(rules, c(hyperConfidence, conviction, rulePowerFactor, hyperLift, cosine, gini, fishersExactTest), transactions = tr)* calculó cada métrica de calidad para cada regla de asociación como se puede ver en los parámetros que acepta la función. El parámetro *rules* es el conjunto de reglas creadas a las que se les va a calcular cada métrica de calidad. El siguiente parámetro es un vector con todas las métricas de calidad que se van a calcular para cada regla del conjunto Idem. El último parámetro en la función es el conjunto de datos transaccional. El conjunto con los valores de cada métrica de calidad se unió al conjunto con cada regla de asociación con la función *cbind()*.
Las métricas de calidad con el valor que reportan para cada regla de asociación son el medio para validar o seleccionar las reglas que son significativas. Cada métrica va a evaluar el interés del patron de asociación y a partir de este se va a rechazar una regla o se va a aceptar. Esta evaluación va a depender de los valores reportados por cada métrica según un intervalo que por lo

general oscila entre [0, 1]. Esta tarea es un paso imprescindible en el filtrado de reglas.
Por otra el paquete ARules proporciona algunas funciones de filtrado que evalúan la redundancia o significancia de una regla de asociación tales como:

Función *is.redundant()*

La función *is.redundant()* se apoya en una métrica de calidad para seleccionar reglas no redundantes. Esta función itera sobre el conjunto de reglas de asociación y determina si la regla se descarta o se conserva según el valor reportado por la métrica de calidad y devuelve un conjunto con valores falso para reglas no redundantes y verdadero para reglas redundantes.

Función *is.significant()*

La función *is.significant()* usa el método de Fisher, un valor α de 0.01 y el ajuste de Bonferroni para corregir el error que surge de las multiples comparaciones. Cada regla representa una prueba estadística y puede ser necesaria la corrección. Esta función determina cuál regla es estadísticamente significativa y cuál no.

Función *is.maximal()*

La función *is.maximal()* reporta solo conjuntos de reglas que son maximales. Una regla maximal es creada con un conjunto de elementos frecuentes maximales y este conjunto es aquel que no está incluido en ningún otro conjunto de elementos frecuentes.

Como resultado de la selección computacional con las métricas de calidad y funciones se tiene un conjunto de reglas estadisticamente significativas más no biologicamente significativas, (ver figura 4.3).

4.7. Validación biológica

El conjunto con las reglas estadísticamente significativas fue sometido a la inspección biológica por un experto en biología el cuál fue el responsable de la recolección de los datos bajo estudio, para determinar si los patrones descritos en las reglas representan un comportamiento biológico como se observa en la clínica.

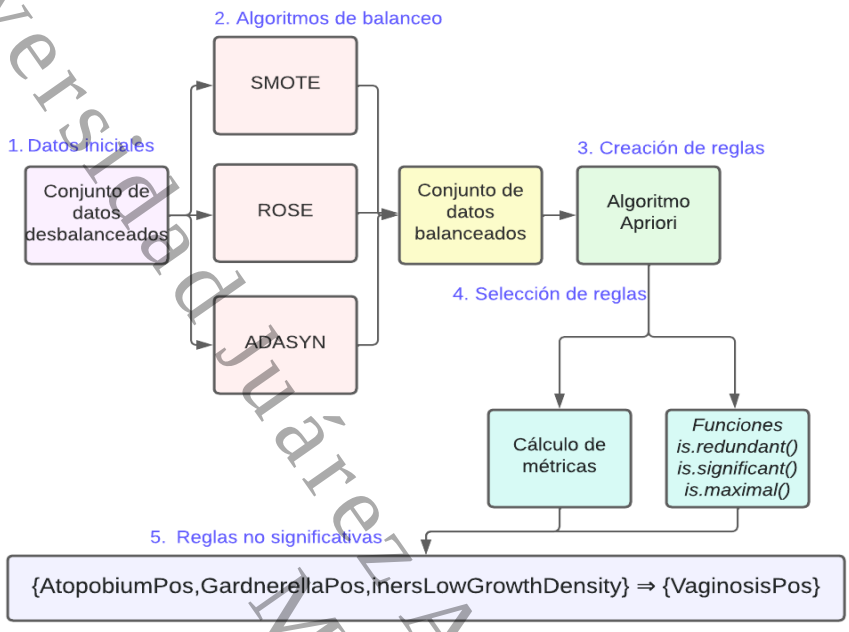

Figura 4.2. Proceso de balanceo y selección de reglas.

4.8. Proceso experimental

El trabajo experimental desarrollado en esta investigación se realizó de manera secuencial/ordenada a como se presenta en la (figura 4.3).

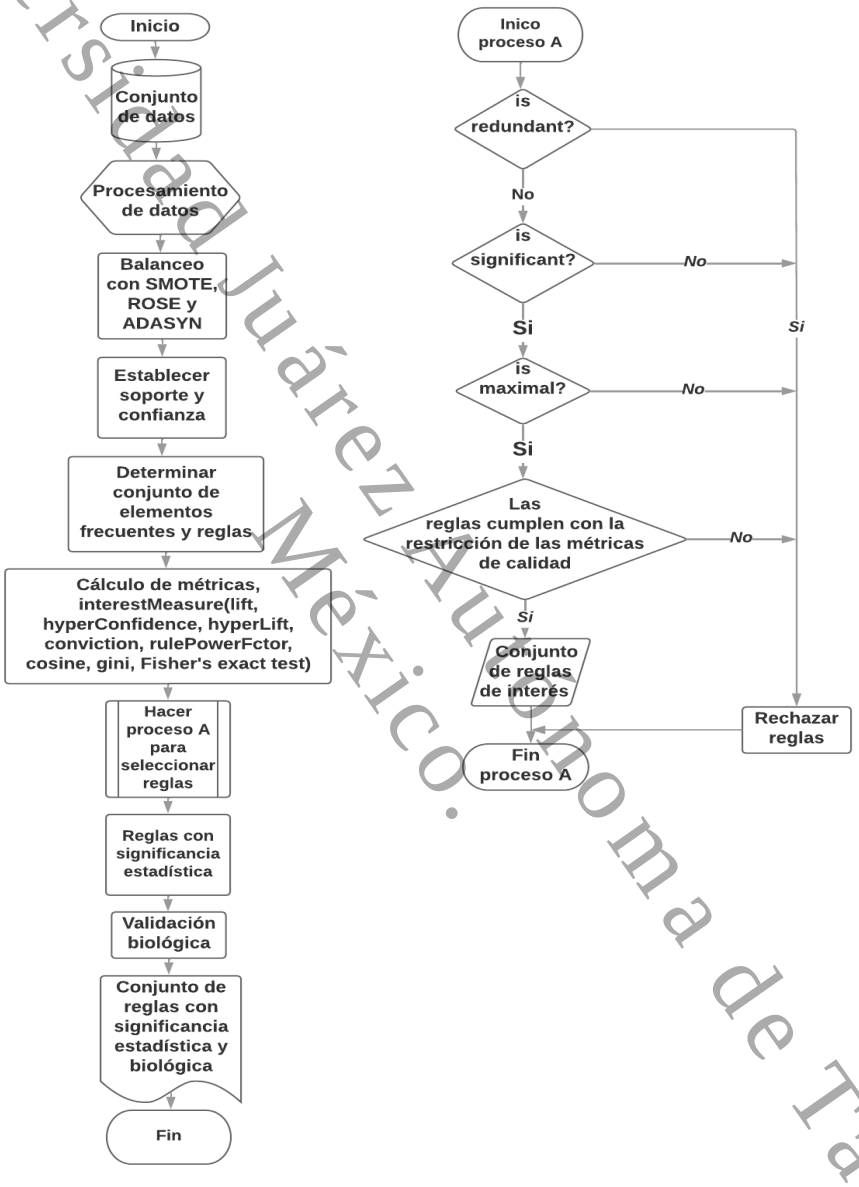

Figura 4.3. Proceso experimental realizado con los algoritmos Apriori, Eclat y FP-Growth.

Capítulo 5

Resultados

Esta sección describe los resultados de las tres etapas de experimentos realizados en esta investigación. Se presenta el modelo de reglas de asociación y se discute la descripción que el modelo realiza con lo que se ha reportado en la clínica.

En la **primera etapa de experimentos** se determina el porcentaje apropiado para el soporte y la confianza para crear las reglas de asociación. También se investiga la mejor métrica de calidad para filtrar las reglas creadas con los algoritmos. Las funciones *is.redundant()*, *is.significant()* e *is.maximal()* del paquete ARules también fueron analizadas para realizar la validación estadística y la selección de las reglas de asociación. Aquí se explora la factibilidad de las reglas de asociación para estudiar a la vaginosis bacteriana.

Durante el diagnóstico de las enfermedades es común observar que el mayor número de pacientes van a tener un diagnóstico negativo comparado con el diagnóstico positivo. Esta característica trae como consecuencia que en los conjuntos de datos haya una clase minoritaria y una clase mayoritaria. Por lo tanto en la **segunda etapa de experimentos** se aborda el desbalanceo que presenta el conjunto de datos bajo estudio. La consecuencia de este desbalanceo es que los algoritmos de minería de datos pueden estar sesgados hacia la clase mayoritaria. Por lo tanto en esta etapa se balancea para determinar si hay mejoras en la creación de reglas de asociación.

En la **tercera etapa de experimentos** se crean las reglas de asociación con los algoritmos Apriori, Eclat y FP-Growth con el conjunto de datos sin balancear y con el conjunto de datos balanceado para analizar cuál de ellos extrae los mejores patrones bacterianos. En esta última etapa de experimentos se usa el lenguaje de programación awk con expresiones regulares para

seleccionar solo las reglas que representan los patrones observados en la clínica.

5.1. *Etapa 1.* Reglas creadas con el conjunto de datos desbalanceado.

Apartir de la revisión en la literatura científica de cada métrica de calidad se determinó y seleccionó las métricas Hyperconfianza, Hyperlift, Lift, Convicción, Coseno, Indice Gini, Prueba exacta de Fisher y RPF [22, 24, 25, 26, 27]. Cada una de estas métricas de calidad evalúan la importancia de una regla de asociación según un intervalo de valores aceptables como se puede ver en la Tabla 5.1.

Tabla 5.1. Métricas de calidad

Métrica	Descripción	Intervalo
Hyperconfianza	Evalúa el nivel de confianza en observaciones con recuentos demasiado altos o bajos. Similar al p-valor de la prueba exacta de Fisher. Valores cercanos a 1 son valores aceptables.	[0, 1]
Hyperlift	Adaptación de la métrica lift, es más robusta para conteos bajos o raros. Los valores cercanos a 1 son valores aceptables.	[0, ∞]
Lift	Mide la frecuencia con la que X y Y ocurren juntos si fueran estadísticamente independientes. 1 significa independencia y >3 inicia ruido aleatorio.	[0, ∞]
Convicción	Evalúa el grado en que el antecedente influye en la ocurrencia del consecuente. Si los elementos son independientes, la convicción es 1.	[0, ∞]
Coseno	Mide la similitud entre LHS y RHS. Los valores de 0,0 a <0,5 significan que no hay correlación.	[0, 1]
Indice Gini	Mide la probabilidad de que una variable en particular se clasifique incorrectamente cuando se elige al azar. 0 significa que la regla no proporciona ninguna información.	[0, 1]
Prueba exacta de Fisher	Prueba de significancia para identificar si una regla representa un patrón verdadero. Cuanto más cerca esté el p-valor de 0, menor será la probabilidad de observar una regla solo por aleatoriedad.	[0, 1]
RPF	Se centra en la importancia de la asociación entre el antecedente y el consecuente de la regla. Cuando aumenta la asociación del antecedente y el consecuente, aumenta la importancia de la regla.	[0, 1]

Para determinar cual de estas métricas selecciona los mejores patrones se

crearon las reglas de asociación para la clase positiva (VB+ 51 casos) con los soportes de 5 %, 10 % y 15 % como ya se describió en la Sección 4.4. La confianza se fijó en 90 % como ya se describió en la sección 5.5 para todos los experimentos realizados.
La función *apriori(tr, parameter = list(supp = 0.05, conf = 0.9, minlen = 2, target = rules"), appearance = list(rhs = "VaginosisPos"))* con los parámetros establecidos creó las reglas de asociación. El parámetro *tr* es el conjunto de datos de tipo transacción. En el parámetro *supp* se exploraron todos los porcentajes de soporte. El parámetro *minlen* se refiere al número de elementos en el antecedente de la regla. El parámetro *target* le dice a la función que cree reglas de asociación y el parámetro *appearence* hace que la función solo reporte reglas que en su consecuente tengan el elemento VaginosisPos.
La función *interestMeasure(rules, c(hyperConfidence, conviction, rulePowerFactor, hyperLift, cosine, gini, fishersExactTest), transactions = tr)* calculó cada métrica de calidad para cada regla de asociación como se puede ver en los parámetros que acepta la función. El parámetro *rules* es el conjunto de reglas creadas a las que se les va a calcular cada métrica de calidad. El siguiente parámetro es un vector con todas las métricas de calidad que se van a calcular para cada regla del conjunto rules. El último parámetro en la función es el conjunto de datos transaccional. El conjunto con los valores de cada métrica de calidad se unió al conjunto con cada regla de asociación con la función *cbind()*. Durante la selección se analizó cuál de las métricas de calidad seleccionaba patrones computacionalmente aceptables.
Las funciones *is.redundant()*, *is.significant()* e *is.maximal()* aportadas por el paquete ARules también fueron utilizadas para realizar la selección de reglas como se describe en la Sección 4.6. La función *is.redundant()* para seleccionar a las reglas necesita una métrica de calidad como uno de sus argumentos. Esta función itera sobre el conjunto de reglas de asociación y verifica el valor reportado por la métrica de calidad y si el valor reportado por la métrica esta en el intervalo aceptable, (ver Tabla 5.1) se conserva la regla de lo contrario se rechaza la regla, (ver figura 4.3).
Para la clase negativa (VB- 134 casos) y para la clase indeterminada (VB? 16 casos) se realizó el mismo procedimiento previamente descrito y lo único que cambió fueron los porcentajes para el soporte. Para la clase negativa los porcentajes de soporte fueron de 20 %, 35 % y 50 % y para la clase indeterminada fueron de 1 %, 2 % y 4 % como se describió en la sección 4.4. También el parámetro appearence de la función *apriori()* cambió. Para las reglas creadas con la clase negativa este parámetro cambió a VaginosisNeg y para la clase

indeterminada cambió a VaginosisInd.

5.1.1. Clase positiva VB+

Esta clase es minoritaria comparada con la clase negativa ya que los casos positivos son 51 comparado con los casos negativos que son 134. Esta característica puede sesgar al algoritmo Apriori hacia la clase mayoritaria. Los valores reportados por las 8 métricas de calidad de la (Tabla 5.1) van a determinar si una regla creada por el algoritmo Apriori es redundante o no lo es.
Las 8 métricas de calidad seleccionadas fueron diseñadas para tener en cuenta el desbalanceo que se presenta en los conjuntos de datos y se describen con mayor detalle en la sección 2.5.
Para cada porcentaje de soporte se crearon las reglas de asociación y se reportó el conjunto de reglas creadas. Se realizó la selección con las funciones y métricas de calidad y se reportó el número de reglas seleccionadas. Los valores reportados por las métricas de calidad se representan como el promedio del conjunto de reglas. Aquí el objetivo es determinar si el porcentaje de soporte usado para crear las reglas de asociación es el adecuado según el valor promedio reportado por cada métrica de calidad.
Los resultados del análisis de VB+ se compilan en las (Tablas 5.2, 5.3). En la primera columna de las Tablas se tiene el porcentaje de soporte/confianza. Para la clase positiva se evalúan los porcentajes de soporte de 5 %, 10 % y 15 %. La segunda columna de la (Tabla 5.2) compila el conjunto de reglas reportado por el algoritmo y en la (Tabla 5.3) el conjunto de reglas después de hacer la selección con las funciones y métricas de calidad. Las siguientes columnas reportan los valores de cada métrica de calidad. Las filas 1, 2 y 3 de la (Tabla 5.2) contienen los conjuntos de reglas por porcentaje de soporte sin hacer la selección. Las filas 1, 2 y 3 de la (Tabla 5.3) contienen los conjuntos de reglas por porcentaje de soporte seleccionadas con las métricas de calidad y las funciones del paquete ARules.
El algoritmo Apriori se ejecutó con soporte de 5 % y reportó 1477 reglas, ver fila 1 de la (Tabla 5.2). Tras realizar la selección con las métricas de calidad y funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.máximal()*) sólo 117 reglas fueron reportadas según los valores de cada métrica de calidad, ver fila 1 de la (Tabla 5.3).

Tabla 5.2. Conjunto de reglas originales creadas con la clase positiva.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] 0.05/0.9	1477	1	1.861	3.857	10.565	0.06987	0.5205	0.08321	1.143^{-07}
[2] 0.1/0.9	148	1	2.117	3.793	11.239	0.11432	0.6702	0.1417	1.091^{-13}
[3] 0.15/0.9	1	1	2.385	3.941	NaN	0.1542	0.7796	0.2031	2.919^{-23}

En la (Tabla 5.3) para el soporte de 5 % en la fila 1 se puede observar que las reglas reportadas después de la selección no cumplen con la restricción de todas las métricas ya que reportan un valor bajo, por ejemplo, la métrica coseno reportó un valor por debajo de 0.5. También se puede ver que el valor del lift es alto ya que reporta un valor de casi 4. Ir a la (Tabla 5.1) para consultar los valores de referencia de cada métrica de calidad. Con base en el resultado de las métricas, se puede decir que las reglas creadas con el soporte de 5 % es un porcentaje no adecuado para crear las reglas.

Tabla 5.3. Conjunto de reglas significativas por el método de Fisher ($\alpha = 0.01$) y ajuste de Bonferroni, no redundantes y maximales.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] 0.05/0.9	117	1	1.795	3.851	9.504	0.05923	0.4811	0.06943	2.412^{-07}
[2] 0.1/0.9	18	1	2.051	3.766	9.664	0.10325	0.6376	0.1256	3.965^{-13}
[3] 0.15/0.9	1	1	2.385	3.941	NaN	0.1542	0.7796	0.2031	2.919^{-23}

Para el soporte de 10 % se reportaron 148 reglas (fila 2 de la Tabla 5.2) y después de hacer la selección quedaron 18 reglas (fila 2 de la Tabla 5.3). Para este porcentaje de soporte, las reglas seleccionadas cumplen con la restricción de todas las métricas, incluso el lift tiende a disminuir. Para el soporte del 15 % solo se reporta una regla (fila 3 de la Tabla 5.2). Después de hacer la selección, la misma regla continúa (fila 3 de la Tabla 5.3). Todas las métricas excepto el lift en este porcentaje de soporte reportan valores aceptables.

5.1.2. Clase negativa VB-

Esta clase es mayoritaria ya que es la que tiene el mayor número de registros que son 134. Esta característica puede sesgar al algoritmo Apriori hacía la clase mayoritaria. Como se describió para la clase positiva las métricas de calidad describen las reglas creadas con el algoritmo Apriori.
Los valores reportados por las métricas de calidad se representan como el promedio del conjunto de reglas.
Los resultados del análisis de VB- se compilan en las (Tablas 5.4, 5.5). En la primera columna de esas Tablas se tiene el porcentaje de soporte/confianza. Para la clase negativa se evalúan los porcentajes de soporte de 20 %, 35 % y 50 %. La segunda columna de la (Tabla 5.4) compila el conjunto de reglas reportado por el algoritmo y en la (Tabla 5.5) el conjunto de reglas después de hacer la selección con las funciones y métricas de calidad. Las siguientes columnas reportan los valores de cada métrica de calidad. Las filas 1, 2 y 3 de la (Tabla 5.4) contienen los conjuntos de reglas por porcentaje de soporte sin hacer la selección. Las filas 1, 2 y 3 de la (Tabla 5.5) contienen los conjuntos de reglas por porcentaje de soporte seleccionadas con las métricas de calidad y las funciones del paquete ARules.
Para el soporte del 20 % el algoritmo reportó 1478 reglas, (fila 1 de la Tabla 5.4). Luego de hacer la selección con las métricas de calidad y funciones del paquete ARules el número de reglas se redujo a 96, (fila 1 de la Tabla 5.5).

Tabla 5.4. Conjunto de reglas originales creadas con la clase negativa.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] **0.20/0.9**	**1478**	1	1.225	1.447	8.782	0.2691	0.6293	0.07762	1.156^{-06}
[2] **0.35/0.9**	**239**	1	1.283	1.447	10.812	0.4197	0.7905	0.15373	5.731^{-11}
[3] **0.50/0.9**	**56**	1	1.318	1.452	12.502	0.5300	0.8913	0.2384	7.927^{-22}

Todas las métricas de calidad reportan valores ideales, incluso el lift se estabilizó. Ir a la (Tabla 5.1) para consultar los valores de referencia de cada métrica de calidad. La explicación de este comportamiento es el porcentaje de soporte utilizado. El porcentaje utilizado depende de las ocurrencias de la clase, ya que un conteo alto en las instancias de la clase permite utilizar un soporte alto. Mayores porcentajes de soporte implican reglas más específicas; sin embargo, los porcentajes altos no se pueden usar con clases que representan recuentos bajos en sus ocurrencias.

Tabla 5.5. Conjunto de reglas significativas por el método de Fisher ($\alpha = 0.01$) y ajuste de Bonferroni, no redundantes y maximales.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] **0.20/0.9**	**96**	1	1.184	1.422	7.675	0.2167	0.5689	0.05070	1.767^{-06}
[2] **0.35/0.9**	**20**	1	1.255	1.426	12.372	0.3729	0.7453	0.11974	3.033^{-06}
[3] **0.50/0.9**	**7**	1	1.302	1.435	10.505	0.5103	0.8744	0.2141	4.774^{-21}

El soporte del 35 % reportó 239 reglas (fila 2 de la Tabla 5.4), y estas disminuyeron a 20 reglas después de hacer la selección ver fila 2 de la (Tabla 5.5). Para el soporte del 50 % se crearon 56 reglas (fila 3 de la Tabla 5.4) y luego de hacer la selección con las métricas de calidad (hiper-confianza, lift, hyper-lift, convicción, coseno, rpf, gini, y Prueba exacta de Fisher) y funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*) quedaron 7 reglas (fila 3 de la Tabla 5.5). En la (Tabla 5.5) se puede observar que para estos porcentajes de soporte las métricas de calidad se estabilizan al máximo.
Cuando el conjunto de datos en estudio está balanceado, todas las métricas funcionan correctamente. Sin embargo, cuando el conjunto de datos está desbalanceado debido al recuento bajo en sus instancias, métricas como el lift informan valores fuera de rango.

5.1.3. Clase indeterminada VB?

Esta clase también es minoritaria ya que tiene solo 16 casos. El sesgo del algoritmo Apriori es muy marcado. Para esta clase la métrica lift reporta valores por arriba de 12 lo que refleja el grado de desbalanceo.
Para cada porcentaje de soporte se crean las reglas de asociación y se reporta el número de reglas creadas sin realizar la selección. Del mismo modo se realiza la selección con las funciones y métricas de calidad y se reporta el número de reglas. Los valores reportados por las métricas de calidad se representan como el promedio del conjunto de reglas. Aquí el objetivo es determinar el comportamiento del soporte al momento de crear las reglas.
Los resultados del análisis de VB se compilan en las (Tablas 5.6, 5.7). En la primera columna de las Tablas se tiene el porcentaje de soporte/confianza.

Para la clase indeterminada se evalúan los porcentajes de soporte de 1 %, 2 % y 4 %. La segunda columna de la (Tabla 5.6) compila el conjunto de reglas reportado por el algoritmo y en la (Tabla 5.7) se muestra el conjunto de reglas después de hacer la selección con las funciones y métricas de calidad. Las siguientes columnas reportan los valores de cada métrica de calidad. Las filas 1, 2 y 3 de la (Tabla 5.6) contienen los conjuntos de reglas por porcentaje de soporte sin hacer la selección. Las filas 1, 2 y 3 de la (Tabla 5.7) contienen los conjuntos de reglas por porcentaje de soporte seleccionadas con las métricas de calidad y las funciones del paquete ARules.
Para el soporte del 1 %, se crearon 2028 reglas, (fila 1 de la Tabla 5.6) de las cuales solo quedó una regla en el conjunto después de la selección, (fila 1 de la Tabla 5.7).

Tabla 5.6. Conjunto de reglas originales creadas con la clase indeterminada.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] 0.01/0.9	2028	0.9996	1.562	12.56	NaN	0.01554	0.4413	0.02676	3.741^{-04}
[2] 0.02/0.9	16	1	2.5	12.56	NaN	0.02488	0.559	0.04322	1.68^{-06}
[3] 0.04/0.9	0	0	0	0	0	0	0	0	0

La función *is.redundant()*, al iterar sobre el conjunto de reglas, descarta casi todas, ya que de las 2028 reglas producidas por el algoritmo solo queda 1. Esto muestra que el bajo soporte crea reglas altamente redundantes.

Tabla 5.7. Conjunto de reglas significativas por el método de Fisher ($\alpha = 0.01$) y ajuste de Bonferroni, no redundantes y maximales.

Supp/Conf	Reglas	HyperConf	HyperLift	Lift	Convicción	RPF	Coseno	Gini	Fisher
[1] 0.01/0.9	1	1	2.5	12.56	NaN	0.02488	0.559	0.04322	1.68^{-06}
[2] 0.02/0.9	1	1	2.5	12.56	NaN	0.02488	0.559	0.04322	1.68^{-06}
[3] 0.04/0.9	0	0	0	0	0	0	0	0	0

El lift se dispara al máximo, (Tablas 5.6, 5.7). Ir a la (Tabla 5.1) para consultar los valores de referencia de cada métrica de calidad. Para el soporte del 2 % se crearon 16 reglas, (fila 2 de la Tabla 5.6) y solo quedó una después

de la selección, (fila 2 de la Tabla 5.7). Para el soporte del 4 % se crearon 0 reglas, (fila 3 en las Tablas 5.6, 5.7). El bajo conteo significa que para este porcentaje de soporte no se encontró ningún conjunto de elementos ya que la frecuencia con la que están presentes en la base de datos transaccional es muy inferior al umbral mínimo establecido.
A partir de los experimentos realizados se observó que la métrica lift en conjuntos de datos balanceados, reporta valores ideales que oscilan entre 1 y 2. Sin embargo, en conjuntos de datos desbalanceados reporta valores fuera de rango por ejemplo de 4, 5, 12 según el grado de desbalanceo que existe. Con frecuencia el desbalanceo es consecuencia de que por lo general la gran mayoría de pacientes reporta un estado saludable comparado con los pacientes enfermos. Esta caraterística podría ser la principal causa por la cual se generan conjunto de datos desbalanceados.

5.1.4. Selección de reglas de asociación a partir del valor reportado por cada métrica de calidad

Para evaluar cuál de las métricas seleccionadas reporta el mayor número de reglas primero se ejecutó el algoritmo Apriori para crear el conjunto de reglas a partir de los conjuntos de elementos frecuentes. Con frecuencia el algoritmo Apriori crea una gran cantidad de reglas de asociación. De todas las reglas creadas la mayoría son redundantes o no significativas.
El medio para seleccionar reglas son las métricas de calidad. A través de la función *interesMeasure()* se calcularon las 8 métricas de calidad para cada regla creada por el algoritmo Apriori. El último paso de esta tarea fue unir los valores de cada métrica de calidad con su respectiva regla de asociación. Para seleccionar las reglas de asociación a partir del valor reportado por cada métrica de calidad; se hizo con la función *is.redundant(rules.sub, measure = fishersExactTest)*. El parámetro *rules.sub* contiene el conjunto de reglas de asociación. El parámetro *fishersExactTest* es la métrica de calidad que va a determinar cuál de las reglas del conjunto rules.sub se aceptan o rechazan.
La función compara la redundancia de la regla en función del valor calculado para la métrica de calidad. Posteriormente informa un vector de valores lógicos en el que hay dos valores posibles: Verdadero para una regla redundante y Falso para una regla no redundante.
Las reglas seleccionadas por cada métrica de calidad se compilan en las (Ta-

blas 5.8, 5.9) y tienen la siguiente estructura: analizar las Tablas por bloques horizontales para cada métrica de calidad. La fila 1 para la métrica hyperconfianza contiene a la clase vaginosis positiva para tres diferentes porcentajes de soporte. La fila 2 contiene el conjunto de reglas computacionalmente significativas por cada porcentaje de soporte evaluado. Para analizar a las clases vaginosis negativa e indeterminada seguir el mismo procedimiento que para la clase vaginosis positiva.

Tabla 5.8. Reglas reportadas por cada métrica de calidad y por cada clase según los % de soporte evaluados.

Hyperconfianza	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	16	8	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	18	6	3
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Hyperlift	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	26	14	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	18	6	3
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Lift	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	24	10	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	33	9	5
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Convicción	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	20	7	0
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	24	8	4
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	0	0	0

Para las demás métricas seguir el mismo procedimiento que para la métrica hyeperconfianza. Las (Tablas 5.8, 5.9) evalúan las reglas reportadas por

cada métrica de calidad y por cada clase según el % de soporte evaluado.

Tabla 5.9. Reglas reportadas por cada métrica de calidad y por cada clase según los % de soporte evaluados.

Factor de poder de la regla	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2]Reglas significativas	16	8	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	18	6	3
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Coseno	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	16	8	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	18	6	3
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Indice gini	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	16	8	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	18	6	3
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0
Prueba exacta de Fisher	% de soporte evaluados		
[1] **VB+**	**5 %**	**10 %**	**15 %**
[2] Reglas significativas	117	18	1
[3] **VB-**	**20 %**	**35 %**	**50 %**
[4] Reglas significativas	96	20	7
[5] **VB?**	**1 %**	**2 %**	**4 %**
[6] Reglas significativas	1	1	0

El principal objetivo de estas Tablas es determinar cuál de las métricas de calidad selecciona el mayor número de reglas según el valor calculado por la función *ineteresMeasure()* para cada regla a tavés de la función *is.redundant()*.
Las métricas hiperconfianza, factor de poder de la regla, coseno y gini mostraron el mismo resultado, consulte las (Tablas 5.8, 5.9). La métrica hiperlift solo varía con respecto a la hiperconfianza, el factor de poder de la regla, el coseno y el ídice gini en las reglas reportadas para la clase positiva (26, 14 y 1), en las demás clases tienen un desempeño idéntico, ver (Tablas 5.8, 5.9).

El desempeño de la métrica lift es diferente en términos del número de reglas reportadas para la clase positiva y negativa, sin embargo, para la clase indeterminada el desempeño es idéntico al de las otras métricas, consulte las (Tablas 5.8, 5.9).
La métrica convicción es la única que presenta un desempeño diferente de las demás métricas, por ejemplo, no reportan ninguna regla para la clase indeterminada después de hacer la selección, ver (Tabla 5.8).
De todas las métricas, la prueba exacta de Fisher es la que reporta el mayor número de reglas, ver la (Tabla 5.9). Esta métrica calcula la probabilidad exacta de un conjunto específico de frecuencias en Tablas de contingencia 2 * 2. La función *is.redundant()* itera sobre cada regla del conjunto sometido a selección, y cada regla se evalúa en la Tabla de contingencia utilizada por la prueba exacta de Fisher para seleccionar las reglas. Cada regla representa una prueba estadística. El hecho de calcular la frecuencia exacta de cada regla le permite a esta métrica seleccionar el mayor número de reglas comparada con las otras métricas de calidad.

5.1.5. Patrones implicados en el desarrollo de la vaginosis

En este apartado se analiza la clase vaginosis positiva ya que es de nuestro interés conocer los patrones bacterianos que la desarrollan. Para extraer los patrones bacterianos involucrados en el desarrollo de la vaginosis, se eligió el soporte de 7 % del intervalo analizado en la Sección 5.1.1. Elegimos este porcentaje para el soporte al observar que representaban los mejores patrones del experimento para la clase VB+ según la métrica Prueba exacta de Fisher. Se habla de mejores patrones del experimento para la clase VB+ en el sentido de que con este porcentaje de soporte se obtuvieron reglas específicas y no redundantes según las métricas de calidad.
Se extrajo un subconjunto de la base de datos transaccional conservando todas las variables excepto la variable **Lactobacillus** con el enfoque cualitativo, se conservó la variable **Lactobacillus** con el enfoque cuantitativo. La razón por la que se preservó la variable **Lactobacillus** con el enfoque cuantitativo fue la precisión con la calcula la densidad de crecimiento de estas bacterias durante la reacción de PCR en tiempo real. También se mantienen las clases VB+ con 51 instancias, la clase VB- con 134 instancias y la clase

VB con 16 instancias.
Dado que la métrica de calidad Prueba exacta de Fisher fue la que mejor desempeño tuvo, como se describió en la Sección 5.1.4, se eligió para realizar la selección de las reglas de asociación. Sin embargo, en los resultados de este experimento también se presentan los valores reportados por las otras métricas de calidad con la intensión de contrastar el resultado de cada métrica.
Los resultados de este experimento se compilan en la (Tabla 5.10) y presenta la siguiente estructura: en la columna 1 están registradas todas las reglas de asociación y en la columna 2 la frecuencia de cada regla en el conjunto de datos. Esta Tabla solo contiene las 2 columnas mencionadas. Las filas contienen la extención de las reglas de asociación. La regla es una implicación si ⇒ entonces, a la parte izquierda de la flecha se le conoce como antecedente (LHS por sus siglas en ingles) y a la parte derecha de la flecha se le conoce como consecuente (RHS por sus siglas en ingles). En el antecedente el algoritmo Apriori ubica a las bacterias que interactúan entre sí para desarrollar la infección. En el consecuente el algoritmo ubica a la infección que se desarrolla como consecuencia de la interacción bacteriana. En este caso es la vaginosis bacteriana positiva VB+; también contiene una f con un valor numérico. Esto se refiere a la frecuencia con la que la regla esta presente en el conjunto de datos.

Tabla 5.10. Reglas de asociación con significancia estadística y frecuencia (f) de la regla en el conjunto de datos. ♣

[1]{AtopobiumPos,MegasphaeraPos,MycoplasGeniNeg,UreaplasUreaNeg} ⇒	{VaginosisPos} f 25
[2]{AtopobiumPos,inersHighGrowthDensity,MegasphaeraPos,UreaplasUreaNeg} ⇒	{VaginosisPos} f 16
[3]{AtopobiumPos,GardnerellaPos,MegasphaeraNeg,MycoplasHomiNeg} ⇒	{VaginosisPos} f 16
[4]{AtopobiumPos,GardnerellaPos,MegasphaeraNeg,UreaplasUreaNeg} ⇒	{VaginosisPos} f 18
[5]{AtopobiumPos,gasseriUndetectable,MycoplasGeniNeg,UreaplasUreaNeg} ⇒	{VaginosisPos} f 22
[6]{AtopobiumPos,GardnerellaNeg,MegasphaeraPos} ⇒	{VaginosisPos} f 17
[7]{AtopobiumPos,GardnerellaPos,inersHighGrowthDensity} ⇒	{VaginosisPos} f 17
[8]{AtopobiumPos,GardnerellaPos,MycoplasGeniNeg,UreaplasParPos} ⇒	{VaginosisPos} f 17
[9]{AtopobiumPos,GardnerellaPos,MycoplasGeniNeg,MycoplasHomiNeg,UreaplasUreaNeg} ⇒	{VaginosisPos} f 17
[10]{AtopobiumPos,MegasphaeraPos,UreaplasParPos} ⇒	{VaginosisPos} f 15
[11]{AtopobiumPos,gasseriUndetectable,MegasphaeraPos} ⇒	{VaginosisPos} f 15
[12]{AtopobiumPos,GardnerellaPos,gasseriUndetectable} ⇒	{VaginosisPos} f 15
[13]{AtopobiumPos,GardnerellaPos,UreaplasParPos,UreaplasUreaNeg} ⇒	{VaginosisPos} f 15
[14]{AtopobiumPos,crisLowGrowthDensity,GardnerellaPos,UreaplasUreaNeg} ⇒	{VaginosisPos} f 15
[15]{AtopobiumPos,MycoplasHomiPos,UreaplasUreaNeg} ⇒	{VaginosisPos} f 18
[16]{AtopobiumPos,inersHighGrowthDensity,UreaplasParPos} ⇒	{VaginosisPos} f 16
[17]{AtopobiumPos,inersHighGrowthDensity,MycoplasHomiNeg} ⇒	{VaginosisPos} f 16

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

Cargado el conjunto de datos bajo estudio en memoria como un objeto de

tipo transacción, el algoritmo Apriori leyó 201 transacciones y 29 elementos. Al crear las reglas con el soporte del 7 % y confianza del 90 %, el número mínimo de elementos igual a 2 en el antecedente y restringido el consecuente al elemento VaginosisPos, el algoritmo creó 58 reglas.
Después de la selección de las reglas creadas con las funciones del paquete ARules y la métrica de calidad prueba exacta de Fihser, solo quedaron 17 reglas en el conjunto, ver (Tabla 5.10). Debido a que la (Tabla 5.10) es muy densa los valores de cada métrica de calidad calculados para cada regla de asociación se reportan en la (Tabla 5.11).
Estas 17 reglas son significativas estadísticamente o aceptables desde el punto de vista computacional, sin embargo, aún se debe someter a este conjunto a selección biológica. Dicha selección es realizada por el experto en Biología y su tarea es validar que los patrones detectados por el algoritmo Apriori describan lo que se observa en la clínica.
Con respecto a cada métrica calculada para cada regla, la (Tabla 5.11) muestra valores aceptables en los datos reportados, excepto la métrica lift que reporta valores altos (valores de casi 4 unidades).

Tabla 5.11. Métricas de las 17 reglas con significancia estadística.

No.	Lift	Hyperconf	Hyperlift	Convicción	RPF	Coseno	Gini	Prueba exacta de Fisher
[1]	**3.941176**	1	2.272727	NA	0.12437811	0.7001400	0.15821503	4.801688^{-18}
[2]	**3.941176**	1	2.000000	NA	0.07960199	0.5601120	0.09633157	3.903820^{-11}
[3]	**3.941176**	1	2.000000	NA	0.07960199	0.5601120	0.09633157	3.903820^{-11}
[4]	**3.941176**	1	2.000000	NA	0.08955224	0.5940885	0.10955742	1.364731^{-12}
[5]	**3.612745**	1	2.000000	8.955224	0.10033167	0.6288281	0.11918190	2.432086^{-13}
[6]	**3.941176**	1	1.888889	NA	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[7]	**3.941176**	1	1.888889	NA	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[8]	**3.941176**	1	1.888889	NA	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[9]	**3.941176**	1	1.888889	NA	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[10]	**3.941176**	1	1.875000	NA	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[11]	**3.941176**	1	1.875000	NA	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[12]	**3.941176**	1	1.875000	NA	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[13]	**3.941176**	1	1.875000	NA	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[14]	**3.941176**	1	1.875000	NA	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[15]	**3.547059**	1	1.800000	7.462687	0.08059701	0.5636019	0.09230125	1.781025^{-10}
[16]	**3.709343**	1	1.777778	12.686567	0.07491952	0.5433885	0.08732471	5.454796^{-10}
[17]	**3.709343**	1	1.777778	12.686567	0.07491952	0.5433885	0.08732471	5.454796^{-10}

Ir a la (Tabla 5.1) para consultar los valores de referencia de cada métrica de calidad. Sin embargo, la métrica hiperlift corrige el sesgo de la métrica lift e informa valores más realistas.
El hiperlift es una adaptación de la métrica lift donde en lugar de dividir por el conteo esperado bajo independencia ($E[C_{XY}] = n_X/n \times n_Y/n$) se utiliza

un cuantil más alto de la distribución de conteo hipergeométrico. Esto es más sólido para conteos bajos y da como resultado menos falsos positivos cuando se usa hiper-lift para el filtrado de reglas.
La validación biológica del conjunto de 17 reglas de asociación resultó en solo 5 reglas, ver (Tabla 5.12). Para la validación biológica, el experto (biólogo) analizó los patrones de cada regla para determinar si representaban el comportamiento bacteriano observado en la clínica.
Esta inspección por parte de este experto permitió descartar reglas de asociación que aunque eran estadísticamente significativas o computacionalmente aceptables no eran biológicamente significativas, ver (Tabla 5.12).

Tabla 5.12. Reglas con significancia estadística y biológica y frecuencia (f) de la regla de asociación en el conjunto de datos. ♣

[1]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒	{VaginosisPos} f 17
[2]{AtopobiumPos, MegasphaeraPos, UreaplasmaParPos} ⇒	{VaginosisPos} f 15
[3]{AtopobiumPos, gasseriUndetectable, MegasphaeraPos} ⇒	{VaginosisPos} f 15
[4]{AtopobiumPos, GardnerellaPos, gasseriUndetectable} ⇒	{VaginosisPos} f 15
[5]{AtopobiumPos, inersHighGrowthDensity, UreaplasmaParPos} ⇒	{VaginosisPos} f 16

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

¿Qué representa cada regla? La *regla número uno* dice que ***Atopobium vaginae*** (AtopobiumPos) y ***Gardnerella vaginalis*** (GardnerellaPos) deben estar presentes en la mucosa vaginal de la paciente junto con ***Lactobacillus iners*** (inersHighGrowthDensity) a una alta densidad de crecimiento para detonar VB+.
Lactobacillus iners (inersHighGrowthDensity) ha sido reportado en la literatura médica [3, 5, 6, 7, 8, 10] asociado con flora normal alterada, por lo que predispone al desarrollo de vaginosis.
La *regla número dos* nos dice que ***Atopobium vaginae*** (AtopobiumPos), ***Megasphaera filotipo 1*** (MegasphaeraPos) y ***Ureaplasma parvum*** (UreaplasParPos) deben estar presentes para desarrollar vaginosis bacteriana, (ver Tabla 5.12). La coexistencia de ***Atopobium vaginae*** (AtopobiumPos) y ***Megasphaera filotipo 1*** (MegasphaeraPos)) provoca el desarrollo de vaginosis bacteriana siempre que no este presente ***Lactobacillus gasseri*** (gasseriIndetectable), como se puede observar en la *regla número tres*.
Puede coexistir ***Atopobium vaginae*** (AtopobiumPos) y ***Gardnerella vaginalis*** (GardnerellaPos) para desarrollar vaginosis bacteriana siempre que

Lactobacillus gasseri (gasseriIndetectable) sea indetectable en la mucosa vaginal, *regla número cuatro*.
Atopobium vaginae (AtopobiumPos) y ***Ureaplasma parvum*** (UreaplasParPos) deben estar presentes en la mucosa vaginal de la paciente, ***Lactobacillus iners*** (inersHighGrowthDensity) también debe estar presente en una alta densidad de crecimiento para desarrollar vaginosis bacteriana como se muestra en la *regla cinco*, consulte la (Tabla 5.12).
El coeficiente de correlación de Pearson informó que ***Atopobium vaginae*** (AtopobiumPos) es una de las bacterias más significativas durante el diagnóstico de vaginosis, (ver Tabla 5.13). Las primeras 2 columnas de esta Tabla contienen el intervalo de valores y la correlación a la que corresponde cada intervalo. Las columnas 3 y 4 contienen el valor de la asociación de Pearson para cada bacteria. El valor de Pearson que contiene la etiqueta del número 1 se corresponde con la bacteria con la etiqueta del número 1 que es ***Mycoplasma genitalium***. La (Tabla 5.12) registra 5 reglas, en las 5 reglas ***Atopobium vaginae*** (AtopobiumPos) está presente. Esta bacteria es importante en el desarrollo de la vaginosis bacteriana desde el punto de vista biológico.

Tabla 5.13. Asociación lineal entre las bacterias asociadas con vaginosis y el diagnóstico de vaginosis.

Intervalo	Descripción	Pearson	Bacterias
0.00 - 0.09	Correlación nula.	1.- 0.059347656, 2.- 0.055255975, 3.- 0.082835156.	1.- Mycoplasma genitalium, 2.- Ureaplasma parvum, 3.- Ureaplasma urealyticum.
0.10 - 0.19	Correlación muy débil		
0.20 - 0.49	Correlación débil	4.- 0.344004222, 5.- 0.357351802, 6.- 0.362987094	4.- Gardnerella vaginalis, 5.- Mycoplasma hominis, 6.- Megasphaera phylotype 1.
0.50 - 0.69	Correlación moderada		
0.70 - 0.84	Correlación significativa	7.- 0.750937221	7.- Atopobium vaginae
0.85 - 0.95	Correlación fuerte		
0.96 - 1.0	Correlación perfecta		

5.1.6. Visualización basada en grafos

Se utilizó la visualización basada en grafos para representar las reglas de la (Tabla 5.12). Los elementos que forman parte del antecedente apuntan al círculo marcado con el número de regla y desde este círculo se apunta al consecuente de la regla, como se muestra en la (Figura 5.1).
El círculo con la etiqueta rule 1 en el grafo se corresponde con la flecha de la implicación de la regla número 1. En el grafo las bacterias que forman parte del LHS tienen flechas de color azul que entran al círculo con el número de etiqueta y de ese círculo sale una flecha roja que señala al RHS.

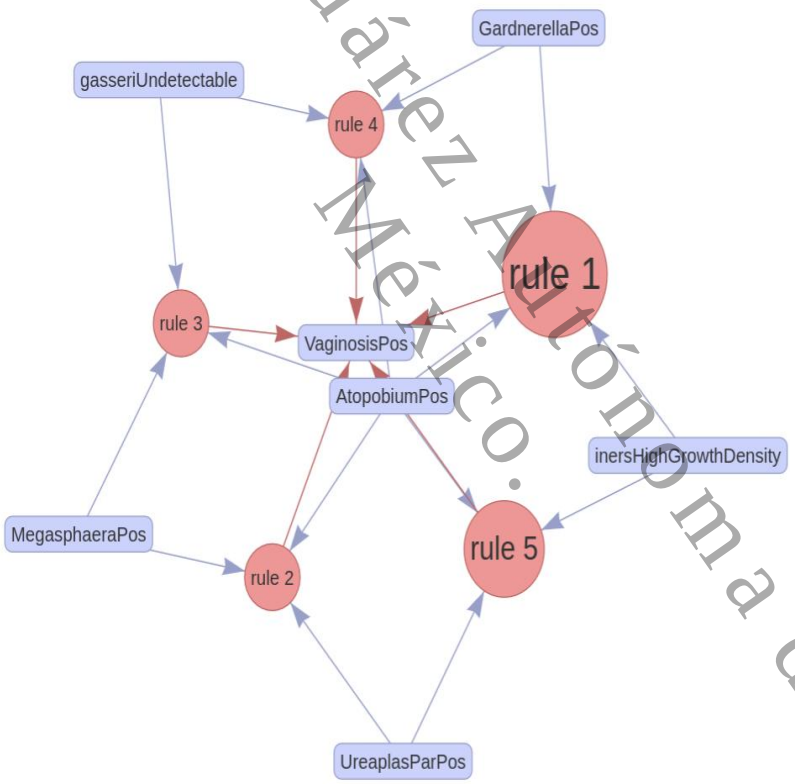

Figura 5.1. Reglas con significancia estadística y biológica.
LHS RHS
[1]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒ {VaginosisPos}
La etiqueta rule con el número se refiere a cada una de las reglas que en este gráfico son 5. Las aristas que entran en un nodo representan en conjunto al antecedente de la regla y las aristas que salen del nodo apuntan a la etiqueta con el consecuente de la regla.

5.2. *Etapa 2.* Reglas creadas con el conjunto de datos desbalanceado y balanceado

En esta etapa se explora si el balanceo del conjunto de datos en estudio mejora la creación de reglas de asociación.
En esta etapa experimental se usó el subconjunto de datos con las clases positiva 51 instancias y negativa 134 instancias. Los algoritmos de balanceo fueron SMOTE, ROSE y ADASYN. El algoritmo Random Forest se usó para determinar el valor K apropiado. Con el algoritmo Apriori se extrajeron las reglas de asociación.

5.2.1. Proceso de balanceo

Al explorar la distribución de clases en el conjunto de datos bajo estudio con la función *table()*, se puede ver en la fila 1 de la (Tabla 5.15) la distribución de clases.
Para el balanceo la función *SMOTE(X = casoPosNeg[, -9], target = casoPosNeg$VBPCR, K = 9, dup_size = 0)* usa los parámetros establecidos. El parámetro X representa el conjunto de datos, *target* la clase a balancear, K es el número de vecinos más cercanos y *dup_size* indica cuántas veces la función SMOTE hará un bucle en la instancia original. La función *ADAS(X = casoPosNeg[, -9], target = casoPosNeg$VBPCR, K = 9)* usa los parámetros establecidos en la función. X representa el conjunto de datos, *target* la clase a balancear y K el número de vecinos más cercanos. Aquí en este parámetro se evaluaron los distintos valores de K para determinar el valor apropiado para los algoritmos de balanceo.
El valor K determina la base para interpolar ejemplos de sobremuestreo sintético en el espacio de características del vecindario a partir de las instancias minoritarias. Este parámetro K no esta presente en el algoritmo ROSE, este algoritmo balancea con la función *ovun.sample(VBPCR ~., data = casoPosNeg, method = over, p = 0.5, seed = 1)*. El parámetro *VBPCR* representa la clase a balancear, *data* el conjunto de datos, *method* indica que se haga un sobremuestreo, el valor p representa la probabilidad de remuestrear la clase minoritaria y *seed* es interpretado como un número entero, para especificar las semillas y mantener un seguimiento de la muestra.
Para determinar cuál es el valor apropiado de K para el algoritmo SMO-

TE y ADASYN se exploró el siguiente intervalo: $K = i$ donde $i = 3$, $i = 5$, $i = 7$, $i = 9$ e $i = 11$. Se utilizó la función *randomForest(VBPCR~., data=prueba, proximity=TRUE)* con los parámetros establecidos para esta función para realizar esta tarea con un conjunto de entrenamiento del 66 % y un conjunto de prueba del 34 %. Este algoritmo evalúa proximidades entre puntos de datos y a partir de esa evaluación determina el valor K apropiado o con mejor rendimiento. El valor $K = 9$ tuvo el mejor rendimiento según el OOB error y el 1-OOB reportado por el algoritmo Random forest para el algoritmo SMOTE, ver fila 4 en la (Tabla 5.14).
El algoritmo SMOTE balanceó al conjunto de datos con los siguientes parámetros: X = casoPosNeg[, -9], *target* = casoPosNeg$VBPCR, $K = 9$, $dup\_size = 0$.
El algoritmo ADASYN balanceó el conjunto de datos con los siguientes parámetros: X = casoPosNeg[, -9], *target* = casoPosNeg$VBPCR, $K = 9$. Para determinar el valor K para el algoritmo ADASYN se usó el mismo procedimiento que SMOTE y el valor K apropiado fue igual a 9 según el OOB error y el 1-OOB reportado por el algoritmo Random forest para el algoritmo ADASYN, ver fila 9 en la (Tabla 5.14). Con respecto al algoritmo ROSE el valor p fue de 0.5, el valor predeterminado en la función. Estos algoritmos realizaron la tarea de balanceo sobremuestreando la clase minoritaria con 51 casos con respecto a la clase mayoritaria con 134 casos.

Tabla 5.14. Número de vecinos K-cercanos durante el proceso de muestreo.

Conjunto de datos balanceado	Conjunto de entrenamiento 66 %		Conjunto de prueba 34 %				
K-valor usado	OOB error	1-OOB	Precisión	Precisión del balanceo	Kappa	Sensibilidad	Especificidad
SMOTE							
[1] K = 3	3.8	96.2	1	1.0000	1	1.0000	1.0000
[2] K = 5	3.8	96.2	1	1.0000	1	1.0000	1.0000
[3] K = 7	3.8	96.2	1	1.0000	1	1.0000	1.0000
[4] K = 9	2.53	97.47	1	1.0000	1	1.0000	1.0000
[5] K = 11	5.06	94.94	1	1.0000	1	1.0000	1.0000
ADASYN							
[6] K = 3	2.2	97.8	1	1.0000	1	1.0000	1.0000
[7] K = 5	1.1	98.9	1	1.0000	1	1.0000	1.0000
[8] K = 7	1.1	98.9	1	1.0000	1	1.0000	1.0000
[9] K = 9	0	100	1	1.0000	1	1.0000	1.0000
[10] K = 11	1.09	98.91	1	1.0000	1	1.0000	1.0000

5.2.2. Resultados del proceso de balanceo

El nombre de casoPosNegP se refiere a una nomenclatura usada para identificar el procesamiento del conjunto de datos realizado. Este procesamiento tuvo varias etapas y en cada una el conjunto de datos tuvo varios nombres para identificar la etapa de procesamiento. Este procesamiento incluye desde tener el conjunto de datos en formato numérico y desbalanceado, el conjunto de datos en formato numérico balanceado y el conjunto de datos discretizado balanceado. El nombre casoPosNegP representa el conjunto de datos con todas las etapas de procesamiento realizadas.
La ejecución del algoritmo SMOTE con el subconjunto casoPosNegA 51/134 desbalanceado como uno de los argumentos necesarios para realizar el balanceo y K = 9 reportó el subconjunto casoPosNegP 102/134 balanceado, ver primera columna de la (Tabla 5.15).
El algoritmo ROSE al balancear el subconjunto casoPosNegB 51/134 no balanceado reportó el subconjunto casoPosNegP 132/134 balanceado, ver la segunda columna de la (Tabla 5.15).
Cuando el subconjunto casoPosNegC 51/134 desbalanceado se balanceo con el algoritmo ADASYN y K = 9, informó el subconjunto casoPosNegP 140/134 balanceado, consulte la tercera columna de la (Tabla 5.15).
En la (Tabla 5.15) se puede observar que los tres algoritmos de balanceo dejan fija la clase mayoritaria y sobremuestrean la clase minoritaria, evitando así la pérdida de información. El conjunto de datos original sobre vaginosis se balanceo con el algoritmo SMOTE incluido en el paquete DMwR versión 0.4.1 y se obtuvo el siguiente resultado, clases Positivo-Negativo: desbalanceado (51/134), balanceado (102/102). Hay sobremuestreo y submuestreo para emparejar las clases con la consecuencia de pérdida de información.
Observamos que el algoritmo SMOTE tuvo el rendimiento más bajo y el algoritmo ADASYN tuvo el mejor rendimiento. El algoritmo ROSE tuvo un desempeño intermedio, ver (Tabla 5.15).

Tabla 5.15. Conjunto de datos sobre vaginosis original en formato numérico y subconjunto balanceado en formato categórico.

Conjunto de datos original		
[1] Clase positiva 51	Clase negativa 134	Clase indeterminada 16
Subconjuntos con las clases positiva/negativa balanceados con los algoritmos de balanceo		
[2] SMOTE 102/134	ROSE 132/134	ADASYN 140/134

5.2.3. Resultados de la creación de reglas de asociación con significancia estadística y biológica

Las (Tablas 5.16, 5.17 y 5.18) solo muestran las métricas de calidad para cada regla de asociación creada con el algoritmo Apriori. Las reglas mostradas tienen significancia estadística y biológica. Las reglas para cada métrica se presentan en Tablas diferentes con el mismo número en cada Tabla con el fin de no cargar en exceso a las Tablas.

Las reglas creadas con los conjuntos de datos balanceados se compilan en las (Tablas 5.16, 5.17 y 5.18), tienen la siguiente estructura: la primera columna tiene el número de regla para el conjunto de reglas creadas con el conjunto de datos balanceado como primer conjunto y para el conjunto de reglas creadas con el conjunto de datos desbalanceado como segundo conjunto. En las siguientes columnas se compila el valor de cada métrica de calidad calculada para cada regla para el conjunto de reglas creadas con el conjunto de datos balanceado como primer conjunto y para el conjunto de reglas creadas con el conjunto de datos desbalanceado como segundo conjunto.

En la (Tabla 5.16) hay 2 conjuntos de reglas. El conjunto que va de la fila 1 hasta la 14 de la Tabla fue creado con el conjunto de datos balanceado. En la (Tabla 5.17) hay 2 conjuntos de reglas. El conjunto que va de la fila 1 hasta la 6 de la Tabla fue creado con el conjunto de datos balanceado. En la (Tabla 5.18) hay 2 conjuntos de reglas. El conjunto que va de la fila 1 hasta la 14 de la Tabla fue creado con el conjunto de datos balanceado. El conjunto de datos que va de la fila 1 a la 5 en las 3 Tablas mencionadas fue creado con el conjunto de datos desbalanceado.

El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo SMOTE, 12 % para el soporte y 90 % para la confianza creó 232 reglas de asociación. Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*), solo 39 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, y solo quedaron 14 reglas, ver el primer conjunto de la (Tabla 5.16).

El algoritmo Apriori con el conjunto de datos original, 7 % para el soporte y 90 % para la confianza creó 58 reglas de asociación. Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules, solo 17 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, quedaron 5 reglas, ver el segundo conjunto de la (Tabla 5.16).

Si comparamos los valores reportados para la métrica lift, en el segundo conjunto los valores son casi 4, lo que refleja que existe un desbalanceo en el conjunto de datos. En el primer conjunto, el valor más alto es 2.3, esto muestra que el balanceo mejora la creación de reglas, consulte la (Tabla 5.16).

Tabla 5.16. Reglas creadas con Apriori y balanceado con SMOTE, clases positiva 102/negativa 134. †

Número regla	Lift	hyper Conf	conviction	RPF	hyperLift	cosine	gini	fishersExactTest
[1]	**2.313725**	1	NA	0.1271186	1.578947	0.5423261	0.09390087	6.993240^{-13}
[2]	**2.313725**	1	NA	0.1228814	1.611111	0.5332108	0.09033234	1.983015^{-12}
[3]	**2.313725**	1	NA	0.1228814	1.611111	0.5332108	0.09033234	1.983015^{-12}
[4]	**2.313725**	1	NA	0.1313559	1.631579	0.5512908	0.09750422	2.444239^{-13}
[5]	**2.313725**	1	NA	0.1228814	1.611111	0.5332108	0.09033234	1.983015^{-12}
[6]	**2.247619**	1	19.872881	0.1399516	1.619048	0.5690426	0.10126102	2.325638^{-13}
[7]	**2.239089**	1	17.601695	0.1230180	1.578947	0.5335072	0.08673998	1.434633^{-11}
[8]	**2.239089**	1	17.601695	0.1230180	1.578947	0.5335072	0.08673998	1.434633^{-11}
[9]	**2.236601**	1	17.033898	0.1187853	1.526316	0.5242486	0.08319932	3.921004^{-11}
[10]	**2.181513**	1	9.936441	0.1318402	1.571429	0.5523061	0.09081447	7.744330^{-12}
[11]	**2.177624**	1	9.652542	0.1276171	1.523810	0.5433885	0.08720616	2.109994^{-11}
[12]	**2.173500**	1	9.368644	0.1233950	1.550000	0.5343239	0.08363541	5.686245^{-11}
[13]	**2.173500**	1	9.368644	0.1233950	1.550000	0.5343239	0.08363541	5.686245^{-11}
[14]	**2.096814**	1	6.056497	0.1113612	1.450000	0.5076015	0.07050045	2.798537^{-09}
Reglas creadas con el algoritmo Apriori con el conjunto de datos original desbalanceado. ‡								
[1]	**3.941176**	1	NA	0.08457711	1.888889	0.5773503	0.10290856	7.385605^{-12}
[2]	**3.941176**	1	NA	0.07462687	1.875000	0.5423261	0.08982531	2.016973^{-10}
[3]	**3.941176**	1	NA	0.07462687	1.875000	0.5423261	0.08982531	2.016973^{-10}
[4]	**3.941176**	1	NA	0.07462687	1.875000	0.5423261	0.08982531	2.016973^{-10}
[5]	**3.709343**	1	12.68657	0.07491952	1.777778	0.5433885	0.08732471	5.454796^{-10}

†El conjunto después del encabezado de la Tabla (primer conjunto) contiene los valores de cada métrica de calidad calculada para 14 reglas de asociación.
‡El conjunto después del subtitulo de la Tabla (segundo conjunto) contiene los valores de cada métrica de calidad calculada para 5 reglas de asociación.
♣ De los dos conjuntos de la Tabla, el primero tuvo la mejor calidad en comparación con el segundo, comparar los valores de la métrica lift.

El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo de balanceo ROSE, 17 % para el soporte y 90 % para la confianza creó 74 reglas de asociación. Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*), solo 18 reglas resultaron significativas.
El conjunto resultante se sometió a validación biológica, quedaron 6 reglas, ver el primer conjunto en la segunda fila de la (Tabla 5.17).
El algoritmo Apriori con el conjunto de datos original, 7 % para el soporte y 90 % para la confianza creó 58 reglas de asociación. Luego de hacer la se-

lección de las reglas con las métricas de calidad y las funciones del paquete ARules, solo 17 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, quedaron 5 reglas, ver el segundo conjunto de la (Tabla 5.17).

Tabla 5.17. Reglas creadas con Apriori y balanceado con ROSE, clases positiva 132/negativa 134. †

Número regla	Lift	conviction	hyperlift	hyper Conf	RPF	cosine	gini	fishersExactTest
[1]	**2.015152**	NA	1.516129	1	0.1766917	0.5967081	0.10892563	3.214640^{-17}
[2]	**1.972276**	23.676692	1.488871	1	0.1692529	0.5840122	0.09991885	2.386310^{-15}
[3]	**1.934545**	12.593985	1.500000	1	0.1732331	0.5908392	0.09957073	5.908434^{-15}
[4]	**1.891775**	8.228070	1.437500	1	0.1623446	0.5719694	0.08844258	3.538426^{-13}
[5]	**1.852639**	6.246617	1.461538	1	0.1970046	0.6300747	0.10881952	1.271290^{-15}
[6]	**1.828563**	5.440602	1.441176	1	0.1671540	0.5803797	0.08612396	1.415026^{-12}
Reglas creadas con el algoritmo Apriori con el conjunto de datos original desbalanceado. ‡								
[1]	**3.941176**	NA	1.888889	1	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[2]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[3]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[4]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[5]	**3.709343**	12.68657	1.777778	1	0.07491952	0.5433885	0.08732471	5.454796^{-10}

†El conjunto después del encabezado de la Tabla (primer conjunto) contiene los valores de cada métrica de calidad calculada para 6 reglas de asociación.
‡El conjunto después del subtitulo de la Tabla (segundo conjunto) contiene los valores de cada métrica de calidad calculada para 5 reglas de asociación.
♣ De los dos conjuntos de la Tabla, el primero tuvo la mejor calidad en comparación con el segundo, comparar los valores de la métrica lift.

Al igual que con el subconjunto balanceado con SMOTE, el subconjunto balanceado con ROSE mejoró la creación de reglas de asociación. La métrica lift incluso mejoró sus valores, que oscilan entre 1,8 y 2, consulte la columna 2 de la (Tabla 5.17).
El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo de balanceo ADASYN, 14 % para el soporte y 90 % para la confianza creó 341 reglas de asociación.
Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules(*is.redundant()*, *is.significant()* e *is.maximal()*), solo 46 reglas resultaron significativas desde el punto de vista estadístico y computacional. El conjunto resultante se sometió a validación biológica por el experto, y solo quedaron 14 reglas, ver el primer conjunto de la (Tabla 5.18).
El algoritmo Apriori con el conjunto de datos original, 7 % para el soporte y 90 % para la confianza creó 58 reglas de asociación. Luego de hacer la se-

lección de las reglas con las métricas de calidad y las funciones del paquete ARules, solo 17 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, quedaron 5 reglas, ver el segundo conjunto de la (Tabla 5.18).

Tabla 5.18. Reglas creadas con Apriori balanceado con ADASYN y clases positiva 140 / negativa 134. †

Número regla	Lift	conviction	hyperlift	hyper Conf	RPF	cosine	gini	fishersExactTest
[1]	**1.957143**	NA	1.444444	1	0.1423358	0.5277987	0.07938441	2.071304^{-13}
[2]	**1.957143**	NA	1.444444	1	0.1423358	0.5277987	0.07938441	2.071304^{-13}
[3]	**1.957143**	NA	1.464286	1	0.1496350	0.5411628	0.08417176	3.804358^{-14}
[4]	**1.957143**	NA	1.444444	1	0.1423358	0.5277987	0.07938441	2.071304^{-13}
[5]	**1.918000**	24.452555	1.484848	1	0.1752555	0.5856620	0.09821827	1.030317^{-15}
[6]	**1.913651**	22.007299	1.466667	1	0.1570154	0.5543479	0.08564903	7.636734^{-14}
[7]	**1.911628**	21.029197	1.448276	1	0.1497199	0.5413162	0.08077494	4.082915^{-13}
[8]	**1.908214**	19.562044	1.444444	1	0.1387774	0.5211594	0.07362168	4.813357^{-12}
[9]	**1.877259**	11.981752	1.468750	1	0.1645315	0.5674607	0.08750939	8.609090^{-14}
[10]	**1.868182**	10.759124	1.448276	1	0.1463172	0.5351296	0.07528894	5.262349^{-12}
[11]	**1.861672**	10.025547	1.392857	1	0.1353926	0.5147646	0.06821774	5.717473^{-11}
[12]	**1.861672**	10.025547	1.392857	1	0.1353926	0.5147646	0.06821774	5.717473^{-11}
[13]	**1.837318**	7.987835	1.437500	1	0.1576046	0.5553871	0.07972217	1.925741^{-12}
[14]	**1.823701**	7.172749	1.413793	1	0.1394326	0.5223884	0.06777184	1.001861^{-10}
Reglas creadas con el algoritmo Apriori con el conjunto de datos original desbalanceado. ‡								
[1]	**3.941176**	NA	1.88889	1	0.08457711	0.5773503	0.10290856	7.385605^{-12}
[2]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[3]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[4]	**3.941176**	NA	1.875000	1	0.07462687	0.5423261	0.08982531	2.016973^{-10}
[5]	**3.709343**	12.68657	1.777778	1	0.07491952	0.5433885	0.08732471	5.454796^{-10}

†El conjunto después del encabezado de la Tabla (primer conjunto) contiene los valores de cada métrica de calidad calculada para 14 reglas de asociación.
‡El conjunto después del subtitulo de la Tabla (segundo conjunto) contiene los valores de cada métrica de calidad calculada para 5 reglas de asociación.
♣ De los dos conjuntos de la Tabla, el primero tuvo la mejor calidad en comparación con el segundo, comparar los valores de la métrica lift.

Al comparar los valores reportados por la métrica lift en el subconjunto balanceado, se reportan valores por debajo de 2, a diferencia del conjunto de datos desbalanceado que reportó valores de casi 4, consulte la (Tabla 5.18). De los tres algoritmos de balanceo, el que tuvo el mejor resultado fue el algoritmo ADASYN y se puede apreciar en los valores reportados por la métrica lift, ver (Tabla 5.19).

Tabla 5.19. Valores reportados por la métrica lift para reglas creadas con conjunto de datos no balanceado y balanceado con tres algoritmos de balanceo. †

Numero regla	Lift (Datos no balanceados)	Lift (SMOTE)	Lift (ROSE)	Lift (ADASYN)
[1]	3.941176	2.313725	2.015152	1.957143
[2]	3.941176	2.313725	1.972276	1.957143
[3]	3.941176	2.313725	1.934545	1.957143
[4]	3.941176	2.313725	1.891775	1.957143
[5]	3.709343	2.313725	1.852639	1.918000
[6]		2.247619	1.828563	1.913651
[7]		2.239089		1.911628
[8]		2.239089		1.908214
[9]		2.236601		1.877259
[10]		2.181513		1.868182
[11]		2.177624		1.861672
[12]		2.173500		1.861672
[13]		2.173500		1.837318
[14]		2.096814		1.823701

†1 significa independencia y 3 inicia el ruido aleatorio e indica que hay desbalanceo en el conjunto de datos.

5.2.4. Patrones bacterianos que desencadenan vaginosis bacteriana con significancia estadística y biológica

Las reglas presentadas aquí describen los patrones involucrados en el antecedente (LHS) y el consecuente (RHS) de la regla a diferencia de la Sección 5.2.3 que solo presenta los valores calculados para cada métrica de calidad de cada regla.

5.2.5. Reglas creadas con el subconjunto casoPosNegP balanceado con SMOTE.

El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo SMOTE, 12 % para el soporte y 90 % para la confianza creó 232 reglas de asociación. Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*), solo 39 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, y solo quedaron 14 reglas, ver (Tabla 5.20).

Tabla 5.20. Reglas creadas con datos balanceados con SMOTE y frecuencia (f) de la regla.

[1]{gasseriUndetectable,MegasphaeraPos} ⇒	{VaginosisPos}	f 30
[2]{crisLowGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 29
[3]{GardnerellaPos,MegasphaeraPos,MycoplasHomiPos} ⇒	{VaginosisPos}	f 29
[4]{MegasphaeraPos,MycoplasHomiPos,UreaplasParPos} ⇒	{VaginosisPos}	f 31
[5]{AtopobiumPos,GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos,UreaplasParPos} ⇒	{VaginosisPos}	f 29
[6]{AtopobiumPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 34
[7]{GardnerellaPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 30
[8]{AtopobiumPos,gasseriUndetectable,UreaplasParPos} ⇒	{VaginosisPos}	f 30
[9]{AtopobiumPos,jensHighGrowthDensity,UreaplasParPos} ⇒	{VaginosisPos}	f 29
[10]{gassLowGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 33
[11]{AtopobiumPos,gassLowGrowthDensity,UreaplasParPos} ⇒	{VaginosisPos}	f 32
[12]{inersHighGrowthDensity,MycoplasHomiPos} ⇒	{VaginosisPos}	f 31
[13]{AtopobiumPos,jensLowGrowthDensity} ⇒	{VaginosisPos}	f 31
[14]{GardnerellaPos,jensLowGrowthDensity} ⇒	{VaginosisPos}	f 29

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

5.2.6. Reglas creadas con el subconjunto casoPosNegP balanceado con ROSE.

El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo de balanceo ROSE, 17 % para el soporte y 90 % para la confianza creó 74 reglas de asociación. Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*), solo 18 reglas resultaron significativas. El conjunto resultante se sometió a validación biológica, quedaron 6 reglas, ver (Tabla 5.21).

Tabla 5.21. Reglas creadas con datos balanceados con ROSE y frecuencia (f) de la regla.

[1]{AtopobiumPos,MegasphaeraPos,UreaplasParPos} ⇒	{VaginosisPos}	f 47
[2]{AtopobiumPos,inersHighGrowthDensity,UreaplasParPos} ⇒	{VaginosisPos}	f 46
[3]{AtopobiumPos,jensLowGrowthDensity} ⇒	{VaginosisPos}	f 48
[4]{MycoplasHomiPos} ⇒	{VaginosisPos}	f 46
[5]{AtopobiumPos,gassLowGrowthDensity} ⇒	{VaginosisPos}	f 57
[6]{GardnerellaPos,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 49

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

5.2.7. Reglas creadas con el subconjunto casoPosNegP balanceado con ADASYN.

El algoritmo Apriori con el subconjunto casoPosNegP balanceado con el algoritmo de balanceo ADASYN, 14 % para el soporte y 90 % para la confianza creó 341 reglas de asociación.
Luego de hacer la selección de las reglas con las métricas de calidad y las funciones del paquete ARules(*is.redundant()*, *is.significant()* e *is.maximal()*), solo 46 reglas resultaron significativas desde el punto de vista estadístico y computacional. El conjunto resultante se sometió a validación biológica por el experto, y solo quedaron 14 reglas, ver (Tabla 5.22).

Tabla 5.22. Reglas creadas con datos balanceados con ADASYN y frecuencia (f) de la regla.

[1]{GardnerellaPos,gassLowGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 39
[2]{crisLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 39
[3]{crisLowGrowthDensity,GardnerellaPos,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 41
[4]{GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos,MycoplasHomiPos} ⇒	{VaginosisPos}	f 39
[5]{AtopobiumPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 49
[6]{GardnerellaPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 44
[7]{crisLowGrowthDensity,MycoplasHomiPos} ⇒	{VaginosisPos}	f 42
[8]{AtopobiumPos,gasseriUndetectable,inersHighGrowthDensity} ⇒	{VaginosisPos}	f 39
[9]{AtopobiumPos,jensLowGrowthDensity} ⇒	{VaginosisPos}	f 47
[10]{gassLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 42
[11]{GardnerellaPos,gasseriUndetectable,UreaplasParPos} ⇒	{VaginosisPos}	f 39
[12]{AtopobiumPos,gassLowGrowthDensity,UreaplasParPos} ⇒	{VaginosisPos}	f 39
[13]{GardnerellaPos,jensLowGrowthDensity} ⇒	{VaginosisPos}	f 46
[14]{crisHighGrowthDensity,MegasphaeraPos} ⇒	{VaginosisPos}	f 41

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

5.2.8. Bacterias detectadas en los patrones reportados por las reglas de asociación

Las reglas de asociación se componen de dos conjuntos, uno en el antecedente (LHS) y otro en el consecuente (RHS). Los elementos en el conjunto

antecedente son bacterias Gram- que interactuarán entre sí para desencadenar el elemento en el conjunto consecuente que es la vaginosis bacteriana, consulte las Tablas 5.12, 5.20, 5.21 y 5.22.

En la literatura médica [8, 10, 11] se ha informado que ***Lactobacillus crispatus*** está asociado con la protección de la mucosa vaginal y ***Lactobacillus inners*** está asociado con mucosa vaginal alterada. Por lo tanto es común encontrar en los patrones reportados por los algoritmos que ***Lactobacillus crispatus*** está en una baja densidad de crecimiento y ***Lactobacillus inners*** en una alta densidad de crecimiento, ver Tablas 5.12, 5.20, 5.21 y 5.22.

Lactobacillus jensenii y ***Lactobacillus gasseri*** pueden tener una densidad de crecimiento alta o baja o ser indetectables y eso se debe a que pueden estar en diferentes niveles durante la transición de salud a enfermedad.

Hay cuatro especies de lactobacilos presentes en la mucosa vaginal y al analizar la fisiopatología en la que están involucradas cuando las bacterias Gram-oportunistas están presentes en una alta densidad de crecimiento proporciona información para el entendimiento de esta condición clínica.

Las bacterias Gram- oportunistas involucradas en el desarrollo de la vaginosis bacteriana reportadas por el algoritmo Apriori del subconjunto balanceado fueron las siguientes: ***Atopobium vaginae***, ***Gardnerella vaginalis***, ***Megasphaera filotipo 1***, ***Mycoplasma hominis*** y ***Ureaplasma parvum***, consulte las Tablas 5.20, 5.21 y 5.22.

Las bacterias Gram- oportunistas involucradas en el desarrollo de la vaginosis bacteriana reportadas por el algoritmo Apriori del conjunto de datos original fueron las siguientes: ***Atopobium vaginae***, ***Gardnerella vaginalis***, ***Megasphaera filotipo 1***, y ***Ureaplasma parvum***. Las bacterias ***Mycoplasma hominis*** y ***Lactobacillus crispatus*** no fueron reportadas por el algoritmo Apriori del conjunto de datos original multiclase desbalanceado 5.12. Este resultado nos dice que el balanceo en realidad mejoró la creación de reglas de asociación.

Estas bacterias están presentes en la mucosa vaginal en el estado de salud solo en un estado de crecimiento limitado, sin embargo, cuando se pierde la homeostasis que ejerce el ***Lactobacillus crispatus***, estas bacterias crecen a un nivel de densidad que provoca la vaginosis bacteriana, consulte las Tablas 5.12, 5.20, 5.21 y 5.22.

5.3. *Etapa 3.* Reglas creadas con el conjunto de datos desbalanceado y balanceado con los algoritmos Apriori, Eclat y FP-Growth

En esta última etapa de experimentos se crean las reglas de asociación con los algoritmos Apriori, Eclat y FP-Growth. El principal objetivo es determinar cuál de los tres algoritmos crea los mejores patrones según el conjunto de datos balanceado y sin balancear.

5.3.1. Reglas creadas con el algoritmo Apriori

El algoritmo Apriori se ejecutó con los parámetros descritos en la Subsección 5.1 con el conjunto de datos desbalanceado, 7 % para el soporte, 90 % para la confianza y luego de hacer la selección con la métricas de calidad y las funciones del paquete ARules (*is.redundant()*, *is.significant()* e *is.maximal()*) reportó 5 reglas, (Tabla 5.23). Con respecto a la métrica prueba exacta de Fisher, estas 5 reglas son aceptables ya que reporta valores cercanos a 0, la (Tabla 5.1 compila el intervalo de cada métrica. Por otro lado, la métrica lift reporta valores de casi 4, esto es consecuencia del desbalanceo entre las clases (positiva, negativa e indeterminada).

Tabla 5.23. Reglas creadas con los datos no balanceado y el algoritmo Apriori. †

Reglas	Lift	Fisher's exact test
[1]{AtopobiumPos, GardnerellaPos, inersHighGrowthDensity} ⇒ {VaginosisPos}	3.941176	7.385605^{-12}
[2]{AtopobiumPos, MegasphaeraPos, UreaplasmaParPos} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[3]{AtopobiumPos, gasseriUndetectable, MegasphaeraPos} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[4]{AtopobiumPos, GardnerellaPos, gasseriUndetectable} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[5]{AtopobiumPos, inersHighGrowthDensity, UreaplasmaParPos} ⇒ {VaginosisPos}	3.709343	5.454796^{-10}

†Patrón bacteriano que desencadena vaginosis bacteriana con significancia biológica y estadística
♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

El algoritmo Apriori con el conjunto de datos balanceado, 14 % para soporte, 90 % para confianza y luego de hacer la selección reportó 13 reglas, Tabla 5.24.
Estas 13 reglas son patrones altamente precisos porque la prueba exacta de

Fisher reporta valores cercanos a 0 y el lift reporta valores que fluctúan entre 1 y 2. Los valores cercanos a 0 reportados por la prueba exacta de Fisher dicen que es muy poco probable que los patrones sean producto de la aleatoriedad. Los valores reportados por la métrica lift muestran la dependencia del LHS y RHS de la regla y que no hay desbalanceo.

Tabla 5.24. Reglas creadas con Apriori y el subconjunto balanceado con ADASYN. †

Reglas	Lift	Fisher's exact test
[1]{GardnerellaPos,gassLowGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.957143	2.071304^{-13}
[2]{crisLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.957143	2.071304^{-13}
[3]{crisLowGrowthDensity,GardnerellaPos,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.957143	3.804358^{-14}
[4]{GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos,MycoplasHomiPos} ⇒ {VaginosisPos}	1.957143	2.071304^{-13}
[5]{AtopobiumPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.918000	1.030317^{-15}
[6]{GardnerellaPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.913651	7.636734^{-14}
[7]{crisLowGrowthDensity,MycoplasHomiPos} ⇒ {VaginosisPos}	1.911628	4.082915^{-13}
[8]{AtopobiumPos,gasseriUndetectable,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.908214	4.813357^{-12}
[9]{AtopobiumPos,jensLowGrowthDensity} ⇒ {VaginosisPos}	1.877259	8.609090^{-14}
[10]{gassLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.868182	5.262349^{-12}
[11]{GardnerellaPos,gasseriUndetectable,UreaplasParPos} ⇒ {VaginosisPos}	1.861672	5.717473^{-11}
[12]{AtopobiumPos,gassLowGrowthDensity,UreaplasParPos} ⇒ {VaginosisPos}	1.861672	5.717473^{-11}
[13]{GardnerellaPos,jensLowGrowthDensity} ⇒ {VaginosisPos}	1.837318	1.925741^{-12}

†Patrón bacteriano que desencadena vaginosis bacteriana con significancia biológica y estadística

♣ Un mayor número de bacterias (≥ 3) en LHS aumenta la precisión del diagnóstico.

5.3.2. Reglas creadas con el algoritmo Eclat

El algoritmo Eclat (*Equivalence Class Clustering and bottom up Lattice Traversal*) extrae conjuntos de elementos frecuentes mediante operaciones de intersección simple para el agrupamiento de clases de equivalencia para buscar los conjuntos de elementos frecuentes y reporta los conjuntos de elementos frecuentes a partir de los cuales se extraen las reglas de asociación. La función *eclat(tr, parameter = list(supp = 0.07, maxlen = 5)* con sus parámetros extrajó los conjuntos de elementos frecuentes. El parámetro *supp* establece el porcentaje de soporte para extraer los conjuntos de elementos frecuentes y el parámetro *maxlen* la longitud máxima de elementos frecuentes.

Para extraer las reglas de asociación a partir de los conjuntos de elementos frecuentes creados por el algoritmo ECLAT se usó la función *ruleInduction(itemsets, confidence = .9)* para generar reglas a partir de los conjuntos de elementos encontrados.
El parámetro *itemsets* representa los conjuntos de elementos frecuentes y *confianza* representa a la métrica con la que la función se guía para crear las reglas. Por lo tanto el soporte se usa primero para encontrar conjuntos de elementos frecuentes (significativos) con el algoritmo ECLAT. Luego, la confianza se usa en un segundo paso para producir reglas a partir de los conjuntos de elementos frecuentes que exceden un mínimo umbral de confianza con la función *ruleInduction()*.
La función *is.redundant()* en los algoritmos ECLAT y FP-Growth no está habilitada por lo tanto para completar la selección de reglas de asociación se usó el lenguaje de programación AWK acompañado de expresiones regulares. AWK es un lenguaje de programación cuya operación básica es buscar patrones en un conjunto de datos y realizar acciones específicas en las líneas o campos que contienen instancias de esos patrones.
Cuando se ejecuta AWK con una expresión regular, este lenguaje busca en el archivo que contiene a los datos línea por línea el patrón que describe la expresión regular. Encontrado el patron el lenguaje realiza una acción. En el caso específico de esta investigación, imprime las líneas que coinciden con el patrón buscado.
El algoritmo Eclat con el conjunto de datos desbalanceado, 7 % para el soporte, 90 % para la confianza y luego de hacer la selección reportó 6 reglas, Tabla 5.25. El algoritmo Eclat comparado con el algoritmo Apriori con los mismos parámetros y conjunto de datos desbalanceado tuvo mejor desempeño con respecto al número de reglas reportadas, Tablas 5.23 y 5.25. Las 6 reglas reportadas por el algoritmo Eclat son aceptables según la métrica de la prueba exacta de Fisher, pero el desbalanceo se ve según los valores reportados por la métrica lift.

Tabla 5.25. Reglas creadas con los datos no balanceado y el algoritmo Eclat. †

Reglas	Lift	Fisher's exact test
[1]{AtopobiumPos,inersHighGrowthDensity,UreaplasParPos} ⇒ {VaginosisPos}	3.709342	5.454796^{-10}
[2]{AtopobiumPos,gasseriUndetectable,MegasphaeraPos} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[3]{AtopobiumPos,MegasphaeraPos,UreaplasParPos} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[4]{AtopobiumPos,GardnerellaPos,gasseriUndetectable} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}
[5]{AtopobiumPos,MycoplasHomiPos} ⇒ {VaginosisPos}	3.565826	3.582408^{-11}
[6]{AtopobiumPos,GardnerellaPos,inersHighGrowthDensity} ⇒ {VaginosisPos}	3.941176	7.385604^{-12}

†Patrón bacteriano que desencadena vaginosis bacteriana con significancia biológica y estadística
♣ Un mayor número de bacterias (3) en LHS aumenta la precisión del diagnóstico.

El algoritmo Eclat con el conjunto de datos balanceado, 14 % para soporte, 90 % para confianza y después de hacer la selección reportó 14 reglas, ver la Tabla 5.26. Estos patrones son confiables y precisos según la prueba exacta de Fisher y la métrica de calidad lift.

Tabla 5.26. Reglas creadas con el algoritmo Eclat y el subconjunto balanceado con el algoritmo ADASYN. †

Reglas	Lift	Fisher's exact test
[1]{GardnerellaPos,gasseriUndetectable,UreaplasParPos} ⇒ {VaginosisPos}	1.861672	5.717473^{-11}
[2]{AtopobiumPos,gassLowGrowthDensity,UreaplasParPos} ⇒ {VaginosisPos}	1.861672	5.717473^{-11}
[3]{gassLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.868181	5.262348^{-12}
[4]{AtopobiumPos,gasseriUndetectable,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.908214	4.813356^{-12}
[5]{crisLowGrowthDensity,MycoplasHomiPos} ⇒ {VaginosisPos}	1.911627	4.082914^{-13}
[6]{GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos,MycoplasHomiPos} ⇒ {VaginosisPos}	1.957142	2.071303^{-13}
[7]{GardnerellaPos,gassLowGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.957142	2.071303^{-13}
[8]{crisLowGrowthDensity,inersHighGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.957142	2.071303^{-13}
[9]{GardnerellaPos,MegasphaeraPos,MycoplasHomiPos,UreaplasParPos} ⇒ {VaginosisPos}	1.957142	8.902198^{-14}
[10]{GardnerellaPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.913650	7.636734^{-14}
[11]{crisLowGrowthDensity,GardnerellaPos,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.957142	3.804358^{-14}
[12]{AtopobiumPos,gassLowGrowthDensity,inersHighGrowthDensity} ⇒ {VaginosisPos}	1.918	1.030316^{-15}
[13]{AtopobiumPos,inersHighGrowthDensity,MegasphaeraPos,UreaplasParPos} ⇒ {VaginosisPos}	1.957142	2.932692^{-21}
[14]{AtopobiumPos,GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos} ⇒ {VaginosisPos}	1.957142	4.130373^{-22}

†Patrón bacteriano que desencadena vaginosis bacteriana con significancia biológica y estadística
♣ Un mayor número de bacterias (3) en LHS aumenta la precisión del diagnóstico.

Todos los patrones reportados por este algoritmo son consistentes con lo que se observa en la clínica.

5.3.3. Reglas creadas con el algoritmo FP-Growth

Para ejecutar el algoritmo FP-Growth en el lenguaje de programación R a través del paquete ARules se hace a través de la función *fim4r(tr, method = fpgrowth, target = rules, supp = 7, conf = 90)*. Para que cree las reglas necesita los parámetros ya establecidos en la función. El parámetro *tr* representa el conjunto de datos tipo transacción, *method* representa al algoritmo que en este caso es el algoritmo FP-Growth, el parámetro *target* le pide al algoritmo que cree reglas de asociación, *supp* representa el umbral de soporte para buscar los conjuntos de elementos frecuentes y *conf* respresenta el umbral de confianza para que el algoritmo cree las reglas.
El algoritmo FP-Growth con el conjunto de datos desbalanceado, 7 % para el soporte, 90 % para la confianza y después de hacer la selección reportó 2 reglas, Tabla 5.27. En la selección mencionada se hicieron 2 etapas, por una parte se usaron las métricas de calidad y las funciones *is.significant()* e *is.maximal()* para validar estadísticamente y por otra para validar bilógicamente se usó el lenguaje de programación AWK con expresiones regulares para seleccionar solo los patrones que representan el comportamiento observado en la clínica.
El algoritmo FP-Growth tuvo un rendimiento más bajo en comparación con el algoritmo Apriori con los mismos parámetros y conjunto de datos desbalanceado, Tabla 5.23. De los 3 algoritmos, este tuvo el rendimiento más bajo con el conjunto de datos desbalanceado.

Tabla 5.27. Reglas creadas con el conjunto de datos no balanceado y el algoritmo FP-Growth. †

Rules	Lift	Fisher's exact test
[1]{AtopobiumPos,gasseriUndetectable,UreaplasParPos} ⇒ {VaginosisPos}	3.678431	1.246707^{-08}
[2]{AtopobiumPos,GardnerellaPos,gasseriUndetectable} ⇒ {VaginosisPos}	3.941176	2.016973^{-10}

†Patrón bacteriano que desencadena vaginosis bacteriana con significancia biológica y estadística
♣ Un mayor número de bacterias (3) en LHS aumenta la precisión del diagnóstico.

El algoritmo FP-Growth con el conjunto de datos balanceado, 14 % para

soporte, 90 % para confianza y después de hacer la selección reportó 2 reglas, Tabla 5.28. En la selección mencionada se hicieron 2 etapas de selección, por una parte se usarón las métricas de calidad y las funciones *is.significant()* e *is.maximal()* para validar estadísticamente y por otra para validar biológicamente se usó el lenguaje de programación AWK con expresiones regulares para seleccionar solo los patrones que representan el comportamiento observado en la clínica.
Estos patrones son altamente confiables y precisos de acuerdo con la prueba exacta de Fisher y la métrica de calidad lift. El algoritmo FP-Growth con el conjunto de datos balanceado creó un patrón más específico con respecto a la calidad ya que identificó a las bacterias hasta ahora relacionadas con la vaginosis bacteriana en su LHS.

Tabla 5.28. Reglas creadas con el algoritmo FP-Growth y el subconjunto balanceado con el algoritmo ADASYN. †

Rules	Lift	Fisher's exact test
[1]{AtopobiumPos,GardnerellaPos,gasseriUndetectable,UreaplasParPos} ⇒ {VaginosisPos}	1.957142	2.071303^{-13}
[2]{AtopobiumPos,GardnerellaPos,inersHighGrowthDensity,MegasphaeraPos,MycoplasHomiPos} ⇒ {VaginosisPos}	1.957142	2.071303^{-13}

†Patrón bacteriano que desencadena vaginosis bacteriana con significaancia biológica y estadística
♣ A higher number of bacteria (3) in LHS increases the accuracy of the diagnosis.

Estos patrones son computacionalmente significativos y biológicamente consistentes con lo que sucede en la clínica. Apriori, Eclat y FP-Growth reportan en su LHS todas las bacterias reportadas en la clínica [1, 3, 6, 7, 8, 10]. Apriori y Eclat producen un mayor número de reglas, principalmente con el conjunto de datos balanceado. Todos estos patrones representan diferentes combinaciones bacterianas que pueden ocurrir en pacientes que desarrollan vaginosis bacteriana. Esta variedad de patrones bacterianos dependerá de las condiciones fisiológicas particulares de cada paciente.
Por otro lado, el algoritmo FP-Growth con el conjunto de datos balanceado reportó 2 reglas de asociación. En esas 2 reglas reportó a las bacterias que reportaron los otros 2 algoritmos en sus diversos patrones. En el lado izquierdo de la regla este modelo de reglas de asociación ubica las bacterias Gram- que interactúan entre sí y apunta al lado derecho de la regla donde se ubica la enfermedad desencadenada por la interacción bacteriana, (ver figura 5.2).

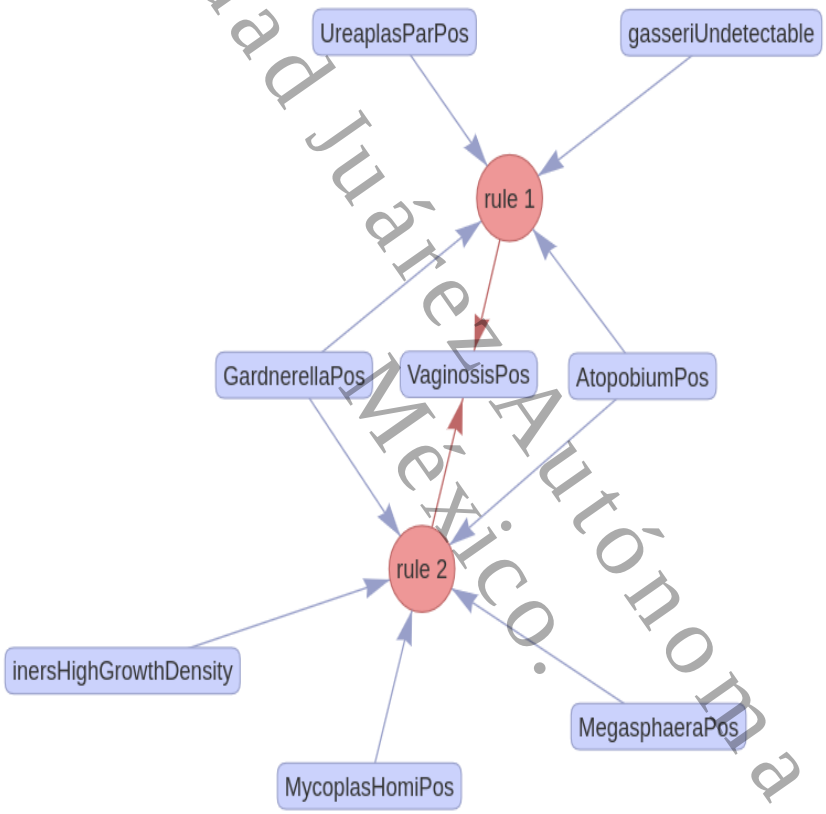

Figura 5.2. Reglas con significancia estadística y biológica.
La etiqueta rule con el número se refiere a cada una de las reglas que en este gráfico son 2. Las aristas que entran en un nodo representan en conjunto al antecedente de la regla y las aristas que salen del nodo apuntan a la etiqueta con el consecuente de la regla.

Por lo tanto, este modelo de reglas de asociación representa de manera específica y restrictiva la asociación que se desarrolla entre las bacterias Gram- que interactúan entre sí para desencadenar la vaginosis bacteriana.
Si analizamos la regla 14 de la (Tabla 5.26), lo que dice biológicamente la regla es que dada una paciente, desarrollará vaginosis bacteriana solo si la combinación de ***Atopobium vaginae***, ***Gardnerella vaginalis*** , ***Latobacillus iners*** con una alta densidad de crecimiento y ***Megasphaera philotype 1*** está presente.
Lactobacillus iners está relacionada con la flora vaginal alterada. Por lo tanto, en un patrón de vaginosis bacteriana, ***Lactobacillus iners*** tendrá una densidad de crecimiento alta Todos los patrones reportados por estos algoritmos son biológicamente consistentes.

Capítulo 6

Conclusiones

En esta investigación se estudió un conjunto de datos desbalanceado con tres clases (positivo 51 casos, negativo 134 casos e indeterminado 16 casos). En la primera etapa de experimentos se prepararon los datos con el formato que aceptan los algoritmos para crear reglas de asociación. Se renombraron las variables con una etiqueta apropiada para facilitar el análisis de cada bacteria y sus interrelaciones. Se investigó experimentalmente el mejor porcentaje para el soporte y la confianza. Se encontro que el procentaje de 7 % para el soporte y 90 % para la confianza reportaron los mejores patrones (reglas de asociación).

Para la validación estadística se investigaron las siguientes métricas: Lif, Convicción, RPF, Coseno, Índice Gini, Hiperconfianza, Hiperlift y Prueba exacta de Fisher. De todas las métricas de calidad analizadas en esta investigación la Prueba exacta de Fisher fue la que tuvo el mejor rendimiento.

El algoritmo usado en esta etapa experimental para crear las reglas fue el algoritmo clásico Apriori. Las reglas reportadas por este algoritmo biológicamente son consistentes con lo que se reporta en la clínica. Con los resultados de esta etapa experimental se demuestra la factibilidad de las reglas de asociación para modelar las posibles interacciones entre las bacterias asociadas a vaginosis bacteriana.

En la segunda etapa de experimentos se estudió un subconjunto de datos obtenido a partir del conjunto de datos original del cual se eliminó la clase indeterminada. Este subconjunto de datos se caracteriza porque está desbalanceado (clase positiva 51 casos, clase negativa 134 casos). Dado que el desbalanceo de datos puede generar ruido aleatorio, este subconjunto se balanceó con los algoritmos de balanceo de datos SMOTE, ROSE y ADASYN.

Para que los algoritmos SMOTE y ADASYN balanceen un conjunto de datos hay que determinar el vecino K más cercano.
Para determinar el valor K apropiado se usó el algoritmo Random Forest. Este algoritmo reportó el valor de 9 como el valor más apropiado para el número de vecinos K más cercanos. Durante el balanceo de datos el algoritmo ADASYN fue el que tuvo el mejor rendimeinto.
Una vez balanceado el subconjunto de datos se crearon las reglas de asociación con el algoritmo clásico Apriori. Las métricas usadas fueron la Prueba exacta de Fisher para realizar la validación estadística y el Lift para medir el ruido aleatorio que ocasiona el desbalanceo. En el conjunto de datos original desbalanceado el lift reportó valores por arriba de 4 dependiendo del grado de desbalanceo entre las clases.
En el subconjunto balanceado el lift reportó valores entre 1 y 2, lo que representa que no hay ruido aleatorio y como consecuencia mejores patrones (reglas de asociación). Los patrones reportados en este bloque de experimentos superan a los patrones reportados en el primer bloque de experimentos.
En la tercera etapa de experimentos se usó el subconjunto de datos balanceado y las métricas de calidad Lift y Prueba exacta de Fisher. Los algoritmos Apriori, Eclat y FP-growth se usaron para crear las reglas de asociación.
Con los algoritmos ECLAT y FP-Growth los cuales no tienen habilitada la función *is.redundant()* se utilizó el lenguaje de programación AWK apoyado con expresiones regulares para filtrar las reglas con significancia biológica. Los algoritmos con mayor rendimiento fueron Eclat y FP-Growth. El algoritmo Eclat con respecto al número de reglas presentadas y el FP-Growth con respecto a la calidad de las reglas.
El modelo de reglas de asociación representa de manera eficiente la asociación que se desarrolla entre las bacterias Gram- oportunistas que interactúan entre sí para desencadenar la vaginosis bacteriana. Conocer las bacterias que estos algoritmos ubican en el antecedente de la regla de asociación es de suma importancia ya que esto orienta al médico para atacar objetivamente el problema que representa la vaginosis bacteriana y evitar las recurrencias caraterísticas de esta infección.

6.1. Contribuciones

1. Definición de criterios para la selección de las reglas de asociación con base en las métricas de evaluación.

2. Descripción del desarrollo de la VB con base en los hallazgos derivados de esta investigación.

3. Modelo en forma de reglas de asociación que describe la coexistencia de bacterias en los casos positivos de VB.

4. Significancia clínica del modelo de VB basado en reglas de asociación.

6.2. Trabajos futuros

Las bacterias Gram- oportunistas involucradas en el desarrollo de vaginosis bacteriana reportadas por los algoritmos Apriori, Eclat y FP-Growth en el conjunto de datos fueron las siguientes: ***Atopobium vaginae***, ***Gardnerella vaginalis***, ***Megasphaera filotipo 1***, ***Mycoplasma hominis*** y ***Ureaplasma parvum***.
Debido a que la vaginosis bacteriana es un síndrome polimicrobiano se justifica en un trabajo futuro aumentar el número de bacterias a investigar para tener una mayor comprension de la etiología de esta infección.

Bibliografía

[1] Ravel, J., Gajer, P., Abdo, Z., Schneider, G. M., Koenig, S. S., McCulle, S. L., and Forney, L. J. (2011). Vaginal microbiome of reproductive-age women. Proceedings of the National Academy of Sciences, 108(Supplement 1), 4680-4687, https://doi.org/10.1073/pnas.1002611107

[2] Morris M, Nicoll A, Simms I, Wilson J, Catchpole M. (2001 May) Bacterial vaginosis: a public health review. BJOG. 108(5):439-50, https://doi.org/10.1111/j.1471-0528.2001.00124.x

[3] Onderdonk, A. B., Delaney, M. L., and Fichorova, R. N. (2016). The human microbiome during bacterial vaginosis. Clinical microbiology reviews, 29(2), 223-238, https://doi.org/10.1128/CMR.00075-15

[4] Hernández, J. A. S., García, L. L. C., González, E. V., Gordillo, L. V., and Tapia, J. A. R. (2007). Diagnóstico clínico, de laboratorio y tratamiento de la vaginosis por Gardnerella vaginilis. Universitas médica, 48(4), 382-395.

[5] Bagnall, P., Rizzolo, D. (2017). Bacterial vaginosis: a practical review. Journal of the American Academy of PAs, 30(12), 15-21. https://doi.org:10.1097/01.JAA.0000526770.60197.fa

[6] Gad[1], G. F., El-Adawy, A. R., Mohammed, M. S., Ahmed, A. F., and Mohamed[1], H. A. (2014). Evaluation of different diagnostic methods of bacterial vaginosis.

[7] Beverly, E. S., Chen, H. Y., Wang, Q. J., Zariffard, M. R., Cohen, M. H., and Spear, G. T. (2005). Utility of Amsel criteria, Nugent score, and quantitative PCR for Gardnerella vaginalis, Mycoplasma hominis, and

Lactobacillus spp. For diagnosis of bacterial vaginosis in human immunodeficiency virus-infected women. Journal of clinical microbiology, 43(9), 4607-4612, https://doi.org/10.1128/JCM.43.9.4607-4612.2005

[8] Kusters, J. G., Reuland, E. A., Bouter, S., Koenig, P., and Dorigo-Zetsma, J. W. (2015). A multiplex real-time PCR assay for routine diagnosis of bacterial vaginosis. European Journal of Clinical Microbiology and Infectious Diseases, 34(9), 1779-1785, https://doi.org/10.1007/s10096-015-2412-z

[9] Aguilera, P., Tachiquín, M. R., Graciela, M., Munive, R., and Olvera, B. P. (2014). PCR en tiempo real. Herramientas moleculares aplicadas en ecología: aspectos teóricos y prácticos. México DF: SEMARNAT, INECC, UAM-I, 175-201.

[10] Sanchez-Garcia, E. K., Contreras-Paredes, A., Martinez-Abundis, E., Garcia-Chan, D., Lizano, M., and de la cruz-Hernandez, E. (2019). Molecular epidemiology of bacterial vaginosis and its association with genital micro-organisms in asymptomatic women. Journal of medical microbiology, 68(9), 1373-1382, https://doi.org/10.1099/jmm.0.001044

[11] Zariffard, M. R., Saifuddin, M., Sha, B. E., and Spear, G. T. (2002). Detection of bacterial vaginosis-related organisms by real-time PCR for Lactobacilli, Gardnerella vaginalis and Mycoplasma hominis. FEMS Immunology and Medical Microbiology, 34(4), 277-281, https://doi.org/10.1111/j.1574-695X.2002.tb00634.x

[12] Baker, Y. S., Agrawal, R., Foster, J. A., Beck, D., and Dozier, G. (2014, March). Detecting bacterial vaginosis using machine learning. In Proceedings of the 2014 ACM Southeast Regional Conference (pp. 1-4), https://doi.org/10.1145/2638404.2638521

[13] Baker, Y. S., Agrawal, R., Foster, J. A., Beck, D., and Dozier, G. (2014, July). Applying machine learning techniques in detecting Bacterial Vaginosis. In 2014 International Conference on Machine Learning and Cybernetics (Vol. 1, pp. 241-246). IEEE, DOI: 10.1109/ICMLC.2014.7009123

[14] National Human Genome Research Institute. Reacción en Cadena de la Polimerasa (PCR). (August 26, 2022). https://www.genome.gov/

es/genetics-glossary/Reaccion-en-cadena-de-la-polimerasa. Accesado27deagostodel2022.

[15] Beck, D., and Foster, J. A. (2015). Machine learning classifiers provide insight into the relationship between microbial communities and bacterial vaginosis. BioData mining, 8(1), 1-9, https://doi.org/10.1186/s13040-015-0055-3

[16] Beck, D., and Foster, J. A. (2014). Machine learning techniques accurately classify microbial communities by bacterial vaginosis characteristics. PloS one, 9(2), e87830, https://doi.org/10.1371/journal.pone.0087830

[17] Marchán, E., Salcedo, J., Aza, T., Figuera, L., de Pisón, F. M., and Guillén, P. (2011). Reglas de asociación para determinar factores de riesgo epidemiológico de transmisión de la enfermedad de Chagas. Ciencia e Ingeniería, 32(2), 55-60, http://www.redalyc.org/articulo.oa?id=507550794009

[18] Chausa Fernández, P., Gómez Aguilera, E. J., Cáceres Taladriz, C., García Alcaide, F., and Gatell Artigas, J. M. (2006). Extracción de reglas de asociación en una base de datos clínicos de pacientes con VIH/SIDA, https://oa.upm.es/13883/

[19] Hahsler, M., Grün, B., and Hornik, K. (2005). arules-A computational environment for mining association rules and frequent item sets. Journal of statistical software, 14, 1-25.

[20] Hernández, J. A. R., Herrera, D. M. R., and Rodríguez, J. E. (2016). A research comparative among association rules algorithms. Visión electrónica, 10(2), 7.

[21] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining associations between sets of items in large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data (pp. 207-216).

[22] Hahsler, M. (2015). A probabilistic comparison of commonly used interest measures for association rules. United States. Southern Methodist University.

[23] Lenca, P., Vaillant, B., Meyer, P., and Lallich, S. (2007). Association rule interestingness measures: Experimental and theoretical studies. In Quality Measures in Data Mining (pp. 51-76). Springer, Berlin, Heidelberg.

[24] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997, June). Dynamic itemset counting and implication rules for market basket data. In Proceedings of the 1997 ACM SIGMOD international conference on Management of data (pp. 255-264).

[25] Hahsler, M., and Hornik, K. (2007). New probabilistic interest measures for association rules. Intelligent Data Analysis, 11(5), 437-455.

[26] Tan, P. N., Kumar, V., and Srivastava, J. (2004). Selecting the right objective measure for association analysis. Information Systems, 29(4), 293-313.

[27] Kumar, S., and Joshi, N. (2016). Rule power factor: a new interest measure in associative classification. Procedia Computer Science, 93, 12-18.

[28] Bayardo, R. J., Agrawal, R., and Gunopulos, D. (2000). Constraint-based rule mining in large, dense databases. Data mining and knowledge discovery, 4(2), 217-240.

[29] Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A., and Alvarado-Pérez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional, 63-86, doi: http://dx.doi.org/10.16925/9789587600490

[30] Agrawal, R., Imieliński, T., and Swami, A. (1993, June). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).

[31] Sánchez, O., Moyano, J. M., Sánchez, L., and Alcála-Fádez, J. (2017, July). Mining association rules in R using the package RKEEL. In 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-6). IEEE, https://doi.org/10.1109/FUZZ-IEEE.2017.8015572

[32] Moyano, J., and Sanchez, L. (2016, July). RKEEL: using KEEL in R code. In 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 257-264). IEEE.

[33] Mercaderes, R. D., and Llavori, R. B. Búsqueda de Reglas de Asociación en bases de datos y colecciones de textos.

[34] Naranjo Cuervo, R. C., and Sierra Martínez, L. M. (2009). Herramienta software para el análisis de canasta de mercado sin selección de candidatos. Ingeniería e Investigación, 29(1), 60-68.

[35] Heaton, J. (2016, March). Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms. In SoutheastCon 2016 (pp. 1-7). IEEE.

[36] Hernández, J. A. R., Herrera, D. M. R., and Rodríguez, J. E. (2016). A research comparative among association rules algorithms. Visión electrónica, 10(2), 7, http://dx.doi.org/10.14483/22484728.11654

[37] Longadge, R., and Dongre, S. (2013). Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707

[38] Hahsler, M., Buchta, C., Gruen, B., Hornik, K., Johnson, I., Borgelt, C., and Hahsler, M. M. (2021). Package 'arules', doi:10.18637/jss.v014.i15

[39] Agrawal, R., and Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. Very large data bases, VLDB (Vol. 1215, pp. 487-499).

[40] Borgelt, C. (2003, November). Efficient implementations of apriori and eclat. In FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations (p. 90).

[41] Borgelt, C. (2012). Frequent item set mining. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2(6), 437-456, doi: 10.1002/widm.1074

[42] Zaki, M. J., Parthasarathy, S., Ogihara, M., and Li, W. (1997, August). New algorithms for fast discovery of association rules. In KDD (Vol. 97, pp. 283-286).

[43] Hahsler, M., Buchta, C., and Hornik, K. (2008). Selective association rule generation. Computational Statistics, 23(2), 303-315.

[44] Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. ACM sigmod record, 29(2), 1-12.

[45] Fernández, A., Garcia, S., Herrera, F., and Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905, https://doi.org/10.1613/jair.1.11192

[46] Siriseriwan, W. (2019). A collection of oversampling techniques for class imbalance problem based on SMOTE.

[47] Lunardon, N., Menardi, G., Torelli, N., Lunardon, M. N., and Suggests, M. A. S. S. (2021). Package 'ROSE'.

[48] Zhang, J., and Chen, L. (2019). Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. Computer Assisted Surgery, 24(sup2), 62-72, https://doi.org/10.1080/24699322.2019.1649074

[49] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE, DOI: 10.1109/IJCNN.2008.4633969

[50] Biau, G., and Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

[51] Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). Random forests. In Ensemble machine learning (pp. 157-175). Springer, Boston, MA, https://doi.org/10.1007/978-1-4419-9326-7-5

[52] RColorBrewer, S., and Liaw, M. A. (2018). Package 'randomForest'. University of California, Berkeley: Berkeley, CA, USA.

[53] Candanedo, I. S., González, S. R., and Muñoz, L. (2019). Extracción de patrones para la Industria 4.0 a través de un modelo predictivo. I+ D Tecnológico, 15(2), 5-12.

[54] Mirabal Sosa, MayelÍn, Robaina García, Maytee, and Uranga Piña, Rolando. (2010). R: una herramienta poco difundida y muy útil para la investigación clínica. Revista Cubana de Investigaciones Biomédicas, 29(2), 302-308.

[55] Aho, A. V., Kernighan, B. W., and Weinberger, P. J. (1979). Awk—a pattern scanning and processing language. Software: Practice and Experience, 9(4), 267-279.

Freddy de la Crus Ruiz.pdf

Universidad Juárez Autónoma de Tabasco

Detalles del documento

Identificador de la entrega
trn:oid:::3117:582720111

Fecha de entrega
24 abr 2026, 1:55 p.m. GMT-6

Fecha de descarga
24 abr 2026, 3:27 p.m. GMT-6

Nombre del archivo
Freddy de la Crus Ruiz.pdf

Tamaño del archivo
6.1 MB

119 páginas

24.879 palabras

175.469 caracteres

7% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Coincidencias menores (menos de 10 palabras)
- Abstract

Fuentes principales

7% Fuentes de Internet
3% Publicaciones
0% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

Caracteres reemplazados
294 caracteres sospechosos en N.º de páginas
Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Fuentes principales

7% Fuentes de Internet
3% Publicaciones
0% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1 Internet
doczz.net <1%

2 Internet
ijcopi.org <1%

3 Internet
docplayer.es <1%

4 Internet
revistas.utp.ac.pa <1%

5 Internet
rstudio-pubs-static.s3.amazonaws.com <1%

6 Internet
hdl.handle.net <1%

7 Publicación
Freddy de la Cruz-Ruiz, Juana Canul-Reich, Rafael Rivera-López, Erick de la Cruz-H... <1%

8 Internet
academia-lab.com <1%

9 Internet
repositorio.ugto.mx <1%

10 Internet
repositorio.xoc.uam.mx <1%

11 Internet
www.researchgate.net <1%

12 Internet
sedici.unlp.edu.ar <1%

13 Internet
bibliotecadigital.udea.edu.co <1%

14 Internet
ri.ujat.mx <1%

15 Internet
ouci.dntb.gov.ua <1%

16 Internet
bookdown.org <1%

17 Internet
www.cienciadedatos.net <1%

18 Publicación
José Luis González-Pimentel, Alba Cuecas, Consolación Álvarez, Vicente Mariscal. ... <1%

19 Internet
vdoc.pub <1%

20 Internet
www.buenastareas.com <1%

21 Internet
rua.ua.es <1%

22 Internet
idus.us.es <1%

23 Internet
biocontainer-doc.readthedocs.io <1%

24 Internet
repositorio.uci.cu <1%

25 Internet
dehesa.unex.es <1%

26 Internet

dspace.lib.ntua.gr <1%

27 Publicación

"Proceedings of International Conference on Communication and Computational... <1%

28 Publicación

de Sousa, Ricardo Miguel Oliveira Pires. "Extraccao de Regras de Associacao com ... <1%

29 Internet

dskalizzy.github.io <1%

30 Internet

repositorio.ucv.edu.pe <1%

31 Internet

www2.adicciones.es <1%

32 Internet

ingenius.ups.edu.ec <1%

33 Internet

lists.nongnu.org <1%

34 Internet

research.vumc.nl <1%

35 Publicación

Philippe Nitsche, Pete Thomas, Rainer Stuetz, Ruth Welsh. "Pre-crash scenarios at... <1%

36 Internet

pubmed.ncbi.nlm.nih.gov <1%

37 Internet

diversitas.emnuvens.com.br <1%

38 Internet

archivos.ujat.mx <1%

39 Internet

crea.ujaen.es <1%

40 Internet
www.scribd.com <1%

41 Internet
id.123dok.com <1%

42 Internet
retos-operaciones-logistica.eae.es <1%

43 Internet
www.coursehero.com <1%

44 Internet
www.ptolomeo.unam.mx:8080 <1%

45 Internet
www.slideshare.net <1%

46 Internet
1library.co <1%

47 Internet
actitudsaludable.net <1%

48 Internet
escholarship.org <1%